

**APLICACION DE LA ARQUITECTURA DE CONSTRUCCION DE SOFTWARE  
POR COMPONENTES Y LA METODOLOGIA ORIENTADA A OBJETOS EN EL  
DESARROLLO DEL SISTEMA DE GESTION DEL AREA DE MONITORIA  
INFORMATICA, SECRETARIA E INVENTARIO DE LA UNIVERSIDAD DE  
NARIÑO EXTENSION IPIALES (SISGAMISI)**

**CLAUDIA JIMENA MEJIA NOGUERA  
NUBIA ROCIO VALLEJO BENAVIDES  
FREDY EDGARDO HERNANDEZ PARRA**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
INGENIERIA DE SISTEMAS  
SAN JUAN DE PASTO  
2006**

**APLICACION DE LA ARQUITECTURA DE CONSTRUCCION DE SOFTWARE  
POR COMPONENTES Y LA METODOLOGIA ORIENTADA A OBJETOS EN EL  
DESARROLLO DEL SISTEMA DE GESTION DEL AREA DE MONITORIA  
INFORMATICA, SECRETARIA E INVENTARIO DE LA UNIVERSIDAD DE  
NARIÑO EXTENSION IPIALES (SISGAMISI)**

**CLAUDIA JIMENA MEJIA NOGUERA  
NUBIA ROCIO VALLEJO BENAVIDES  
FREDY EDGARDO HERNANDEZ PARRA**

**Trabajo de Grado para optar el título de  
Ingeniero de Sistemas**

**Director  
HENRY ALONSO YELA  
Ingeniero de Sistemas**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
INGENIERIA DE SISTEMAS  
SAN JUAN DE PASTO  
2006**

**“Las ideas y conclusiones aportadas en la tesis de grado son responsabilidad exclusiva de los autores”.**

**“Artículo 1 del acuerdo 324 de octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño”.**

**Nota de Aceptación**

---

---

---

---

---

---

---

**Jurado**

---

**Jurado**

San Juan de Pasto Junio de 2006

## CONTENIDO

	pág.
INTRODUCCION.....	38
1. REFERENTES DEL PROYECTO.....	39
1.1 TITULO.....	39
1.2 MODALIDAD.....	39
2. MARCO REFERENCIAL.....	40
2.1 ESTADO DEL ARTE .....	41
2.2 MARCO HISTORICO.....	41
3. PLANTEAMIENTO DEL PROBLEMA.....	42
3.1 DESCRIPCION DEL PROBLEMA .....	42
3.2 FORMULACION DEL PROBLEMA.....	43
3.3 SISTEMATIZACION DEL PROBLEMA.....	43
4. JUSTIFICACION.....	45
5. ALCANCE Y DELIMITACION DEL SISTEMA.....	46
6. OBJETIVOS.....	48
6.1 OBJETIVO GENERAL.....	48
6.2 OBJETIVOS ESPECIFICOS.....	48
7. MARCO TEORICO.....	49
7.1 METODOLOGIA ORIENTADO A OBJETOS.....	49
7.2 LENGUAJE UNIFICADO DE MODELADO.....	50

<b>7.2.1 Artefactos de UML.....</b>	<b>51</b>
<b>7.2.1.1 Casos de Uso.....</b>	<b>51</b>
<b>7.2.1.2 Diagramas de Casos de Uso.....</b>	<b>52</b>
<b>7.2.1.3 Diagramas de Clases.....</b>	<b>53</b>
<b>7.2.1.4 Diagramas de Actividades.....</b>	<b>53</b>
<b>7.2.1.5 Diagrama de Despliegue.....</b>	<b>53</b>
<b>8. METODOLOGIA .....</b>	<b>54</b>
<b>9. ANALISIS.....</b>	<b>55</b>
<b>9.1 CASOS DE USO.....</b>	<b>55</b>
<b>9.1.1 Control de Acceso Al Sistema.....</b>	<b>55</b>
<b>9.1.1.1 Primera Iteración – Fachada.....</b>	<b>55</b>
<b>9.1.2 Administración de Reservas de Internet.....</b>	<b>57</b>
<b>9.1.2.1 Primera Iteración – Fachada .....</b>	<b>57</b>
<b>9.1.2.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>59</b>
<b>9.1.3 Administración de Reservas de Computadores.....</b>	<b>70</b>
<b>9.1.3.1 Primera Iteración – Fachada.....</b>	<b>70</b>
<b>9.1.3.2 Segunda Iteración – Llenado (Diagrama General) .....</b>	<b>73</b>
<b>9.1.4 Administración de Reservas Audiovisuales.....</b>	<b>84</b>
<b>9.1.4.1 Primera Iteración – Fachada.....</b>	<b>84</b>
<b>9.1.4.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>87</b>
<b>9.1.5 Administración de Horarios de Monitoria Informatica.....</b>	<b>98</b>
<b>9.1.5.1 Primera Iteración – Fachada.....</b>	<b>98</b>
<b>9.1.5.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>100</b>

<b>9.1.6 Administración de Horarios de Aulas de Informatica.....</b>	<b>105</b>
<b>9.1.6.1 Primera Iteración – Fachada.....</b>	<b>105</b>
<b>9.1.6.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>108</b>
<b>9.1.7 Administración de Servicios Audiovisuales.....</b>	<b>113</b>
<b>9.1.7.1 Primera Iteración – Fachada.....</b>	<b>114</b>
<b>9.1.7.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>116</b>
<b>9.1.8 Administración de Reservas de Aulas de Informatica.....</b>	<b>123</b>
<b>9.1.8.1 Primera Iteración – Fachada.....</b>	<b>123</b>
<b>9.1.8.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>125</b>
<b>9.1.9 Administración de Novedades.....</b>	<b>130</b>
<b>9.1.9.1 Primera Iteración – Fachada.....</b>	<b>130</b>
<b>9.1.9.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>131</b>
<b>9.1.10 Administración de Preguntas de Urna Virtual.....</b>	<b>135</b>
<b>9.1.10.1 Primera Iteración – Fachada.....</b>	<b>135</b>
<b>9.1.10.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>138</b>
<b>9.1.11 Administración de Opciones de Urna Virtual.....</b>	<b>147</b>
<b>9.1.11.1 Primera Iteración – Fachada.....</b>	<b>147</b>
<b>9.1.11.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>148</b>
<b>9.1.12 Administración de Servicios Recreativos.....</b>	<b>152</b>
<b>9.1.12.1 Primera Iteración – Fachada.....</b>	<b>152</b>
<b>9.1.12.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>154</b>
<b>9.1.13 Administración de Datos Personales de Usuarios.....</b>	<b>158</b>
<b>9.1.13.1 Primera Iteración – Fachada.....</b>	<b>158</b>

9.1.13.2 Segunda Iteración – Llenado (Diagrama General).....	160
9.1.14 Administración de Personal de Salud.....	166
9.1.14.1 Primera Iteración – Fachada.....	167
9.1.14.2 Segunda Iteración – Llenado (Diagrama General) .....	168
9.1.15 Administración de Personal Administrativo y Operativo.....	172
9.1.15.1 Primera Iteración – Fachada.....	172
9.1.15.2 Segunda Iteración – Llenado (Diagrama General).....	174
9.1.16 Administración de Profesores.....	179
9.1.16.1 Primera Iteración – Fachada.....	179
9.1.16.2 Segunda Iteración – Llenado (Diagrama General).....	181
9.1.17 Administración de Cargos.....	186
9.1.17.1 Primera Iteración – Fachada.....	186
9.1.17.2 Segunda Iteración – Llenado (Diagrama General).....	188
9.1.18 Administración de Lugares.....	193
9.1.18.1 Primera Iteración – Fachada.....	193
9.1.18.2 Segunda Iteración – Llenado (Diagrama General).....	194
9.1.19 Administración de Reservas de Citas de Salud.....	200
9.1.19.1 Primera Iteración – Fachada.....	200
9.1.19.2 Segunda Iteración – Llenado (Diagrama General).....	202
9.1.20 Administración de Horarios de Salud.....	208
9.1.20.1 Primera Iteración – Fachada.....	208
9.1.20.2 Segunda Iteración – Llenado (Diagrama General) .....	209
9.1.21 Administración de Estudiantes.....	211



9.1.21.1	Primera Iteración – Fachada.....	211
9.1.21.2	Segunda Iteración – Llenado (Diagrama General) .....	213
9.1.22	Administración de Elementos de Inventario.....	216
9.1.22.1	Primera Iteración – Fachada.....	216
9.1.22.2	Segunda Iteración – Llenado (Diagrama General) .....	218
9.1.23	Administración Tipos Elementos.....	224
9.1.23.1	Primera Iteración – Fachada.....	224
9.1.23.2	Segunda Iteración – Llenado (Diagrama General) .....	226
9.1.24	Administración Ubicaciones.....	231
9.1.24.1	Primera Iteración – Fachada.....	231
9.1.24.2	Segunda Iteración – Llenado (Diagrama General).....	233
9.1.25	Administración De Computadores.....	238
9.1.25.1	Primera Iteración – Fachada.....	238
9.1.25.2	Segunda Iteración – Llenado (Diagrama General).....	241
9.1.26	Administración de Componentes Computador Inicial.....	249
9.1.26.1	Primera Iteración – Fachada.....	249
9.1.26.2	Segunda Iteración – Llenado (Diagrama General).....	251
9.1.27	Administración de Componentes Computador Actual.....	255
9.1.27.1	Primera Iteración – Fachada.....	255
9.1.27.2	Segunda Iteración – Llenado (Diagrama General).....	257
9.1.28	Administración de Servicios Informaticos.....	263
9.1.28.1	Primera Iteración – Fachada.....	263
9.1.28.2	Segunda Iteración – Llenado (Diagrama General).....	265

<b>9.1.29 Manejo de Restricciones Medicas.....</b>	<b>271</b>
<b>9.1.29.1 Primera Iteración – Fachada.....</b>	<b>271</b>
<b>9.1.30 Administración de Roles.....</b>	<b>273</b>
<b>9.1.30.1 Primera Iteración – Fachada.....</b>	<b>273</b>
<b>9.1.30.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>275</b>
<b>9.1.31 Administración de Privilegios.....</b>	<b>282</b>
<b>9.1.31.1 Primera Iteración – Fachada.....</b>	<b>282</b>
<b>9.1.31.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>283</b>
<b>9.1.32 Administración Login Password.....</b>	<b>287</b>
<b>9.1.32.1 Primera Iteración – Fachada.....</b>	<b>287</b>
<b>9.1.32.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>289</b>
<b>9.1.33 Administración de Entradas de Almacén.....</b>	<b>293</b>
<b>9.1.33.1 Primera Iteración – Fachada.....</b>	<b>293</b>
<b>9.1.33.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>295</b>
<b>9.1.34 Administración de Salidas de Almacén.....</b>	<b>298</b>
<b>9.1.34.1 Primera Iteración – Fachada.....</b>	<b>298</b>
<b>9.1.34.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>299</b>
<b>9.1.35 Administración de Rubros.....</b>	<b>303</b>
<b>9.1.35.1 Primera Iteración – Fachada.....</b>	<b>303</b>
<b>9.1.35.2 Segunda Iteración – Llenado (Diagrama General).....</b>	<b>304</b>
<b>9.1.36 Administración de Gastos de Caja Menor.....</b>	<b>308</b>
<b>9.1.36.1 Primera Iteración - Fachada.....</b>	<b>308</b>
<b>9.1.36.2 Segunda Iteración – Llenado (Diagrama General) .....</b>	<b>310</b>

<b>10. DISEÑO.....</b>	<b>317</b>
<b>10.1 DIAGRAMA DE ARQUITECTURA.....</b>	<b>317</b>
<b>10.2 DIAGRAMAS DE CLASES.....</b>	<b>318</b>
<b>10.2.1 Control de Acceso al Sistema.....</b>	<b>318</b>
<b>10.2.2 Administración de Reservas de Internet.....</b>	<b>319</b>
<b>10.2.3 Administración de Reservas de Computadores.....</b>	<b>320</b>
<b>10.2.4 Administración de Reservas Audiovisuales.....</b>	<b>321</b>
<b>10.2.5 Administración de Horarios de Monitoría Informática.....</b>	<b>322</b>
<b>10.2.6 Administración de Horarios de Aulas de Informática.....</b>	<b>323</b>
<b>10.2.7 Administración de Servicios Audiovisuales.....</b>	<b>324</b>
<b>10.2.8 Administración de Reservas de Aulas de Informatica.....</b>	<b>325</b>
<b>10.2.9 Administración de Novedades.....</b>	<b>326</b>
<b>10.2.10 Administración de Preguntas de Urna Virtual.....</b>	<b>326</b>
<b>10.2.11 Administración de Opciones de Urna Virtual.....</b>	<b>327</b>
<b>10.2.12 Administración de Servicios Recreativos.....</b>	<b>327</b>
<b>10.2.13 Administración de Usuarios .....</b>	<b>328</b>
<b>10.2.14 Administración de Personal de Salud.....</b>	<b>329</b>
<b>10.2.15 Administración de Personal Administrativo y Operativo.....</b>	<b>330</b>
<b>10.2.16 Administración de Profesores.....</b>	<b>331</b>
<b>10.2.17 Administración de Cargos.....</b>	<b>331</b>
<b>10.2.18 Administración de Lugares.....</b>	<b>332</b>
<b>10.2.19 Administración de Citas Médicas.....</b>	<b>333</b>
<b>10.2.20 Administración de Citas Odontológicas.....</b>	<b>334</b>

10.2.21	Administración de Citas Psicológicas.....	335
10.2.22	Administración de Horario Médico.....	336
10.2.23	Administración de Horario Odontológico.....	336
10.2.24	Administración de Horario Psicológico.....	336
10.2.25	Administración de Estudiantes.....	337
10.2.26	Administración de Elementos de Inventario.....	338
10.2.27	Administración Tipos Elementos.....	339
10.2.28	Administración Ubicaciones.....	339
10.2.29	Administración Computadores.....	340
10.2.30	Administración de Componentes Computador Inicial.....	341
10.2.31	Administración De Componentes Computador Actual.....	342
10.2.32	Administración de Servicios Informáticos.....	343
10.2.33	Manejo de Restricciones Médicas.....	344
10.2.34	Administración de Roles.....	344
10.2.35	Administración de Privilegios.....	345
10.2.36	Administración de Login Y Password.....	346
10.2.37	Administración de Entradas de Almacén.....	347
10.2.38	Administración de Salidas de Almacén.....	348
10.2.39	Administración de Rubros.....	349
10.3	DIAGRAMA DE ACTIVIDADES.....	350
10.3.1	Diagrama Control de Acceso al Sistema.....	350
10.3.2	Administración de Reservas de Internet.....	351
10.3.2.1	Diagrama Crear Reservas de Internet.....	351

<b>10.3.2.2 Diagrama Consultar Reservas de Internet.....</b>	<b>352</b>
<b>10.3.2.3 Diagrama Actualizar Reservas de Internet.....</b>	<b>353</b>
<b>10.3.2.4 Diagrama Actualizar Observaciones y Faltas de Reservas de.....</b> <b>Internet</b>	<b>354</b>
<b>10.3.2.5 Diagrama Cancelar Reservas de Internet.....</b>	<b>355</b>
<b>10.3.2.6 Diagrama Verificar Usuarios Sancionados para.....</b> <b>Reservas de Internet</b>	<b>356</b>
<b>10.3.3 Administración de Reservas de Computadores.....</b>	<b>357</b>
<b>10.3.3.1 Crear Reservas de Computadores.....</b>	<b>357</b>
<b>10.3.3.2 Consultar Reservas de Computadores.....</b>	<b>358</b>
<b>10.3.3.3 Actualizar Reservas de Computadores.....</b>	<b>359</b>
<b>10.3.3.4 Diagrama Actualizar Observaciones y Faltas de.....</b> <b>Reservas de Computadores</b>	<b>360</b>
<b>10.3.3.5 Diagrama Cancelar Reservas de Computadores.....</b>	<b>361</b>
<b>10.3.3.6 Diagrama Verificar Usuarios Sancionados para.....</b> <b>Reservas de Computadores</b>	<b>362</b>
<b>10.3.4 Administración de Reservas Audiovisuales.....</b>	<b>363</b>
<b>10.3.4.1 Diagrama Crear Reservas Audiovisuales.....</b>	<b>363</b>
<b>10.3.4.2 Diagrama Consultar Reservas Audiovisuales.....</b>	<b>364</b>
<b>10.3.4.3 Diagrama Actualizar Equipos Asignados a la Reserva.....</b> <b>Audiovisual</b>	<b>365</b>
<b>10.3.4.4 Diagrama Actualizar Observaciones y Faltas de.....</b> <b>Reservas Audiovisuales</b>	<b>366</b>
<b>10.3.4.5 Diagrama Cancelar Reservas Audiovisuales.....</b>	<b>367</b>
<b>10.3.4.6 Diagrama Verificar Usuarios Sancionados para.....</b> <b>Reservas Audiovisuales</b>	<b>368</b>

<b>10.3.5 Administración de Horario de Monitoría Informática.....</b>	<b>369</b>
<b>10.3.5.1 Diagrama Asignar Franjas Horario Monitoría Informática.....</b>	<b>369</b>
<b>10.3.5.2 Diagrama Consultar Horario Monitoría Informática.....</b>	<b>370</b>
<b>10.3.5.3 Diagrama Actualizar Franja Horario Monitoría Informática.....</b>	<b>371</b>
<b>10.3.5.4 Diagrama Inicializar Horario Monitoría Informática.....</b>	<b>371</b>
<b>10.3.6 Administración de Horarios de Aulas de Informática.....</b>	<b>372</b>
<b>10.3.6.1 Diagrama Asignar Franjas Horarios Aulas de Informática.....</b>	<b>372</b>
<b>10.3.6.2 Diagrama Consultar Horario Aulas de Informática.....</b>	<b>373</b>
<b>10.3.6.3 Diagrama Actualizar Franja Horarios Aulas de Informática.....</b>	<b>374</b>
<b>10.3.6.4 Diagrama Inicializar Horarios Aulas de Informática.....</b>	<b>375</b>
<b>10.3.7 Administración de Servicios Audiovisuales.....</b>	<b>375</b>
<b>10.3.7.1 Diagrama Crear Servicio Audiovisual.....</b>	<b>375</b>
<b>10.3.7.2 Diagrama Consultar Servicio Audiovisual.....</b>	<b>376</b>
<b>10.3.7.3 Diagrama Actualizar Servicio Audiovisual.....</b>	<b>377</b>
<b>10.3.7.4 Diagrama Eliminar Servicio Audiovisual.....</b>	<b>378</b>
<b>10.3.7.5 Diagrama Asignar Tipos de Elementos.....</b>	<b>379</b>
<b>10.3.7.6 Diagrama Quitar Tipos de Elementos.....</b>	<b>380</b>
<b>10.3.8 Administración de Reservas de Aulas de Informática.....</b>	<b>381</b>
<b>10.3.8.1 Diagrama Crear Reserva de Aulas de informática.....</b>	<b>381</b>
<b>10.3.8.2 Diagrama Consultar Reservas de Aulas de Informática.....</b>	<b>382</b>
<b>10.3.8.3 Diagrama Cancelar Reservas de Aulas de Informática.....</b>	<b>382</b>
<b>10.3.9 Administración de Novedades.....</b>	<b>383</b>
<b>10.3.9.1 Diagrama Crear Novedades .....</b>	<b>383</b>

<b>10.3.9.2 Consultar Novedades.....</b>	<b>383</b>
<b>10.3.9.3 Actualizar Novedades.....</b>	<b>384</b>
<b>10.3.9.4 Diagrama Eliminar Novedades.....</b>	<b>384</b>
<b>10.3.10 Administración de Preguntas de Urna Virtual.....</b>	<b>385</b>
<b>10.3.10.1 Crear Pregunta Urna.....</b>	<b>385</b>
<b>10.3.10.2 Diagrama Consultar Preguntas.....</b>	<b>386</b>
<b>10.3.10.3 Diagrama Actualizar Pregunta Urna.....</b>	<b>386</b>
<b>10.3.10.4 Diagrama Eliminar Pregunta Urna.....</b>	<b>387</b>
<b>10.3.10.5 Diagrama Asignar Opciones a Pregunta Urna.....</b>	<b>388</b>
<b>10.3.10.6 Diagrama Consultar Resultados Urna.....</b>	<b>389</b>
<b>10.3.10.7 Diagrama Registrar Voto Urna.....</b>	<b>390</b>
<b>10.3.11 Administración de Opciones de Urna Virtual.....</b>	<b>390</b>
<b>10.3.11.1 Diagrama Crear Opciones Urna.....</b>	<b>390</b>
<b>10.3.11.2 Diagrama Consultar Opciones Urna.....</b>	<b>391</b>
<b>10.3.11.3 Diagrama Actualizar Opciones Urna.....</b>	<b>391</b>
<b>10.3.11.4 Diagrama Eliminar Opciones Urna.....</b>	<b>392</b>
<b>10.3.12 Administración de Servicios Recreativos.....</b>	<b>392</b>
<b>10.3.12.1 Diagrama Crear Servicio Recreativo.....</b>	<b>392</b>
<b>10.3.12.2 Diagrama Consultar Servicio Recreativo.....</b>	<b>393</b>
<b>10.3.12.3 Diagrama Actualizar Servicio Recreativo.....</b>	<b>393</b>
<b>10.3.12.4 Diagrama Eliminar Servicio Recreativo.....</b>	<b>394</b>
<b>10.3.13 Administración de Usuarios.....</b>	<b>394</b>
<b>10.3.13.1 Registrar Usuario.....</b>	<b>394</b>

<b>10.3.13.2</b>	<b>Consultar Usuario.....</b>	<b>395</b>
<b>10.3.13.3</b>	<b>Actualizar Usuario.....</b>	<b>395</b>
<b>10.3.13.4</b>	<b>Eliminar Usuario.....</b>	<b>396</b>
<b>10.3.14</b>	<b>Administracion del Personal de Salud.....</b>	<b>397</b>
<b>10.3.14.1</b>	<b>Registrar Personal de Salud.....</b>	<b>397</b>
<b>10.3.14.2</b>	<b>Consultar Personal de Salud.....</b>	<b>398</b>
<b>10.3.14.3</b>	<b>Actualizar Personal de Salud.....</b>	<b>398</b>
<b>10.3.14.4</b>	<b>Eliminar Personal de Salud .....</b>	<b>399</b>
<b>10.3.15</b>	<b>Administracion de Personal Administrativo y Operativo.....</b>	<b>400</b>
<b>10.3.15.1</b>	<b>Registrar Personal Administrativo y Operativo.....</b>	<b>400</b>
<b>10.3.15.2</b>	<b>Consultar Personal Administrativo y Operativo.....</b>	<b>401</b>
<b>10.3.15.3</b>	<b>Actualizar Personal Administrativo y Operativo.....</b>	<b>401</b>
<b>10.3.15.4</b>	<b>Eliminar Personal Administrativo y Operativo.....</b>	<b>402</b>
<b>10.3.16</b>	<b>Administracion de Profesores.....</b>	<b>402</b>
<b>10.3.16.1</b>	<b>Registrar Profesores.....</b>	<b>402</b>
<b>10.3.16.2</b>	<b>Consultar Profesores.....</b>	<b>403</b>
<b>10.3.16.3</b>	<b>Actualizar Profesores.....</b>	<b>403</b>
<b>10.3.16.4</b>	<b>Eliminar Profesores.....</b>	<b>404</b>
<b>10.3.17</b>	<b>Administracion de Cargos.....</b>	<b>404</b>
<b>10.3.17.1</b>	<b>Registrar Cargos.....</b>	<b>404</b>
<b>10.3.17.2</b>	<b>Consultar Cargos.....</b>	<b>405</b>
<b>10.3.17.3</b>	<b>Actualizar Cargos.....</b>	<b>405</b>
<b>10.3.17.4</b>	<b>Eliminar Cargos.....</b>	<b>406</b>



<b>10.3.18 Administracion de Lugares.....</b>	<b>407</b>
<b>10.3.18.1 Registrar Lugares.....</b>	<b>407</b>
<b>10.3.18.2 Consultar Lugares.....</b>	<b>408</b>
<b>10.3.18.3 Actualizar Lugares.....</b>	<b>409</b>
<b>10.3.18.4 Eliminar Lugares.....</b>	<b>410</b>
<b>10.3.19 Administracion de Citas.....</b>	<b>411</b>
<b>10.3.19.1 Crear Cita .....</b>	<b>411</b>
<b>10.3.19.2 Consultar Cita .....</b>	<b>412</b>
<b>10.3.19.3 Actualizar Cita .....</b>	<b>413</b>
<b>10.3.19.4 Cancelar Cita .....</b>	<b>414</b>
<b>10.3.20 Administracion de Horarios de Salud.....</b>	<b>415</b>
<b>10.3.20.1 Registrar Horario de Salud.....</b>	<b>415</b>
<b>10.3.20.2 Consultar Horario de Salud.....</b>	<b>416</b>
<b>10.3.21 Administración de Estudiantes.....</b>	<b>417</b>
<b>10.3.21.1 Diagrama Descargar Información de Estudiantes de la Extensión.....</b>	<b>417</b>
<b>10.3.21.2 Diagrama Consultar Datos de Estudiantes.....</b>	<b>418</b>
<b>10.3.22 Administración de Elementos de Inventario.....</b>	<b>418</b>
<b>10.3.22.1 Adicionar Elemento.....</b>	<b>418</b>
<b>10.3.22.2 Consultar Elemento.....</b>	<b>419</b>
<b>10.3.22.3 Actualizar Elemento.....</b>	<b>420</b>
<b>10.3.22.4 Eliminar Elemento.....</b>	<b>421</b>
<b>10.3.23 Administración Tipos de Elementos.....</b>	<b>421</b>

<b>10.3.23.1 Crear Tipo Elemento.....</b>	<b>421</b>
<b>10.3.23.2 Consultar Tipo Elemento.....</b>	<b>422</b>
<b>10.3.23.3 Actualizar Tipo Elemento.....</b>	<b>422</b>
<b>10.3.23.4 Eliminar Tipo Elemento.....</b>	<b>423</b>
<b>10.3.24 Administración Ubicaciones.....</b>	<b>424</b>
<b>10.3.24.1 Crear Ubicación.....</b>	<b>424</b>
<b>10.3.24.2 Consultar Ubicación.....</b>	<b>425</b>
<b>10.3.24.3 Actualizar Ubicación.....</b>	<b>425</b>
<b>10.3.24.4 Eliminar Ubicación.....</b>	<b>426</b>
<b>10.3.25 Administración de Computadores.....</b>	<b>427</b>
<b>10.3.25.1 Adicionar Computador.....</b>	<b>427</b>
<b>10.3.25.2 Consultar Computador.....</b>	<b>428</b>
<b>10.3.25.3 Actualizar Computador.....</b>	<b>429</b>
<b>10.3.25.4 Eliminar Computador.....</b>	<b>430</b>
<b>10.3.25.5 Consultar Registro Cambios Computador.....</b>	<b>431</b>
<b>10.3.26 Administración de Componentes Computador Inicial.....</b>	<b>432</b>
<b>10.3.26.1 Asignar Componentes Iniciales.....</b>	<b>432</b>
<b>10.3.26.2 Consultar Componente por Computador.....</b>	<b>433</b>
<b>10.3.26.3 Quitar Componente Inicial.....</b>	<b>433</b>
<b>10.3.27 Administración de Componentes Computador Actual.....</b>	<b>434</b>
<b>10.3.27.1 Asignar Componentes Actuales.....</b>	<b>434</b>
<b>10.3.27.2 Consultar Componente por Computador.....</b>	<b>435</b>
<b>10.3.27.3 Quitar Componente Actual.....</b>	<b>435</b>

<b>10.3.27.4 Consultar Registro Cambio Componente.....</b>	<b>436</b>
<b>10.3.28 Administración Servicios Informáticos.....</b>	<b>437</b>
<b>10.3.28.1 Actualizar Restricciones.....</b>	<b>437</b>
<b>10.3.28.2 Actualizar Sanciones.....</b>	<b>437</b>
<b>10.3.28.3 Adicionar Franja Horaria.....</b>	<b>438</b>
<b>10.3.28.4 Consultar Franja Horaria.....</b>	<b>438</b>
<b>10.3.28.5 Eliminar Franja Horaria.....</b>	<b>439</b>
<b>10.3.29 Actualizar Restricciones Médicas.....</b>	<b>440</b>
<b>10.3.30 Administración de Roles.....</b>	<b>440</b>
<b>10.3.30.1 Crear Rol.....</b>	<b>440</b>
<b>10.3.30.2 Consultar Rol.....</b>	<b>441</b>
<b>10.3.30.3 Actualizar Rol.....</b>	<b>441</b>
<b>10.3.30.4 Eliminar Rol.....</b>	<b>442</b>
<b>10.3.30.5 Asignar Privilegios Rol.....</b>	<b>442</b>
<b>10.3.30.6 Asignar Roles Usuario.....</b>	<b>443</b>
<b>10.3.31 Administración de Privilegios.....</b>	<b>444</b>
<b>10.3.31.1 Crear Privilegio.....</b>	<b>444</b>
<b>10.3.31.2 Consultar Privilegios.....</b>	<b>444</b>
<b>10.3.31.3 Actualizar Privilegio.....</b>	<b>445</b>
<b>10.3.31.4 Eliminar Privilegio.....</b>	<b>446</b>
<b>10.3.32 Administración de Login y Password.....</b>	<b>447</b>
<b>10.3.32.1 Registrar Login y Password.....</b>	<b>447</b>
<b>10.3.32.2 Consultar Login y Password.....</b>	<b>448</b>

10.3.32.3	Actualizar Password.....	449
10.3.33	Administración de Entradas de Almacén.....	450
10.3.33.1	Diagrama Registrar Entrada Almacén.....	450
10.3.33.2	Diagrama Consultar Entrada Almacén.....	451
10.3.34	Administración de Salidas de Almacén.....	452
10.3.34.1	Diagrama Registrar Salida Almacén.....	452
10.3.34.2	Diagrama Consultar Salida Almacén.....	453
10.3.35	Administración Rubros.....	454
10.3.35.1	Diagrama Registrar Rubro.....	454
10.3.35.2	Diagrama Consultar Rubro.....	455
10.3.35.3	Diagrama Actualizar Rubro.....	455
10.3.35.4	Diagrama eliminar Rubro.....	456
10.4	DIAGRAMA FISICO DE LA BASE DE DATOS.....	457
11.	IMPLEMENTACION.....	458
11.1	DOCUMENTACION DE CLASES.....	458
11.1.1	Monitoría Informática.....	458
11.1.1.1	Reserva Internet Value Object.....	458
11.1.1.2	Reserva Internet Visitor.....	459
11.1.1.3	Reserva Internet Manager.....	460
11.1.1.4	Reserva Internet Data Access Object.....	461
11.1.1.5	Reserva Computador Value Object.....	462
11.1.1.6	Reserva Computador Visitor.....	463
11.1.1.7	Reserva Computador Manager.....	464

11.1.1.8 Reserva Computador Data Access Object.....	466
11.1.1.9 Reserva Audiovisual Value Object.....	467
11.1.1.10 Reserva Audiovisual Visitor.....	468
11.1.1.11 Reserva Audiovisual Manager.....	469
11.1.1.12 Reserva Audiovisual Data Access Object.....	471
11.1.1.13 Reserva Audiovisual Value Object.....	473
11.1.1.14 Reserva Audiovisual Visitor.....	474
11.1.1.15 Reserva Audiovisual Manager.....	475
11.1.1.16 Reserva Audiovisual Data Access Object.....	477
11.1.1.17 Elemento Audiovisual Reservado Value Object.....	478
11.1.1.18 Elemento Audiovisual Reservado Visitor.....	479
11.1.1.19 Elemento Audiovisual Reservado Manager.....	480
11.1.1.20 Elemento Audiovisual Reservado Data Access Object.....	480
11.1.2 Administración Informática.....	481
11.1.2.1 Horario Monitor Informática Value Object.....	481
11.1.2.2 Horario Monitor Informática Visitor.....	482
11.1.2.3 Horario Monitor Informática Manager.....	482
11.1.2.4 Horario Monitor Informática Data Access Object.....	484
11.1.2.5 Horario Aula Informática Value Object.....	484
11.1.2.6 Horario Aula Informática Visitor.....	485
11.1.2.7 Horario Aula Informática Manager.....	486
11.1.2.8 Horario Aula Informática Data Access Object.....	488
11.1.2.9 Estado Aula Informática Value Object.....	489

<b>11.1.2.10</b>	<b>Estado Aula Informática Manager</b> .....	<b>489</b>
<b>11.1.2.11</b>	<b>Estado Aula Informática Data Access Object</b> .....	<b>490</b>
<b>11.1.2.12</b>	<b>Reserva Aula Informática Value Object</b> .....	<b>490</b>
<b>11.1.2.13</b>	<b>Reserva Aula Informática Visitor</b> .....	<b>491</b>
<b>11.1.2.14</b>	<b>Reserva Aula Informática Manager</b> .....	<b>492</b>
<b>11.1.2.15</b>	<b>Reserva Aula Informática Data Access Object</b> .....	<b>493</b>
<b>11.1.2.16</b>	<b>Servicio Audiovisual Value Object</b> .....	<b>494</b>
<b>11.1.2.17</b>	<b>Servicio Audiovisual Manager</b> .....	<b>495</b>
<b>11.1.2.18</b>	<b>Servicio Audiovisual Data Access Object</b> .....	<b>496</b>
<b>11.1.2.19</b>	<b>Elemento Servicio Audiovisual Value Object</b> .....	<b>496</b>
<b>11.1.2.20</b>	<b>Elemento Servicio Audiovisual Visitor</b> .....	<b>497</b>
<b>11.1.2.21</b>	<b>Elemento Servicio Audiovisual Manager</b> .....	<b>497</b>
<b>11.1.2.22</b>	<b>Elemento Servicio Audiovisual Data Access Object</b> .....	<b>498</b>
<b>11.1.3</b>	<b>Información</b> .....	<b>499</b>
<b>11.1.3.1</b>	<b>Novedad Value Object</b> .....	<b>499</b>
<b>11.1.3.2</b>	<b>Novedad Manager</b> .....	<b>500</b>
<b>11.1.3.3</b>	<b>Novedad Data Access Object</b> .....	<b>501</b>
<b>11.1.3.4</b>	<b>Servicio Recreativo Value Object</b> .....	<b>502</b>
<b>11.1.3.5</b>	<b>Servicio Recreativo Manager</b> .....	<b>502</b>
<b>11.1.3.6</b>	<b>Servicio Recreativo Data Access Object</b> .....	<b>503</b>
<b>11.1.3.7</b>	<b>Opción Urna Value Object</b> .....	<b>504</b>
<b>11.1.3.8</b>	<b>Opción Urna Manager</b> .....	<b>504</b>
<b>11.1.3.9</b>	<b>Opción Urna Data Access Object</b> .....	<b>505</b>

11.1.3.10	Pregunta Urna Value Object.....	506
11.1.3.11	Pregunta Urna Manager.....	507
11.1.3.12	Pregunta Urna Data Access Object.....	507
11.1.3.13	Opinion Urna Value Object.....	508
11.1.3.14	Opinion Urna Visitor.....	509
11.1.3.15	Opinion Urna Manager.....	509
11.1.3.16	Opinion Urna Data Access Object.....	510
11.1.4	Usuarios.....	511
11.1.4.1	Usuario Value Object.....	511
11.1.4.2	Usuario Visitor.....	514
11.1.4.3	Usuario Manager.....	516
11.1.4.4	Usuario Data Access Object.....	518
11.1.4.5	Tipo Identificación Value Object.....	520
11.1.4.6	Tipo de Identificación Manager.....	521
11.1.4.7	Tipo Identificación Data Access Object.....	521
11.1.4.8	Estado Usuario Value Object.....	522
11.1.4.9	Tipo de Identificación Manager.....	523
11.1.4.10	Tipo Identificación Data Access Object.....	523
11.1.4.11	Cargo Value Object.....	254
11.1.4.12	Cargo Manager.....	524
11.1.4.13	Cargo Data Access Object.....	525
11.1.4.14	Estudiante Value Object.....	526
11.1.4.15	Estudiante Visitor.....	526

11.1.4.16	Estudiante Manager.....	527
11.1.4.17	Estudiante Data Access Object.....	528
11.1.4.18	Estudiante Base Value Object.....	529
11.1.4.19	Estudiante Base Manager.....	532
11.1.4.20	Estudiante Base Data Access Object.....	532
11.1.4.21	Programa Value Object.....	533
11.1.4.22	Programa Visitor.....	534
11.1.4.23	Programa Manager.....	534
11.1.4.24	Programa Data Access Object.....	535
11.1.4.25	Semestre Value Object.....	536
11.1.4.26	Semestre Manager.....	536
11.1.4.27	Semestre Data Access Object.....	537
11.1.4.28	Facultad Value Object.....	537
11.1.4.29	Facultad Manager.....	538
11.1.4.30	Facultad Data Access Object.....	538
11.1.4.31	Lugar Value Object.....	539
11.1.4.32	Lugar Manager.....	540
11.1.4.33	Lugar Data Access Object.....	542
11.1.4.34	Dependencia Value Object.....	543
11.1.4.35	Dependencia Manager.....	543
11.1.4.36	Dependencia Data Access Object.....	544
11.1.4.37	Empleado Value Object.....	545
11.1.4.38	Empleado Visitor.....	545



11.1.4.39 Empleado Manager.....	546
11.1.4.40 Empleado Data Access Object.....	547
11.1.4.41 Médico Value Object.....	548
11.1.4.42 Médico Visitor.....	549
11.1.4.43 Médico Manager.....	550
11.1.4.44 Médico Data Access Object.....	551
11.1.4.45 Profesor Value Object.....	551
11.1.4.46 Profesor Visitor.....	552
11.1.4.47 Profesor Manager.....	553
11.1.4.48 Profesor Data Access Object.....	554
11.1.5 Secretaria.....	554
11.1.5.1 Horario Médico Value Object.....	554
11.1.5.2 Horario Médico Visitor.....	555
11.1.5.3 Horario Médico Manager.....	555
11.1.5.4 Horario Médico Data Access Object.....	556
11.1.5.5 Horario Odontólogo Value Object.....	557
11.1.5.6 Horario Odontólogo Visitor.....	558
11.1.5.7 Horario Odontólogo Manager.....	558
11.1.5.8 Horario Odontólogo Data Access Object.....	559
11.1.5.9 Horario Psicólogo Value Object.....	560
11.1.5.10 Horario Psicólogo Visitor.....	561
11.1.5.11 Horario Psicólogo Manager.....	561
11.1.5.12 Horario Psicólogo Data Access Object.....	562

<b>11.1.5.13 Cita Médica Value Object.....</b>	<b>563</b>
<b>11.1.5.14 Cita Médica Visitor.....</b>	<b>564</b>
<b>11.1.5.15 Cita Medica Manager.....</b>	<b>565</b>
<b>11.1.5.16 Cita Médica Data Access Object.....</b>	<b>566</b>
<b>11.1.5.17 Cita Odontológica Value Object.....</b>	<b>567</b>
<b>11.1.5.18 Cita Odontológica Visitor.....</b>	<b>568</b>
<b>11.1.5.19 Cita Odontológica Manager.....</b>	<b>569</b>
<b>11.1.5.20 Cita Odontológica Data Access Object.....</b>	<b>570</b>
<b>11.1.5.21 Cita Psicológica Value Object.....</b>	<b>571</b>
<b>11.1.5.22 Cita Psicológica Visitor.....</b>	<b>572</b>
<b>11.1.5.23 Cita Psicológica Manager.....</b>	<b>572</b>
<b>11.1.5.24 Cita Psicológica Data Access Object.....</b>	<b>574</b>
<b>11.1.6 Inventario.....</b>	<b>575</b>
<b>11.1.6.1 Inventario Value Object.....</b>	<b>575</b>
<b>11.1.6.2 Inventario Visitor.....</b>	<b>576</b>
<b>11.1.6.3 Inventario Manager.....</b>	<b>577</b>
<b>11.1.6.4 Inventario Data Access Object.....</b>	<b>579</b>
<b>11.1.6.5 Tipo Elemento Value Object.....</b>	<b>580</b>
<b>11.1.6.6 Tipo Elemento Manager.....</b>	<b>581</b>
<b>11.1.6.7 Tipo Elemento Data Access Object.....</b>	<b>582</b>
<b>11.1.6.8 Estado Elemento Value Object.....</b>	<b>582</b>
<b>11.1.6.9 Estado Elemento Manager.....</b>	<b>583</b>
<b>11.1.6.10 Estado Elemento Data Access Object.....</b>	<b>583</b>

<b>11.1.6.11</b>	<b>Tipo Ubicación Value Object.....</b>	<b>584</b>
<b>11.1.6.12</b>	<b>Tipo Ubicación Manager.....</b>	<b>585</b>
<b>11.1.6.13</b>	<b>Tipo Ubicación Data Access Object.....</b>	<b>585</b>
<b>11.1.6.14</b>	<b>Ubicación Value Object.....</b>	<b>586</b>
<b>11.1.6.15</b>	<b>Ubicación Visitor.....</b>	<b>586</b>
<b>11.1.6.16</b>	<b>Ubicación Manager.....</b>	<b>587</b>
<b>11.1.6.17</b>	<b>Ubicación Data Access Object.....</b>	<b>588</b>
<b>11.1.7</b>	<b>Administración de Computadores.....</b>	<b>589</b>
<b>11.1.7.1</b>	<b>Computador Value Object.....</b>	<b>589</b>
<b>11.1.7.2</b>	<b>Computador Visitor.....</b>	<b>590</b>
<b>11.1.7.3</b>	<b>Computador Manager.....</b>	<b>591</b>
<b>11.1.7.4</b>	<b>Computador Data Access Object.....</b>	<b>593</b>
<b>11.1.7.5</b>	<b>Elemento Computador Value Object.....</b>	<b>594</b>
<b>11.1.7.6</b>	<b>Elemento Computador Visitor.....</b>	<b>595</b>
<b>11.1.7.7</b>	<b>Elemento Computador Inicial Manager.....</b>	<b>595</b>
<b>11.1.7.8</b>	<b>Elemento Computador Actual Manager.....</b>	<b>596</b>
<b>11.1.7.9</b>	<b>Elemento Computador Inicial Data Access Object.....</b>	<b>597</b>
<b>1.1.7.10</b>	<b>Elemento Computador Actual Data Access Object.....</b>	<b>598</b>
<b>11.1.7.11</b>	<b>Registro Cambio Computador Value Object.....</b>	<b>598</b>
<b>11.1.7.12</b>	<b>Registro Cambio Computador Visitor.....</b>	<b>599</b>
<b>11.1.7.13</b>	<b>Registro Cambio Computador Manager.....</b>	<b>601</b>
<b>11.1.7.14</b>	<b>Registro Cambio Computador Data Access Object.....</b>	<b>602</b>
<b>11.1.7.15</b>	<b>Registro Cambio Componente Value Object.....</b>	<b>602</b>

11.1.7.16 Registro Cambio Componente Visitor.....	603
11.1.7.17 Registro Cambio Componente Manager.....	605
11.1.7.18 Registro Cambio Componente Data Access Object.....	606
11.1.8 Administración.....	607
11.1.8.1 Servicio General Value Object.....	607
11.1.8.2 Servicio General Manager.....	608
11.1.8.3 Servicio General Data Access Object.....	608
11.1.8.4 Franja Value Object.....	609
11.1.8.5 Franja Manager.....	610
11.1.8.6 Franja Data Access Object.....	610
11.1.8.7 Servicio Informático Value Object.....	611
11.1.8.8 Servicio Informático Manager.....	612
11.1.8.9 Servicio Informático Data Access Object.....	613
11.1.8.10 Privilegio Value Object.....	614
11.1.8.11 Privilegio Manager.....	615
11.1.8.12 Privilegio Data Access Object.....	615
11.1.8.13 Rol Value Object.....	616
11.1.8.14 Rol Manager.....	617
11.1.8.15 Rol Data Access Object.....	617
11.1.8.16 Rol Privilegio Value Object.....	618
11.1.8.17 Rol Privilegio Visitor.....	619
11.1.8.18 Rol Privilegio Manager.....	619
11.1.8.19 Rol Privilegio Data Access Object.....	620

11.1.8.20 Rol Usuario Value Object.....	621
11.1.8.21 Rol Usuario Visitor.....	621
11.1.8.22 Rol Usuario Manager.....	622
11.1.8.23 Rol Usuario Data Access Object.....	623
11.1.9 Almacén.....	624
11.1.9.1 Entrada Almacén Value Object.....	624
11.1.9.2 Entrada Almacén Visitor.....	625
11.1.9.3 Entrada Almacén Manager.....	626
11.1.9.4 Entrada Almacén Data Access Object.....	627
11.1.9.5 Salida Almacén Value Object.....	627
11.1.9.6 Salida Almacén Visitor.....	628
11.1.9.7 Salida Almacén Manager.....	629
11.1.9.8 Salida Almacén Data Access Object.....	630
11.1.10 Rubro.....	631
11.1.10.1 Rubro Value Object.....	631
11.1.10.2 Rubro Manager.....	632
11.1.10.3 Rubro Data Access Object.....	633
11.2 DIAGRAMAS DE SECUENCIA.....	634
11.2.1 Administración de Reservas de Internet.....	634
11.2.1.1 Consulta Disponibilidad Reservas de Internet.....	635
11.2.1.2 Creación Reserva de Internet.....	636
11.2.1.3 Actualización Reserva de Internet.....	637
11.2.1.4 Eliminación Reserva de Internet.....	638

11.2.1.5	Actualización Observaciones y Faltas de Reservas de Internet.....	639
11.2.1.6	Verificar Usuarios Sancionados para Reservas de Internet.....	640
11.2.2	Administración de Reservas de Computadores.....	641
11.2.2.1	Consulta Disponibilidad Reservas de Computadores.....	642
11.2.2.2	Creación Reserva de Computadores.....	643
11.2.2.3	Actualización Reserva de Computadores.....	644
11.2.2.4	Eliminación Reserva de Computadores.....	645
11.2.2.5	Actualización Observaciones y Faltas de Reservas de Computadores	646
11.2.2.6	Verificar Usuarios Sancionados para Reservas de Computadores	647
11.2.3	Administración de Reservas de Audiovisuales.....	648
11.2.3.1	Consulta Disponibilidad Reservas de Audiovisuales.....	648
11.2.3.2	Creación Reservas de Audiovisuales.....	649
11.2.3.3	Consulta Reservas de Audiovisuales.....	650
11.2.3.4	Cancelación Reservas de Audiovisuales.....	651
11.2.3.5	Actualización Observaciones y Faltas Reservas de Audiovisuales	652
11.2.3.6	Verificar Usuarios Sancionados para Reservas de Audiovisuales	653
11.2.4	Administración de Horarios de Monitoria Informática.....	654
11.2.4.1	Asignar Franjas Horario Monitoría Informática.....	654
11.2.4.2	Consultar Horario Monitoría Informática.....	655
11.2.4.3	Actualizar Franja Horario Monitoría Informática.....	656
11.2.4.4	Inicializar Horario Monitoría Informática.....	657

<b>11.2.5 Administración de Horarios de Aulas De Informática.....</b>	<b>658</b>
<b>11.2.5.1 Asignar Franjas Horario Aula Informática.....</b>	<b>658</b>
<b>11.2.5.2 Consultar Horario Aula Informática.....</b>	<b>659</b>
<b>11.2.5.3 Actualizar Franjas Horario Aula Informática.....</b>	<b>660</b>
<b>11.2.5.4 Inicializar Horario Aula Informática.....</b>	<b>661</b>
<b>11.2.6 Administración de Servicios Audiovisuales.....</b>	<b>662</b>
<b>11.2.6.1 Creación Servicio Audiovisual.....</b>	<b>662</b>
<b>11.2.6.2 Consulta Servicio Audiovisual.....</b>	<b>663</b>
<b>11.2.6.3 Actualizar Servicio Audiovisual.....</b>	<b>664</b>
<b>11.2.6.4 Eliminar Servicio Audiovisual.....</b>	<b>665</b>
<b>11.2.6.5 Asignar Tipos de Elementos.....</b>	<b>666</b>
<b>11.2.6.6 Quitar Tipos de Elementos.....</b>	<b>667</b>
<b>11.2.7 Administración de Reservas de Aulas de Informática.....</b>	<b>668</b>
<b>11.2.7.1 Creación Reserva Aula Informática.....</b>	<b>668</b>
<b>11.2.7.2 Consultar Reserva Aula Informática.....</b>	<b>669</b>
<b>11.2.7.3 Cancelar Reserva Aula Informática.....</b>	<b>670</b>
<b>11.2.8 Administración Novedades.....</b>	<b>671</b>
<b>11.2.8.1 Crear Novedad.....</b>	<b>671</b>
<b>11.2.8.2 Consultar Novedad.....</b>	<b>672</b>
<b>11.2.8.3 Actualizar Novedad.....</b>	<b>673</b>
<b>11.2.8.4 Eliminar Novedad.....</b>	<b>674</b>
<b>11.2.9 Administración de Usuarios.....</b>	<b>675</b>
<b>11.2.9.1 Registrar Usuario.....</b>	<b>675</b>

11.2.9.2	Consultar Usuario.....	676
11.2.9.3	Actualizar Usuario.....	677
11.2.9.4	Eliminar Usuario.....	678
11.2.10	Administración del Personal de Salud.....	679
11.2.10.1	Registrar Personal de Salud.....	679
11.2.10.2	Consultar Personal de Salud.....	680
11.2.10.3	Actualizar Personal de Salud.....	681
11.2.10.4	Eliminar Personal de Salud.....	682
11.2.11	Administración del Personal Administrativo y Operativo.....	683
11.2.11.1	Registrar Personal Administrativo y Operativo.....	683
11.2.11.2	Consultar Personal Administrativo y Operativo.....	684
11.2.11.3	Actualizar Personal Administrativo y Operativo.....	685
11.2.11.4	Eliminar Personal Administrativo y Operativo.....	686
11.2.12	Administración de Profesores.....	687
11.2.12.1	Registrar Profesor.....	687
11.2.12.2	Consultar Profesor.....	688
11.2.12.3	Actualizar Profesor.....	689
11.2.12.4	Eliminar Profesor.....	690
11.2.13	Administración de Cargos.....	691
11.2.13.1	Registrar Cargos.....	691
11.2.13.2	Consultar Cargo.....	692
1.2.13.3	Actualizar Cargo.....	693
11.2.13.4	Eliminar Cargo.....	694



<b>11.2.14 Administración de Lugares.....</b>	<b>695</b>
<b>11.2.14.1 Registrar País.....</b>	<b>695</b>
<b>11.2.14.2 Registrar Departamento.....</b>	<b>696</b>
<b>11.2.14.3 Registrar Ciudad.....</b>	<b>697</b>
<b>11.2.14.4 Consultar Lugares.....</b>	<b>698</b>
<b>11.2.14.5 Actualizar País.....</b>	<b>699</b>
<b>11.2.14.6 Actualizar Departamento.....</b>	<b>700</b>
<b>11.2.14.7 Actualizar Ciudad.....</b>	<b>701</b>
<b>11.2.14.8 Eliminar País.....</b>	<b>702</b>
<b>11.2.14.9 Eliminar Departamento.....</b>	<b>703</b>
<b>11.2.14.10 Eliminar Ciudad.....</b>	<b>704</b>
<b>11.2.15 Administración de Citas Médicas.....</b>	<b>705</b>
<b>11.2.15.1 Crear Cita Médica.....</b>	<b>705</b>
<b>11.2.15.2 Consultar Disponibilidad de Citas Médicas.....</b>	<b>706</b>
<b>11.2.15.3 Consultar Cita Médica.....</b>	<b>707</b>
<b>11.2.15.4 Actualizar Cita Médica.....</b>	<b>708</b>
<b>11.2.15.5 Cancelar Cita Médica.....</b>	<b>709</b>
<b>11.2.16 Administración de Citas Odontológicas.....</b>	<b>710</b>
<b>11.2.16.1 Crear Cita Odontológica.....</b>	<b>710</b>
<b>11.2.16.2 Consultar Disponibilidad de Citas Odontológicas.....</b>	<b>711</b>
<b>11.2.16.3 Consultar Cita Odontológica.....</b>	<b>712</b>
<b>11.2.16.4 Actualizar Cita Odontológica.....</b>	<b>713</b>
<b>11.2.16.5 Cancelar Cita Odontológica.....</b>	<b>714</b>

<b>11.2.17 Administración De Citas Psicológicas.....</b>	<b>715</b>
<b>11.2.17.1 Crear Cita Psicológica.....</b>	<b>715</b>
<b>11.2.17.2 Verificar Disponibilidad Citas Psicológicas.....</b>	<b>716</b>
<b>11.2.17.3 Consultar Cita Psicológica.....</b>	<b>717</b>
<b>11.2.17.4 Actualizar Cita Psicológica.....</b>	<b>718</b>
<b>11.2.18 Administración de Horarios Médicos.....</b>	<b>719</b>
<b>11.2.18.1 Asignar Horario Médico.....</b>	<b>719</b>
<b>11.2.18.2 Consultar Horario Médico.....</b>	<b>720</b>
<b>11.2.19 Administración de Horarios Odontológicos.....</b>	<b>721</b>
<b>11.2.19.1 Asignar Horario Odontológico.....</b>	<b>721</b>
<b>11.2.19.2 Consultar Horario Odontológico.....</b>	<b>722</b>
<b>11.2.20 Administración de Horarios Psicológicos.....</b>	<b>723</b>
<b>11.2.20.1 Asignar Horario Psicológico.....</b>	<b>723</b>
<b>11.2.20.2 Consultar Horario Psicológico.....</b>	<b>724</b>
<b>11.2.21 Administración de Estudiantes.....</b>	<b>725</b>
<b>11.2.21.1 Descargar Información Estudiantes de la Extensión.....</b>	<b>725</b>
<b>11.2.21.2 Consultar Datos de Estudiantes.....</b>	<b>726</b>
<b>11.2.22 Administración de Elementos de Inventario.....</b>	<b>727</b>
<b>11.2.22.1 Adicionar Elementos de Inventario.....</b>	<b>727</b>
<b>11.2.22.2 Consultar Elemento de Inventario.....</b>	<b>728</b>
<b>11.2.22.3 Actualizar Elemento de Inventario.....</b>	<b>729</b>
<b>11.2.22.4 Eliminar Elemento de Inventario.....</b>	<b>730</b>
<b>11.2.22.5 Consultar Elemento Actualizados.....</b>	<b>731</b>

<b>11.2.22.6 Consultar Elemento Eliminados.....</b>	<b>732</b>
<b>11.2.23 Administración Tipos de Elementos.....</b>	<b>733</b>
<b>11.2.23.1 Crear Tipo Elemento.....</b>	<b>733</b>
<b>11.2.23.2 Consultar Tipo Elemento.....</b>	<b>734</b>
<b>11.2.23.3 Actualizar Tipo Elemento.....</b>	<b>735</b>
<b>11.2.23.4 Eliminar Tipo Elemento.....</b>	<b>736</b>
<b>11.2.24 Administración Ubicaciones.....</b>	<b>737</b>
<b>11.2.24.1 Crear Ubicación.....</b>	<b>737</b>
<b>11.2.24.2 Consultar Ubicación.....</b>	<b>738</b>
<b>11.2.24.3 Actualizar Ubicación.....</b>	<b>739</b>
<b>11.2.24.4 Eliminar Ubicación.....</b>	<b>740</b>
<b>11.2.25 Administración de Computadores.....</b>	<b>741</b>
<b>11.2.25.1 Adicionar Computador.....</b>	<b>741</b>
<b>11.2.25.2 Consultar Computador.....</b>	<b>742</b>
<b>11.2.25.3 Actualizar Computador.....</b>	<b>743</b>
<b>11.2.25.4 Eliminar Computador.....</b>	<b>744</b>
<b>11.2.25.5 Consultar Cambios Computador.....</b>	<b>745</b>
<b>11.2.25.6 Consultar Eliminación Computador.....</b>	<b>746</b>
<b>11.2.26 Administración de Componentes Computador Inicial.....</b>	<b>747</b>
<b>11.2.26.1 Asignar Componente Iniciales.....</b>	<b>747</b>
<b>11.2.26.2 Consultar Componente por Computador.....</b>	<b>748</b>
<b>11.2.26.3 Quitar Componente Inicial.....</b>	<b>749</b>
<b>11.2.27 Administración de Componentes Computador Actual.....</b>	<b>750</b>

<b>11.2.27.1 Asignar Componente Computador Actual.....</b>	<b>750</b>
<b>11.2.27.2 Consultar Componente por Computador.....</b>	<b>751</b>
<b>11.2.27.3 Quitar Componente Actual.....</b>	<b>752</b>
<b>11.2.27.4 Consulta Registro Cambio Componente.....</b>	<b>753</b>
<b>11.2.28 Administración de Servicios Informáticos.....</b>	<b>754</b>
<b>11.2.28.1 Actualizar Restricciones.....</b>	<b>754</b>
<b>11.2.28.2 Actualizar Sanciones.....</b>	<b>755</b>
<b>11.2.28.3 Adicionar Franja Horaria.....</b>	<b>756</b>
<b>11.2.28.4 Consultar Franja Horaria.....</b>	<b>757</b>
<b>11.2.28.5 Eliminar Franja Horaria.....</b>	<b>758</b>
<b>11.2.29 Actualizar Restricciones Médicas.....</b>	<b>759</b>
<b>11.2.30 Administración de Roles.....</b>	<b>760</b>
<b>11.2.30.1 Crear Rol.....</b>	<b>760</b>
<b>11.2.30.2 Consultar Rol.....</b>	<b>761</b>
<b>11.2.30.3 Actualizar Rol.....</b>	<b>762</b>
<b>11.2.30.4 Eliminar Rol.....</b>	<b>763</b>
<b>11.2.30.5 Asignar Privilegios Rol.....</b>	<b>764</b>
<b>11.2.30.6 Asignar Roles Usuario.....</b>	<b>765</b>
<b>11.2.31 Administración Privilegios.....</b>	<b>766</b>
<b>11.2.31.1 Crear Privilegio.....</b>	<b>766</b>
<b>11.2.31.2 Consultar Privilegio.....</b>	<b>767</b>
<b>11.2.31.3 Actualizar Privilegio.....</b>	<b>768</b>
<b>11.2.31.4 Eliminar Privilegio.....</b>	<b>769</b>

<b>11.2.32 Administración Login y Password.....</b>	<b>770</b>
<b>11.2.32.1 Consultar Login y Password.....</b>	<b>770</b>
<b>11.2.32.2 Actualizar Password.....</b>	<b>771</b>
<b>11.2.33 Administración de Entradas de Almacén.....</b>	<b>772</b>
<b>11.2.33.1 Registrar Entrada Almacén.....</b>	<b>772</b>
<b>11.2.33.2 Consultar Entrada Almacén.....</b>	<b>773</b>
<b>11.2.34 Administración de Salidas de Almacén.....</b>	<b>774</b>
<b>11.2.34.1 Registrar Salida Almacén.....</b>	<b>774</b>
<b>11.2.34.2 Consultar Salida Almacén.....</b>	<b>775</b>
<b>11.2.35 Administración Rubro.....</b>	<b>776</b>
<b>11.2.35.1 Registrar Rubro.....</b>	<b>776</b>
<b>11.2.35.2 Consultar Rubro.....</b>	<b>777</b>
<b>11.2.35.3 Actualizar Rubro.....</b>	<b>778</b>
<b>11.2.35.4 Eliminar Rubro.....</b>	<b>779</b>
<b>11.2.36 Administración de Gastos de Caja Menor.....</b>	<b>780</b>
<b>11.2.36.1 Registro Gasto Caja Menor.....</b>	<b>780</b>
<b>11.2.36.2 Consulta Gasto Caja Menor.....</b>	<b>781</b>
<b>11.2.36.3 actualización Gasto Caja Menor.....</b>	<b>782</b>
<b>11.2.36.4 Eliminación Gasto Caja Menor.....</b>	<b>783</b>
<b>11.2.36.5 Asignación Presupuesto Rubro.....</b>	<b>784</b>
<b>11.2.36.6 Consulta Asignación Presupuesto.....</b>	<b>785</b>
<b>11.2.36.7 Actualización Asignación Presupuesto.....</b>	<b>786</b>
<b>11.2.36.8 Eliminación Asignación Presupuesto.....</b>	<b>787</b>

<b>11.2.36.9 Consulta Relación Presupuestal.....</b>	<b>788</b>
<b>11.3 DIAGRAMA DE DESPLIEGUE.....</b>	<b>789</b>
<b>12. CONCLUSIONES.....</b>	<b>790</b>
<b>13. RECOMENDACIONES.....</b>	<b>791</b>
<b>BIBLIOGRAFIA.....</b>	<b>792</b>

## **RESUMEN**

El mundo moderno exige la aplicación de nuevas técnicas para dar soluciones oportunas y eficientes a los diferentes problemas que se presentan actualmente en nuestra sociedad. El desarrollo de un software que se ajuste a los patrones de calidad debe soportar características de portabilidad, reusabilidad, modularidad y escalabilidad.

El software se convierte en una herramienta de apoyo que permite optimizar y agilizar los procesos que actualmente se manejan en las dependencias de Monitoría Informática, Secretaria e Inventario de la Universidad de Nariño Extensión Ipiales.

La metodología aplicada para el desarrollo: la Programación Orientada a Objetos, la construcción de Software por Componentes y la arquitectura J2EE, permiten crear sistemas fiables que sean flexibles, mantenibles y capaces de evolucionar para cumplir las necesidades del cambio.

## **ABSTRACT**

The modern world demands the application of new techniques to give opportune and efficient solutions to the different problems that are presented at the moment in our society. So that the developed software is adjusted to the patterns of quality it should be portable, reusable, to modulate and scalable.

The software becomes a support tool that allows to optimize and to speed up the processes that at the moment are managed in Monitoría Computer science Secretary and Inventory dependences, of the University of Nariño Extensión Ipiales.

The methodology applied for the development: the Object-Oriented Programming, the construction of Software for Components and the architecture J2EE, they allow to create reliable systems that are flexible, easy to maintain and able to evolve to complete the necessities of the change.



## INTRODUCCION

El perfil integral de todo Ingeniero de Sistemas debe estar enfocado en la búsqueda de nuevas tendencias o técnicas para la solución de problemas sin quedarse atrás del avance tecnológico de nuestra sociedad. El análisis, diseño e implementación de software generalmente se basan en los lineamientos de una programación estructurada cuyo desempeño ha tenido buena acogida hasta la actualidad; sin embargo surgen nuevas alternativas que brindan características de portabilidad, reusabilidad, modularidad y escalabilidad; esto se refleja en el desarrollo de un producto final que se pueda acoplar de mejor manera a los diferentes cambios tecnológicos.

El proyecto busca generar un sistema de información a través del cual se pretende optimizar y agilizar los procesos que actualmente se llevan de forma manual en la Universidad de Nariño Extensión Ipiales tales como: Administración de Reservas de Internet, Computadores, Audiovisuales, Aulas de Informática, Citas Médicas, Citas Odontológicas, Citas Psicológicas, Administración de Datos de Usuario (estudiantes, docentes, personal operativo y administrativo), Administración de Caja Menor, Administración de Computadores y sus Componentes, y Administración de Inventario.

Para el modelamiento del sistema se ha optado por utilizar el Lenguaje Unificado de Modelado (UML), el cual capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

Para aprovechar las características de este lenguaje se aplicará la Metodología Orientada a Objetos que se acopla de mejor manera al enfoque que se le pretende dar al desarrollo del proyecto.

## **1. REFERENTES DEL PROYECTO**

### **1.1 TITULO**

Aplicación de la Arquitectura de Construcción de Software por Componentes y la Metodología Orientada a Objetos en el Desarrollo del Sistema de Gestión del Area de Monitoría Informática, Secretaría e Inventario de la Universidad de Nariño Extensión Ipiales.

### **1.2 MODALIDAD**

Desarrollo de Software

## 2. MARCO REFERENCIAL

El proceso de proyección de la Universidad de Nariño Extensión Ipiales, inicia a partir de 1988, surge como respuesta a los requerimientos educativos del departamento y a la necesidad de influir en el desarrollo integral del mismo.

El primero de febrero de 1991 se constituye “La Fundación para la regionalización de la Universidad de Nariño Región Obando – Sede Ipiales” con estatutos y requerimientos legales.

La conformación y puesta en marcha de la sede Ipiales fue un proceso que duro aproximadamente siete años; entre los pronunciamientos oficiales encaminados a afianzar este proceso, cabe destacar:

- Acuerdo 135 de noviembre 9 y 143 de noviembre 19 de 1993, mediante los cuales se autoriza la apertura y ofrecimiento de los programas de Administración de Empresas, Economía e Ingeniería de Sistemas.
- Convenio Interinstitucional suscrito entre el Municipio de Ipiales y la Universidad de Nariño para la extensión o seccional de la misma en la ciudad de Ipiales, firmado el 27 de octubre de 1994.

En términos generales la Universidad de Nariño, mediante estos convenios adquiere las obligaciones de: implementar programas de pregrado, ejercer la coordinación académica, mantener su cuerpo docente con las más altas calidades académicas y culturales.

Por su parte, el municipio adquiere las obligaciones de: aportar una cantidad de dinero previamente establecida para cofinanciar los programas que entren en funcionamiento, proporcionar la planta física para el normal desarrollo de los programas establecidos.

Actualmente, la universidad de Nariño Extensión Ipiales se establece en las instalaciones ubicadas en el barrio La Floresta, ofreciendo a la comunidad una

educación integral que busca formar profesionales capaces de liderar el desarrollo económico y social de su entorno. Dado el impacto que ha tenido la Universidad en la región, ahora se cuenta con los programas de Ingeniería de Sistemas, Ingeniería Civil, Comercio Internacional y Mercadeo, Administración de Empresas, Economía, Psicología, Ingeniería Electrónica, Diseño Gráfico e Ingeniería Agroindustrial.

## **2.1 ESTADO DEL ARTE**

Debido a la existencia de diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero con distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de dichas herramientas, además de pugnas entre enfoques. UML surge como un estándar para el modelamiento de sistemas de software principalmente, pero con posibilidades de ser aplicado a todo tipo de proyectos.

En la actualidad UML es una herramienta capaz de modelar no sólo sistemas de software sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos (OO).

## **2.2 MARCO HISTORICO**

Los procesos que aborda el trabajo en las diferentes dependencias de la Universidad de Nariño Extensión Ipiales hasta la actualidad se han llevado a cabo de forma manual o con la ayuda de herramientas de software existentes como el paquete de office, no diseñadas para cubrir las necesidades específicas de la extensión.

La investigación realizada permite asegurar que no existe un precedente de un sistema de información o software que se haya diseñado para dar soporte a los procesos que abarca el proyecto tales como: Administración de Reservas de Internet, Computadores, Audiovisuales, Aulas de Informática, Citas Médicas, Citas Odontológicas, Citas Psicológicas, Administración de Datos de Usuario(estudiantes, docentes, personal operativo y administrativo), Administración de Caja Menor, Administración de Computadores y sus Componentes, y Administración de Inventario.

### **3. PLANTEAMIENTO DEL PROBLEMA**

#### **3.1 DESCRIPCION DEL PROBLEMA**

Al analizar el entorno de la Universidad de Nariño extensión Ipiales y la proyección social y educativa que ha tenido en la región, se puede observar, particularmente el ingreso de un gran número de nuevos estudiantes a la institución. Desde el punto de vista académico y administrativo de la Extensión, este incremento determina el manejo de un volumen de información cada vez más amplio, que proporcionalmente requiere de un procesamiento mayor que solvante las necesidades de información que se generan en el día a día.

Realizando un análisis más detallado, a continuación se mencionará algunas debilidades referidas a la actual administración de la información; dichas debilidades ocasionan un inadecuado rendimiento en el desarrollo de los procesos concernientes a monitoría, secretaría y manejo de inventario de la Universidad de Nariño Extensión Ipiales:

- No hay integración de los procesos que se llevan a cabo en las diferentes dependencias de la Universidad de Nariño Extensión Ipiales, generando islas de Información.
- Carencia de una Base de Datos estructurada y consistente que le permita a los diferentes usuarios que interactúan con algún proceso obtener resultados ágiles y oportunos.
- La Universidad de Nariño Extensión Ipiales carece de un software desarrollado a la medida, que proporcione las herramientas adecuadas para la administración de la información, y que, fácilmente se adecue a los nuevos requerimientos que se presenten por la expansión de la sede en sus diferentes dependencias.
- Desarrollo de procesos de forma manual (Reservas Informáticas y Administración de Inventario), ocasionando un acceso lento y costoso a la información, debido al alto consumo de recursos de impresión, papelería, etc.

- Falta de un registro detallado que permita llevar un control periódico y adecuado de los recursos físicos, humanos y financieros, como también de los servicios informáticos (Internet, audiovisuales, computadores) y de salud (médico, odontológico, psicológico) ofrecidos por la Universidad de Nariño Extensión Ipiales.
- Limitación en el momento de toma de decisiones que requieran de la información en forma veraz y oportuna.
- Regular aprovechamiento de recursos informáticos en el manejo de los diferentes servicios por parte de los estudiantes.

### **3.2 FORMULACION DEL PROBLEMA**

¿De que manera se pueden optimizar los procesos de monitoría informática, secretaría e inventario de la Universidad de Nariño Extensión Ipiales, de tal forma que la información se maneje de forma veraz y oportuna?

### **3.3 SISTEMATIZACION DEL PROBLEMA**

- ¿Cómo ofrecer la solución a la necesidad de un sistema de información escalable, portable y de bajo costo que se ajuste a los patrones de calidad del software?
- ¿Cómo llevar un control de las existencias y el estado de los diferentes elementos y recursos con los que cuenta la Universidad de Nariño Extensión Ipiales?
- ¿Cómo brindar la información personal de los usuarios vinculados a la Universidad de Nariño Extensión Ipiales de manera rápida y oportuna?
- ¿Cómo ofrecer a los usuarios información actualizada concerniente a los diferentes servicios que le ofrece la Universidad de Nariño Extensión Ipiales?

- ¿Cómo lograr un desarrollo eficiente de los procesos para asegurar una óptima utilización de los recursos que en ellos se utilizan?

#### **4. JUSTIFICACION**

El sistema de información a desarrollar está dirigido a automatizar e integrar los diferentes procesos que se llevan a cabo en el área administrativa y operativa de la Universidad de Nariño Extensión Ipiales, incluyendo el manejo de monitoría, secretaria e inventario. Uno de los aspectos fundamentales del software radica en la metodología aplicada para su desarrollo: la Programación Orientada a Objetos (OOP), la construcción de Software por Componentes y el Ambiente Distribuido de las Aplicaciones, las cuales ofrecen nuevas y mejores alternativas para el desarrollo de software. Se pretende aprovechar las características distribuidas inherentes a la arquitectura a utilizar (J2EE), la cual permite llevar a cabo una Distribución Lógica de Componentes capaces de atender las solicitudes de los clientes y dar respuestas por sí solos, como también interactuando con otros componentes, algunas veces residentes en contenedores y servidores diferentes; esto permite distribuir procesamiento y atención favoreciendo el rendimiento del sistema. Físicamente, en principio la aplicación estará distribuida en dos servidores; uno de ellos será el servidor de Base de Datos y el otro será el servidor Web.

La elección de esta metodología busca construir un sistema que se ajuste a los patrones de calidad del software, escalabilidad, portabilidad, extensibilidad y bajo costo. Acompañado de técnicas de modelado (Lenguaje Unificado de Modelado) para conceptualizar mejor el mundo en el cual se desarrollará la aplicación.

De acuerdo con lo anterior, se ha propuesto construir un software que satisfaga las necesidades de manejo de Información existentes en la Universidad de Nariño Extensión Ipiales. La construcción de este software exige en los desarrolladores un cambio de filosofía en el momento de analizar, diseñar y desarrollar aplicaciones por lo que es necesario explorar otras metodologías de desarrollo de software diferentes a las tradicionales técnicas de Programación Estructurada ya utilizadas.



## 5. ALCANCE Y DELIMITACION DEL SISTEMA

De manera general el sistema está compuesto por cuatro dependencias básicas a saber, coordinación académica, secretaría, monitoría informática y biblioteca. Las dependencias de coordinación académica y biblioteca no se tendrán en cuenta en el estudio y desarrollo del sistema planteado, debido a que el módulo de biblioteca ya fue desarrollado y los procesos referentes a coordinación académica son manejados exclusivamente en la sede principal de Pasto. Además de cubrir el área de monitoría informática y secretaría, se abarcará también la administración del inventario de la extensión para un adecuado control de los recursos físicos y computacionales existentes.

El proyecto propuesto abarcará las etapas de especificación de los requerimientos, análisis, diseño, desarrollo e implementación del software; esta última se llevará a cabo siempre y cuando la Universidad de Nariño Extensión Ipiales cuente con los recursos financieros necesarios para la adecuación óptima del sistema propuesto, en caso de que no se den estas condiciones la implementación se llevará a cabo únicamente para la dependencia de Monitoría ya que esta cuenta con la infraestructura apropiada.

El producto final que se desarrollará estará en capacidad de brindar los siguientes beneficios:

- Integración de la Información manejada en las dependencias de monitoría, secretaría y administración de inventario.
- Sistematización de los diferentes procesos manuales que se llevan a cabo en las áreas de Monitoría Informática, Secretaría e Inventario de la Universidad de Nariño Extensión Ipiales tales como: Administración de Reservas de Internet, Computadores, Audiovisuales, Aulas de Informática, Citas Médicas, Citas Odontológicas, Citas Psicológicas, Administración de Datos de Usuario(estudiantes, docentes, personal operativo y administrativo), Administración de Caja Menor, Administración de Computadores y sus Componentes, y Administración de Inventario.

- Gestión rápida y oportuna de las diferentes solicitudes realizadas por los usuarios de los servicios informáticos y de salud tales como: reservas de internet, audiovisuales, computadores, citas médicas, psicológicas y odontológicas.
- Consolidación de un inventario detallado de los elementos existentes en la Universidad de Nariño Extensión Ipiales.
- Optimización en la utilización de recursos de oficina, informáticos y financieros.
- Extensibilidad, portabilidad y escalabilidad del software.

Es importante destacar que se tiene gran parte de conocimiento acerca de la forma en que se maneja la información actualmente y de los problemas existentes sobre todo en el área de monitoría informática. Partiendo de esto se buscará obtener un óptimo desempeño de las diferentes funciones que se desarrollan en la Universidad de Nariño Extensión Ipiales.

## **6. OBJETIVOS**

### **6.1 OBJETIVO GENERAL**

Modelar y Desarrollar un Software basado en el Análisis Orientado a Objetos para la administración de la información de las dependencias de Monitoría Informática, Secretaría y control de Inventario de la Universidad de Nariño Extensión Ipiales.

### **6.2 OBJETIVOS ESPECIFICOS**

- Construir un sistema que se ajuste a los patrones de calidad del software, escalabilidad, portabilidad y bajo costo. Acompañado de técnicas de modelado para conceptualizar mejor el mundo para el cual se desarrollará la aplicación.
- Administrar de forma detallada y veraz las existencias y el estado de los diferentes elementos y recursos con los que cuenta la Universidad de Nariño Extensión Ipiales.
- Permitir un acceso rápido y oportuno a la información personal de los usuarios (estudiantes, docentes, personal operativo y administrativo) vinculados a la Universidad de Nariño Extensión Ipiales a través de interfaces gráficas de fácil manejo.
- Permitir que el usuario interactúe con el sistema consultando la información actualizada concerniente a los diferentes servicios que le ofrece la Universidad de Nariño Extensión Ipiales tales como: Servicios Informáticos (internet, audiovisuales, computadores, aulas de informática), Servicios de Salud(médico, odontológico, psicológico).
- Optimizar el rendimiento de los procesos y los recursos que en ellos se utilizan.

## **7. MARCO TEORICO**

### **7.1 METODOLOGIA ORIENTADO A OBJETOS**

La ingeniería del software orientada a objetos se ha convertido en el paradigma de elección para muchos desarrolladores de software y para un creciente número de sistemas de información.

Las tecnologías de objetos llevan a reutilizar, y la reutilización (de componente de software) lleva a un desarrollo de software más rápido y a programas de mejor calidad. El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente poco acoplada. Esto lleva a menores efectos colaterales cuando se deben hacer cambios. Los sistemas orientados a objetos son más fáciles de adaptar y más fácilmente escalables (pueden crearse grandes sistemas ensamblando subsistemas reutilizables).

El análisis orientado a objetos y su diseño se enfoca en los objetos. Los objetos tienen ciertos comportamientos y atributos que determinan la manera en que interactúan y funcionan. La programación orientada a objetos ayuda a los desarrolladores a crear objetos que reflejan escenarios del mundo real.

En un enfoque orientado a objetos, los objetos, por definición, son modulares en su construcción. Esto quiere decir que son entidades completas y, por lo tanto, tienden a ser altamente reutilizables. Las aplicaciones orientadas a objetos se construyen sobre el paso de mensajes entre objetos.

La orientación a objetos fuerza a reconsiderar el pensamiento sobre la sistematización de la información y la forma de percibir el mundo.

La orientación a objetos puede describirse como el conjunto de disciplinas que desarrollan y modelan software que facilitan la construcción de sistemas complejos a partir de componentes.

Las técnicas orientadas a objetos proporcionan mejoras y metodologías para construir sistemas de software complejos a partir de unidades de software modularizadas y reutilizables. Se necesita un nuevo enfoque para construir software en la actualidad. Este nuevo enfoque debe ser capaz de manipular tanto sistemas grandes como pequeños y debe crear sistemas fiables que sean flexibles, mantenibles y capaces de evolucionar para cumplir las necesidades del cambio.

Un objeto es la instancia de una clase. Una clase es la representación abstracta de un concepto en el mundo real, y proporciona la base a partir de la cual creamos instancias de objetos específicos.

La orientación a objetos trata de cubrir las necesidades de los usuarios finales, así como las propias de los desarrolladores de productos software. Las tareas de visualización, especificación, construcción y documentación de sistemas orientados a objetos son ampliamente soportadas por el lenguaje unificado de modelado (UML).

## **7.2 LENGUAJE UNIFICADO DE MODELADO**

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

Se basa en las anteriores especificaciones BOOCH, RUMBAUGH y COAD-YOURDON. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto.

UML es ahora un estándar, no existe otra especificación de diseño orientado a objetos, ya que es el resultado de las tres opciones existentes en el mercado. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

**7.2.1 Artefactos de UML.** En todos los ámbitos de la ingeniería se construyen modelos, en realidad, simplificaciones de la realidad, para comprender mejor el sistema que se va a desarrollar: los arquitectos utilizan y construyen planos (modelos) de los edificios, los grandes diseñadores de coches preparan modelos en sistemas CAD/CAM con todos los detalles y los ingenieros de software deberían igualmente construir modelos de los sistemas software.

Para la construcción de modelos, hay que centrarse en los detalles relevantes mientras se ignoran los demás, por lo cual con un único modelo no se tiene bastante. Varios modelos aportan diferentes vistas de un sistema los cuales ayudan a comprenderlo desde varios frentes. Así, UML recomienda la utilización de varios diagramas para representar las distintas vistas de un sistema.

Los artefactos utilizados para el desarrollo del sistema son:

**7.2.1.1 Casos de Uso.** Un caso de uso es un documento narrativo que describe la secuencia de eventos cuando uno o más actores utilizan un sistema para llevar a cabo un proceso. Un caso de uso cuenta una historia del uso del sistema. Para el desarrollo del sistema propuesto se utiliza la siguiente plantilla para la captura de los respectivos requerimientos funcionales:

<b>IDENTIFICADOR:</b> <i>código (jerarquizado: R1, R1.1, R1.1.1)</i>		<b>NOMBRE:</b> <i>nombre (debe comenzar por un verbo)</i>
<b>TIPO:</b> <i>Necesario / deseable</i>	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> <i>Se utiliza para requerimientos que corresponden a un caso de uso que se introdujo mediante la relación &lt;&lt;include&gt;&gt; o &lt;&lt;extends&gt;&gt;</i>	<b>CRITICO?</b> <i>Sí o no, dependiendo de si es una operación sobre la cual existen requerimientos de eficiencia. Es una de las operaciones que realiza el usuario en el día a día? Es la razón de la existencia del sistema?</i>

<b>PRIORIDAD DE DESARROLLO:</b> <i>Capa, dentro del esquema de desarrollo incremental, en la cual se incluye el requerimiento (fachada, llenado, enfocado, terminado)</i>	<b>ACTORES:</b> <i>Usuarios del requerimiento</i>	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> <i>Formatos, documentos, pantallas, etc. asociados con el requerimiento. Cómo ve el usuario el mundo?</i>
<b>ENTRADA:</b> <i>Información que se necesita para definir y satisfacer el requerimiento</i>	<b>SALIDA:</b> <i>Resultado del requerimiento. A qué situación se llega al terminar el requerimiento?</i>	
<b>DESCRIPCIÓN:</b> <i>Descripción informal del requerimiento</i>		
<b>OBSERVACIONES:</b> <i>Descripción de anotaciones que se quieran resaltar</i>		
<b>CURSO BÁSICO DE EVENTOS</b> <i>Descripción detallada de cada uno de los pasos involucrados en el desarrollo del proceso</i>		
<b>CAMINOS DE EXCEPCION:</b> <i>Descripción de cada situación anormal posible y el comportamiento asociado del sistema</i>		
<b>TRIGGERS:</b> <i>Proceso no visibles para el usuario que se ejecutan automáticamente ante un determinado evento</i>		
<b>Precondición:</b> <i>(condiciones que debe satisfacer la información de entrada o la aplicación antes del requerimiento)</i> <b>Postcondición:</b> <i>(características de la salida en función del estado inicial de la aplicación y del valor de las entradas)</i>		
<b>CRITERIOS DE ACEPTACIÓN</b> <i>Se supondrá por defecto que hay al menos un criterio de aceptación:</i>  <i>Para las entradas válidas (definidas en la precondición), la salida debe ser válida (definida en la poscondición)</i>  <i>Poder verificar que se hizo lo especificado de manera persistente (si es el caso). Esto es, que la información efectivamente se guardó de manera correcta</i>  <i>Poder verificar que se cumplen las restricciones de desempeño, utilización de recursos, seguridad, etc. asociadas al requerimiento (si existen)</i>		

**7.2.1.2 Diagramas de Casos de Uso.** Modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado. Se utiliza para entender el uso del sistema. Muestra el conjunto de casos de uso, actores y sus relaciones.

- **Casos de uso.** Es una operación o tarea específica que se realiza tras una orden o estímulo de un agente externo, puede ser un actor o desde la invocación desde otro caso de uso. Se representa por una elipse, cada caso de uso contiene un nombre, que indique su funcionalidad.
- **Actor.** Un actor es una entidad o usuario que utiliza alguno de los casos de uso del sistema. Se representa mediante una figura humana estilizada acompañado de un nombre significativo.

- **Relaciones.** Entre los elementos de un diagrama de casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas o no entre ellos. Estas relaciones son:

Association: Relación (asociación) entre un actor y un caso de uso que denota la participación del actor en dicho caso de uso.

Extend: Relación de dependencia entre dos casos de uso que denota que un caso de uso es una especialización de otro.

Include: Relación de dependencia entre dos casos de uso que denota la inclusión del comportamiento de un escenario en otro.

**7.2.1.3 Diagramas de Clases.** Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño).

- **Clase.** Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase).

**7.2.1.4 Diagramas de Actividades.** Es un caso especial del diagrama de estados, simplifica el diagrama de estados modelando el comportamiento mediante flujos de actividades. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

**7.2.1.5 Diagrama de Despliegue.** Muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo. Se utiliza para identificar Sistemas de Cooperación.



## **8. METODOLOGIA**

Para el desarrollo del software se aplicará la metodología de Ingeniería del Software modelo de ensamblaje de componentes, el cual incorpora características del paradigma de orientación a objetos, haciendo énfasis en la utilización de clases y objetos que permiten la reutilización de código proporcionando ventajas significativas para el desarrollo del producto final.

Recolección de Información a través de entrevistas, observaciones y experiencias en el manejo de la información de la Universidad de Nariño Extensión Ipiales.

Utilización del Lenguaje de Modelamiento Unificado (UML) para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software, de manera que permita representar la realidad del sistema y su entorno.

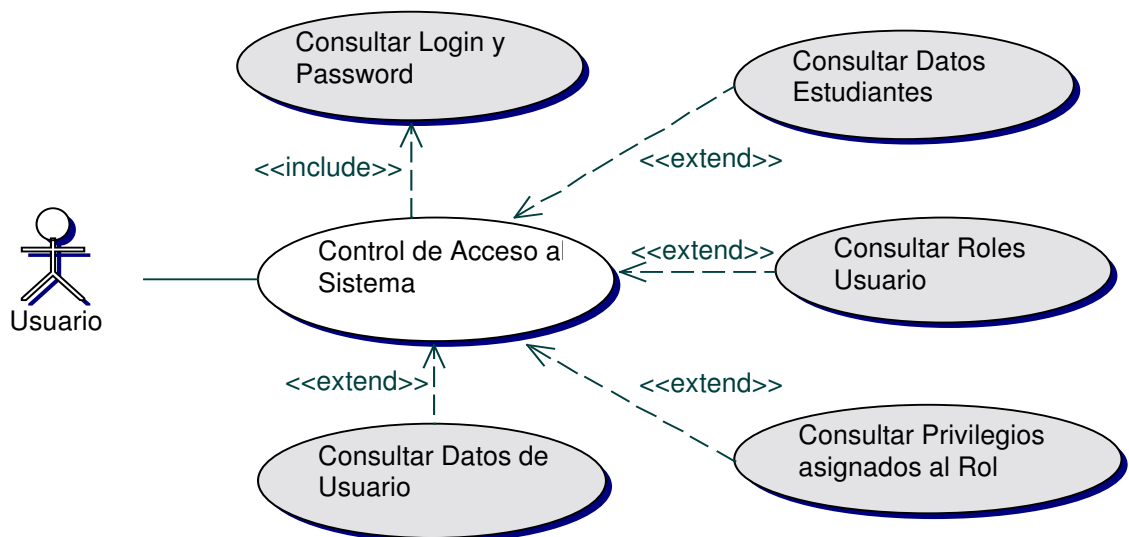
## 9. ANALISIS

### 9.1 CASOS DE USO

#### 9.1.1 Control De Acceso Al Sistema

##### 9.1.1.1 Primera Iteración - Fachada

- Diagrama



- Iteración

<b>IDENTIFICADOR:</b> CASIS1		<b>NOMBRE:</b> Control de Acceso al Sistema
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Login y Password Consultar Datos Estudiantes Consultar Roles Usuario Consultar Privilegios asignados al Rol Consultar Datos de Usuario	<b>CRITICO?</b> Si

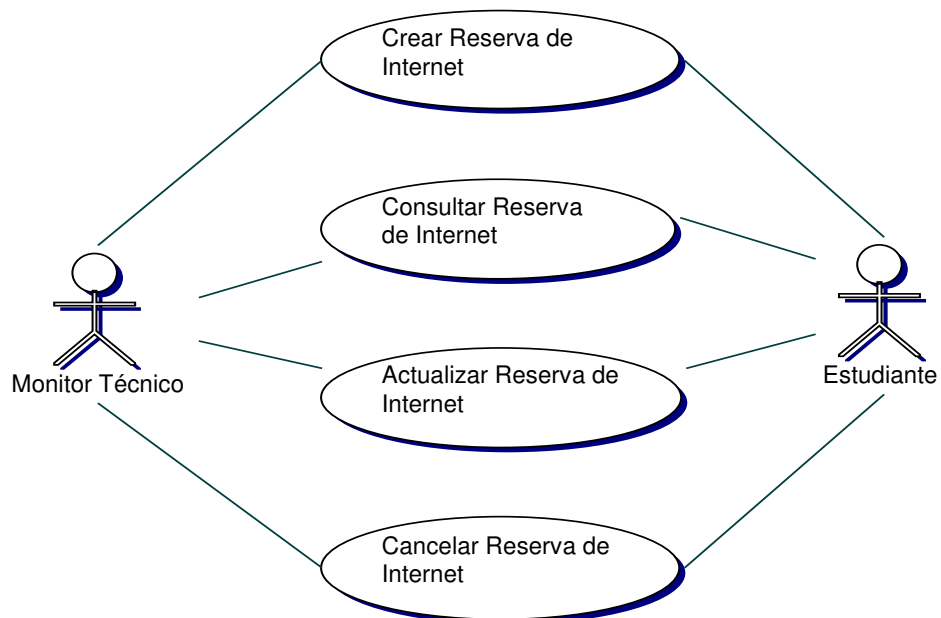
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Login, password.	<b>SALIDA:</b> Usuario logeado o proceso no efectuado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo controlar el acceso de los usuarios a los módulos del sistema. Los usuarios estudiantes podrán acceder a la administración de sus reservas de internet, de computadores, de audiovisuales y citas médicas, así como también modificar su password de acceso. Los demás usuarios podrán acceder a la administración de los módulos privados soportados por el sistema de acuerdo a los privilegios que tengan asignados.		
<b>OBSERVACIONES:</b> Para poder acceder a los módulos del sistema el usuario debe encontrarse registrado con su login y password. Para poder acceder a los módulos privados el usuario debe tener asignado un rol con sus respectivos privilegios. Si un usuario tiene asignado mas de un rol, se le permitirá seleccionar el rol con el que desea ingresar al sistema.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acceso al sistema por el usuario Estudiante</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. El actor estudiante accede a la página principal del sistema.		2. Despliega la información pública y le permite digitar su login y password para acceder al sistema.
3. El actor estudiante digita su login y password.		5. Verifica que el login y password pertenezcan a un usuario registrado en el sistema. <b>(b)</b>
4. El actor estudiante da la orden para acceder al sistema.		6. Verifica que el usuario sea un estudiante de la universidad. <b>(c)</b>
		7. Carga los privilegios asignados al rol estudiante.
		8. Despliega los servicios a los que tiene acceso el estudiante.
<b>Acceso al sistema por un usuario</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. El usuario ingresa en la página de acceso a los módulos privados del sistema.		2. Solicita el login y password del usuario.
3. Suministra los datos solicitados.		5. Verifica que el login y password pertenezcan a un usuario registrado en el sistema. <b>(b)</b>
4. El usuario da la orden para acceder al sistema. <b>(a)</b>		6. Verifica que el usuario tenga asignado un rol que le permita acceder al sistema. <b>(d)</b>
		7. Verifica si el usuario tiene mas de un rol asignado. <b>(e)</b>
		8. Despliega el listado de roles asignados al usuario.
9. El usuario selecciona el rol deseado.		11. Carga los privilegios asignados al rol.
10. El usuario da la orden para ingresar al sistema con ese rol. <b>(a)</b>		12. Despliega los módulos a los que tiene acceso el usuario.

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no ingresar al sistema: El sistema cancela el proceso</p> <p>(b) El login y password es incorrecto el actor no es un usuario del sistema: El sistema le indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(c) El usuario no es un estudiante de la universidad: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(d) El usuario no tiene asignado roles que le permitan acceder al sistema: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(e) El usuario solo tiene un rol asignado. El sistema continua el proceso en el punto 11 del curso básico de eventos de acceso al sistema por el usuario.</p> <ul style="list-style-type: none"> <li>• El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: El usuario debe estar registrado con su login y password El usuario debe tener asignado un rol.</p> <p>Postcondición: Control Eficiente del acceso de los usuarios al sistema.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra controlar el acceso de los usuarios al sistema de forma segura. Manejo adecuado de los roles asignados a los usuarios permitiendo el acceso solo a los privilegios asignados a cada rol. Correcta autenticación de usuarios para acceder a los módulos públicos y privados soportados por el sistema.</p>

## 9.1.2 Administración de Reservas de Internet

### 9.1.2.1 Primera Iteración - Fachada

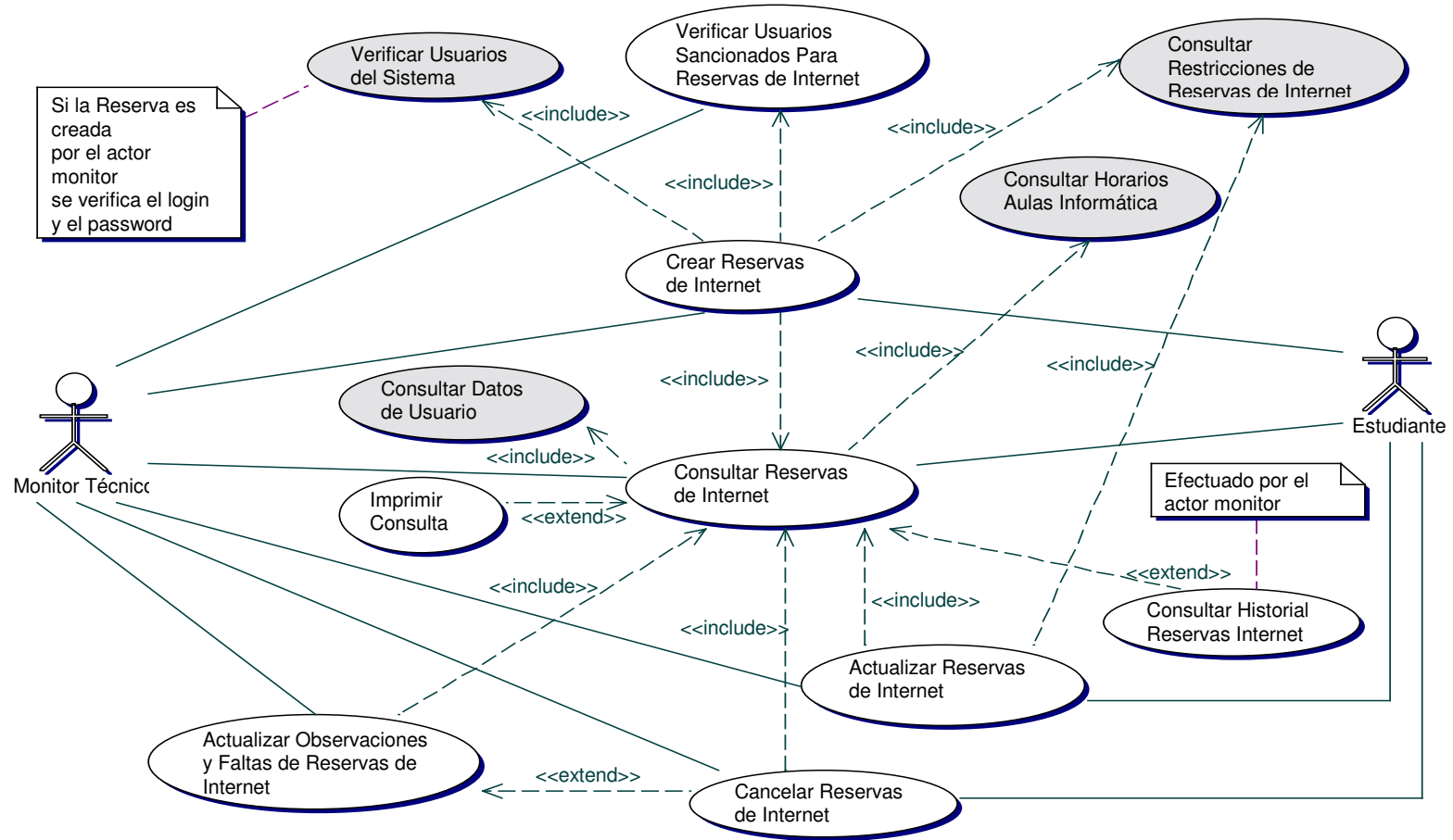
- Diagrama



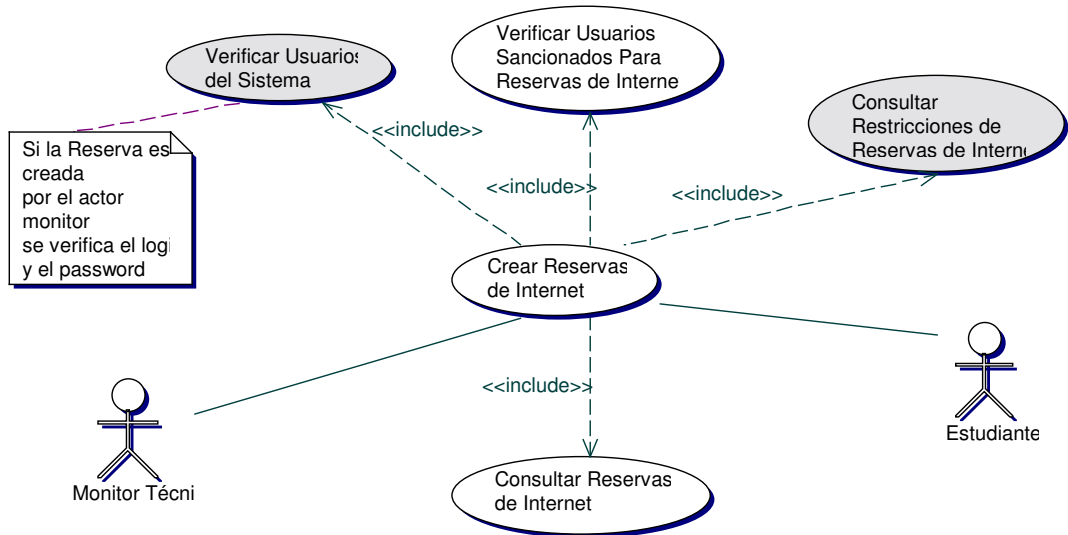
- Iteración

<b>IDENTIFICADOR:</b> ARII		<b>NOMBRE:</b> Administración de Reservas de Internet	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear reservas de Internet Consultar reservas de Internet Verificar Usuarios Sancionados para Reservas de Internet Actualizar reservas de Internet Actualizar Observaciones y Faltas de Reservas de Internet Eliminar reservas de Internet		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las reservas de internet. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y cancelación.			
<b>OBSERVACIONES:</b> El actor monitor puede administrar las reservas de Internet sin ninguna restricción. El usuario deberá encontrarse registrado en el sistema para poder hacer uso del servicio. El período de tiempo para la administración de una reserva será determinado por el administrador y estará comprendido entre la fecha actual y el intervalo de tiempo establecido.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de las reservas de Internet.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de las reservas de internet. La información en la base de datos se maneja de forma persistente y consistente. Aprovechamiento eficiente de equipos y horarios asignados para las franjas de internet. Administración de forma local y remota de reservas para franjas de Internet, brindando mayor acceso a los usuarios en el uso del servicio.			

### 9.1.2.2 Segunda Iteración – Llenado (Diagrama General)



- **Diagrama Crear Reservas de Internet**



- **Iteración Crear Reservas de Internet**

<b>IDENTIFICADOR:</b> ARI.1		<b>NOMBRE:</b> Crear Reservas de Internet	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Verificar usuarios del sistema Verificar Usuarios Sancionados para Reservas de Internet Consultar Restricciones de Reservas de Internet Consultar reservas de Internet		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Estudiante. Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, fecha, horario, aula y equipo.		<b>SALIDA:</b> Reserva registrada o no efectuada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar las reservas de Internet. La creación de una reserva de internet puede ser efectuada por el actor iniciador interactuando directamente con el sistema desde una terminal o a través del monitor, el cual recibe los datos y los suministra al sistema. En ambos casos el actor iniciador recibirá la respuesta a su solicitud. De acuerdo a los datos ingresados, el sistema estará en capacidad de generar información asociada a dichos datos, como son: nombre del usuario solicitante, horario, aulas de informática y los equipos disponibles de acuerdo a la fecha seleccionada.			
<b>OBSERVACIONES:</b> Cuando el proceso es realizado por el actor iniciador, el login y password validados inicialmente, permitirán al sistema obtener automáticamente los datos personales del usuario. Un usuario tiene derecho de crear un determinado número de reservas por día. Si la reserva se realiza a través del monitor se solicitará el código estudiantil o la identificación del usuario solicitante para obtener sus datos.			

## **CURSO BÁSICO DE EVENTOS**

### **Creación efectuada por el actor iniciador**

#### **Acción de los actores**

1. El actor iniciador selecciona la opción crear reserva de internet.
4. Selecciona la fecha para su reserva
6. Selecciona la franja y el aula para su reserva
8. El actor iniciador selecciona el equipo para su reserva.
9. El actor iniciador selecciona la opción registrar reserva. **(a)**

#### **Respuesta del Sistema**

2. Verifica si el actor solicitante no se encuentra sancionado. **(d)**
3. Despliega una pantalla presentando al usuario la fecha, franjas, aulas y equipos disponibles para su reserva.
5. Actualiza el listado de las franjas disponibles en la fecha seleccionada.
7. Actualiza el listado de los equipos disponibles en la fecha, franja y aula seleccionada
10. Verifica si el actor iniciador no tiene realizada una reserva en la fecha y franja seleccionada. **(b)**
11. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(e)**
12. Registra la reserva actualizando la base de datos con la información suministrada.
13. Muestra un mensaje indicando que la reserva se realizó con éxito.

### **Creación efectuada a través del actor monitor**

#### **Acción de los actores**

1. El actor iniciador solicita al actor monitor la creación de una reserva de Internet.
2. El actor monitor selecciona la opción crear reserva de Internet.
4. El actor monitor digita el código del usuario solicitante. **(a)**
8. El actor monitor selecciona la fecha para la reserva.
10. El actor monitor selecciona la franja y aula para la reserva.
12. El actor monitor selecciona el equipo para la reserva.
13. El actor monitor selecciona la opción registrar reserva. **(a)**

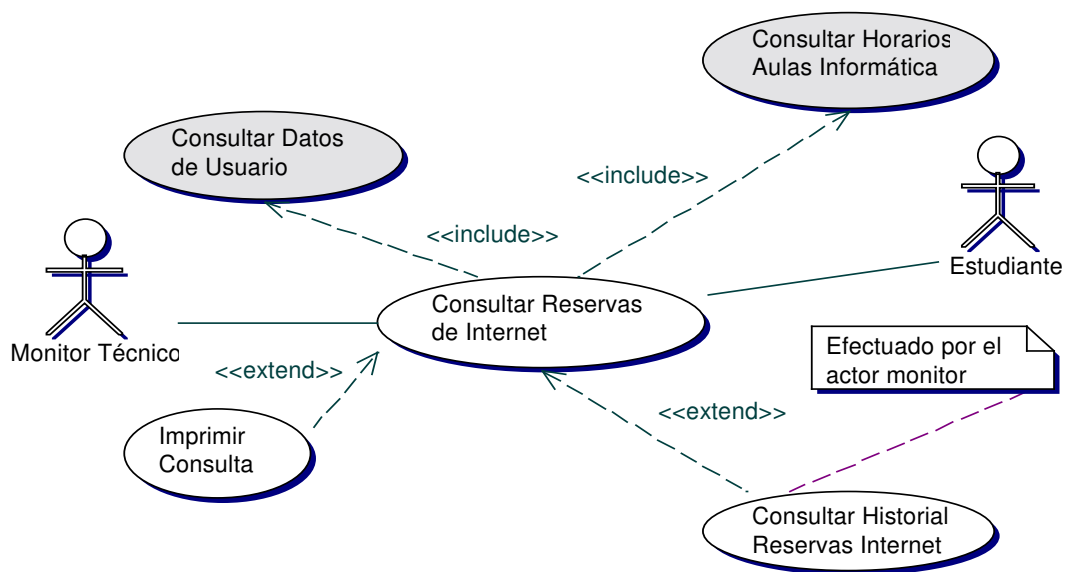
#### **Respuesta del Sistema**

3. Solicita el código del usuario.
5. Verifica si el actor solicitante está registrado como usuario del sistema. **(c)**
6. Verifica si el actor solicitante no se encuentra sancionado. **(d)**
7. Despliega una pantalla presentando el nombre completo del usuario y permite escoger la fecha, franja y aula para la reserva.
9. Actualiza el listado de las franjas disponibles en la fecha seleccionada.
11. Actualiza el listado de los equipos disponibles en la fecha, franja y aula seleccionada
14. Verifica si el actor iniciador no tiene realizada una reserva en la fecha y franja seleccionada. **(b)**
15. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(e)**
16. Registra la reserva actualizando la base de datos con la información suministrada.
17. Muestra un mensaje con los datos de la reserva indicando que la reserva se realizó con éxito



<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no registrar la reserva: El sistema cancela el proceso</p> <p>(b) El actor iniciador ya tiene una reserva en la fecha y franja seleccionada: El sistema le indica este suceso a través de un mensaje.</p> <p>(c) El actor iniciador no se encuentra registrado como usuario del sistema: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(d) El actor iniciador se encuentra sancionado: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(e) Los datos seleccionados ya no se encuentran disponibles. El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p><b>TRIGGERS:</b></p> <p>Verificación de usuario: Este evento se ejecutará para determinar si el actor iniciador se encuentra registrado como usuario del sistema y obtener sus datos.</p> <p>Verificación del periodo de sanción: Si el usuario se encuentra sancionado, el sistema verificará si ya cumplió su periodo de sanción, en caso de que así sea lo habilitará automáticamente para usar el servicio, si no muestra un mensaje informándole que no puede efectuar el proceso.</p> <p>Validar que no exista un registro del usuario en la misma fecha y franja.</p> <p>Confirmación de disponibilidad: Se verificará que los datos seleccionados para la reserva aún se encuentren disponibles.</p>
<p>Precondición: No se debe sobrepasar el número de reservas permitidas para el usuario en el día. El usuario no debe estar sancionado.</p> <p>Postcondición: Registro eficiente de las reservas de Internet. Control eficiente del uso de las reservas de internet.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra crear una reserva de Internet controlando los requerimientos mínimos para poder efectuar el proceso de forma satisfactoria.</p> <p>La información de la base de datos se almacena de forma persistente y consistente.</p>

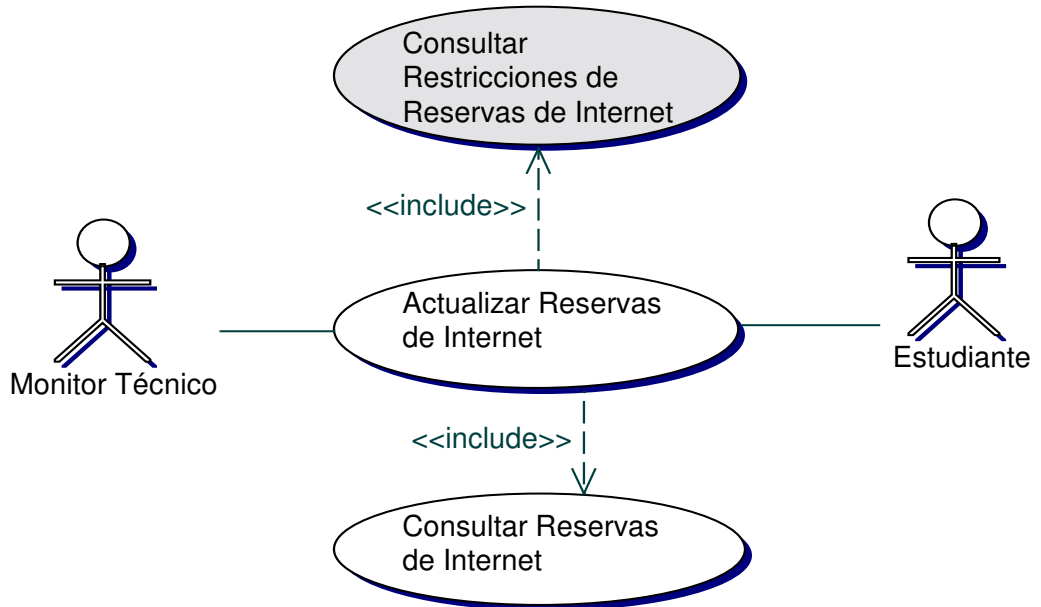
- Diagrama Consultar Reservas de Internet**



• Iteración Consultar Reservas de Internet

<b>IDENTIFICADOR:</b> ARI1.2		<b>NOMBRE:</b> Consultar Reservas de Internet	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario Consultar Horarios Aulas Informática Imprimir Consulta Consultar Historial Reservas Internet		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código de usuario, fecha, franja, aula y número de equipo		<b>SALIDA:</b> Listado de Reservas de Internet	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las reservas de internet en la fecha indicada por el usuario. La consulta de reservas puede ser realizada por cualquiera de los actores involucrados en este caso de uso. Se desplegará la información de las reservas existentes y de las franjas disponibles en el día seleccionado por el actor. La información desplegada incluirá la fecha, nombre del usuario, el número del equipo, el aula, la hora, programa y el semestre. Adicionalmente existirá una consulta que permita visualizar el historial de todas las reservas efectuadas.			
<b>OBSERVACIONES:</b> El historial de las reservas efectuadas podrá ser accedido únicamente por el monitor técnico del aula de informática. La información de entrada dependerá del tipo de consulta que el monitor desee hacer (por usuario, por fecha y por equipo). Para el actor Estudiante, el resultado de la consulta mostrará sus propias reservas.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Consulta efectuada por el monitor</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor iniciador selecciona la opción consultar reservas de Internet.		2. Despliega un menú presentando las opciones para su consulta (por franja, por usuario, por historial).	
3. El actor monitor selecciona el criterio para su consulta. (a)		4. Solicita los datos necesarios para efectuar la consulta.	
5. El actor monitor suministra los datos solicitados. (a)		6. Se despliega un listado de todos los equipos reservados indicando los datos del equipo, horario y datos del usuario, así como también los equipos que se encuentran disponibles.	
<b>Consulta efectuada por el estudiante</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor estudiante selecciona la opción consultar reservas de Internet.		2. Solicita la fecha para la consulta.	
3. El actor iniciador selecciona la fecha deseada. (a)		4. Se despliega un listado de todas sus reservas.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente.			
<b>Postcondición:</b> Información oportuna de las reservas de Internet existentes. Información consolidada de las reservas existentes hasta el momento.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar información de reservas de Internet de forma veraz y oportuna.			
Los reportes generados permiten a la administración la toma de decisiones con respecto al uso del servicio.			

- Diagrama Actualizar Reservas de Internet



- Iteración Actualizar Reservas de Internet

<b>IDENTIFICADOR:</b> ARI1.3		<b>NOMBRE:</b> Actualizar Reservas de Internet	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas de Internet Consultar Restricciones de Reservas de Internet		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática , Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, franja, aula, equipo.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar una determinada reserva de Internet. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> El actor monitor tendrá acceso a la actualización de todas las reservas vigentes en una fecha determinada.			

## **CURSO BÁSICO DE EVENTOS**

### **Actualización efectuada por el actor monitor**

#### **Acción de los actores**

1. El actor monitor selecciona la opción actualizar reservas de Internet.
3. El actor monitor suministra los datos solicitados. **(a)**
5. El actor monitor selecciona la reserva a modificar.
6. Da la orden para mostrar las opciones para actualizar la reserva. **(a)**
8. Selecciona los nuevos datos de la reserva
9. El actor monitor selecciona la opción actualizar reserva. **(a)**
11. El actor monitor confirma su decisión.

#### **Respuesta del Sistema**

2. Solicita la fecha, franja y aula.
4. Muestra un listado con la información correspondiente a todas las reservas en la fecha, franja y aula seleccionada como son nombre, programa, semestre y equipo.
7. Muestra las opciones para actualizar la reserva como son fecha, franja, aula y equipo.
10. Se pide confirmación para llevar a cabo el proceso.
12. Verifica si el usuario no tiene realizada una reserva en la fecha y franja modificada. **(b)**
13. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(c)**
14. Actualiza la reserva modificando la base de datos con la información suministrada.
15. Muestra un mensaje indicando que la reserva se actualizó con éxito

### **Actualización efectuada por el actor estudiante**

#### **Acción de los actores**

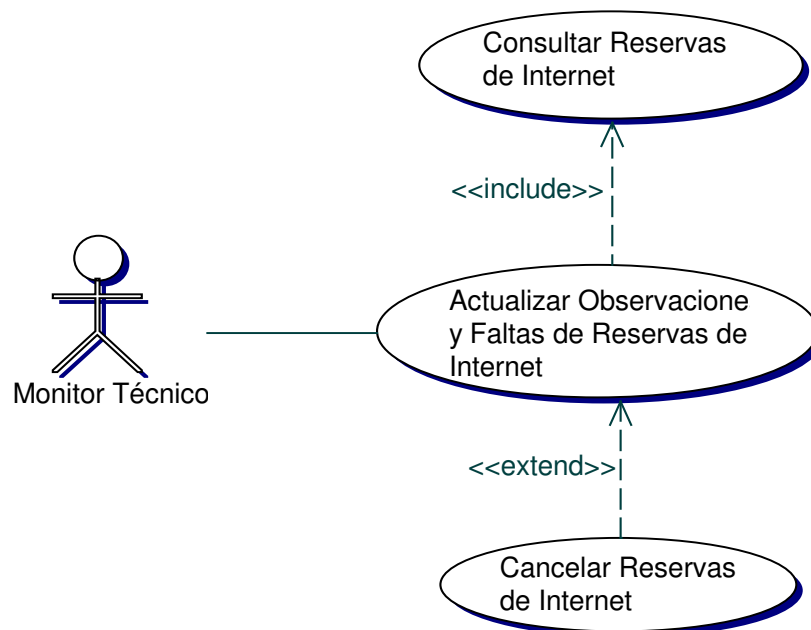
1. EL actor estudiante selecciona la opción actualizar reservas de Internet.
3. El actor estudiante selecciona la fecha. **(a)**
5. El actor estudiante selecciona la reserva a modificar.
6. Da la orden para mostrar las opciones para actualizar la reserva. **(a)**
8. Selecciona los nuevos datos de la reserva
9. El actor estudiante selecciona la opción actualizar reserva. **(a)**
11. El actor monitor confirma su decisión.

#### **Respuesta del Sistema**

2. Solicita la fecha.
4. Muestra un listado con la información correspondiente a sus reservas vigentes en la fecha seleccionada como son franja, equipo y aula.
7. Muestra las opciones para actualizar la reserva como son fecha, franja, aula y equipo.
10. Se pide confirmación para llevar a cabo el proceso.
12. Verifica si el actor estudiante no tiene realizada una reserva en la fecha y franja seleccionada. **(b)**
13. Verifica que los datos seleccionados aun se encuentren disponibles para registrar la reserva. **(c)**
14. Actualiza la reserva modificando la base de datos con la información suministrada.
15. Muestra un mensaje indicando que la reserva se actualizó con éxito.

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.</p> <p>(b) El usuario ya tiene una reserva en la fecha y franja modificada: El sistema le indica este suceso a través de un mensaje y le da la posibilidad de cambiar la fecha.</p> <p>(c) Los datos seleccionados ya no se encuentran disponibles. El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: La información de la base de datos es consistente. El usuario debe tener al menos una reserva creada.</p> <p>Postcondición: Actualización oportuna de las reservas de Internet</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra actualizar la información de reservas de Internet de forma veraz y oportuna.</p> <p>La información de la base de datos se almacena de forma persistente y consistente.</p>

• **Diagrama Actualizar Observaciones y Faltas de Reservas de Internet**

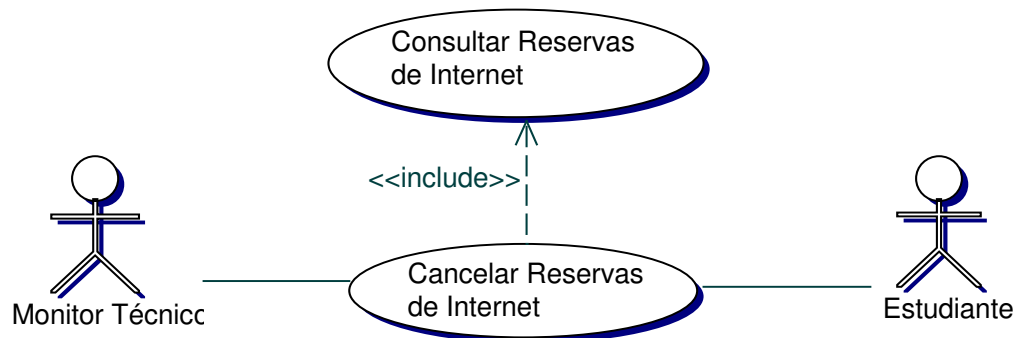


• **Iteración Actualizar Observaciones y Faltas de Reservas de Internet**

<p><b>IDENTIFICADOR:</b> ARI1.4</p>	<p><b>NOMBRE:</b> Actualizar Observaciones y Faltas de Reservas de Internet</p>
---	---

<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas de Internet Cancelar Reservas de Internet	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Fecha, franja, aula, reserva, observaciones y faltas.	<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar las observaciones y las faltas de una determinada reserva. Se pide confirmación para efectuar el proceso.		
<b>OBSERVACIONES:</b> Las observaciones de la reserva se hacen en el caso de que surja alguna eventualidad. El actor monitor será el encargado de establecer cuando el usuario comete una falta cuya acumulación determinará la sanción correspondiente. Se puede considerar como faltas el mal uso de los equipos, la impuntualidad y/o el incumplimiento.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. El actor monitor selecciona la opción de actualizar observaciones y faltas de Reservas de Internet.	2. Solicita la fecha, la franja y el aula.	
3. Selecciona los datos solicitados. (a)	4. Muestra un listado con la información correspondiente a las reservas de la fecha, franja y aula seleccionada como son equipo, nombre, programa y semestre.	
5. El actor monitor selecciona la reserva a modificar.	7. Muestra los datos de la reserva seleccionada y le permite establecer las observaciones y la falta al usuario.	
6. Da la orden para mostrar las opciones para actualizar la reserva. (a)	10. Actualiza la reserva modificando la base de datos con la información suministrada.	
8. El actor monitor modifica los datos correspondientes a observaciones y faltas.	11. Muestra un mensaje indicando que la reserva se actualizó con éxito.	
9. El actor monitor selecciona la opción de actualizar. (a)		
<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.		
<b>TRIGGERS:</b> Aplicar sanción al usuario: El sistema verifica el número de faltas en las que ha incurrido el usuario. Si el número obtenido sobrepasa un límite establecido, el usuario será sancionado durante un lapso de tiempo acordado en el cual no podrá hacer uso del servicio de Internet y se le cancelarán las reservas efectuadas en el periodo correspondiente a la sanción.		
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos una reserva creada.		
<b>Postcondición:</b> Actualización oportuna de observaciones y faltas de las reservas de Internet		
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra actualizar la información de observaciones y faltas de reservas de Internet de forma veraz y oportuna. Se lleva un control del uso de las reservas por parte de los usuarios de tal forma que el sistema permita aplicar sanciones cuando se incumplan las normas. La información de la base de datos se almacena de forma persistente y consistente.		

- Diagrama Cancelar Reservas de Internet

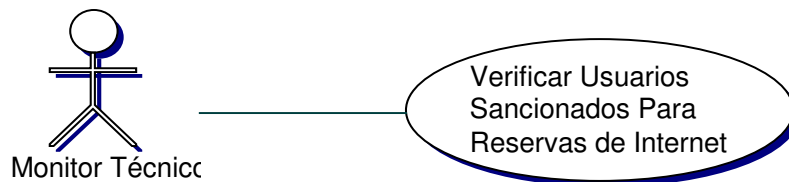


- Iteración Cancelar Reservas de Internet

<b>IDENTIFICADOR:</b> ARI1.5		<b>NOMBRE:</b> Cancelar Reservas de Internet	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas de Internet		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, franja, aula, reserva a cancelar.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo cancelar una determinada reserva de Internet. Se le presentará al actor estudiante un listado de todas sus reservas vigentes en una determinada fecha permitiéndole seleccionar la que desea cancelar. El actor monitor podrá visualizar todas las reservas de una determinada fecha, franja y aula. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> El actor monitor puede cancelar las reservas vigentes de una determinada fecha sin ninguna restricción.			
<b>CURSO BÁSICO DE EVENTOS</b> <i>Eliminación efectuada por el actor monitor</i>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor monitor selecciona la opción de eliminar reservas de Internet.	2. Solicita la fecha, franja y aula..		
3. EL actor monitor selecciona los datos solicitados. (a)	4. Muestra un listado de todas las reservas vigentes.		
5. El actor monitor selecciona la reserva a cancelar.	7. Se pide confirmación para llevar a cabo el proceso.		
6. El actor monitor selecciona la opción cancelar reserva. (a)	9. Cancela la reserva actualizando la base de datos.		
8. El actor monitor confirma su decisión. (a)	10. Muestra un mensaje indicando que la reserva se canceló con éxito.		

<b>Eliminación efectuada por el actor estudiante</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor estudiante selecciona la opción de eliminar reservas de Internet.	2. Solicita la fecha de la reserva..
3. El actor estudiante selecciona la fecha deseada. (a)	4. Muestra un listado de sus reservas vigentes en esa fecha.
5. El actor estudiante selecciona la reserva a cancelar.	
6. El actor estudiante selecciona la opción cancelar reserva. (a)	7. Se pide confirmación para llevar a cabo el proceso.
8. El actor estudiante confirma su decisión. (a)	9. Cancela la reserva actualizando la base de datos.
	10. Muestra un mensaje indicando que la reserva se canceló con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente. El usuario debe tener al menos una reserva creada	
Postcondición: Cancelación satisfactoria de las reservas de Internet existentes.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra cancelar las reservas de Internet de forma consistente.	

- Diagrama Verificar Usuarios Sancionados para Reservas de Internet



- Iteración Verificar Usuarios Sancionados para Reservas de Internet

<b>IDENTIFICADOR:</b> ARI1.6		<b>NOMBRE:</b> Verificar Usuarios Sancionados para Reservas de Internet	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Faltas.		<b>SALIDA:</b> Listado de los usuarios sancionados con faltas para acceder al Servicio de Internet	

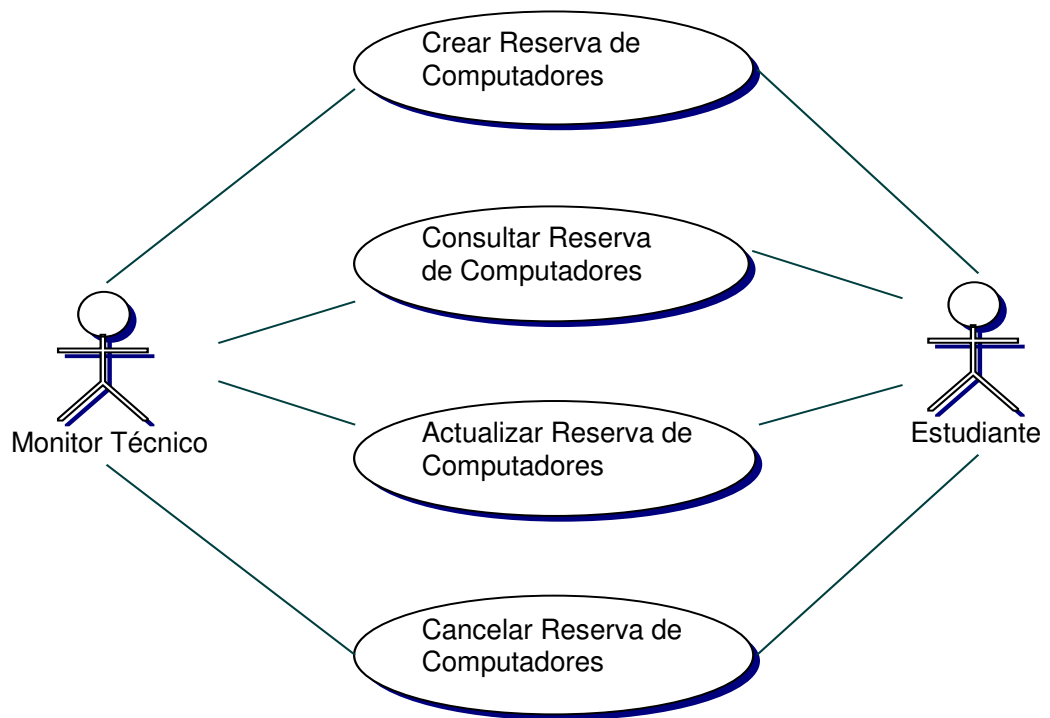


<p><b>DESCRIPCIÓN:</b>  <i>Se satisface este requerimiento permitiendo consultar los usuarios que tengan faltas para el servicio de Internet. Si el número de faltas es igual al máximo permitido el usuario se encuentra sancionado para acceder al servicio de internet durante un periodo de tiempo establecido.</i>  <i>La información desplegada incluirá el nombre del usuario, programa, semestre, la fecha a partir de la cual inicia la sanción y la fecha en la cual termina. Adicionalmente se mostrará el número de faltas que tiene el usuario permitiendo ser modificado por el actor monitor manualmente.</i>  <i>Se pide confirmación para efectuar el proceso de actualización de faltas.</i></p>													
<p><b>OBSERVACIONES:</b> <i>El actor monitor podrá cambiar manualmente el número de faltas para aplicar o quitar una sanción automáticamente.</i></p>													
<p><b>CURSO BÁSICO DE EVENTOS</b></p> <table border="0"> <thead> <tr> <th style="text-align: center;"><i>Acción de los actores</i></th> <th style="text-align: center;"><i>Respuesta del Sistema</i></th> </tr> </thead> <tbody> <tr> <td>1. <i>EL actor monitor selecciona la opción verificar usuarios sancionados para reservas de Internet.</i></td> <td>2. <i>Despliega el listado de todos los usuarios que tengan faltas para el servicio de Internet.</i></td> </tr> <tr> <td>3. <i>El actor monitor modifica el número de faltas de un usuario.</i></td> <td></td> </tr> <tr> <td>4. <i>El actor monitor da la orden para actualizar faltas. (a)</i></td> <td>5. <i>Se pide confirmación para llevar a cabo el proceso.</i></td> </tr> <tr> <td>6. <i>El actor monitor confirma su decisión.</i></td> <td>7. <i>Actualiza las faltas del usuario modificando la base de datos con la información suministrada.</i></td> </tr> <tr> <td></td> <td>8. <i>Muestra un mensaje indicando que los datos se actualizaron con éxito.</i></td> </tr> </tbody> </table>		<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>	1. <i>EL actor monitor selecciona la opción verificar usuarios sancionados para reservas de Internet.</i>	2. <i>Despliega el listado de todos los usuarios que tengan faltas para el servicio de Internet.</i>	3. <i>El actor monitor modifica el número de faltas de un usuario.</i>		4. <i>El actor monitor da la orden para actualizar faltas. (a)</i>	5. <i>Se pide confirmación para llevar a cabo el proceso.</i>	6. <i>El actor monitor confirma su decisión.</i>	7. <i>Actualiza las faltas del usuario modificando la base de datos con la información suministrada.</i>		8. <i>Muestra un mensaje indicando que los datos se actualizaron con éxito.</i>
<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>												
1. <i>EL actor monitor selecciona la opción verificar usuarios sancionados para reservas de Internet.</i>	2. <i>Despliega el listado de todos los usuarios que tengan faltas para el servicio de Internet.</i>												
3. <i>El actor monitor modifica el número de faltas de un usuario.</i>													
4. <i>El actor monitor da la orden para actualizar faltas. (a)</i>	5. <i>Se pide confirmación para llevar a cabo el proceso.</i>												
6. <i>El actor monitor confirma su decisión.</i>	7. <i>Actualiza las faltas del usuario modificando la base de datos con la información suministrada.</i>												
	8. <i>Muestra un mensaje indicando que los datos se actualizaron con éxito.</i>												
<p><b>CAMINOS DE EXCEPCION:</b>  <b>(a)</b> <i>El actor iniciador decide cancelar la consulta: El sistema cancela el proceso.</i>  <ul style="list-style-type: none"> <li>• <i>El sistema no responde. El proceso no se efectúa.</i></li> </ul> </p>													
<p><b>TRIGGERS:</b>  <i>Aplicar o quitar sanción al usuario: El sistema verifica el número de faltas modificado por el actor monitor. Si el número obtenido sobrepasa un límite establecido, el usuario será sancionado durante un lapso de tiempo acordado en el cual no podrá hacer uso del servicio de Internet y se le cancelarán las reservas efectuadas en el periodo correspondiente a la sanción. Si el usuario estaba sancionado y el número de faltas se disminuye, se le habilitará nuevamente para utilizar el servicio.</i></p>													
<p><i>Precondición: El usuario debe tener por lo menos una falta.</i>  <i>Postcondición: Información oportuna de los usuarios sancionados para acceder al servicio de internet.</i>  <i>Actualización de las faltas de un usuario para el servicio de internet.</i></p>													
<p><b>CRITERIOS DE ACEPTACIÓN</b>  <i>Se cumple el requerimiento cuando se logra consultar los usuarios que tienen faltas para el uso del servicio de Internet de forma veraz y oportuna.</i>  <i>El monitor estará en capacidad de aplicar o quitar una sanción a un determinado usuario.</i></p>													

### 9.1.3 Administración de Reservas de Computadores

#### 9.1.3.1 Primera Iteración - Fachada

- Diagrama

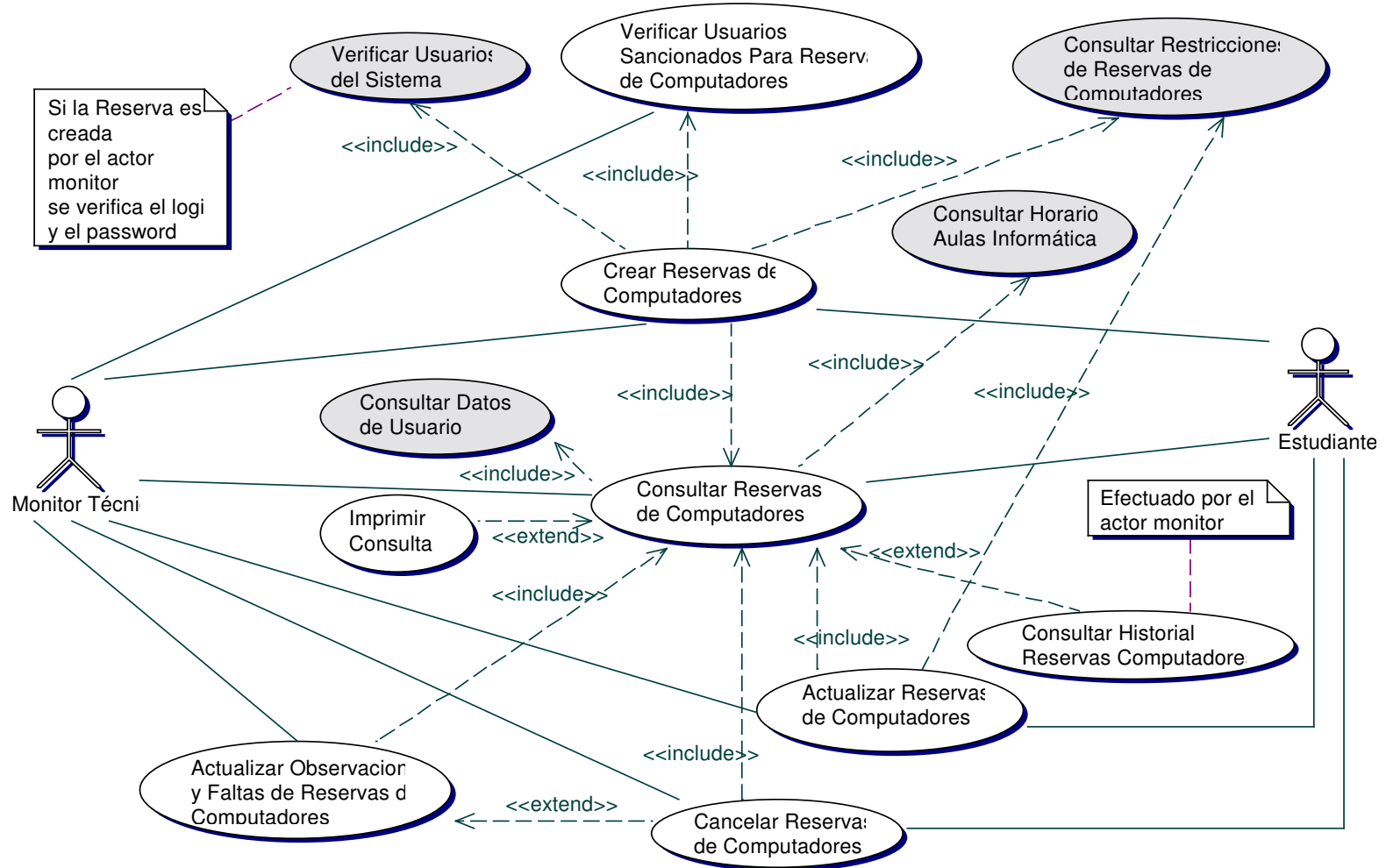


- Iteración

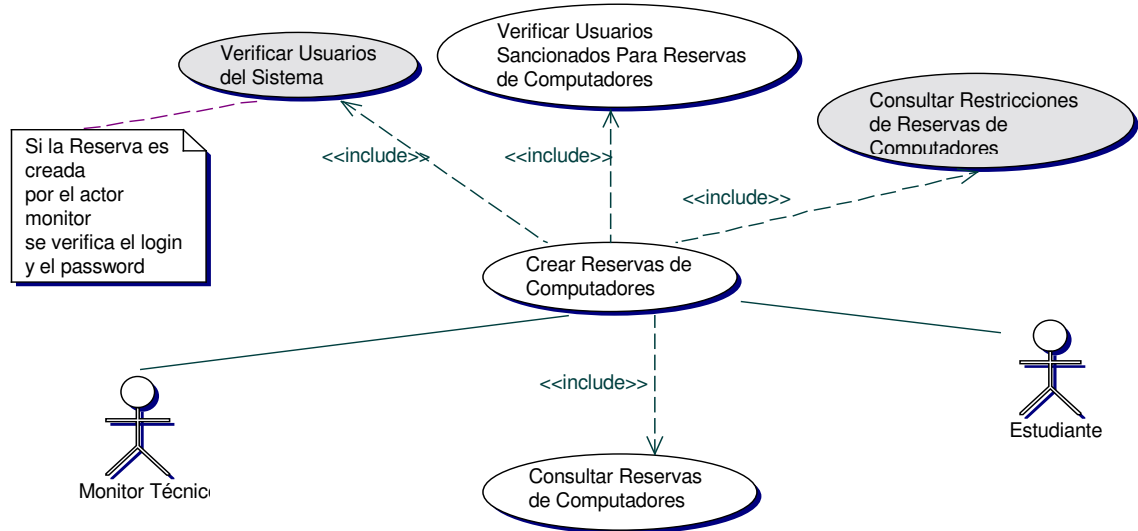
<b>IDENTIFICADOR:</b> ARCI		<b>NOMBRE:</b> Administración de Reservas de Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear reservas de Computadores Consultar reservas de Computadores Verificar Usuarios Sancionados para Reservas de Computadores Actualizar reservas de Computadores Actualizar Observaciones y Faltas de Reservas de Computadores Eliminar reservas de Computadores		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Monitores técnicos del Aula de Informática, Estudiantes.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las reservas de computadores para trabajos varios a los actores estudiantes. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y cancelación.			
<b>OBSERVACIONES:</b> El actor monitor puede administrar las reservas de computadores sin ninguna restricción. El usuario deberá encontrarse registrado en el sistema para poder hacer uso del servicio. El período de tiempo para la administración de una reserva será determinado por el administrador y estará comprendido entre la fecha actual y el intervalo de tiempo establecido.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una			

<i>respuesta del sistema.</i>
<i>Precondición: El actor ha iniciado una sesión en la aplicación.</i>
<i>Postcondición: Administración eficiente de Reservas de Computadores para Trabajos varios.</i>
<b>CRITERIOS DE ACEPTACIÓN</b> <i>Permite llevar una administración sistemática de las reservas de computadores.</i> <i>La información en la base de datos se maneja de forma persistente y consistente.</i> <i>Aprovechamiento eficiente de equipos y horarios asignados para las reservas de computadores.</i> <i>Administración de forma local y remota de reservas para computadores, brindando mayor acceso a los usuarios en el uso del servicio.</i>

### 9.1.3.2 Segunda Iteración – Llenado (Diagrama General)



- **Diagrama Crear Reservas de Computadores**



- **Iteración Crear Reservas de Computadores**

<b>IDENTIFICADOR:</b> ARCI.1		<b>NOMBRE:</b> Crear Reserva de Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Verificar usuarios del sistema Verificar Usuarios Sancionados para Reservas de Computadores Consultar Restricciones de Reservas de Computadores Consultar reservas de Computadores		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Estudiante. Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, fecha, horario, aula y equipo.		<b>SALIDA:</b> Reserva registrada o no efectuada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar las reservas de computadores para trabajos varios. La creación de una reserva de computadores puede ser efectuada por el actor iniciador interactuando directamente con el sistema desde una terminal o a través del monitor, el cual recibe los datos y los suministra al sistema. En ambos casos el actor iniciador recibirá la respuesta a su solicitud. De acuerdo a los datos ingresados, el sistema estará en capacidad de generar información asociada a dichos datos, como son: nombre del usuario solicitante, horario, aulas de informática y los equipos disponibles de acuerdo a la fecha seleccionada			
<b>OBSERVACIONES:</b> Cuando el proceso es realizado por el actor iniciador, el login y password validados inicialmente, permitirán al sistema obtener automáticamente los datos personales del usuario. Un usuario tiene derecho de crear un determinado número de reservas por día. Si la reserva se realiza a través del monitor se solicitará el código estudiantil o la identificación del usuario solicitante para obtener sus datos.			

## **CURSO BÁSICO DE EVENTOS**

### **Creación efectuada por el actor iniciador**

#### **Acción de los actores**

1. El actor iniciador selecciona la opción crear reserva de computadores.
4. Selecciona la fecha para su reserva
6. Selecciona la franja y el aula para su reserva
8. El actor iniciador selecciona el equipo para su reserva.
9. El actor iniciador selecciona la opción registrar reserva. **(a)**

#### **Respuesta del Sistema**

2. Verifica si el actor solicitante no se encuentra sancionado. **(d)**
3. Despliega una pantalla presentando al usuario la fecha, franjas, aulas y equipos disponibles para su reserva.
5. Actualiza el listado de las franjas disponibles en la fecha seleccionada.
7. Actualiza el listado de los equipos disponibles en la fecha, franja y aula seleccionada
10. Verifica si el actor iniciador no tiene realizada una reserva en la fecha y franja seleccionada. **(b)**
11. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(e)**
12. Registra la reserva actualizando la base de datos con la información suministrada.
13. Muestra un mensaje indicando que la reserva se realizó con éxito.

### **Creación efectuada a través del actor monitor**

#### **Acción de los actores**

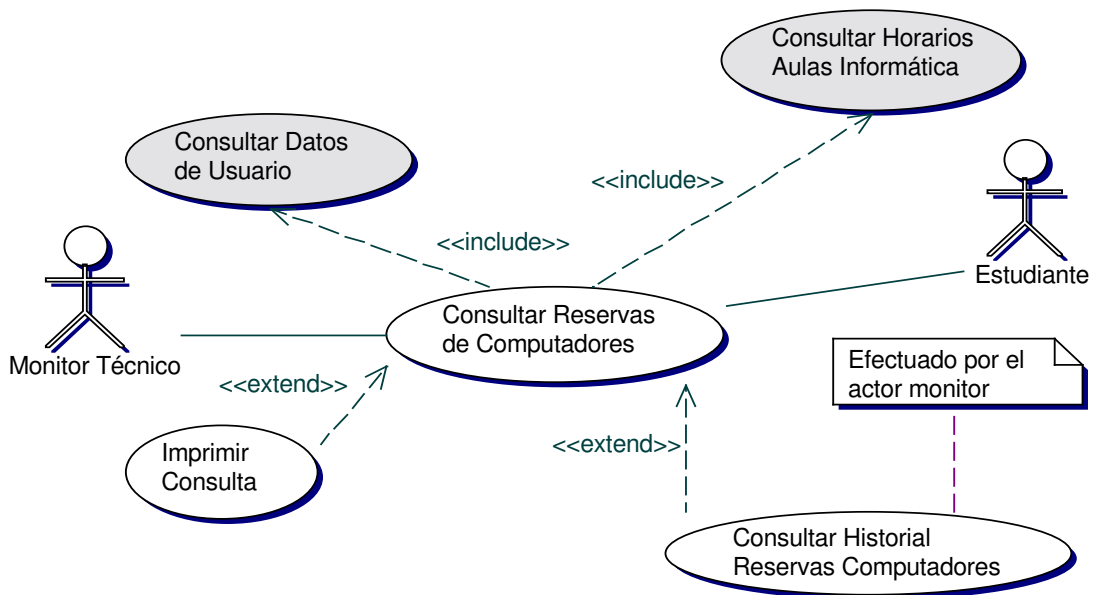
1. El actor iniciador solicita al actor monitor la creación de una reserva de Computadores.
2. El actor monitor selecciona la opción crear reserva de Computadores.
4. El actor monitor digita el código del usuario solicitante. **(a)**
8. El actor monitor selecciona la fecha para la reserva.
10. El actor monitor selecciona la franja y aula para la reserva.
12. El actor monitor selecciona el equipo para la reserva.
13. El actor monitor selecciona la opción registrar reserva. **(a)**

#### **Respuesta del Sistema**

3. Solicita el código del usuario.
5. Verifica si el actor solicitante está registrado como usuario del sistema. **(c)**
6. Verifica si el actor solicitante no se encuentra sancionado. **(d)**
7. Despliega una pantalla presentando el nombre completo del usuario y permite escoger la fecha, franja y aula para la reserva.
9. Actualiza el listado de las franjas disponibles en la fecha seleccionada.
11. Actualiza el listado de los equipos disponibles en la fecha, franja y aula seleccionada
14. Verifica si el actor iniciador no tiene realizada una reserva en la fecha y franja seleccionada. **(b)**
15. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(e)**
16. Registra la reserva actualizando la base de datos con la información suministrada.
17. Muestra un mensaje con los datos de la reserva indicando que la reserva se realizó con éxito

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no registrar la reserva: El sistema cancela el proceso</p> <p>(b) El actor iniciador ya tiene una reserva en la fecha y franja seleccionada: El sistema le indica este suceso a través de un mensaje.</p> <p>(c) El actor iniciador no se encuentra registrado como usuario del sistema: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(d) El actor iniciador se encuentra sancionado: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(e) Los datos seleccionados ya no se encuentran disponibles. El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p><b>TRIGGERS:</b></p> <p>Verificación de usuario: Este evento se ejecutará para determinar si el actor iniciador se encuentra registrado como usuario del sistema y obtener sus datos.</p> <p>Verificación del periodo de sanción: Si el usuario se encuentra sancionado, el sistema verificará si ya cumplió su periodo de sanción, en caso de que así sea lo habilitará automáticamente para usar el servicio, si no muestra un mensaje informándole que no puede efectuar el proceso.</p> <p>Validar que no exista un registro del usuario en la misma fecha y franja.</p> <p>Confirmación de disponibilidad: Se verificará que los datos seleccionados para la reserva aún se encuentren disponibles.</p>
<p>Precondición: No se debe sobrepasar el número de reservas permitidas para el usuario en el día. El usuario no debe estar sancionado.</p> <p>Postcondición: Registro eficiente de las reservas de Computadores. Control eficiente del uso de las reservas de Computadores.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra crear una reserva de Computadores controlando los requerimientos mínimos para poder efectuar el proceso de forma satisfactoria.</p> <p>La información de la base de datos se almacena de forma persistente y consistente.</p>

• **Diagrama Consultar Reservas de Computadores**

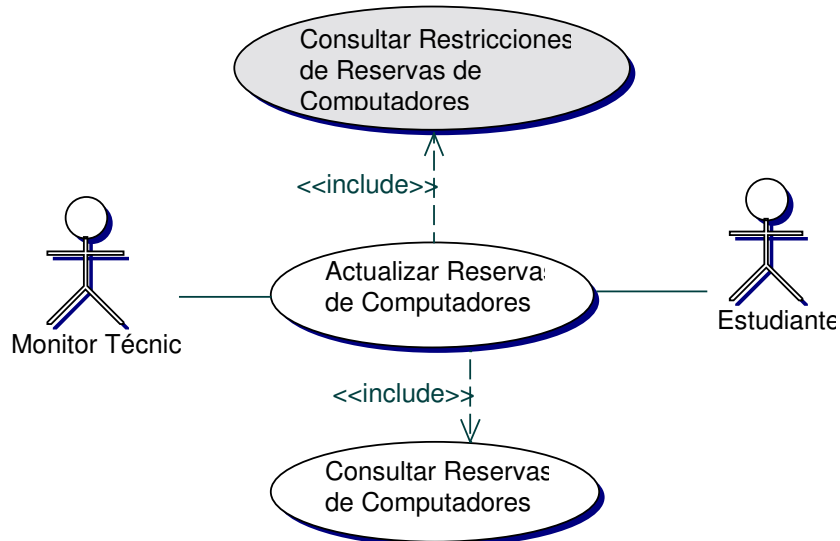


- Iteración Consultar Reservas de Computadores

<b>IDENTIFICADOR:</b> ARCI.2		<b>NOMBRE:</b> Consultar Reservas de Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario Consultar Horarios Aulas Informática Imprimir Consulta Consultar Historial Reservas Computadores		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código de usuario, fecha, franja, aula y número de equipo		<b>SALIDA:</b> Listado de Reservas de Computadores	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las reservas de computadores en la fecha indicada por el usuario. La consulta de reservas puede ser realizada por cualquiera de los actores involucrados en este caso de uso. Se desplegará la información de las reservas existentes y de las franjas disponibles en el día seleccionado por el actor. La información desplegada incluirá la fecha, nombre del usuario, el número del equipo, el aula, la hora, programa y el semestre. Adicionalmente existirá una consulta que permita visualizar el historial de todas las reservas efectuadas.			
<b>OBSERVACIONES:</b> El historial de las reservas efectuadas podrá ser accedido únicamente por el monitor técnico del aula de informática. La información de entrada dependerá del tipo de consulta que el monitor desee hacer (por usuario, por fecha y por equipo). Para el actor Estudiante, el resultado de la consulta mostrará sus propias reservas.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Consulta efectuada por el monitor</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor iniciador selecciona la opción consultar reservas de Computadores.	2. Despliega un menú presentando las opciones para su consulta (por franja, por usuario, por historial).	3. El actor monitor selecciona el criterio para su consulta. (a)	4. Solicita los datos necesarios para efectuar la consulta.
5. El actor monitor suministra los datos solicitados. (a)	6. Se despliega un listado de todos los equipos reservados indicando los datos del equipo, horario y datos del usuario, así como también los equipos que se encuentran disponibles.		
<b>Consulta efectuada por el estudiante</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor estudiante selecciona la opción consultar reservas de Computadores.	2. Solicita la fecha para la consulta.	3. El actor iniciador selecciona la fecha deseada. (a)	4. Se despliega un listado de todas sus reservas.
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Información oportuna de las reservas de Computadores existentes. Información consolidada de las reservas existentes hasta el momento.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar información de reservas de Computadores de forma veraz y oportuna. Los reportes generados permiten a la administración la toma de decisiones con respecto al uso del servicio.			



- **Diagrama Actualizar Reservas de Computadores**



- **Iteración Actualizar Reservas de Computadores**

<b>IDENTIFICADOR:</b> ARC1.3		<b>NOMBRE:</b> Actualizar Reservas de Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas de Computadores Consultar Restricciones de Reservas de Computadores		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, franja, aula, equipo.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar una determinada reserva de Computadores. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> El actor monitor tendrá acceso a la actualización de todas las reservas vigentes en una fecha determinada.			

## **CURSO BÁSICO DE EVENTOS**

### **Actualización efectuada por el actor monitor**

#### **Acción de los actores**

1. El actor monitor selecciona la opción actualizar reservas de Computadores.
3. El actor monitor suministra los datos solicitados. **(a)**
5. El actor monitor selecciona la reserva a modificar.
6. Da la orden para mostrar las opciones para actualizar la reserva. **(a)**
8. Selecciona los nuevos datos de la reserva
9. El actor monitor selecciona la opción actualizar reserva. **(a)**
11. El actor monitor confirma su decisión.

#### **Respuesta del Sistema**

2. Solicita la fecha, franja y aula.
4. Muestra un listado con la información correspondiente a todas las reservas en la fecha, franja y aula seleccionada como son nombre, programa, semestre y equipo.
7. Muestra las opciones para actualizar la reserva como son fecha, franja, aula y equipo.
10. Se pide confirmación para llevar a cabo el proceso.
12. Verifica si el usuario no tiene realizada una reserva en la fecha y franja modificada. **(b)**
13. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(c)**
14. Actualiza la reserva modificando la base de datos con la información suministrada.
15. Muestra un mensaje indicando que la reserva se actualizó con éxito

### **Actualización efectuada por el actor estudiante**

#### **Acción de los actores**

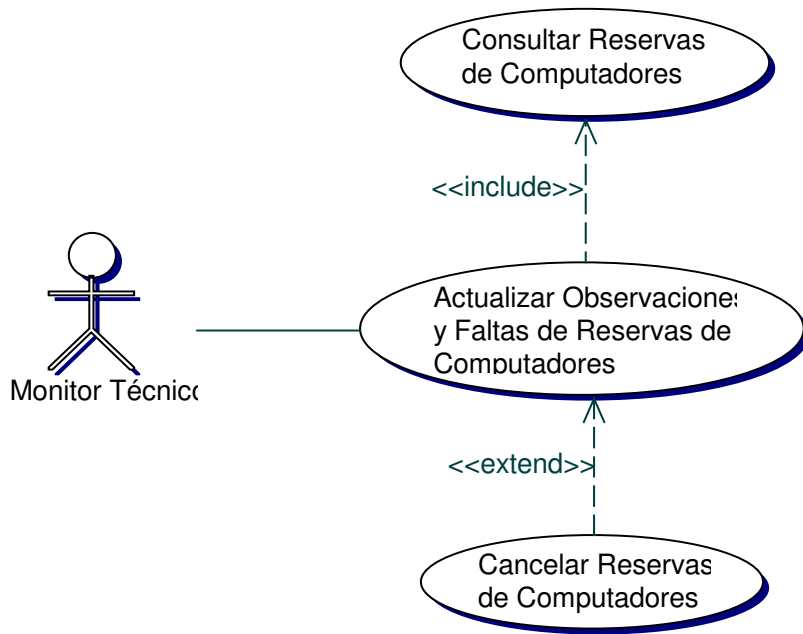
1. EL actor estudiante selecciona la opción actualizar reservas de Computadores.
3. El actor estudiante selecciona la fecha. **(a)**
5. El actor estudiante selecciona la reserva a modificar.
6. Da la orden para mostrar las opciones para actualizar la reserva. **(a)**
8. Selecciona los nuevos datos de la reserva
9. El actor estudiante selecciona la opción actualizar reserva. **(a)**
11. El actor monitor confirma su decisión.

#### **Respuesta del Sistema**

2. Solicita la fecha.
4. Muestra un listado con la información correspondiente a sus reservas vigentes en la fecha seleccionada como son franja, equipo y aula.
7. Muestra las opciones para actualizar la reserva como son fecha, franja, aula y equipo.
10. Se pide confirmación para llevar a cabo el proceso.
12. Verifica si el actor estudiante no tiene realizada una reserva en la fecha y franja seleccionada. **(b)**
13. Verifica que los datos seleccionados aun se encuentren disponibles para registrar la reserva. **(c)**
14. Actualiza la reserva modificando la base de datos con la información suministrada.
15. Muestra un mensaje indicando que la reserva se actualizó con éxito.

<b>CAMINOS DE EXCEPCION:</b>		
(a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.		
(b) El usuario ya tiene una reserva en la fecha y franja modificada: El sistema le indica este suceso a través de un mensaje y le da la posibilidad de cambiar la fecha.		
(c) Los datos seleccionados ya no se encuentran disponibles. El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.		
• El sistema no responde. El proceso no se efectúa.		
Precondición: La información de la base de datos es consistente. El usuario debe tener al menos una reserva creada.		
Postcondición: Actualización oportuna de las reservas de Computadores		
<b>CRITERIOS DE ACEPTACIÓN</b>		
Se cumple el requerimiento cuando se logra actualizar la información de reservas de Computadores de forma veraz y oportuna.		
La información de la base de datos se almacena de forma persistente y consistente.		

• **Diagrama Actualizar Observaciones y Faltas de Reservas de Internet**

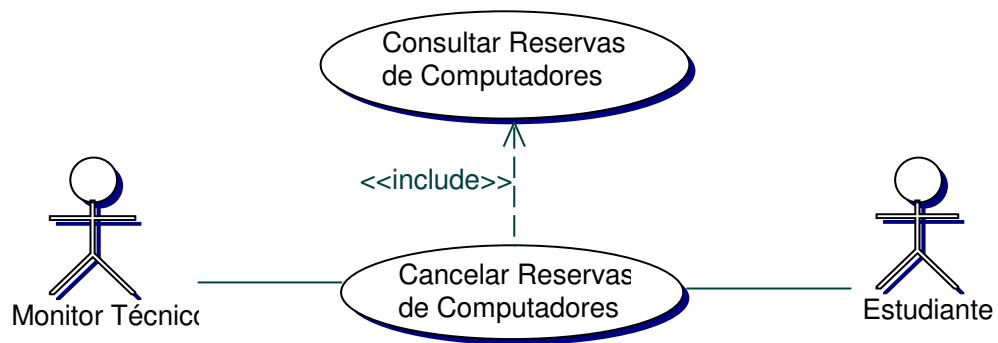


• **Iteración Actualizar Observaciones y Faltas de Reservas de Computadores**

<b>IDENTIFICADOR:</b> ARC1.4		<b>NOMBRE:</b> Actualizar Observaciones y Faltas de Reservas de Computadores
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas de Computadores Cancelar Reservas de Computadores	<b>CRITICO?</b> Si

<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Fecha, franja, aula, reserva, observaciones y faltas.	<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar las observaciones y las faltas de una determinada reserva. Se pide confirmación para efectuar el proceso.		
<b>OBSERVACIONES:</b> Las observaciones de la reserva se hacen en el caso de que surja alguna eventualidad. El actor monitor será el encargado de establecer cuando el usuario comete una falta cuya acumulación determinará la sanción correspondiente. Se puede considerar como faltas el mal uso de los equipos, la impuntualidad y/o el incumplimiento.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. EL actor monitor selecciona la opción de actualizar observaciones y faltas de Reservas de Computadores.		2. Solicita la fecha, la franja y el aula.
3. Selecciona los datos solicitados. (a)		4. Muestra un listado con la información correspondiente a las reservas de la fecha, franja y aula seleccionada como son equipo, nombre, programa y semestre.
5. El actor monitor selecciona la reserva a modificar.		
6. Da la orden para mostrar las opciones para actualizar la reserva. (a)		7. Muestra los datos de la reserva seleccionada y le permite establecer las observaciones y la falta al usuario.
8. El actor monitor modifica los datos correspondientes a observaciones y faltas.		
9. El actor monitor selecciona la opción de actualizar. (a)		10. Actualiza la reserva modificando la base de datos con la información suministrada.
		11. Muestra un mensaje indicando que la reserva se actualizó con éxito.
<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.		
<b>TRIGGERS:</b> Aplicar sanción al usuario: El sistema verifica el número de faltas en las que ha incurrido el usuario. Si el número obtenido sobrepasa un límite establecido, el usuario será sancionado durante un lapso de tiempo acordado en el cual no podrá hacer uso del servicio de Internet y se le cancelarán las reservas efectuadas en el periodo correspondiente a la sanción.		
Precondición: La información de la base de datos es consistente. Debe existir al menos una reserva creada.		
Postcondición: Actualización oportuna de observaciones y faltas de las reservas de Computadores		
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra actualizar la información de observaciones y faltas de reservas de Computadores de forma veraz y oportuna. Se lleva un control del uso de las reservas por parte de los usuarios de tal forma que el sistema permita aplicar sanciones cuando se incumplan las normas. La información de la base de datos se almacena de forma persistente y consistente.		

- **Diagrama Cancelar Reservas de Computadores**



• **Iteración Cancelar Reservas de Computadores**

<b>IDENTIFICADOR:</b> ARI1.5		<b>NOMBRE:</b> Cancelar Reservas de Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas de Computadores		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática, Estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, franja, aula, reserva a cancelar		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo cancelar una determinada reserva de Computadores. Se le presentará al actor estudiante un listado de todas sus reservas vigentes en una determinada fecha permitiéndole seleccionar la que desea cancelar. El actor monitor podrá visualizar todas las reservas de una determinada fecha, franja y aula. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> El actor monitor puede cancelar las reservas vigentes de una determinada fecha sin ninguna restricción			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Cancelación efectuada por el actor monitor</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor monitor selecciona la opción de eliminar reservas de Computadores.	2.	Solicita la fecha, franja y aula..
3.	EL actor monitor selecciona los datos solicitados. (a)	4.	Muestra un listado de todas las reservas vigentes.
5.	El actor monitor selecciona la reserva a cancelar.	7.	Se pide confirmación para llevar a cabo el proceso.
6.	El actor monitor selecciona la opción cancelar reserva. (a)	9.	Cancela la reserva actualizando la base de datos.
8.	El actor monitor confirma su decisión. (a)	10.	Muestra un mensaje indicando que la reserva se canceló con éxito.

<b>Cancelación efectuada por el actor estudiante</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor estudiante selecciona la opción de eliminar reservas de Computadores.	2. Solicita la fecha de la reserva.
3. El actor estudiante selecciona la fecha deseada. (a)	4. Muestra un listado de sus reservas vigentes en esa fecha.
5. El actor estudiante selecciona la reserva a cancelar.	7. Se pide confirmación para llevar a cabo el proceso.
6. El actor estudiante selecciona la opción cancelar reserva. (a)	9. Cancela la reserva actualizando la base de datos.
8. El actor estudiante confirma su decisión. (a)	10. Muestra un mensaje indicando que la reserva se canceló con éxito.

**CAMINOS DE EXCEPCION:**

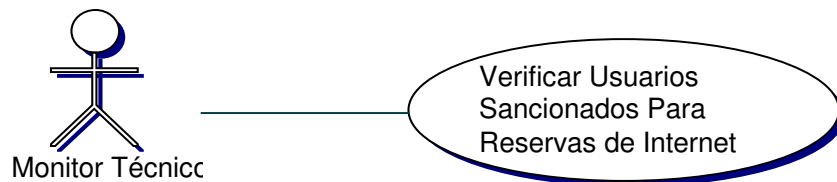
- (a) El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.
- El sistema no responde. El proceso no se efectúa.

**Precondición:** La información de la base de datos es consistente.  
El usuario debe tener al menos una reserva creada

**Postcondición:** Cancelación satisfactoria de las reservas de Internet existentes.

**CRITERIOS DE ACEPTACIÓN**  
Se cumple el requerimiento cuando se logra cancelar las reservas de Computadores de forma consistente.

• **Diagrama Verificar Usuarios Sancionados para Reservas de Internet**



• **Iteración Verificar Usuarios Sancionados para Reservas de Computadores**

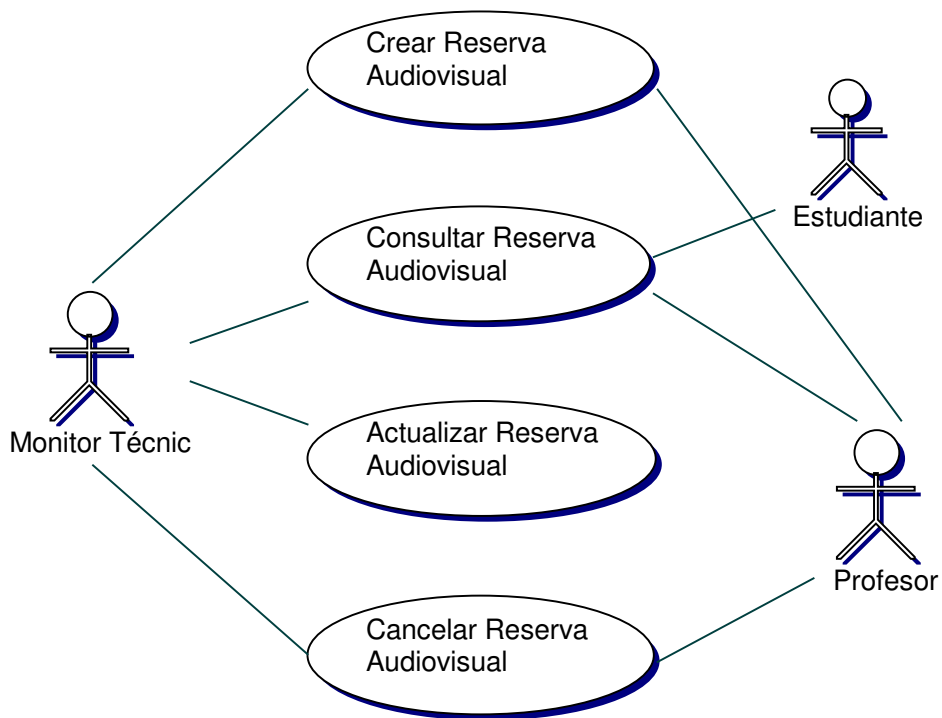
<b>IDENTIFICADOR:</b> ARC1.6		<b>NOMBRE:</b> Verificar Usuarios Sancionados para Reservas de Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Faltas		<b>SALIDA:</b> Listado de los usuarios sancionados con faltas para acceder al Servicio de Computadores	

<p><b>DESCRIPCIÓN:</b>  <i>Se satisface este requerimiento permitiendo consultar los usuarios que tengan faltas para el servicio de Computadores.</i>  <i>Si el número de faltas es igual al máximo permitido el usuario se encuentra sancionado para acceder al servicio de Computadores durante un periodo de tiempo establecido.</i>  <i>La información desplegada incluirá el nombre del usuario, programa, semestre, la fecha a partir de la cual inicia la sanción y la fecha en la cual termina. Adicionalmente se mostrará el número de faltas que tiene el usuario permitiendo ser modificado por el actor monitor manualmente.</i>  <i>Se pide confirmación para efectuar el proceso de actualización de faltas.</i></p>													
<p><b>OBSERVACIONES:</b> <i>El actor monitor podrá cambiar manualmente el número de faltas para aplicar o quitar una sanción automáticamente.</i></p>													
<p><b>CURSO BÁSICO DE EVENTOS</b></p> <table border="0"> <thead> <tr> <th style="text-align: center;"><i>Acción de los actores</i></th> <th style="text-align: center;"><i>Respuesta del Sistema</i></th> </tr> </thead> <tbody> <tr> <td>1. <i>EL actor monitor selecciona la opción verificar usuarios sancionados para reservas de Computadores.</i></td> <td>2. <i>Despliega el listado de todos los usuarios que tengan faltas para el servicio de Computadores.</i></td> </tr> <tr> <td>3. <i>El actor monitor modifica el número de faltas de un usuario.</i></td> <td></td> </tr> <tr> <td>4. <i>El actor monitor da la orden para actualizar faltas. (a)</i></td> <td>5. <i>Se pide confirmación para llevar a cabo el proceso.</i></td> </tr> <tr> <td>6. <i>El actor monitor confirma su decisión.</i></td> <td>7. <i>Actualiza las faltas del usuario modificando la base de datos con la información suministrada.</i></td> </tr> <tr> <td></td> <td>8. <i>Muestra un mensaje indicando que los datos se actualizaron con éxito.</i></td> </tr> </tbody> </table>		<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>	1. <i>EL actor monitor selecciona la opción verificar usuarios sancionados para reservas de Computadores.</i>	2. <i>Despliega el listado de todos los usuarios que tengan faltas para el servicio de Computadores.</i>	3. <i>El actor monitor modifica el número de faltas de un usuario.</i>		4. <i>El actor monitor da la orden para actualizar faltas. (a)</i>	5. <i>Se pide confirmación para llevar a cabo el proceso.</i>	6. <i>El actor monitor confirma su decisión.</i>	7. <i>Actualiza las faltas del usuario modificando la base de datos con la información suministrada.</i>		8. <i>Muestra un mensaje indicando que los datos se actualizaron con éxito.</i>
<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>												
1. <i>EL actor monitor selecciona la opción verificar usuarios sancionados para reservas de Computadores.</i>	2. <i>Despliega el listado de todos los usuarios que tengan faltas para el servicio de Computadores.</i>												
3. <i>El actor monitor modifica el número de faltas de un usuario.</i>													
4. <i>El actor monitor da la orden para actualizar faltas. (a)</i>	5. <i>Se pide confirmación para llevar a cabo el proceso.</i>												
6. <i>El actor monitor confirma su decisión.</i>	7. <i>Actualiza las faltas del usuario modificando la base de datos con la información suministrada.</i>												
	8. <i>Muestra un mensaje indicando que los datos se actualizaron con éxito.</i>												
<p><b>CAMINOS DE EXCEPCION:</b>  <b>(a)</b> <i>El actor iniciador decide cancelar la consulta: El sistema cancela el proceso.</i>  <ul style="list-style-type: none"> <li>• <i>El sistema no responde. El proceso no se efectúa.</i></li> </ul> </p>													
<p><b>TRIGGERS:</b>  <i>Aplicar o quitar sanción al usuario: El sistema verifica el número de faltas modificado por el actor monitor. Si el número obtenido sobrepasa un limite establecido, el usuario será sancionado durante un lapso de tiempo acordado en el cual no podrá hacer uso del servicio de Computadores y se le cancelarán las reservas efectuadas en el periodo correspondiente a la sanción. Si el usuario estaba sancionado y el número de faltas se disminuye, se le habilitará nuevamente para utilizar el servicio.</i></p>													
<p><i>Precondición: El usuario debe tener por lo menos una falta.</i>  <i>Postcondición: Información oportuna de los usuarios sancionados para acceder al servicio de internet.</i>  <i>Actualización de las faltas de un usuario para el servicio de computadores.</i></p>													
<p><b>CRITERIOS DE ACEPTACIÓN</b>  <i>Se cumple el requerimiento cuando se logra consultar los usuarios que tienen faltas para el uso del servicio de Computadores de forma veraz y oportuna.</i>  <i>El monitor estará en capacidad de aplicar o quitar una sanción a un determinado usuario.</i></p>													

## 9.1.4 Administración de Reservas Audiovisuales

### 9.1.4.1 Primera Iteración - Fachada

- Diagrama



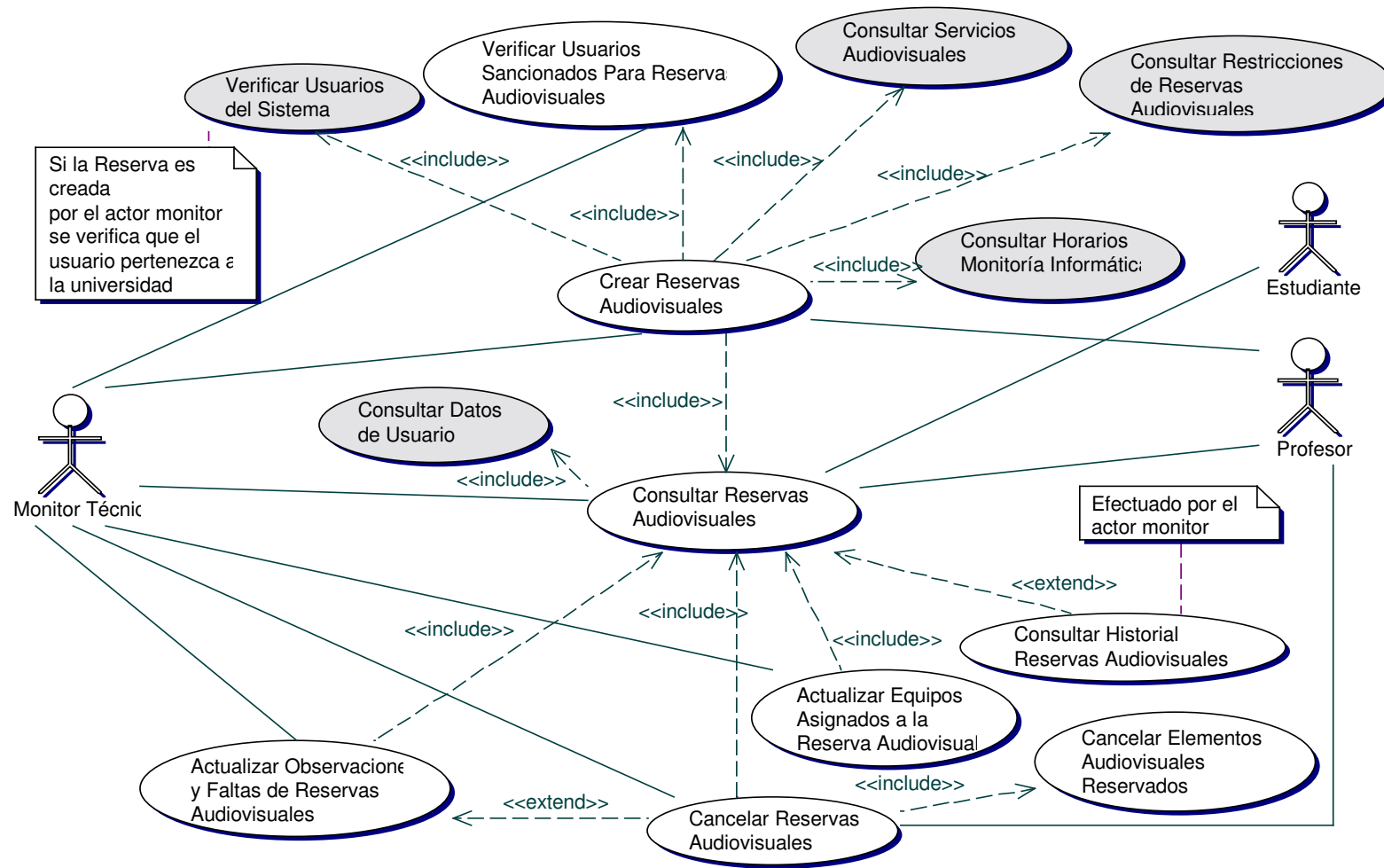
- Iteración

<b>IDENTIFICADOR:</b> ARA1		<b>NOMBRE:</b> Administración de Reservas Audiovisuales	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear reservas audiovisuales Consultar reservas audiovisuales Verificar Usuarios Sancionados para Reservas Audiovisuales Actualizar reservas audiovisuales Actualizar Observaciones y Faltas de Reservas de Internet Eliminar reservas audiovisuales		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador: Estudiante y Profesor Ejecutor: Monitor técnico del Aula de Informática y Profesor.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos necesarios para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar la asignación de elementos audiovisuales, previa solicitud del usuario. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y cancelación.			
<b>OBSERVACIONES:</b> El actor monitor puede administrar las reservas de elementos audiovisuales sin ninguna restricción. El actor estudiante podrá interactuar con el sistema únicamente para la parte de consulta de las reservas de equipos audiovisuales. Para poder llevar a cabo cualquier otro proceso debe hacerlo a través del monitor.			

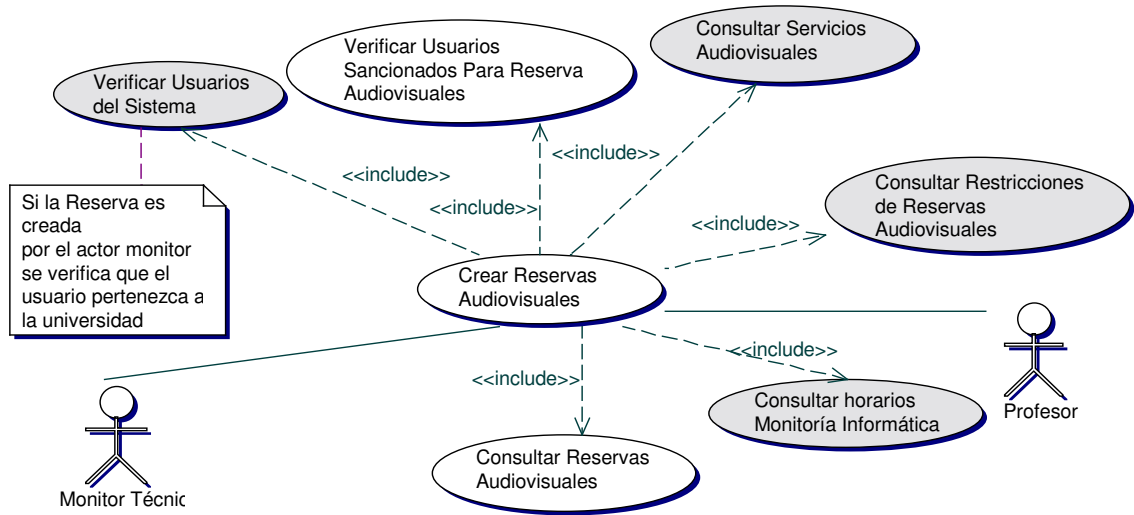


<p><i>El período de tiempo para la administración de una reserva será determinado por el administrador y estará comprendido entre la fecha actual y el intervalo de tiempo establecido.</i></p>
<p><b>TRIGGERS:</b>  <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i></p>
<p><i>Precondición: El actor ha iniciado una sesión en la aplicación.</i>  <i>Postcondición: Administración eficiente de las reservas Audiovisuales.</i></p>
<p><b>CRITERIOS DE ACEPTACIÓN</b>  <i>Permite llevar un registro sistemático de las reservas de elementos audiovisuales mediante el cual se puede controlar el estado de cada elemento antes y después de cada reserva.</i>  <i>La información se guarda en la base de datos de forma persistente y consistente.</i>  <i>Aprovechamiento eficiente de elementos audiovisuales y horarios asignados.</i>  <i>Administración de forma local y remota de reservas audiovisuales, brindando mayor acceso a los usuarios en el uso del servicio.</i></p>

### 9.1.4.2 Segunda Iteración – Llenado (Diagrama General)



• **Diagrama Crear Reservas Audiovisuales**



• **Iteración Crear Reservas Audiovisuales**

<b>IDENTIFICADOR:</b> ARA1.1		<b>NOMBRE:</b> Crear Reservas Audiovisuales.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar reservas audiovisuales Verificar Usuarios del Sistema Verificar Usuarios Sancionados para Reservas Audiovisuales Consultar Servicios Audiovisuales Consultar Restricciones de Reservas Audiovisuales		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Estudiante y Profesor. Ejecutor: Monitor técnico del Aula de Informática y Profesor.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, fecha, servicio audiovisual, horario, programa, semestre, observaciones.		<b>SALIDA:</b> Reserva registrada o no efectuada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar las reservas de elementos audiovisuales. La creación de una reserva de elementos audiovisuales será efectuada por el actor monitor o profesor interactuando directamente con el sistema. Para el actor estudiante, la creación de una reserva debe hacerse a través del monitor. De acuerdo a los datos ingresados, el sistema estará en capacidad de generar información asociada a dichos datos, como son: nombre del usuario solicitante, servicios audiovisuales y horarios disponibles de acuerdo a la fecha seleccionada. También se visualizará las reservas actuales del usuario y los programas y semestres existentes para realizar la reserva.			
<b>OBSERVACIONES:</b> Las observaciones de la reserva permitirán realizar anotaciones referentes al aula audiovisual a utilizar o si el uso de los equipos se extenderá en un horario donde no haya servicio de monitoría informática.			

## **CURSO BÁSICO DE EVENTOS**

### **Creación efectuada por el actor monitor**

#### **Acción de los actores**

1. El actor iniciador solicita la creación de una reserva de elementos audiovisuales al actor monitor.
2. El actor monitor selecciona la opción crear reserva audiovisual.
4. El actor monitor digita el código del usuario solicitante. **(a)**
  
9. El actor monitor selecciona la fecha y el servicio audiovisual para la reserva.
11. El actor monitor determina las franjas, el programa, semestre y observaciones de la reserva
12. El actor monitor selecciona la opción registrar reserva. **(a)**

#### **Respuesta del Sistema**

3. Solicita el código del actor iniciador.
5. Verifica si el actor solicitante es un usuario valido de la universidad. **(b)**
7. Verifica si el actor solicitante no se encuentra sancionado. **(c)**
8. Despliega una pantalla presentando el nombre completo del actor solicitante y sus reservas efectuadas para ese día, y permite escoger la fecha y el servicio audiovisual para la reserva.
10. Actualiza el listado de las franjas disponibles para la fecha y servicio seleccionado.
  
13. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(d)**
14. Registra la reserva actualizando la base de datos con la información suministrada.
15. Muestra un mensaje indicando que la reserva se actualizó con éxito

### **Creación efectuada por el actor profesor**

#### **Acción de los actores**

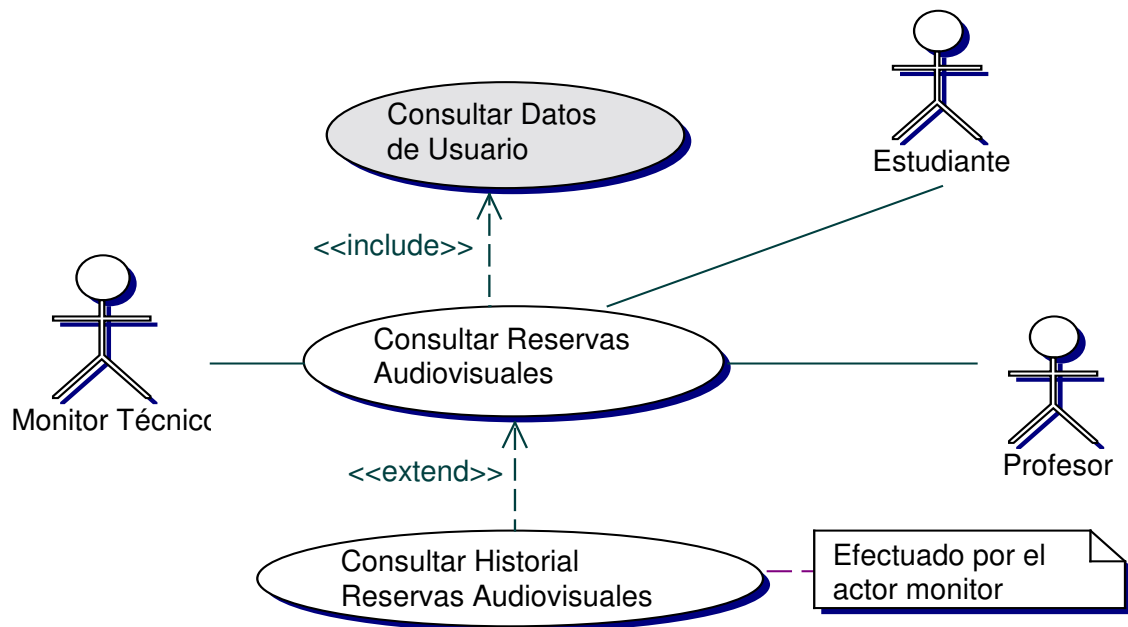
1. El actor profesor selecciona la opción crear reserva audiovisual.
  
9. El actor profesor selecciona la fecha y el servicio audiovisual para la reserva.
11. El actor profesor determina las franjas, el programa, semestre y observaciones de la reserva
12. El actor profesor selecciona la opción registrar reserva. **(a)**

#### **Respuesta del Sistema**

7. Verifica si el actor solicitante no se encuentra sancionado. **(c)**
8. Despliega una pantalla presentando el nombre completo del actor solicitante y sus reservas efectuadas para ese día, y permite escoger la fecha y el servicio audiovisual para la reserva.
10. Actualiza el listado de las franjas disponibles para la fecha y servicio seleccionado.
  
13. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. **(d)**
14. Registra la reserva actualizando la base de datos con la información suministrada.
15. Muestra un mensaje indicando que la reserva se actualizó con éxito

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no registrar la reserva: El sistema cancela el proceso</p> <p>(b) El actor iniciador no es un usuario de la universidad: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(c) El actor iniciador se encuentra sancionado: El sistema indica este suceso a través de un mensaje y cancela el proceso.</p> <p>(d) Los datos seleccionados ya no se encuentran disponibles. El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p><b>TRIGGERS:</b></p> <p>Verificación de usuario: Este evento se ejecutará para determinar si el actor iniciador es un usuario válido de la universidad.</p> <p>Verificación del periodo de sanción: Si el usuario se encuentra sancionado, el sistema verificará si ya cumplió su periodo de sanción, en caso de que así sea lo habilitará automáticamente para usar el servicio, si no muestra un mensaje informándole que no puede efectuar el proceso.</p> <p>Confirmación de disponibilidad: Se verificará que los datos seleccionados para la reserva aún se encuentren disponibles.</p>
<p>Precondición: El actor iniciador debe ser un usuario de la universidad. El usuario no debe estar sancionado..</p> <p>Postcondición: Registro eficiente de las reservas de equipos audiovisuales Control eficiente del estado de los elementos audiovisuales durante la reserva.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra crear una reserva de elementos audiovisuales controlando los requerimientos mínimos para poder efectuar el proceso de forma satisfactoria.</p> <p>La información de la base de datos se almacena de forma persistente y consistente.</p> <p>Control eficiente en la asignación de elementos audiovisuales dependiendo de su disponibilidad.</p>

• **Diagrama Consultar Reservas de Audiovisuales**

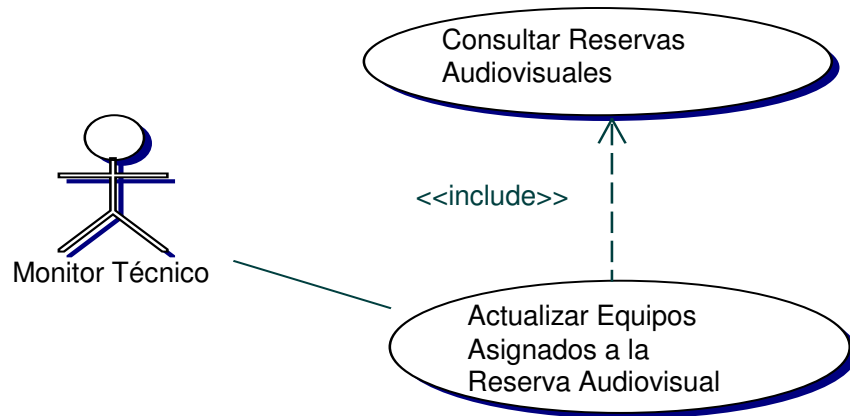


- Iteración Consultar Reservas de Audiovisuales

<b>IDENTIFICADOR:</b> ARA1.2		<b>NOMBRE:</b> Consultar Reservas Audiovisuales	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de usuario Consultar Horarios Monitoría Informática Imprimir Consulta Consultar Historial Reservas Audiovisuales		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del aula de Informática, Estudiante y Profesor.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, fecha, franja.		<b>SALIDA:</b> Listado de reservas de elementos audiovisuales	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las reservas de elementos audiovisuales. La consulta de reservas puede ser realizada por cualquiera de los actores involucrados en este caso de uso. Para el actor monitor se desplegarán todas las reservas realizadas; para los actores estudiante y profesor se desplegarán sus reservas realizadas. La información desplegada incluirá la fecha, el horario, usuario, equipos audiovisuales, programa y semestre. Adicionalmente existirá una consulta que permita visualizar el historial de todas las reservas efectuadas.			
<b>OBSERVACIONES:</b> Las observaciones de la reserva se hacen en el caso de que surja alguna eventualidad. El actor monitor será el encargado de establecer cuando el usuario comete una falta cuya acumulación determinará la sanción correspondiente. Se puede considerar como faltas el mal uso de los equipos, la impuntualidad y/o el incumplimiento. El historial de las reservas efectuadas podrá ser accedido únicamente por el monitor técnico del aula de informática. La información de entrada dependerá del tipo de consulta que el monitor desee hacer (por fecha, por usuario).			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Consulta efectuada por el actor monitor</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor monitor selecciona la opción consultar reservas de audiovisuales.	2. Despliega un menú presentando las opciones para su consulta (por fecha, por usuario, por historial).	3. El actor monitor selecciona la opción para su consulta. (a)	4. Solicita los datos necesarios para efectuar la consulta.
5. El actor monitor suministra los datos solicitados. (a)	6. Se despliega un listado de todas las reservas indicando el servicio audiovisual, horario y datos del usuario.		
<b>Consulta efectuada por el actor profesor o estudiante</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar reservas de elementos audiovisuales.	2. Despliega un listado de todas las reservas del actor en la fecha actual indicando el horario, equipos audiovisuales, programa y semestre de la reserva.	3. El actor selecciona la fecha deseada. (a)	4. Actualiza la información desplegando un listado de los datos de todas las reservas del actor efectuadas en la fecha seleccionada.

<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.
Precondición: La información de la base de datos es consistente. Postcondición: Información oportuna de las reservas de elementos audiovisuales existentes. Información consolidada de las reservas existentes hasta el momento.
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar información de reservas de elementos audiovisuales de forma veraz y oportuna. Los reportes generados permiten a la administración la toma de decisiones con respecto al uso del servicio.

• **Diagrama Actualizar Equipos Asignados a la Reserva Audiovisual**

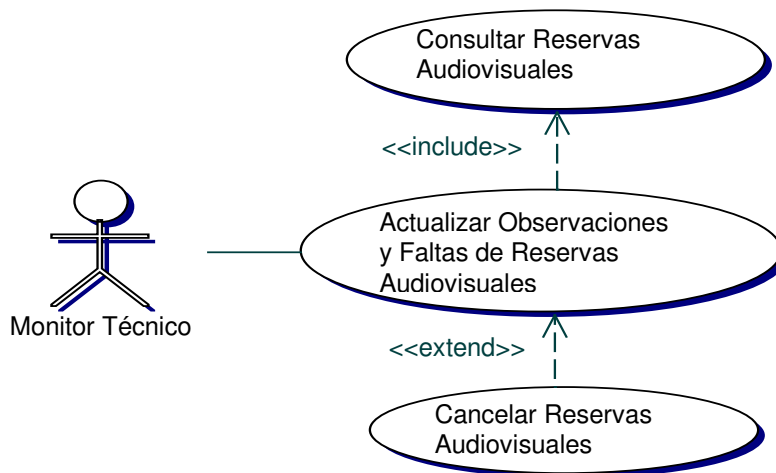


• **Iteración Actualizar Equipos Asignados a la Reserva Audiovisual**

<b>IDENTIFICADOR:</b> ARA1.3		<b>NOMBRE:</b> Actualizar Equipos Asignados a la reserva Audiovisual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas Audiovisuales		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, horario, reserva, equipo audiovisual.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los equipos audiovisuales que se asignan a una reserva audiovisual. La reserva tiene asignado un servicio audiovisual conformado por una serie de elementos o equipos audiovisuales, este requerimiento permite asignar un determinado equipo para prestar el servicio audiovisual reservado.			
<b>OBSERVACIONES:</b> EL proceso de asignación lo realizará el actor monitor técnico antes de la fecha de la reserva o durante la entrega de los equipos audiovisuales al usuario dueño de la reserva.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción asignar equipos a la reserva audiovisual.	2. Solicita la fecha y el horario de la reserva a la que se asignarán los equipos audiovisuales.
3. Selecciona la fecha y el horario de la reserva. <b>(a)</b>	4. Muestra un listado con la información correspondiente a todas las reservas en la fecha y horario seleccionado.
5. Selecciona la reserva deseada. <b>(a)</b>	6. Muestra los datos de la reserva y permite seleccionar los equipos audiovisuales que se asignarán al servicio audiovisual reservado.
7. Selecciona los elementos audiovisuales disponibles.	9. Se pide confirmación para llevar a cabo el proceso.
8. Selecciona la opción asignar equipos audiovisuales <b>(a)</b>	11. Verifica que los datos seleccionados aún se encuentren disponibles para efectuar el proceso. <b>(b)</b>
10. Confirma su decisión.	12. Actualiza la asignando los elementos audiovisuales seleccionados.
	13. Muestra un mensaje indicando que la reserva se actualizó con éxito
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor decide cancelar la actualización: El sistema cancela el proceso.	
<b>(b)</b> Los datos seleccionados ya no se encuentran disponibles. El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.	
• El sistema no responde. El proceso no se efectúa.	
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos una reserva creada en la fecha.	
<b>Postcondición:</b> Asignación eficiente de elementos audiovisuales	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra asignar los elementos audiovisuales disponibles a una reserva audiovisual de forma veraz y oportuna.	
La información de la base de datos se almacena de forma persistente y consistente.	

• **Diagrama Actualizar Observaciones y Faltas de Reservas Audiovisuales**



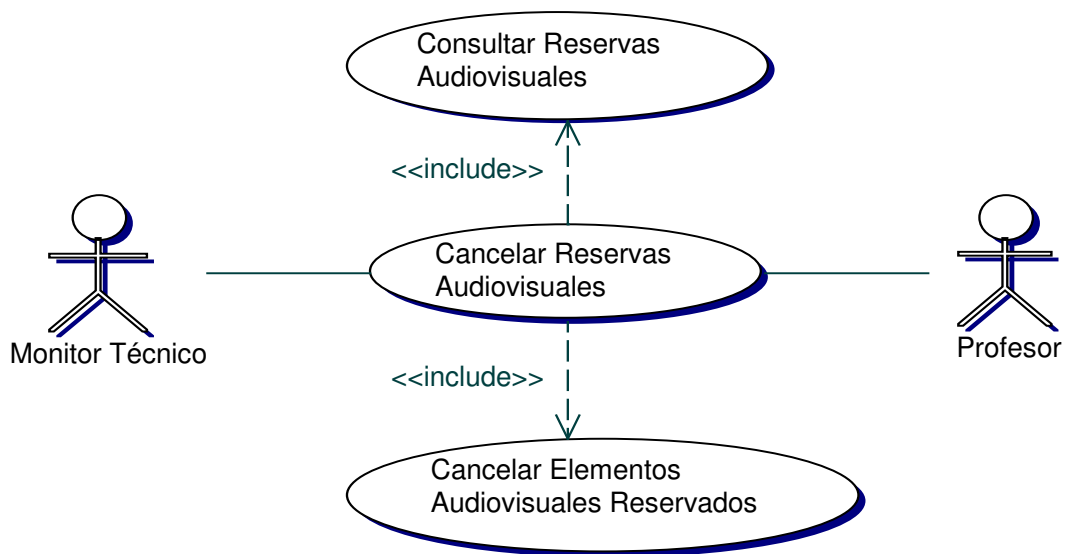


• Iteración Actualizar Observaciones y Faltas de Reservas Audiovisuales

<b>IDENTIFICADOR:</b> ARA1.4		<b>NOMBRE:</b> Actualizar observaciones y faltas de Reservas Audiovisuales	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas Audiovisuales Cancelar Reservas Audiovisuales		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, horario, reserva, observaciones y faltas.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar las observaciones y las faltas de una determinada reserva. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> Las observaciones de la reserva se hacen en el caso de que surja alguna eventualidad. El actor monitor será el encargado de establecer cuando el usuario comete una falta cuya acumulación determinará la sanción correspondiente. Se puede considerar como faltas el mal uso de los equipos, la impuntualidad y/o el incumplimiento.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Actualización efectuada por el actor monitor</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor monitor selecciona la opción de actualizar observaciones y faltas de Reservas de Audiovisuales.	2. Solicita la fecha y el horario de la reserva a actualizar.	3. Selecciona los datos solicitados. (a)	4. Muestra un listado con la información correspondiente a todas las reservas en la fecha y horario seleccionado como son franja, nombre usuario, servicio audiovisual, programa y semestre.
5. El actor monitor selecciona la reserva a actualizar.	6. Muestra los datos de la reserva y permite actualizar las observaciones y faltas.	6. Da la orden para actualizar la reserva. (a)	7. Muestra los datos de la reserva y permite actualizar las observaciones y faltas.
8. El actor monitor modifica los datos correspondientes a observaciones y faltas.	10. Se pide confirmación para llevar a cabo el proceso.	9. El actor monitor selecciona la opción de actualizar. (a)	12. Actualiza la reserva modificando la base de datos con la información suministrada.
11. El actor monitor confirma su decisión	15. Muestra un mensaje indicando que la reserva se actualizó con éxito		
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la actualización: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>TRIGGERS:</b> Aplicar sanción al usuario: El sistema verifica el número de faltas en las que ha incurrido el usuario. Si el número obtenido sobrepasa un límite establecido, el usuario será sancionado durante un lapso de tiempo acordado en el cual no podrá hacer uso del servicio de Audiovisuales y se le cancelarán las reservas efectuadas en el periodo correspondiente a la sanción.			

<p><i>Precondición: La información de la base de datos es consistente. El usuario debe tener al menos una reserva creada.</i></p> <p><i>Postcondición: Actualización oportuna de observaciones y faltas de reservas audiovisuales.</i></p> <p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p><i>Se cumple el requerimiento cuando se logra actualizar la información de observaciones y faltas de reservas Audiovisuales de forma veraz y oportuna.</i></p> <p><i>Se lleva un control del uso de las reservas por parte de los usuarios de tal forma que el sistema permita aplicar sanciones cuando se incumplan las normas.</i></p> <p><i>La información de la base de datos se almacena de forma persistente y consistente.</i></p>
---

• **Diagrama Cancelar Reservas Audiovisuales**

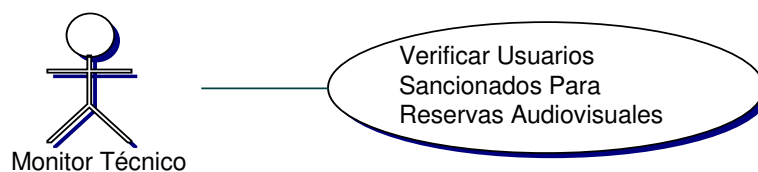


• **Iteración Cancelar Reservas de Audiovisuales**

<b>IDENTIFICADOR:</b> ARA1.5		<b>NOMBRE:</b> Cancelar Reservas Audiovisuales	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas Audiovisuales Cancelar Elementos Audiovisuales Reservados		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Monitores Técnicos del aula de Informática, Estudiantes, Profesores Ejecutor: Monitores Técnicos del aula de Informática, Profesores	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, reserva a eliminar		<b>SALIDA:</b> Proceso efectuada o cancelado	

<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo cancelar una determinada reserva de elementos audiovisuales. Se le presentará al usuario un listado de todas sus reservas vigentes permitiéndole seleccionar aquellas que desea cancelar. El actor monitor podrá visualizar todas las reservas de una determinada fecha. Se pide confirmación para efectuar el proceso.	
<b>OBSERVACIONES:</b> El actor monitor puede cancelar las reservas vigentes de una determinada fecha sin ninguna restricción.	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Cancelación efectuada por el actor monitor</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor monitor selecciona la opción de cancelar reservas audiovisuales.	2. Muestra un listado de todas las reservas vigentes en la fecha actual, permitiendo elegir una fecha diferente.
3. El actor monitor selecciona la reserva a cancelar.	
4. El actor monitor selecciona la opción cancelar reserva. <b>(a)</b>	5. Se pide confirmación para llevar a cabo el proceso.
6. El actor monitor confirma su decisión. <b>(a)</b>	7. Cancela la reserva actualizando la base de datos.
	8. Muestra un mensaje indicando que la reserva se canceló con éxito
<b>Cancelación efectuada por el actor profesor</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor profesor selecciona la opción de cancelar reservas audiovisuales.	2. Muestra un listado de sus reservas vigentes.
3. El actor profesor selecciona la reserva a cancelar.	
4. El actor profesor selecciona la opción cancelar reserva. <b>(a)</b>	5. Se pide confirmación para llevar a cabo el proceso.
6. El actor profesor confirma su decisión.	8. Cancela la reserva actualizando la base de datos.
	9. Muestra un mensaje indicando que la reserva se canceló con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
<b>TRIGGERS:</b>	
Eliminar Elementos Audiovisuales Reservados: El sistema verifica los elementos audiovisuales individuales que se hayan asignado a la reserva y elimina los registros de esta asignación	
Precondición: La información de la base de datos es consistente. El usuario debe tener al menos una reserva creada.	
Postcondición: Cancelación satisfactoria de las reservas de Audiovisuales existentes.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra cancelar las reservas de Audiovisuales de forma consistente.	

• **Diagrama Verificar Usuarios Sancionados para Reservas Audiovisuales**



- Iteración Verificar Usuarios Sancionados para Reservas Audiovisuales

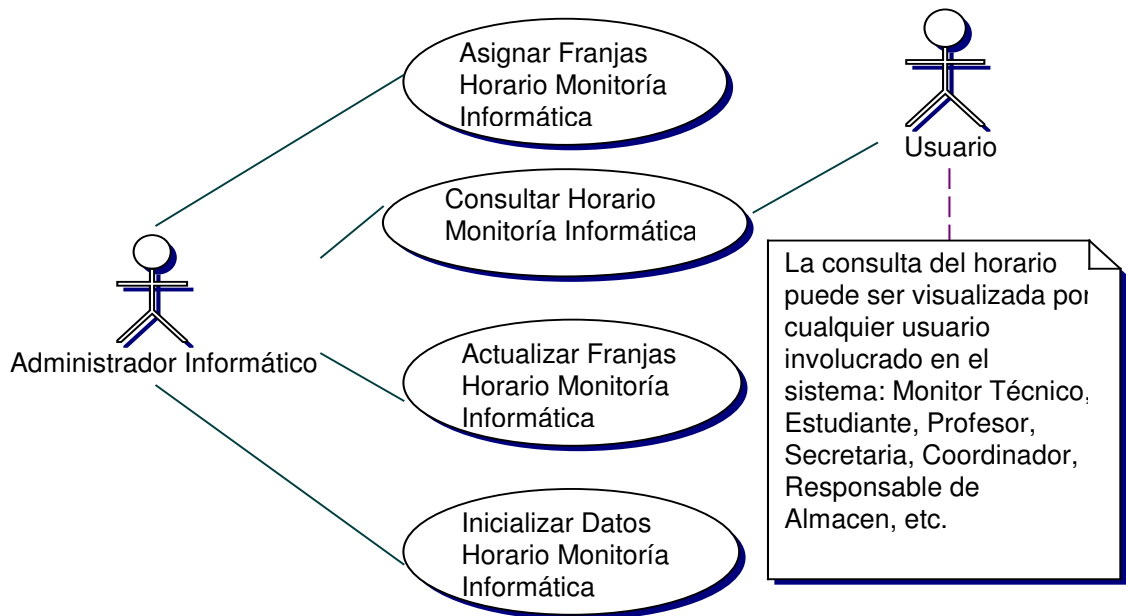
<b>IDENTIFICADOR:</b> ARA1.6		<b>NOMBRE:</b> Verificar Usuarios Sancionados para Reservas Audiovisuales	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor técnico del Aula de Informática.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Faltas		<b>SALIDA:</b> Listado de los usuarios sancionados con faltas para acceder al Servicio de Audiovisuales	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los usuarios que tengan faltas para el servicio de Audiovisuales. Si el número de faltas es igual al máximo permitido el usuario se encuentra sancionado para acceder al servicio de Audiovisuales durante un periodo de tiempo establecido. La información desplegada incluirá el nombre del usuario, programa, semestre, la fecha a partir de la cual inicia la sanción y la fecha en la cual termina. Adicionalmente se mostrará el número de faltas que tiene el usuario permitiendo ser modificado por el actor monitor manualmente. Se pide confirmación para efectuar el proceso de actualización de faltas.			
<b>OBSERVACIONES:</b> El actor monitor podrá cambiar manualmente el número de faltas para aplicar o quitar una sanción automáticamente.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor monitor selecciona la opción verificar usuarios sancionados para reservas de Audiovisuales.		2. Despliega el listado de todos los usuarios que tengan faltas para el servicio de Audiovisuales.	
3. El actor monitor modifica el número de faltas de un usuario.		5. Se pide confirmación para llevar a cabo el proceso.	
4. El actor monitor da la orden para actualizar faltas. (a)		7. Actualiza las faltas del usuario modificando la base de datos con la información suministrada.	
6. El actor monitor confirma su decisión.		8. Muestra un mensaje indicando que los datos se actualizaron con éxito.	
<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.			
<b>TRIGGERS:</b> Aplicar o quitar sanción al usuario: El sistema verifica el número de faltas modificado por el actor monitor. Si el número obtenido sobrepasa un límite establecido, el usuario será sancionado durante un lapso de tiempo acordado en el cual no podrá hacer uso del servicio de Audiovisuales y se le cancelarán las reservas efectuadas en el periodo correspondiente a la sanción. Si el usuario estaba sancionado y el número de faltas se disminuye, se le habilitará nuevamente para utilizar el servicio.			

<p><i>Precondición: El usuario debe tener por lo menos una falta.</i></p> <p><i>Postcondición: Información oportuna de los usuarios sancionados para acceder al servicio de Audiovisuales.</i></p> <p><i>Actualización de las faltas de un usuario para el servicio de Audiovisuales.</i></p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p><i>Se cumple el requerimiento cuando se logra consultar los usuarios que tienen faltas para el uso del servicio de Audiovisuales de forma veraz y oportuna.</i></p> <p><i>El monitor estará en capacidad de aplicar o quitar una sanción a un determinado usuario.</i></p>

## 9.1.5 Administración de Horarios de Monitoría Informática

### 9.1.5.1 Primera Iteración - Fachada

- Diagrama

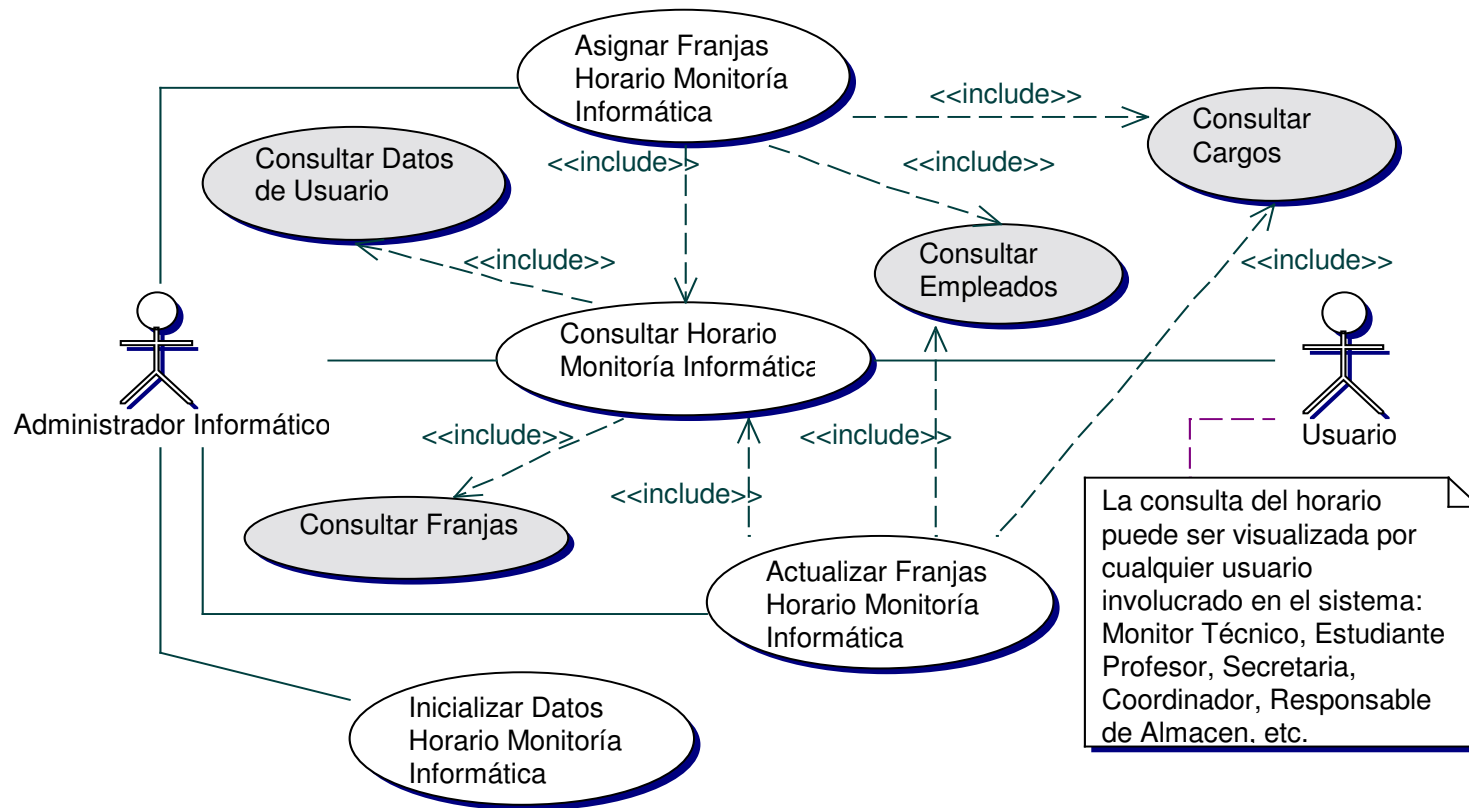


- Iteración

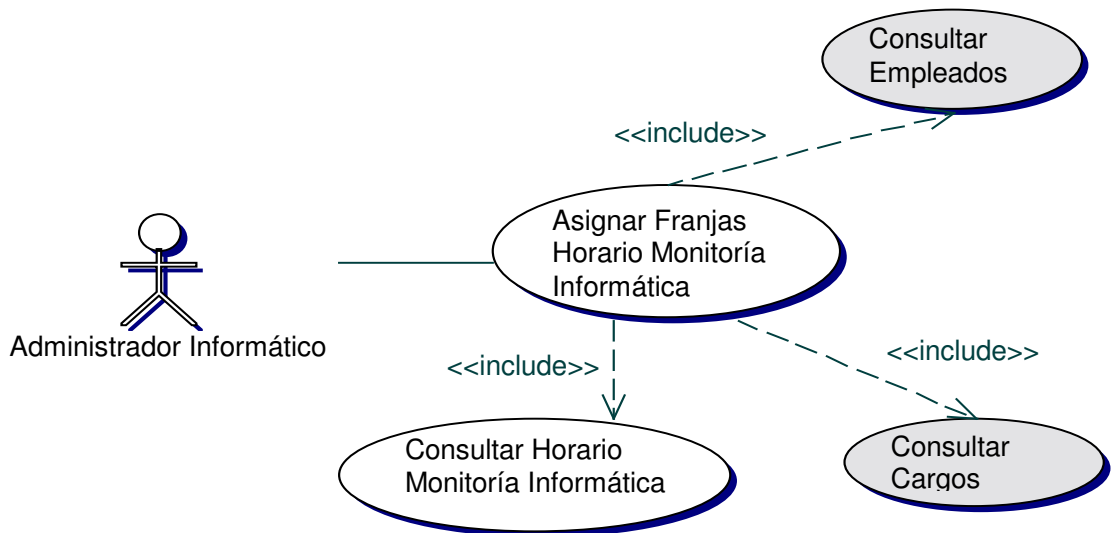
<b>IDENTIFICADOR:</b> AHMI1		<b>NOMBRE:</b> Administración de Horarios de Monitoría Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Asignar Franjas Horario Monitoría Informática Consultar Horario Monitoría Informática Actualizar Franjas Horario Monitoría Informática Inicializar Datos Horario Monitoría Informática		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b>	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b>	

<i>Fachada</i>	<i>Usuario.</i>	<i>Páginas HTML y JSP.</i>
<b>ENTRADA:</b> <i>Datos para efectuar el proceso.</i>		<b>SALIDA:</b> <i>Proceso efectuado o cancelado</i>
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo administrar las franjas del Horario de Monitoría Informática, especificando el personal responsable de prestar el servicio en cada franja o informando si en dicha franja no se presta servicio de Monitoría. El proceso de administración se desarrollará teniendo en cuenta las operaciones de asignación, consulta, actualización e inicialización.</i>		
<b>OBSERVACIONES:</b> <i>El actor administrador informático puede administrar los horarios de monitoría sin ninguna restricción. El actor usuario puede ser cualquier actor involucrado en el sistema y tendrá acceso solamente a la consulta del horario de monitoría informática.</i>		
<b>TRIGGERS:</b> <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i>		
<i>Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de las franjas de Horario de Monitoría informática.</i>		
<b>CRITERIOS DE ACEPTACIÓN</b> <i>Permite llevar una administración sistemática de las franjas de prestación del servicio de Monitoría Informática. La información en la base de datos se maneja de forma persistente y consistente. Información oportuna de las franjas de prestación del servicio de monitoría informática.</i>		

### 9.1.5.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Asignar Franjas Horario Monitoría Informática



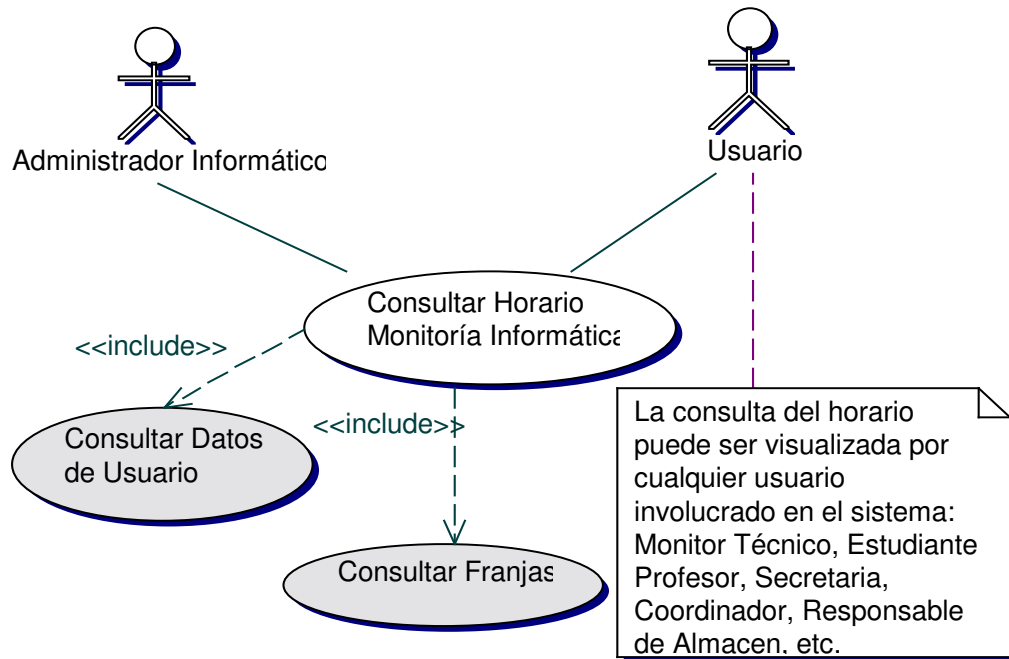
- Iteración Asignar Franjas Horario Monitoría Informática

<b>IDENTIFICADOR:</b> AHMI1.1		<b>NOMBRE:</b> Asignar Franjas Horario Monitoría Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Horario Monitoría Informática Consultar Empleados Consultar Cargos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Día, Estado del servicio, Cargo, Empleado, Franjas.		<b>SALIDA:</b> Proceso efectuada o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo asignar las franjas del Horario de Monitoría Informática definiendo si se presta o no el servicio de monitoría y el encargado de prestar el servicio.			
<b>OBSERVACIONES:</b> Se podrán asignar al horario las franjas que aún no se hayan utilizado en el horario, si ya están utilizadas se deberá inicializar el horario para liberar las franjas y volverlas a asignar.			



<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. EL actor selecciona la opción asignar franjas horario monitoría informática.</li> <li>3. El actor selecciona los datos para el horario y las franjas deseadas.</li> <li>4. Da la orden para asignar los datos y las franjas al horario <b>(a)</b></li> </ol>	<ol style="list-style-type: none"> <li>2. Muestra un listado con la información correspondiente a los días, cargos, empleados y franjas disponibles para el horario.</li> <li>5. Registra las franjas de horario actualizando la base de datos con la información suministrada.</li> <li>6. Muestra un mensaje indicando que la asignación se realizó con éxito</li> </ol>
<b>CAMINOS DE EXCEPCION:</b>	
<p><b>(a)</b> El actor iniciador decide cancelar la asignación: El sistema cancela el proceso.</p> <ul style="list-style-type: none"> <li>• El sistema no responde. El proceso no se efectúa.</li> </ul>	
<p><i>Precondición:</i> La información de la base de datos es consistente. deben existir franjas libres para asignar al horario.</p> <p><i>Postcondición:</i> Asignación eficiente de las franjas del Horario de Monitoría Informática</p>	
<b>CRITERIOS DE ACEPTACIÓN</b>	
<p>Se cumple el requerimiento cuando se logra asignar los datos y franjas del horario de monitoría informática de forma veraz y oportuna.</p> <p>La información de la base de datos se almacena de forma persistente y consistente.</p>	

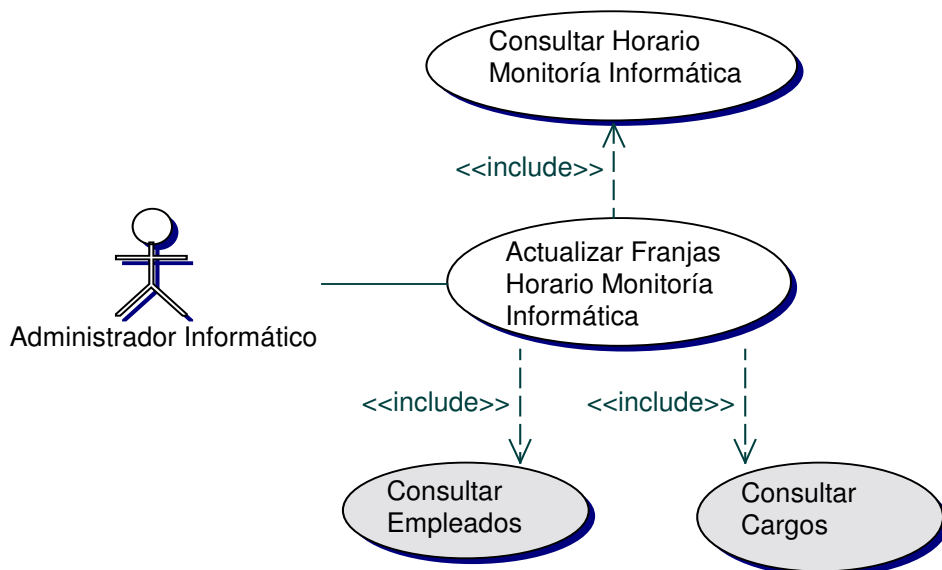
• **Diagrama Consultar Horario Monitoría Informática**



- Iteración Consultar Horario Monitoría Informática

<b>IDENTIFICADOR:</b> AHMI.2		<b>NOMBRE:</b> Consultar Horario Monitoría Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático, Usuario.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Datos de Horario de Monitoría Informática	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las franjas horarias de prestación del servicio de Monitoría Informática.			
<b>OBSERVACIONES:</b> La consulta del horario de Monitoría Informática puede ser visualizada por cualquier usuario involucrado en el sistema.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar Horario de Monitoría Informática.		2. Muestra un listado con la información de todas las franjas horarias del servicio de Monitoría Informática.	
<b>CAMINOS DE EXCEPCION:</b>			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente.			
<b>Postcondición:</b> Información oportuna de los datos del horario de Monitoría Informática.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar la información de las franjas de prestación del servicio de Monitoría Informática de forma veraz y oportuna.			

- Diagrama Actualizar Franja Horario Monitoría Informática



- Iteración Actualizar Franja Horario Monitoría Informática

<b>IDENTIFICADOR:</b> AHMII.3		<b>NOMBRE:</b> Actualizar Franja Horario Monitoría Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Horario Monitoría Informática Consultar Empleados Consultar Cargos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Día, franja, cargo, empleado.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de prestación del servicio en las franjas del horario de Monitoría Informática. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b>			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción actualizar franjas de Horario de Monitoría Informática.	2. Solicita el día del horario a actualizar.	3. Selecciona el día del horario.	4. Muestra el listado de todas las franjas asignadas al horario.
5. Selecciona la franja a actualizar.	6. Da la orden para mostrar las opciones para actualizar la franja. (a)	7. Muestra los datos de la franja y permite cambiar el estado, cargo y empleado responsable del servicio.	
8. Selecciona las opciones que se desea modificar.	9. Selecciona la opción actualizar franja de horario. (a)	10. Se pide confirmación para llevar a cabo el proceso.	
11. El actor monitor confirma su decisión.		12. Actualiza el horario modificando la base de datos con la información suministrada.	
		13. Muestra un mensaje indicando que el horario se actualizó con éxito	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la actualización: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos una franja asignada al horario.			
<b>Postcondición:</b> Actualización eficiente de franjas de Horario de Monitoría Informática.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar la información de las franjas de prestación del servicio de Monitoría Informática de forma eficiente y oportuna.			

- Diagrama Inicializar Horario Monitoría Informática



Inicializar Datos  
Horario Monitoría  
Informática

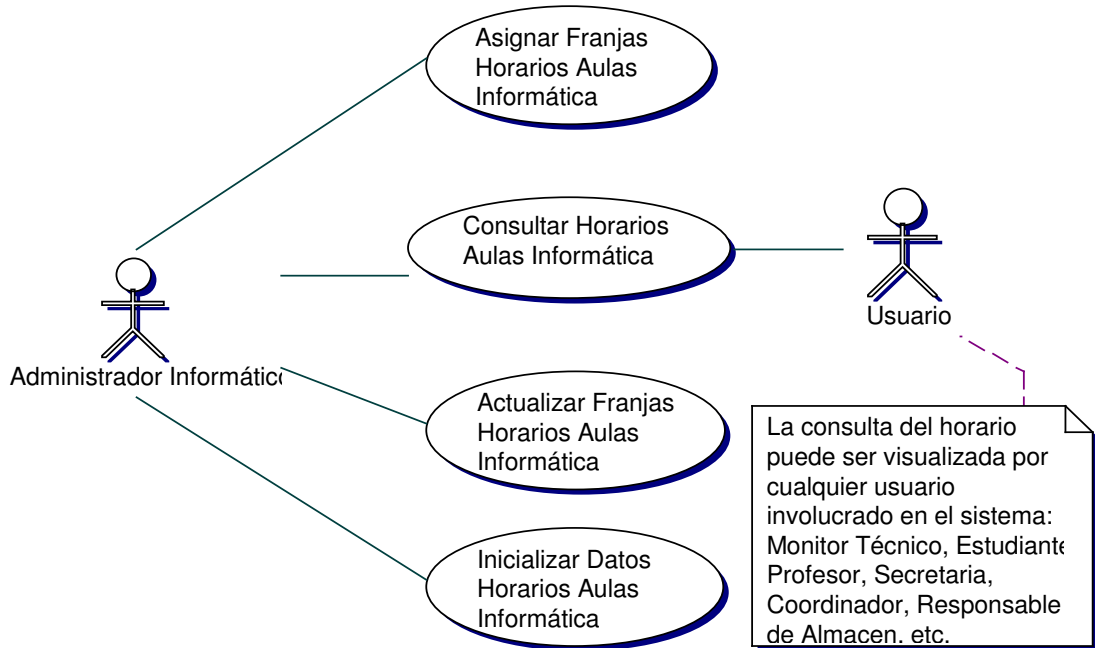
• Iteración Inicializar Horario Monitoría Informática

<b>IDENTIFICADOR:</b> AHMI1.4		<b>NOMBRE:</b> Inicializar Horario Monitoría Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Inicialización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo inicializar los datos de prestación del servicio en las franjas del horario de Monitoría Informática. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> La inicialización permite borrar toda la información del horario de monitoría informática y liberar las franjas para que se puedan volver a asignar al horario y crearlo nuevamente.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción inicializar Horario de Monitoría Informática.	2. Muestra un mensaje informando que se borrarán todos los datos.	3. Selecciona la opción inicializar horario. (a)	4. Se pide confirmación para llevar a cabo el proceso.
5. El actor monitor confirma su decisión.	6. inicializa el horario modificando la base de datos.	7. Muestra un mensaje indicando que el horario se inicializó con éxito	
<b>CAMINOS DE EXCEPCION:</b> (a) El actor decide cancelar la inicialización: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente. Postcondición: Inicialización eficiente de franjas de horario de monitoría informática.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra inicializar la información de las franjas de prestación del servicio de Monitoría Informática de forma eficiente y oportuna.			

9.1.6 Administración de Horarios de Aulas de Informatica

9.1.6.1 Primera Iteración - Fachada

- Diagrama



- Iteración

<b>IDENTIFICADOR:</b> AHAI1		<b>NOMBRE:</b> Administración de Horarios de Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Asignar Franjas Horarios Aulas de Informática Consultar Horarios Aulas de Informática Actualizar Franjas Horarios Aulas de Informática Inicializar Datos Horarios Aulas de Informática		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático, Usuario.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las franjas de los Horarios de Aulas de Informática, especificando el programa, semestre y profesor asignado a cada franja o informando si en dicha franja el aula se encuentra disponible. El proceso de administración se desarrollará teniendo en cuenta las operaciones de asignación, consulta, actualización e inicialización.			
<b>OBSERVACIONES:</b> El actor administrador informático puede administrar los horarios de aulas de informática sin ninguna restricción. El actor usuario puede ser cualquier actor involucrado en el sistema y tendrá acceso solamente a la consulta de los horarios de aulas de informática.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			

*Precondición: El actor ha iniciado una sesión en la aplicación.*

*Postcondición: Administración eficiente de las franjas de Horarios de Aulas de informática.*

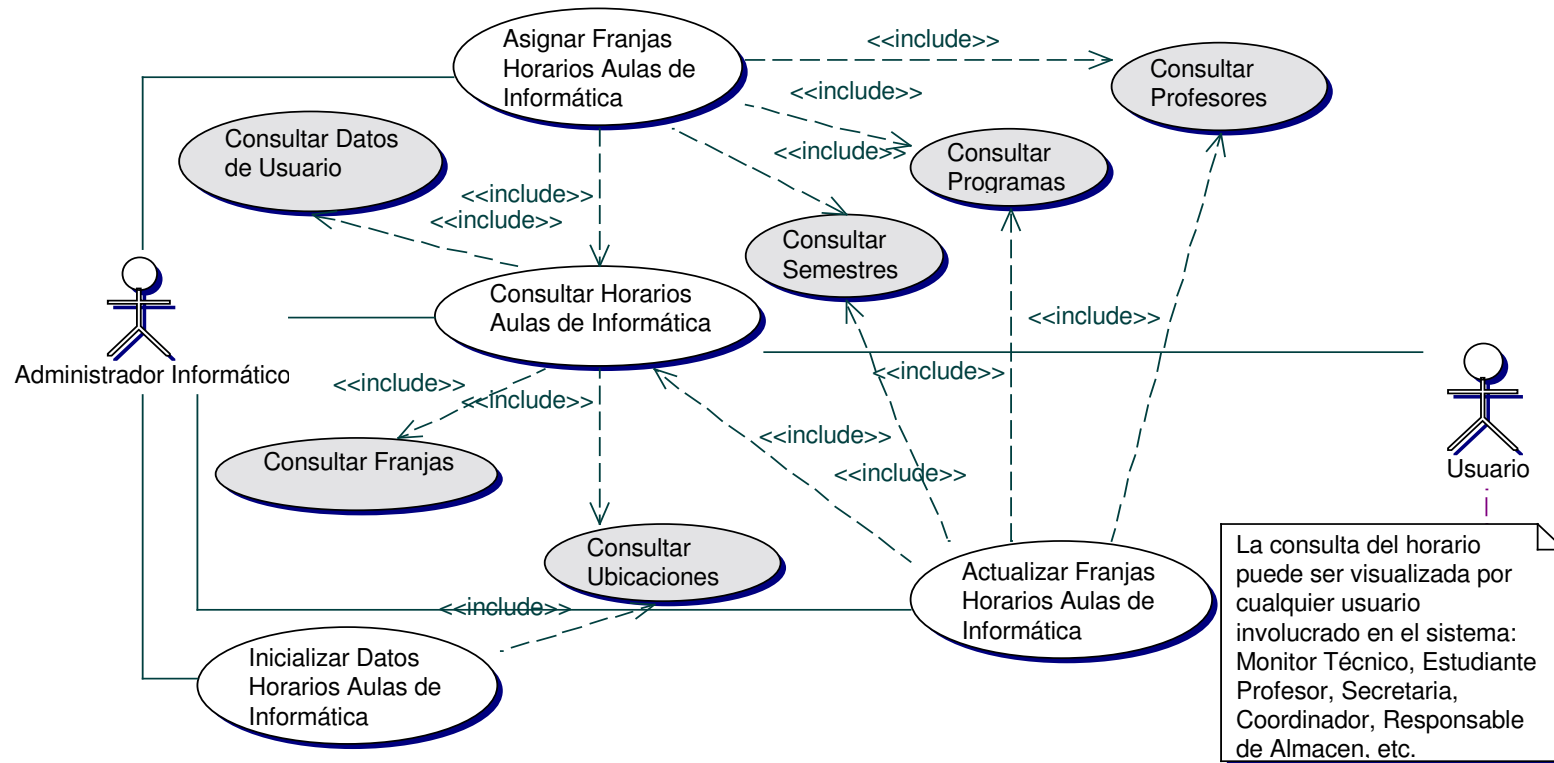
***CRITERIOS DE ACEPTACIÓN***

*Permite llevar una administración sistemática de las franjas disponibles y ocupadas de los horarios de aulas de informática.*

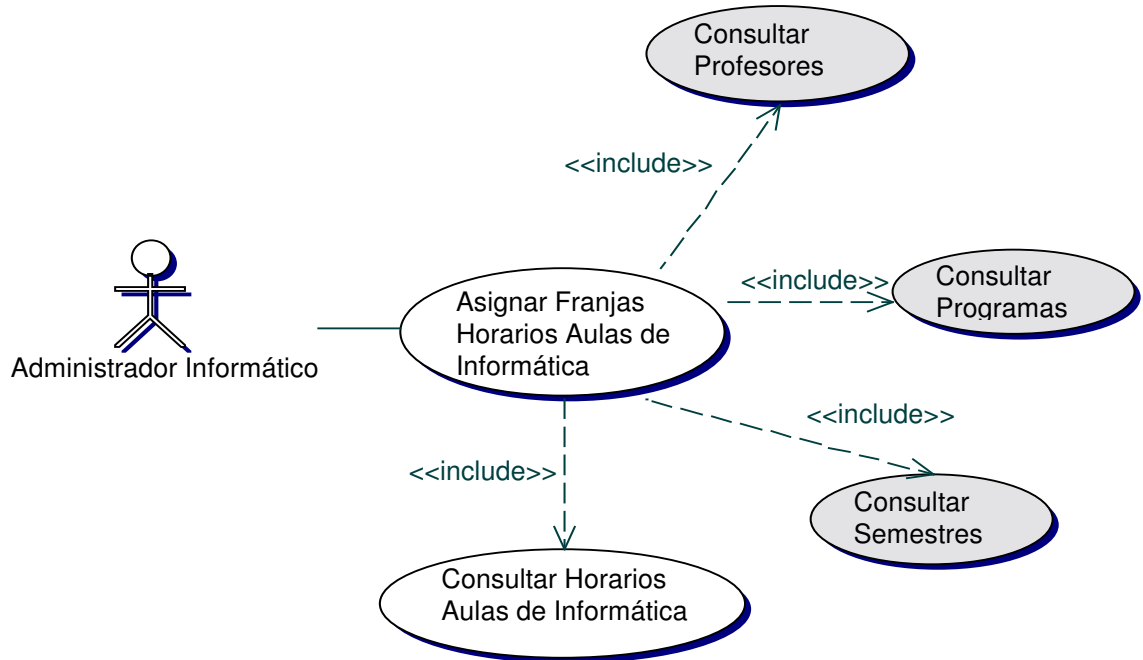
*La información en la base de datos se maneja de forma persistente y consistente.*

*Información oportuna de las franjas disponibles y ocupadas de los horarios de aulas de Informática.*

### 9.1.6.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Asignar Franjas Horarios Aulas de Informática



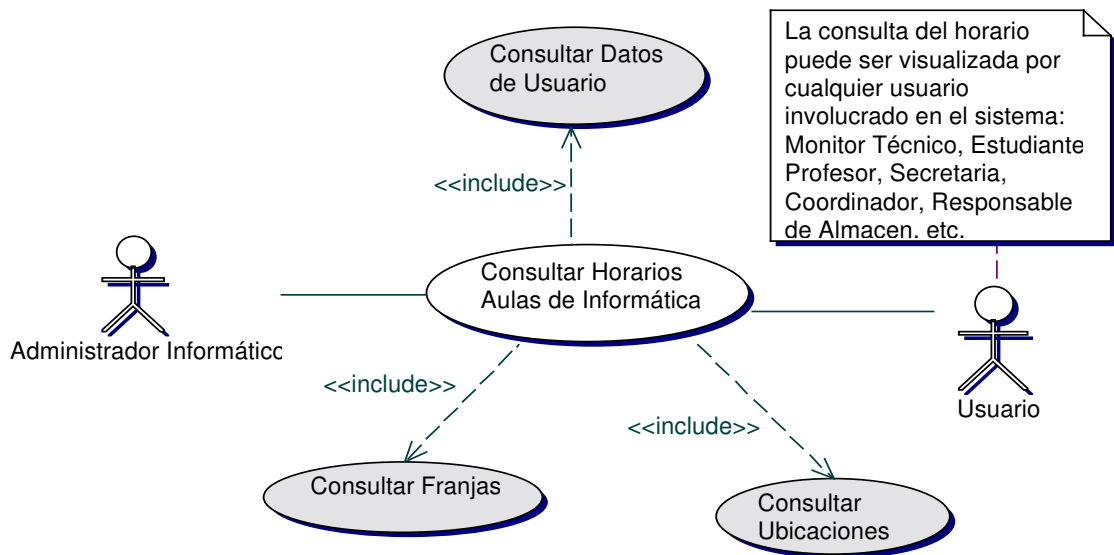
- Iteración Asignar Franjas Horario Aulas Informática

<b>IDENTIFICADOR:</b> AHAI1.1		<b>NOMBRE:</b> Asignar Franjas Horarios Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Horario Aulas de Informática Consultar Profesores Consultar Programas Consultar Semestres		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Aula, día, Programa, Semestre, Profesor, Uso Internet, Franjas.		<b>SALIDA:</b> Proceso efectuada o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo asignar las franjas de los Horarios de Aulas de Informática definiendo si el aula se encuentra ocupada con clases o disponible.			
<b>OBSERVACIONES:</b> Se podrán asignar al horario las franjas que aún no se hayan utilizado en el horario, si ya están utilizadas se deberá inicializar el horario para liberar las franjas y volverlas a asignar. El Uso Internet permite especificar si el aula esta ocupada y hace uso del servicio de Internet.			



<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción asignar franjas horarios de aulas de informática.	2. Solicita el Aula y el día del horario a actualizar.
3. El actor selecciona el aula y el día solicitados.	4. Actualiza el listado de franjas disponibles y permite seleccionar el programa, semestre, profesor y uso Internet que utilizará dichas franjas.
5. Selecciona los datos solicitados.	7. Registra las franjas de horario actualizando la base de datos con la información suministrada.
6. Da la orden para asignar los datos y las franjas al horario (a)	8. Muestra un mensaje indicando que la asignación se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor iniciador decide cancelar la asignación: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente. deben existir franjas libres para asignar al horario.	
Postcondición: Asignación eficiente de las franjas del Horario de Monitoría Informática	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra asignar los datos y franjas del horario de aulas de informática de forma veraz y oportuna.	
La información de la base de datos se almacena de forma persistente y consistente.	

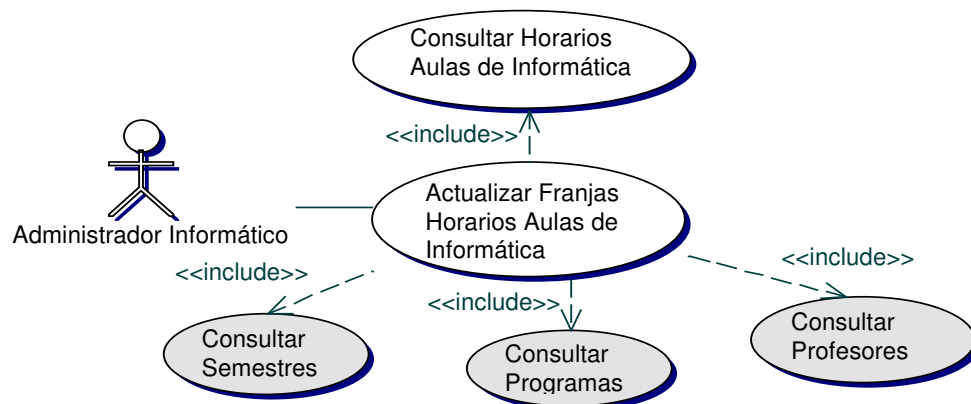
• **Diagrama Consultar Horario Aulas de Informática**



- Iteración Consultar Horario Aulas de Informática

<b>IDENTIFICADOR:</b> AHAI.2		<b>NOMBRE:</b> Consultar Horario Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario Consultar Franjas Consultar Ubicaciones		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático, Usuario.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Aula		<b>SALIDA:</b> Datos de Horario de Aula de informática	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las franjas horarias de los horarios de aulas de informática, especificando si se encuentra disponible u ocupada en clases.			
<b>OBSERVACIONES:</b> La consulta del horario de Monitoría Informática puede ser visualizada por cualquier usuario involucrado en el sistema.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar Horario de Aulas de Informática.	2. Solicita el aula de la que se desea consultar el horario.		
3. Selecciona el aula.	5. Muestra un listado con la información de todas las franjas del horario del aula seleccionada.		
4. Da la orden para visualizar el horario del aula.			
<b>CAMINOS DE EXCEPCION:</b>			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente.			
<b>Postcondición:</b> Información oportuna de los datos de los horarios de Aulas de Informática.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar la información de las franjas de los horarios de Aulas Informática de forma veraz y oportuna.			

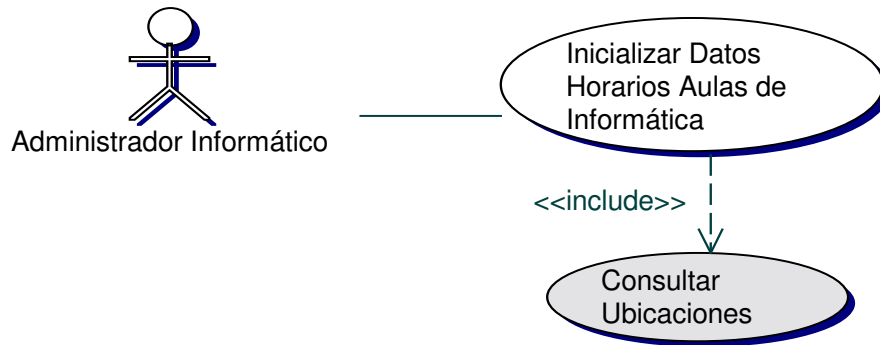
- Diagrama Actualizar Franja Horarios Aulas de Informática



- Iteración Actualizar Franja Horarios Aulas de Informática

<b>IDENTIFICADOR:</b> AHAI1.3		<b>NOMBRE:</b> Actualizar Franja Horarios Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Horarios Aulas de Informática Consultar Profesores Consultar Semestres Consultar Programas		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Día, franja, programa, semestre, profesor, uso Internet.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de las franjas de los horarios de aulas de informática. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> El Uso Internet permite especificar si el aula esta ocupada y hace uso del servicio de Internet.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción actualizar franjas de Horarios de Aulas de Informática.		2. Solicita el día y el aula del horario a actualizar.	
3. Selecciona el día y el aula del horario.		4. Muestra el listado de todas las franjas asignadas al horario de esa aula.	
5. Selecciona la franja a actualizar.			
6. Da la orden para mostrar las opciones para actualizar la franja. (a)		7. Muestra los datos de la franja y permite cambiar el programa, semestre, profesor y uso internet asignado a la franja.	
8. Selecciona las opciones que se desea modificar.			
9. Selecciona la opción actualizar franja de horario. (a)		10. Se pide confirmación para llevar a cabo el proceso.	
11. El actor monitor confirma su decisión.		12. Actualiza el horario modificando la base de datos con la información suministrada.	
		13. Muestra un mensaje indicando que el horario se actualizó con éxito	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la actualización: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos una franja asignada al horario.			
<b>Postcondición:</b> Actualización eficiente de franjas de Horarios de Aulas de Informática.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar la información de las franjas de horarios de Aulas de Informática de forma eficiente y oportuna.			

- Diagrama Inicializar Horarios Aulas de Informática



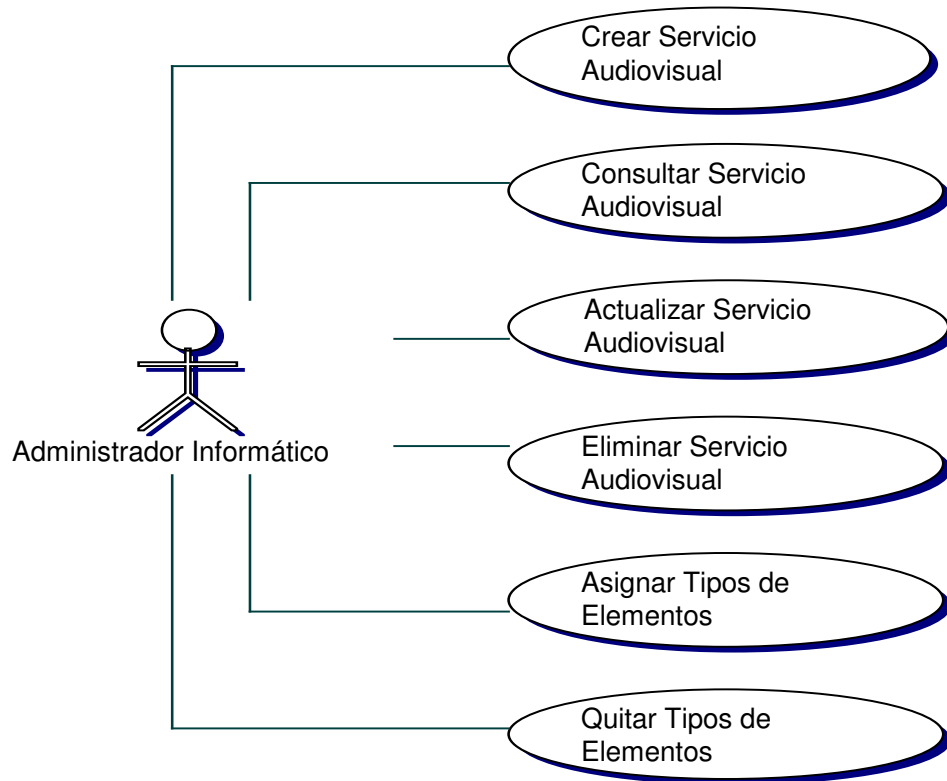
• Iteración Inicializar Horarios Aulas de Informática

<b>IDENTIFICADOR:</b> AHAI.4		<b>NOMBRE:</b> Inicializar Horarios Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Aula		<b>SALIDA:</b> Inicialización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo inicializar los datos de los horarios de Aulas de Informática. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> La inicialización permite borrar toda la información de los horarios de aulas de informática y liberar las franjas para que se puedan volver a asignar al horario y crearlo nuevamente.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción inicializar Horarios de Aulas de Informática.	2. Solicita el aula de la que se desea inicializar el horario.	3. Selecciona el aula deseada.	4. Se pide confirmación para llevar a cabo el proceso.
4. Selecciona la opción inicializar horario. (a)	5. inicializa el horario modificando la base de datos.	5. El actor monitor confirma su decisión.	6. Muestra un mensaje indicando que el horario se inicializó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la inicialización: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Inicialización eficiente de franjas de horarios de aulas de informática.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra inicializar la información de las franjas de los horarios de las aulas de Informática de forma eficiente y oportuna.			

9.1.7 Administración de Servicios Audiovisuales

### 9.1.7.1 Primera Iteración - Fachada

- Diagrama

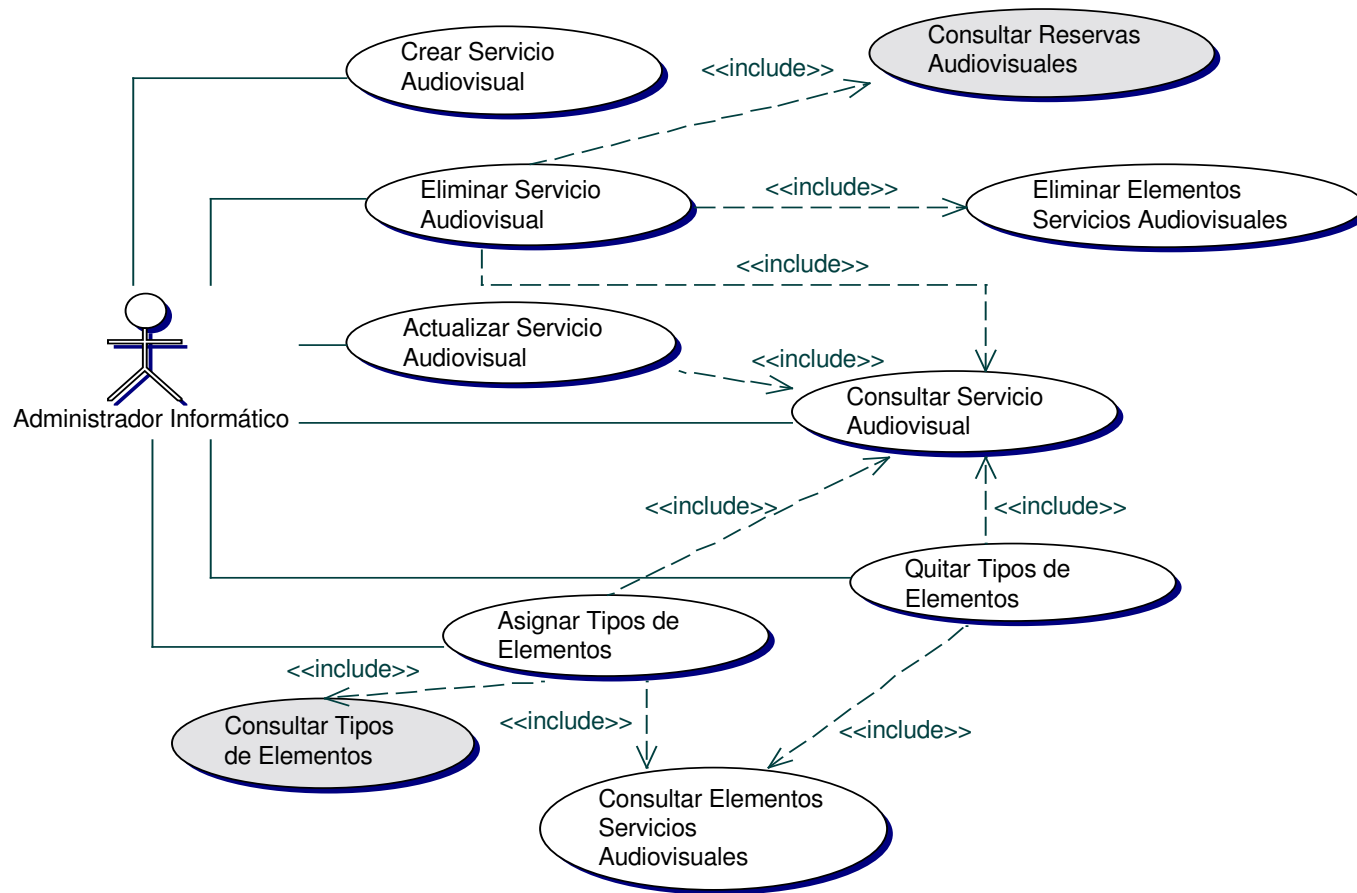


- Iteración

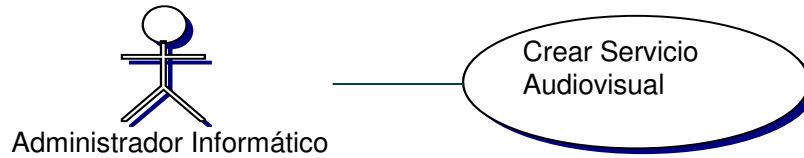
<b>IDENTIFICADOR:</b> ASAUD1		<b>NOMBRE:</b> Administración de Servicios Audiovisuales	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear servicio Audiovisual Consultar servicio Audiovisual Actualizar servicio Audiovisual Eliminar servicio Audiovisual Asignar Tipos de Elementos Quitar Tipos de Elementos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b>			

<p><i>Se satisface este requerimiento permitiendo administrar los Servicios Audiovisuales existentes en la Universidad. Los servicios audiovisuales involucran elementos que permitan realizar exposiciones o presentaciones visuales, y cada servicio puede estar asociado con diferentes tipos de elementos audiovisuales necesarios para la realización de la presentación.</i></p> <p><i>El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización, eliminación y asignación de tipos de elementos al servicio.</i></p>
<p><b>OBSERVACIONES:</b> <i>El actor administrador informático puede administrar los servicios audiovisuales sin ninguna restricción.</i></p>
<p><b>TRIGGERS:</b></p> <p><i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i></p>
<p><i>Precondición: El actor ha iniciado una sesión en la aplicación.</i></p> <p><i>Postcondición: Administración eficiente de los servicios audiovisuales.</i></p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p><i>Permite llevar una administración sistemática de los servicios audiovisuales y los tipos de elementos asociados a ellos.</i></p> <p><i>La información en la base de datos se maneja de forma persistente y consistente.</i></p>

### 9.1.7.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Crear Servicio Audiovisual

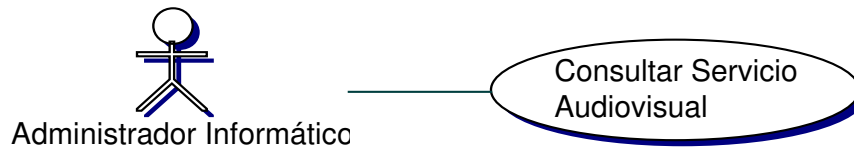


- Iteración Crear Servicio Audiovisual

<b>IDENTIFICADOR:</b> ASAUD1.1		<b>NOMBRE:</b> Crear Servicio Audiovisual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Servicio Audiovisual, Estado.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear un servicio audiovisual, determinando si el servicio quedará activo o inactivo para su uso.			
<b>OBSERVACIONES:</b> Los servicios que se creen como inactivos no quedarán habilitados para ser reservados.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1.	EL actor selecciona la opción crear servicio audiovisual.	2.	Solicita la descripción del servicio audiovisual y el estado del mismo.
3.	El actor suministra los datos solicitados.	5.	Registra el servicio audiovisual actualizando la base de datos con la información suministrada.
4.	Da la orden para crear el servicio audiovisual. (a)	6.	Muestra un mensaje indicando que la creación se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la creación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información en la base de datos es consistente			
<b>Postcondición:</b> Creación eficiente de Servicios Audiovisuales			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra crear los servicios audiovisuales de forma veraz y oportuna.			
La información de la base de datos se almacena de forma persistente y consistente.			

- Diagrama Consultar Servicio Audiovisual

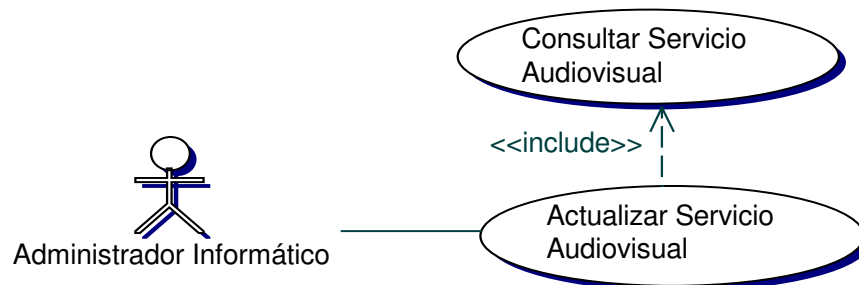




• Iteración Consultar Servicio Audiovisual

<b>IDENTIFICADOR:</b> ASAUD1.2		<b>NOMBRE:</b> Consultar Servicio Audiovisual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático, Usuario.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Listado de Servicios Audiovisuales Existentes	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los servicios audiovisuales existentes.			
<b>OBSERVACIONES:</b> Se desplegarán los servicios audiovisuales activos e inactivos.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar Servicios Audiovisuales.		2. Muestra un listado con la información de todos los servicios audiovisuales existentes.	
<b>CAMINOS DE EXCEPCION:</b>			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente.			
<b>Postcondición:</b> Información oportuna de los servicios audiovisuales.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar la información de los servicios audiovisuales de forma veraz y oportuna.			

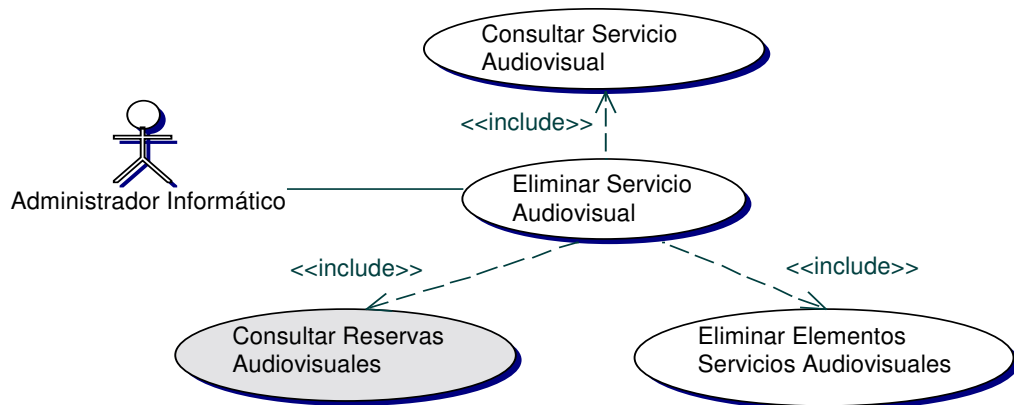
• Diagrama Actualizar Servicio Audiovisual



• Iteración Actualizar Servicio Audiovisual

<b>IDENTIFICADOR:</b> AHMI1.3		<b>NOMBRE:</b> Actualizar Servicio Audiovisual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicio Audiovisual		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Servicio Audiovisual, estado.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar la descripción del servicio audiovisual y el estado activo o inactivo. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> Los servicios que se actualicen a inactivos no quedarán habilitados para ser reservados.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor selecciona la opción actualizar servicios audiovisuales.	2.	Permite seleccionar el servicio audiovisual que se desea actualizar.
3.	Selecciona el servicio Audiovisual.	4.	Muestra los datos del servicio, como son descripción y estado.
5.	Actualiza los datos deseados		
6.	Selecciona la opción actualizar servicio. (a)	7.	Se pide confirmación para llevar a cabo el proceso.
8.	El actor monitor confirma su decisión.	9.	Actualiza el servicio modificando la base de datos con la información suministrada.
		10.	Muestra un mensaje indicando que el servicio se actualizó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la actualización: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos un servicio audiovisual.			
<b>Postcondición:</b> Actualización eficiente de servicios audiovisuales.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar los servicios audiovisuales de forma eficiente y oportuna.			

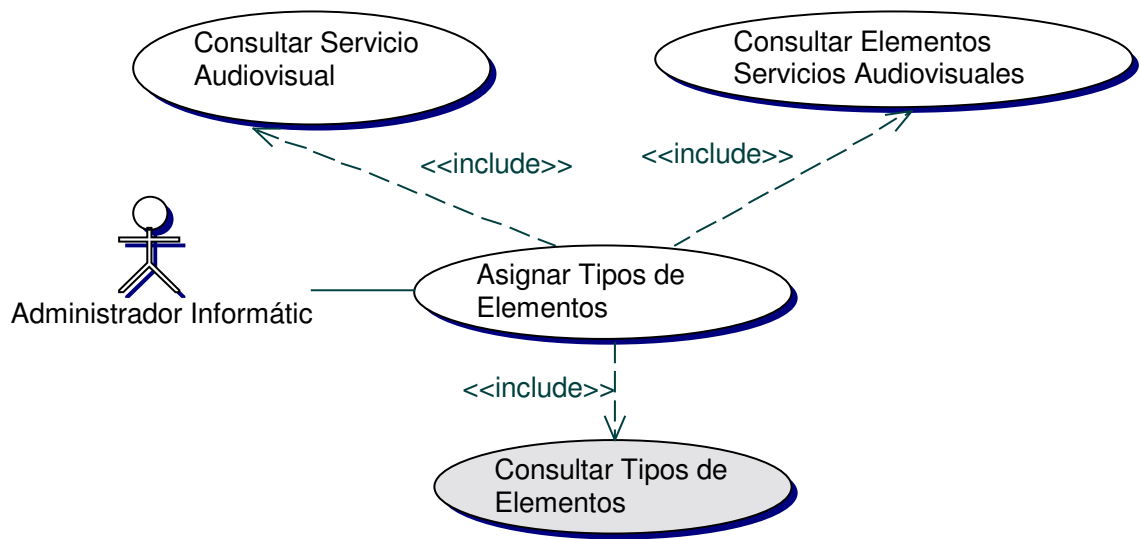
## • Diagrama Eliminar Servicio Audiovisual



- Iteración Eliminar Servicio Audiovisual

<b>IDENTIFICADOR:</b> ASAUDI.4		<b>NOMBRE:</b> Eliminar Servicio Audiovisual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicio Audiovisual Consultar Reservas Audiovisuales Eliminar Elementos Servicios Audiovisuales		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Servicio a eliminar		<b>SALIDA:</b> servicio eliminado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un determinado servicio audiovisual. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> Para eliminar un servicio audiovisual, se verifica que el servicio no haya sido involucrado en una reserva de audiovisuales, si es así no se podrá efectuar la eliminación. Si el servicio se puede eliminar, automáticamente también se eliminarán la asignación de tipos de elementos que ese servicio audiovisual posea.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción eliminar servicio audiovisual.		2. Muestra el listado de todos los servicios audiovisuales existentes.	
3. Selecciona el servicio audiovisual a eliminar.		5. Se pide confirmación para llevar a cabo el proceso.	
4. Selecciona la opción eliminar servicio. <b>(a)</b>		7. Verifica que el servicio audiovisual no se encuentre involucrado en una reserva audiovisual. <b>(b)</b>	
6. El actor monitor confirma su decisión.		8. Elimina el servicio actualizando la base de datos.	
		9. Muestra un mensaje indicando que el servicio se eliminó con éxito	
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El actor decide cancelar la inicialización: El sistema cancela el proceso.			
<b>(b)</b> El Servicio Audiovisual se encuentra involucrado en una reserva audiovisual: El sistema indica este suceso a través de un mensaje y cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>TRIGGERS:</b> Eliminar Elementos Servicios Audiovisuales que estén asignados al servicio a eliminar: El sistema verifica los tipos de elementos que se encuentren asignados al servicio audiovisual y elimina los registros de esta asignación.			
<b>Precondición:</b> La información de la base de datos es consistente. El Servicio audiovisual no debe encontrarse involucrado en una reserva audiovisual.			
<b>Postcondición:</b> Eliminación eficiente de Servicios de Audiovisuales.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra eliminar los servicios audiovisuales de forma consistente.			

- Diagrama Asignar Tipos de Elementos

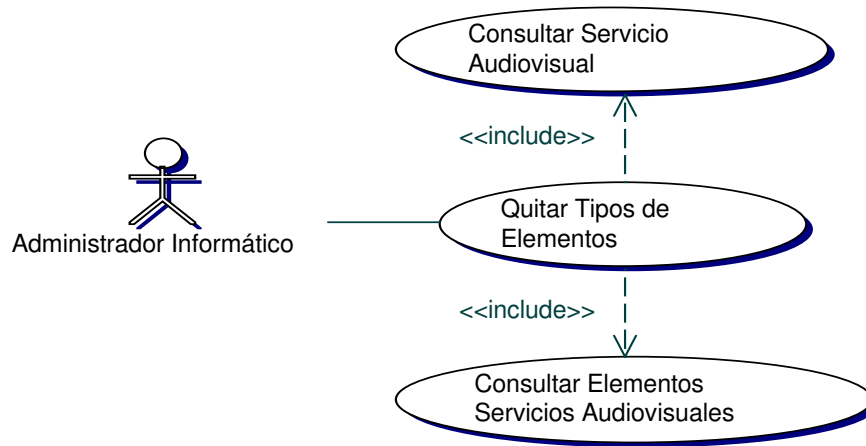


- Iteración Asignar Tipos Elementos

<b>IDENTIFICADOR:</b> ASAUDI.5		<b>NOMBRE:</b> Asignar Tipos de Elementos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicio Audiovisual Consultar Elementos Servicios Audiovisuales Consultar Tipos de Elementos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> servicio audiovisual, tipo de elemento a asignar		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo asignar a un determinado servicio audiovisual los tipos de elementos que ese servicio involucra para su realización. Un servicio Audiovisual puede estar conformado por diferentes elementos audiovisuales que se asocian al servicio audiovisual por el tipo de elemento al que pertenecen. Esta asignación permite determinar los elementos que involucra el servicio para establecer si se encuentran o no disponibles para que el servicio audiovisual pueda ser reservado.			
<b>OBSERVACIONES:</b> Se puede asignar al servicio audiovisual varios tipos de elementos, pero no se puede repetir el mismo tipo de elemento para un mismo servicio.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. EL actor selecciona la opción asignar tipos de elementos.</li> <li>3. El actor selecciona el servicio audiovisual.</li>   <li>5. Selecciona el tipo de elemento a asignar.</li> <li>6. Da la orden para asignar el tipo de elemento (a)</li> </ol>	<ol style="list-style-type: none"> <li>2. Permite seleccionar el servicio audiovisual al que se le asignaran los tipos de elementos.</li> <li>4. Despliega los tipos de elementos que actualmente tiene asignado el servicio y permite seleccionar otro tipo de elemento para asignar.</li>   <li>7. Asigna el tipo de elemento actualizando la base de datos con la información suministrada.</li> <li>8. Muestra un mensaje indicando que la asignación se realizó con éxito</li> </ol>
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor iniciador decide cancelar la asignación: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente.	
Postcondición: Asignación eficiente de los tipos de elementos a un servicio audiovisual.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra asignar los tipos de elementos adecuados para un determinado servicio audiovisual de forma veraz y oportuna.	
La información de la base de datos se almacena de forma persistente y consistente.	

• **Diagrama Quitar Tipos de Elementos**



• **Iteración Quitar Tipos Elementos**

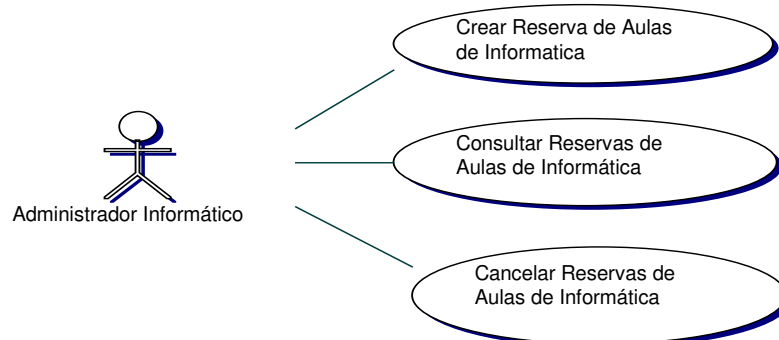
<b>IDENTIFICADOR:</b> ASAUD1.6		<b>NOMBRE:</b> Quitar Tipos de Elementos
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicio Audiovisual Consultar Elementos Servicios Audiovisuales	<b>CRITICO?</b> Si

<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> servicio audiovisual, tipo de elemento a quitar	<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo quitar a un determinado servicio audiovisual los tipos de elementos que ese servicio involucra para su realización.		
<b>OBSERVACIONES:</b>		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción quitar tipos de elementos.		2. Permite seleccionar el servicio audiovisual al que se le quitarán los tipos de elementos.
3. El actor selecciona el servicio audiovisual.		4. Despliega el listado de todos los tipos de elementos que tiene asignado ese servicio audiovisual.
5. Selecciona el tipo de elemento a quitar.		7. Quita el tipo de elemento actualizando la base de datos con la información suministrada.
6. Da la orden para quitar el tipo de elemento (a)		8. Muestra un mensaje indicando que el procesose realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>		
(a) El actor iniciador decide cancelar el proceso: El sistema cancela el proceso.		
• El sistema no responde. El proceso no se efectúa.		
Precondición: La información de la base de datos es consistente.		
Postcondición: Desactivación eficiente de los tipos de elementos de un servicio audiovisual.		
<b>CRITERIOS DE ACEPTACIÓN</b>		
Se cumple el requerimiento cuando se logra quitar los tipos de elementos que tiene un determinado servicio audiovisual de forma veraz y oportuna.		
La información de la base de datos se almacena de forma persistente y consistente.		

## 9.1.8 Administración de Reservas de Aulas de Informatica

### 9.1.8.1 Primera Iteración - Fachada

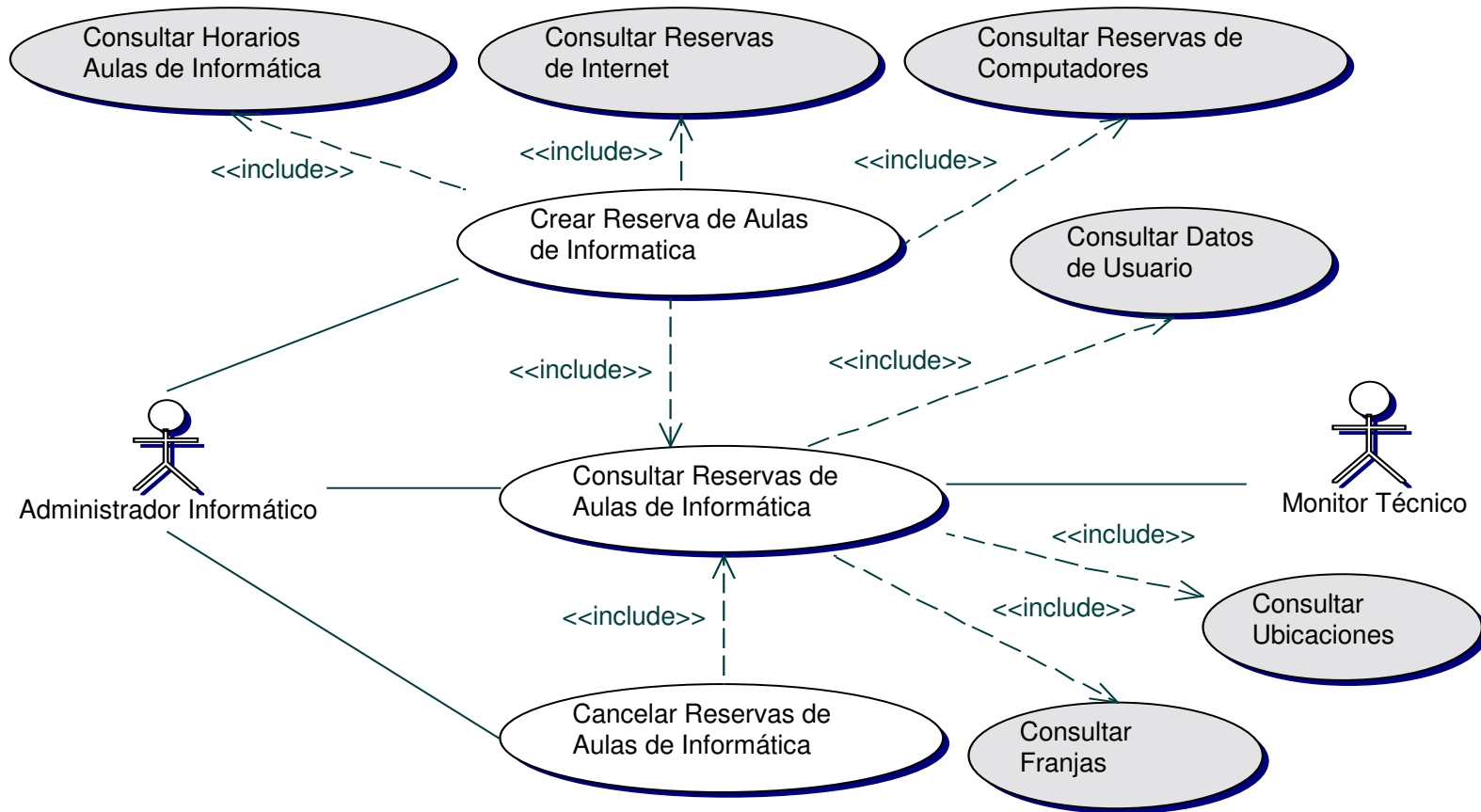
- Diagrama



- Iteración

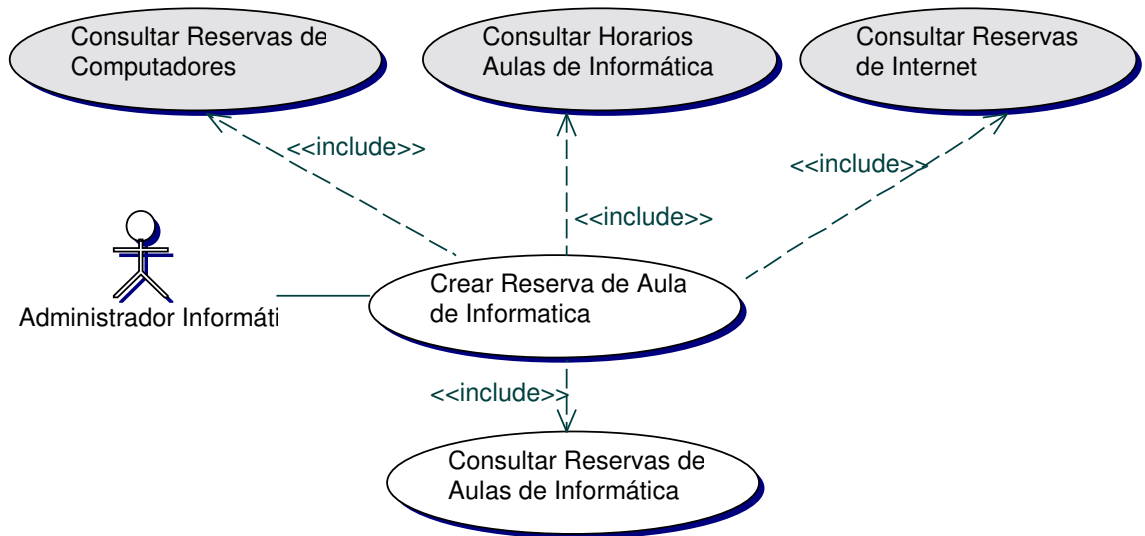
<b>IDENTIFICADOR:</b> ARAINFI		<b>NOMBRE:</b> Administración de Reservas de Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Reserva de Aulas de Informática Consultar Reservas de Aulas de Informática Cancelar Reserva de Aulas de Informática		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador: Estudiante, Profesor. Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las reservas de aulas de informática. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta y eliminación de reservas de aulas de informática.			
<b>OBSERVACIONES:</b> El actor administrador informático puede administrar las reservas de aulas de informática sin ninguna restricción.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: El actor ejecutor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de las reservas de aulas de informática.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de las reservas de aulas de informática. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.8.2 Segunda Iteración – Llenado (Diagrama General)





- **Diagrama Crear Reserva de Aulas de informática**

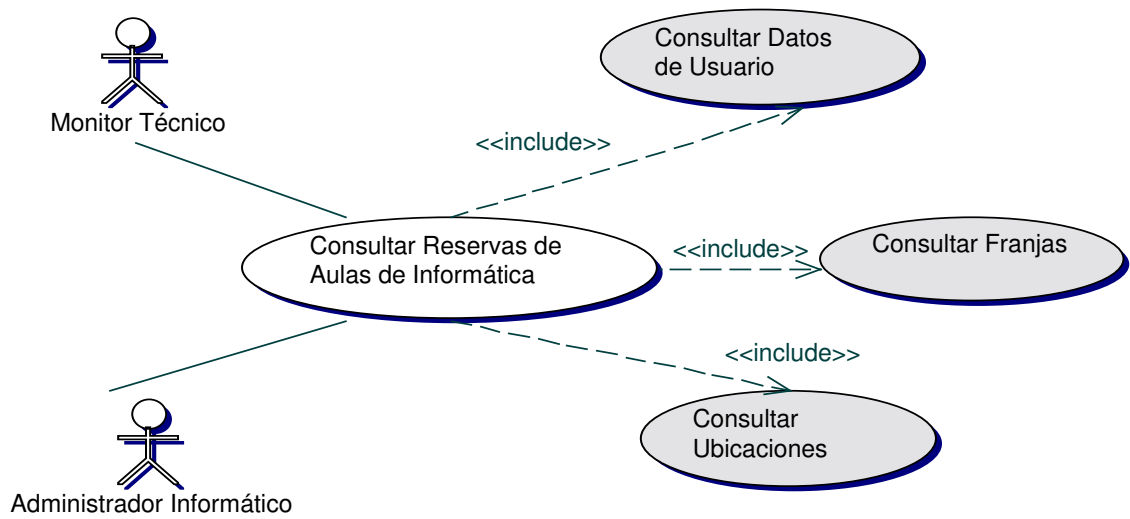


- **Iteración Crear Reservas de Aulas de Informática**

<b>IDENTIFICADOR:</b> ARAINF1.1		<b>NOMBRE:</b> Crear Reservas de Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Horarios Aulas Informática Consultar Reservas de Computadores Consultar reservas de Internet Consultar reservas de Aula de informática		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Profesor, Estudiante. Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, fecha, horario, aula, uso internet, observaciones.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear o registrar una reserva de aulas de informática de acuerdo a la disponibilidad de las mismas. Las reservas permiten separar el aula de informática durante unas franjas de tiempo para dictar clases o algún evento en particular. La reserva permite especificar si la reserva del aula involucrará también que se le preste el servicio de internet.			
<b>OBSERVACIONES:</b> Las observaciones de la reserva permiten especificar alguna eventualidad que se presente durante el uso del aula, o algún requerimiento especial por parte del actor iniciador.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. El actor iniciador solicita al actor ejecutor la creación de una reserva de aulas de informática	3. Solicita la identificación del usuario solicitante de la reserva.
2. EL actor ejecutor selecciona la opción crear reserva de aulas de informática.	5. Verifica si el actor iniciador es un usuario valido de la universidad. <b>(f)</b>
4. El actor ejecutor suministra la identificación.	5. Permite seleccionar la fecha, el horario y el aula para la reserva.
6. El actor ejecutor selecciona los datos para su reserva.	8. Verifica que en el aula seleccionada no se encuentren realizadas reservas de internet o de computadores. <b>(b)</b>
7. El actor ejecutor da la orden para mostrar la disponibilidad del aula. <b>(a)</b>	9. Verifica que el aula no este ocupada con clases. <b>(c)</b>
	10. Verifica que el aula no se encuentre reservada por otro usuario. <b>(d)</b>
	11. Despliega la información sobre la disponibilidad del aula en esa fecha y horario, permitiendo determinar si se utilizará el servicio de internet en la reserva.
12. El actor ejecutor selecciona las opciones de su reserva.	14. Verifica que los datos seleccionados aún se encuentren disponibles para registrar la reserva. <b>(e)</b>
13. Da la orden para crear la reserva del aula. <b>(a)</b>	15. Registra la reserva del aula actualizando la base de datos con la información suministrada.
	16. Muestra un mensaje indicando que la reserva se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor iniciador decide cancelar la creación: El sistema cancela el proceso.	
<b>(b)</b> El aula tiene realizadas reservas de internet o de computadores: El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.	
<b>(c)</b> El aula se encuentra ocupada con clases en el horario de aulas de informática: El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.	
<b>(d)</b> El aula se encuentra reservada por otro usuario: El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.	
<b>(e)</b> Los datos seleccionados ya no se encuentran disponibles. El sistema indica este suceso a través de un mensaje y permite seleccionar nuevos datos para la reserva.	
<b>(f)</b> El actor iniciador no es un usuario valido de la universidad: El sistema indica este suceso a través de un mensaje y cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: El aula no debe encontrarse involucrada en ninguna reserva.	
Postcondición: Creación eficiente de Reservas de Aulas de Informática.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra crear las reservas de aulas de informática de forma veraz y oportuna.	
La información de la base de datos se almacena de forma persistente y consistente.	

- **Diagrama Consultar Reservas de Aulas de Informática**

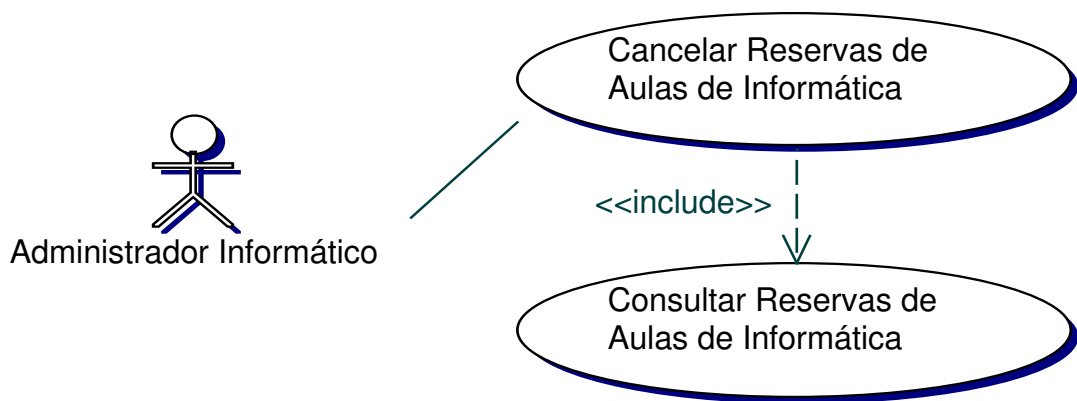


• **Iteración Consultar Reservas de Aulas de Informática**

<b>IDENTIFICADOR:</b> ARAINF1.2		<b>NOMBRE:</b> Consultar Reservas de Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario Consultar Franjas Consultar Ubicaciones		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Administrador Informático, Profesor, Estudiante. Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha.		<b>SALIDA:</b> Listado de Reservas de Aulas de Informática.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las reservas de aulas de informática en una determinada fecha.			
<b>OBSERVACIONES:</b> La consulta permite seleccionar una fecha anterior a la actual para poder acceder al historia de reservas anteriormente realizadas.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. El actor iniciador solicita al actor ejecutor la consulta de una reserva de aulas de informática		3. Solicita la fecha de consulta.	
2. EL actor ejecutor selecciona la opción consultar reserva de aulas de informática.		6. Despliega el listado de todas las reservas realizadas en la fecha seleccionada.	
4. El actor ejecutor selecciona la fecha.			
5. El actor ejecutor da la orden para mostrar las reservas realizadas. (a)			

<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
Precondición: La información de la base de datos es consistente.
Postcondición: Información oportuna de las reservas de Aulas de Informática existentes.
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar la información de las reservas de Aulas de Informática de forma veraz y oportuna.

- Diagrama Cancelar Reservas de Aulas de Informática**



- Iteración Cancelar Reservas de Aulas de Informática**

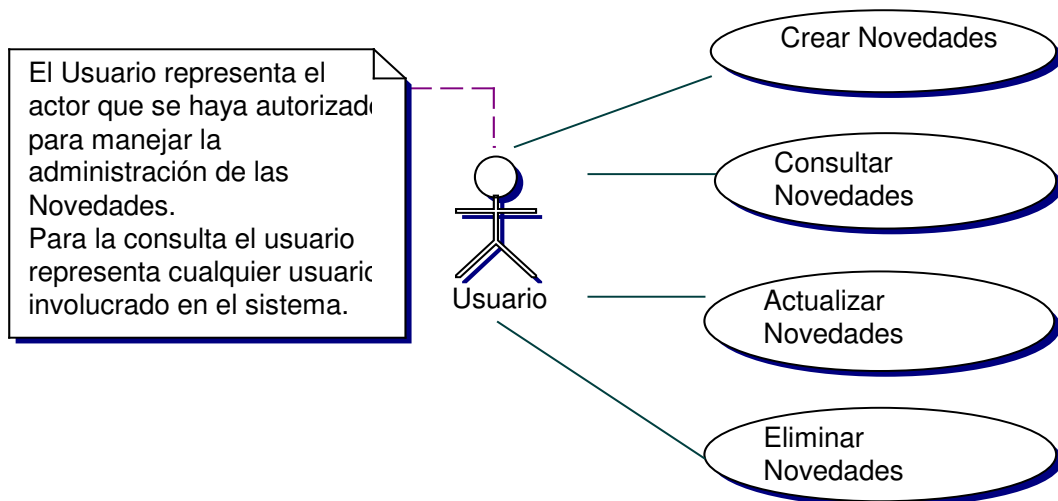
<b>IDENTIFICADOR:</b> ARAINF1.3		<b>NOMBRE:</b> Cancelar Reservas de Aulas de Informática	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Reservas de Aulas de Informática		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Profesor, Estudiante. Ejecutor: Administrador Informático.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, reserva a cancelar.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo cancelar una determinada reserva de Aulas de Informática. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> El actor ejecutor puede cancelar las reservas vigentes de una determinada fecha sin ninguna restricción.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. El actor iniciador solicita al actor ejecutor la cancelación de una reserva de aulas de informática	2. Permite seleccionar la fecha de la reserva a eliminar.
3. EL actor ejecutor selecciona la fecha.	4. Muestra un listado de todas las reservas existentes en la fecha.
5. El actor ejecutor selecciona la reserva a cancelar.	7. Se pide confirmación para llevar a cabo el proceso.
6. El actor ejecutor selecciona la opción cancelar reserva. <b>(a)</b>	9. Cancela la reserva actualizando la base de datos.
8. El actor monitor confirma su decisión. <b>(a)</b>	10. Muestra un mensaje indicando que la reserva se canceló con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente.	
Postcondición: Cancelación satisfactoria de las reservas de Aulas de Informática.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra cancelar las reservas de Aulas de Informática de forma consistente.	

## 9.1.9 Administración de Novedades

### 9.1.9.1 Primera Iteración - Fachada

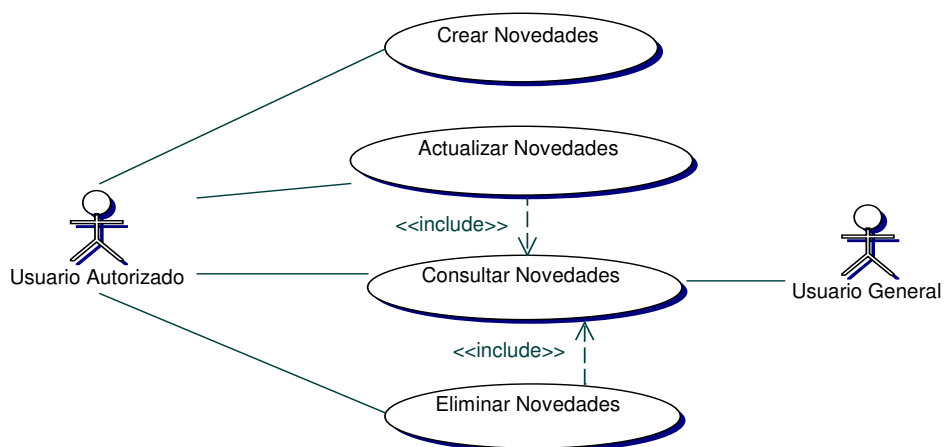
- Diagrama



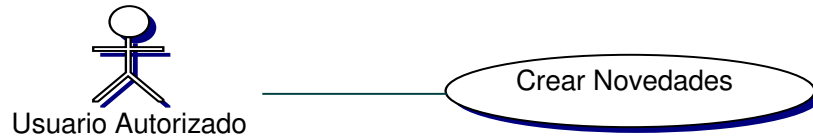
- Iteración

<b>IDENTIFICADOR:</b> ASAUD1		<b>NOMBRE:</b> Administración de Novedades	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Novedades Consultar Novedades Actualizar Novedades Eliminar Novedades		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las Novedades que se quieran informar a los usuarios del sistema. Las novedades permitirán redactar información en general que se quiera dar a conocer a los usuarios que interactúen con el sistema. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y eliminación.			
<b>OBSERVACIONES:</b> El usuario autorizado tendrá acceso a la administración de las novedades sin ninguna restricción. Los usuarios en general podrán acceder solo a la consulta de las novedades activas.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema. Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de las novedades que se quieran dar a conocer.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de las novedades manteniendo informado oportunamente a todos los usuarios del sistema. Permite dar a conocer temas de interés de forma rápida y eficiente. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.9.2 Segunda Iteración – Llenado (Diagrama General)



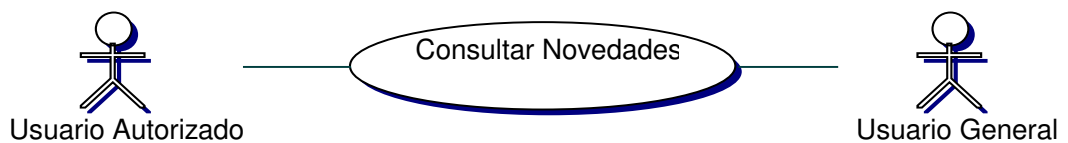
• **Diagrama Crear Novedades**



• **Iteración Crear Novedades**

<b>IDENTIFICADOR:</b> ANOV1.1		<b>NOMBRE:</b> Crear Novedad	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Titulo, Estado, Contenido.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear una novedad estableciendo su titulo y redactando la información que se quiere dar a conocer en la novedad. De igual manera se establecerá si la novedad queda activa o inactiva.			
<b>OBSERVACIONES:</b> Solo las novedades activas quedarán publicadas en la pagina para la consulta de cualquier usuario.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. El actor selecciona la opción crear novedad.	2. Solicita el titulo, el contenido y el estado de la novedad.		
3. El actor suministra los datos solicitados.	5. Registra la novedad actualizando la base de datos con la información suministrada.		
4. Da la orden para crear la novedad (a)	6. Muestra un mensaje indicando que la creación se realizó con éxito		
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la creación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información en la base de datos es consistente			
<b>Postcondición:</b> Creación eficiente de Novedades			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra crear las novedades de forma veraz y oportuna.			
La información de la base de datos se almacena de forma persistente y consistente.			

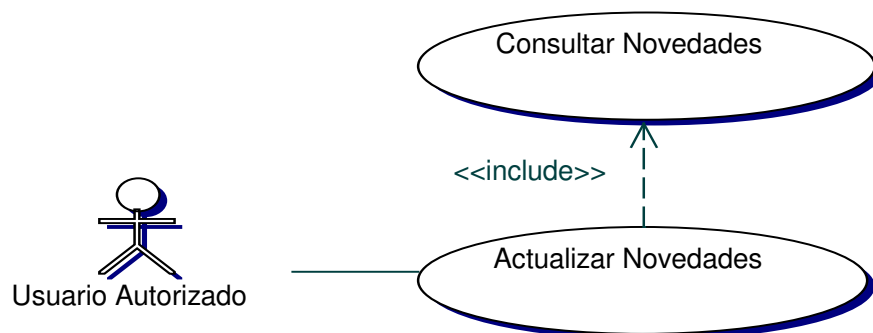
• **Diagrama Consultar Novedades**



- Iteración Consultar Novedades

<b>IDENTIFICADOR:</b> ANOV1.2		<b>NOMBRE:</b> Consultar Novedades	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado, Usuario General.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Titulo Novedad		<b>SALIDA:</b> Listado de Novedades	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las novedades que se hayan creado.			
<b>OBSERVACIONES:</b> Para el usuario autorizado se desplegarán las novedades activas e inactivas; para el usuario general solo las novedades activas.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Consulta efectuada por el usuario autorizado</i>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL usuario autorizado selecciona la opción consultar novedades.	2. Le permite seleccionar el título de la novedad a consultar.		
3. El usuario autorizado selecciona el título para su consulta. (a)	4. Despliega el estado de la novedad y su contenido.		
<i>Consulta efectuada por el usuario general</i>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL usuario general selecciona el título de la novedad a consultar.	2. Despliega el contenido de la novedad seleccionada.		
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente. El usuario ha desplegado la página principal o ha iniciado una sesión en el sistema.			
<b>Postcondición:</b> Información oportuna de las novedades existentes.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar las novedades de forma veraz y oportuna, ofreciendo una forma fácil y eficiente de dar a conocer temas de interés para todos los usuarios del sistema.			

- Diagrama Actualizar Novedades

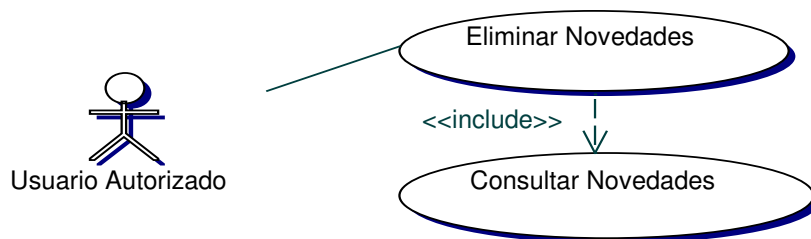




- Iteración Actualizar Novedades

<b>IDENTIFICADOR:</b> ANOV1.3		<b>NOMBRE:</b> Actualizar Novedades	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Novedades		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Novedad a actualizar, título, contenido, estado.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de una novedad como son su título, contenido y estado. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> Las novedades que se actualicen al estado inactivas ya no serán visualizadas por los usuarios en general.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor selecciona la opción actualizar novedad.	2.	Permite seleccionar la novedad que se desea actualizar.
3.	Selecciona la novedad deseada.	4.	Muestra el título, el estado y el contenido de la novedad seleccionada y permite modificar sus datos.
5.	Actualiza los datos deseados	7.	Se pide confirmación para llevar a cabo el proceso.
6.	Selecciona la opción actualizar novedad. (a)	9.	Actualiza la novedad modificando la base de datos con la información suministrada.
8.	El actor confirma su decisión. (a)	10.	Muestra un mensaje indicando que la novedad se actualizó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la actualización: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos una novedad.			
<b>Postcondición:</b> Actualización eficiente de novedades.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar las novedades de forma eficiente y oportuna, permitiendo desactivar las novedades pasadas y actualizarlas con nuevos datos para darlos a conocer.			

- Diagrama Eliminar Novedades



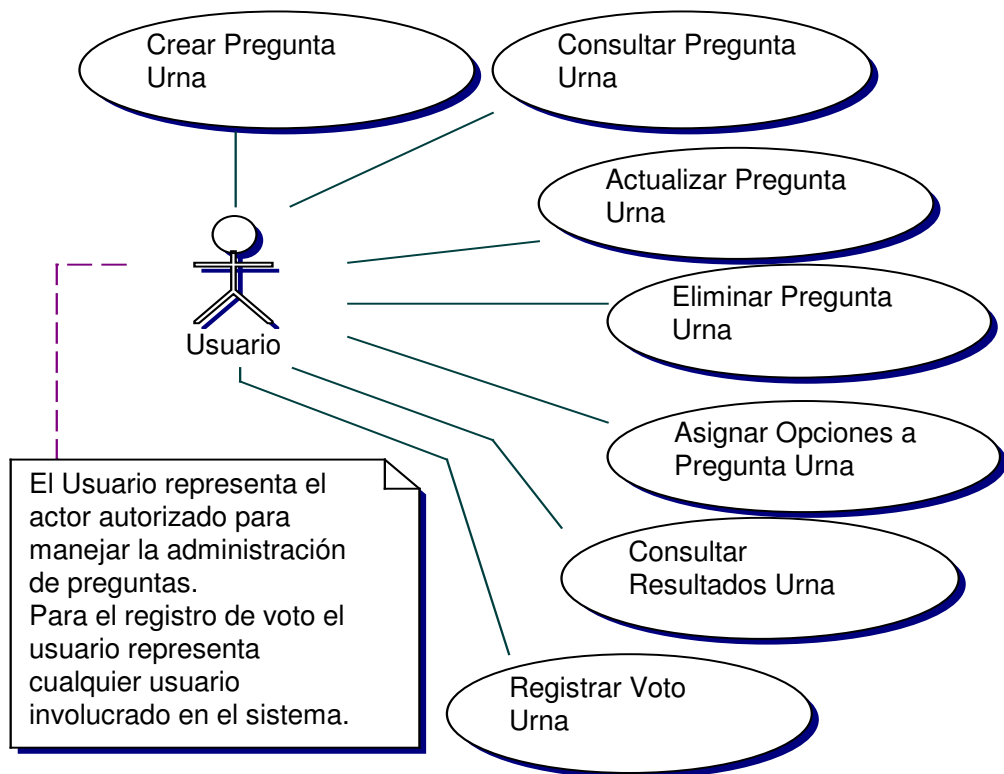
- Iteración Eliminar Novedades

<b>IDENTIFICADOR:</b> ANOV1.4		<b>NOMBRE:</b> Eliminar Novedad	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Novedades		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Novedad a eliminar		<b>SALIDA:</b> Novedad eliminada o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un determinada novedad. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b>			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción eliminar novedad.	2. Permite seleccionar la novedad que se desea eliminar.	3. Selecciona la novedad deseada.	4. Muestra el estado y el contenido de la novedad seleccionada.
5. Selecciona la opción eliminar novedad. (a)	6. Se pide confirmación para llevar a cabo el proceso.	7. El actor confirma su decisión. (a)	8. Elimina la novedad modificando la base de datos con la información suministrada.
	9. Muestra un mensaje indicando que la novedad se eliminó con éxito		
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la eliminación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos una novedad creada.			
<b>Postcondición:</b> Eliminación eficiente de Novedades.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar las novedades de forma consistente.			

## 9.1.10 Administración de Preguntas de Urna Virtual

### 9.1.10.1 Primera Iteración - Fachada

- Diagrama

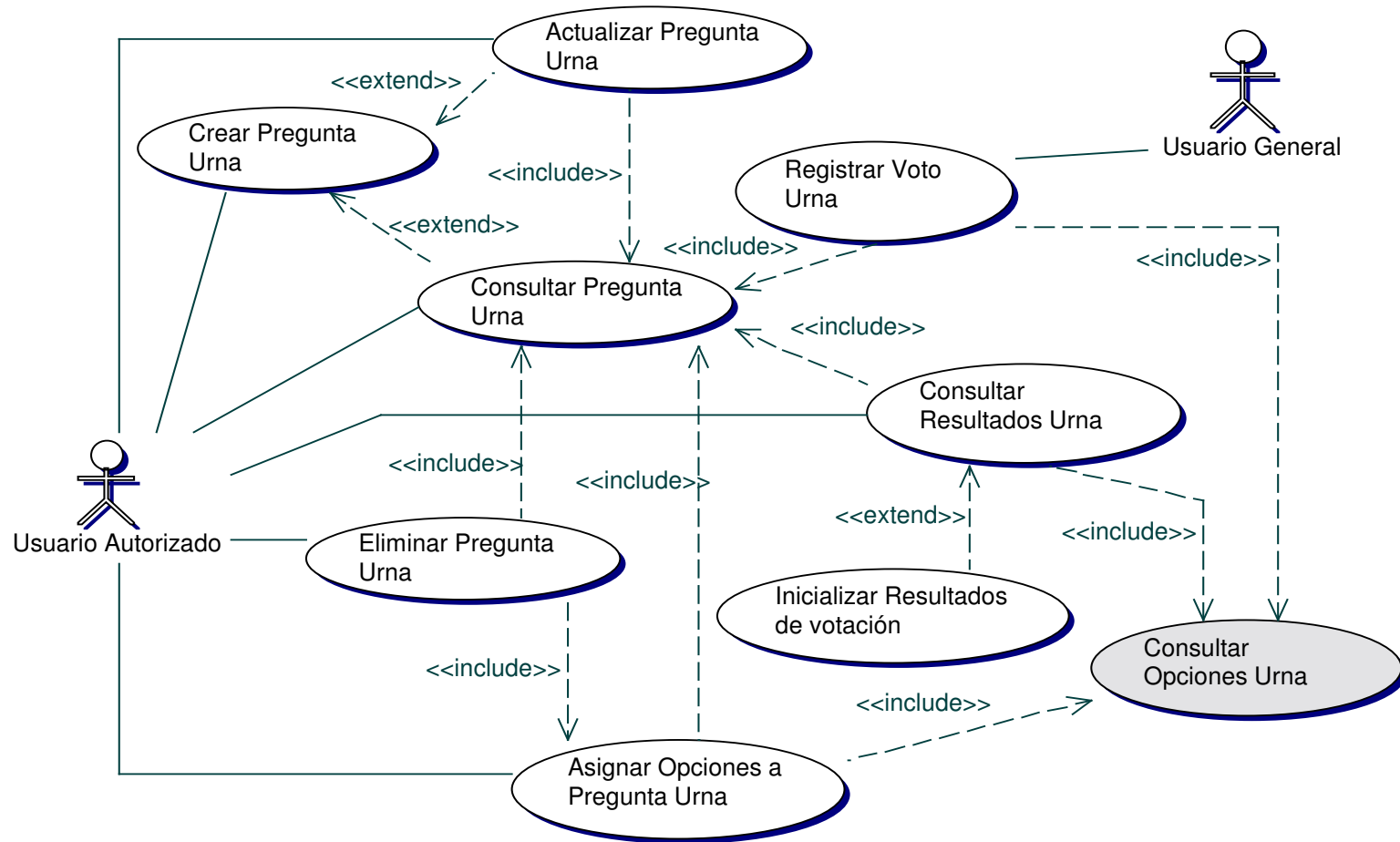


- Iteración

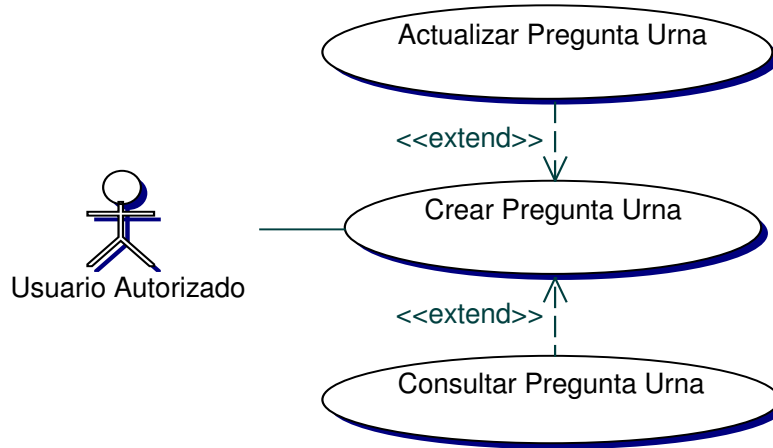
<b>IDENTIFICADOR:</b> APURNA1		<b>NOMBRE:</b> Administración de Preguntas de Urna Virtual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Pregunta Urna Consultar Pregunta Urna Actualizar Pregunta Urna Eliminar Pregunta Urna Asignar Opciones a Pregunta Urna Consultar Resultados Urna Registrar Voto Urna		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las preguntas de la urna virtual estableciendo las opciones que hacen parte de las preguntas, registrando los votos de los usuarios y verificando los resultados de la votación obtenidos.			
<b>OBSERVACIONES:</b> El usuario autorizado tendrá acceso a la administración de las novedades sin ninguna restricción. Los usuarios en general podrán acceder solo a la votación de las opciones de la pregunta.			

<b>TRIGGERS:</b> <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i>
<i>Precondición: El actor ha iniciado una sesión en la aplicación.</i> <i>Postcondición: Administración eficiente de las preguntas de la urna virtual.</i>
<b>CRITERIOS DE ACEPTACIÓN</b> <i>Permite llevar una administración sistemática de las preguntas de la urna virtual, permitiendo obtener la opinión de la comunidad universitaria sobre un determinado tema.</i> <i>La información en la base de datos se maneja de forma persistente y consistente.</i>

### 9.1.10.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Crear Pregunta Urna

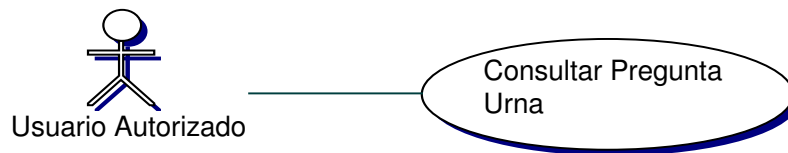


- Iteración Crear Pregunta Urna

<b>IDENTIFICADOR:</b> APURNA1.1		<b>NOMBRE:</b> Crear Pregunta Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Pregunta Urna Actualizar Pregunta Urna		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Pregunta, estado.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear una pregunta de la urna virtual que permita obtener la opinión de los usuarios sobre un determinado tema. De igual manera se establecerá si la pregunta queda activa o inactiva. La pregunta activa quedará publicada para la votación de los usuarios.			
<b>OBSERVACIONES:</b> Solo puede haber una pregunta activa, por tal razón si la nueva pregunta se crea como activa, automáticamente se desactivará la pregunta que se encuentre actualmente publicada. Después de creada la pregunta se debe asignar las opciones de respuesta para que los usuarios puedan elegir una de ellas y efectuar la votación.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor selecciona la opción crear pregunta.	2.	Solicita la descripción de la pregunta y su estado.
3.	El actor suministra los datos solicitados.	5.	Registra la pregunta actualizando la base de datos con la información suministrada.
4.	Da la orden para crear la pregunta (a)	6.	Muestra un mensaje indicando que la creación se realizó con éxito

<b>CAMINOS DE EXCEPCION:</b> <b>(a)</b> El actor iniciador decide cancelar la creación: El sistema cancela el proceso. <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<b>TRIGGERS:</b> Verificar pregunta activa: El sistema verificará si la pregunta creada tiene el estado activo, en ese caso buscará la pregunta que actualmente este activa y la deshabilitará.
Precondición: La información en la base de datos es consistente Postcondición: Creación eficiente de Preguntas
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra crear las preguntas de la urna virtual de forma veraz y oportuna. La información de la base de datos se almacena de forma persistente y consistente.

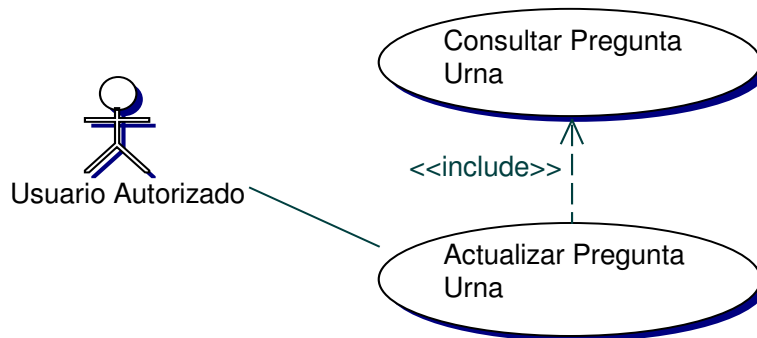
- Diagrama Consultar Preguntas



- Iteración Consultar Pregunta Urna

<b>IDENTIFICADOR:</b> APURNA1.2		<b>NOMBRE:</b> Consultar Pregunta Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Listado de Preguntas	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las preguntas de la Urna Virtual que se hayan creado.			
<b>OBSERVACIONES:</b> El listado mostrado incluirá las preguntas activas e inactivas.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. El actor selecciona la opción consultar preguntas.		2. Despliega un listado de todas las preguntas de la urna virtual especificando su estado.	
<b>CAMINOS DE EXCEPCION:</b> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>			
Precondición: La información de la base de datos es consistente. Postcondición: Información oportuna de las preguntas existentes.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar las preguntas de la urna virtual de forma veraz y oportuna.			

- Diagrama Actualizar Pregunta Urna



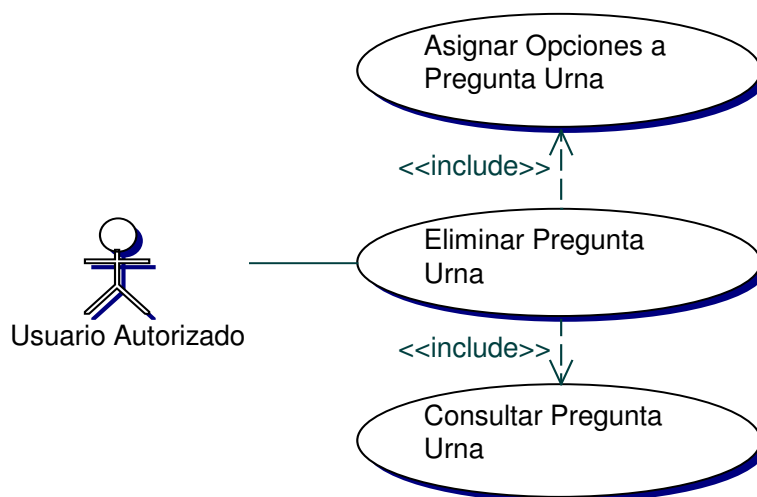
- Iteración Actualizar Pregunta Urna

<b>IDENTIFICADOR:</b> APURNA1.3		<b>NOMBRE:</b> Actualizar Pregunta Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Pregunta Urna		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Pregunta a actualizar, descripción, estado.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de una pregunta como son su descripción y estado. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> La pregunta que se actualice al estado inactivo ya no será visualizada por los usuarios para que efectúen la votación. Si la pregunta se actualiza a estado activa, automáticamente se desactivará la pregunta que se encuentre actualmente publicada.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción actualizar pregunta.		2. Permite seleccionar la pregunta que se desea actualizar.	
3. Selecciona la pregunta deseada.		4. Muestra la descripción y el estado de la pregunta seleccionada y permite modificar sus datos.	
5. Actualiza los datos deseados		7. Se pide confirmación para llevar a cabo el proceso.	
6. Selecciona la opción actualizar pregunta. (a)		9. Actualiza la pregunta modificando la base de datos con la información suministrada.	
8. El actor confirma su decisión. (a)		10. Muestra un mensaje indicando que la pregunta se actualizó con éxito	



<p><b>CAMINOS DE EXCEPCION:</b>  <b>(a)</b> El actor decide cancelar la actualización: El sistema cancela el proceso.  <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul> </p>
<p><b>TRIGGERS:</b>  Verificar pregunta activa: El sistema verificará si la pregunta actualizada paso a estado activo, en ese caso buscará la pregunta que actualmente este activa y la deshabilitará.</p>
<p>Precondición: La información de la base de datos es consistente.  Debe existir al menos una pregunta creada.</p> <p>Postcondición: Actualización eficiente de preguntas.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b>  Se cumple el requerimiento cuando se logra actualizar las preguntas de forma eficiente y oportuna, permitiendo desactivar o activar las preguntas para ser publicadas.</p>

### Diagrama Eliminar Pregunta Urna

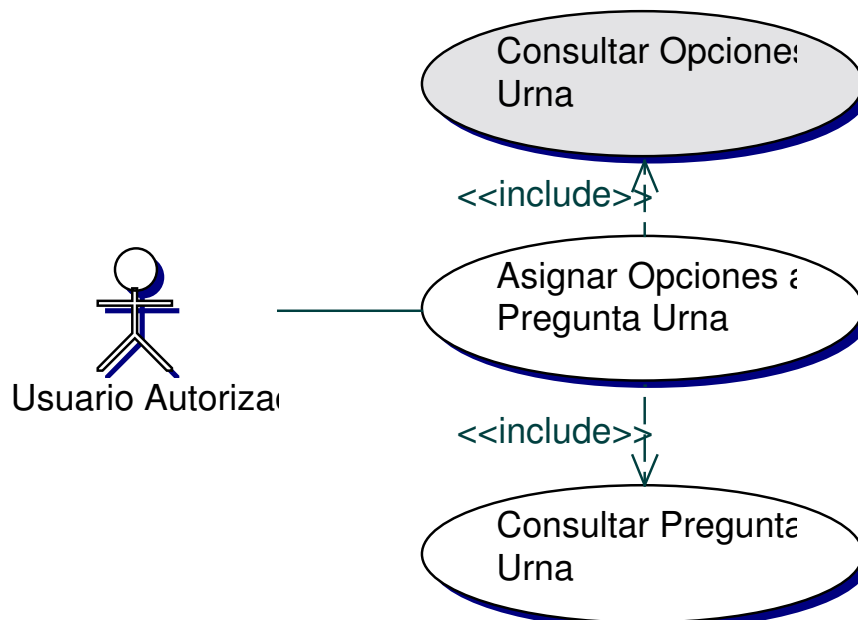


### • Iteración Eliminar Pregunta Urna

<b>IDENTIFICADOR:</b> APURNA1.4		<b>NOMBRE:</b> Eliminar Pregunta Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Pregunta Urna Asignar Opciones a Pregunta Urna		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> pregunta a eliminar		<b>SALIDA:</b> Pregunta eliminada o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar una determinada pregunta de la urna virtual. Se pide confirmación para efectuar el proceso.			

<b>OBSERVACIONES:</b> Al eliminar una pregunta, automáticamente se eliminarán los registros donde se asocia la pregunta y sus respectivas opciones de respuesta.	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción eliminar pregunta.	2. Permite seleccionar la pregunta que se desea eliminar.
3. Selecciona la pregunta deseada.	5. Se pide confirmación para llevar a cabo el proceso.
4. Selecciona la opción eliminar pregunta. (a)	7. Elimina la pregunta modificando la base de datos con la información suministrada.
6. El actor confirma su decisión. (a)	8. Muestra un mensaje indicando que la pregunta se eliminó con éxito
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor decide cancelar la eliminación: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
<b>TRIGGERS:</b>	
Eliminar registro de opciones asociadas a la pregunta: El sistema eliminará el registro de opciones de respuesta que se han asignado a la pregunta que se está eliminando.	
Precondición: La información de la base de datos es consistente. Debe existir al menos una pregunta creada.	
Postcondición: Eliminación eficiente de Preguntas.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra eliminar las preguntas de la urna virtual de forma consistente.	

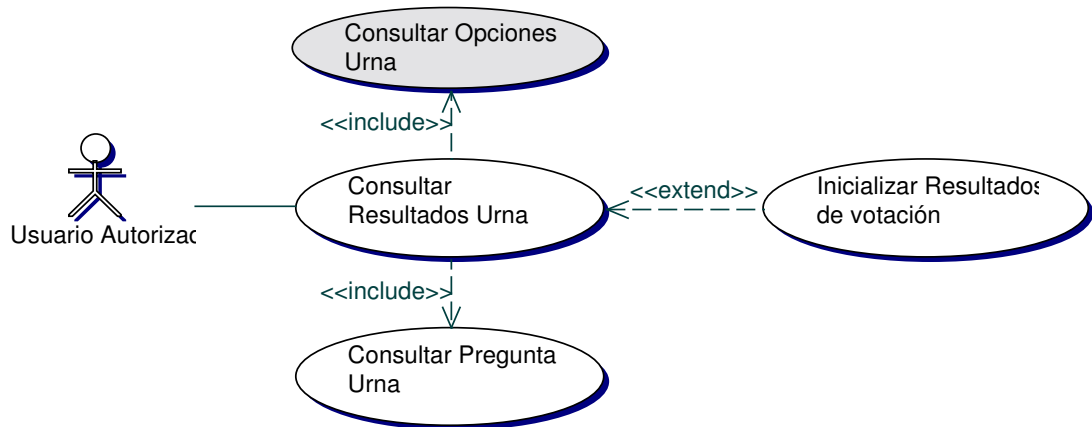
• Diagrama Asignar Opciones a Pregunta Urna



- Iteración Asignar Opciones a Pregunta Urna

<b>IDENTIFICADOR:</b> APURNA1.5		<b>NOMBRE:</b> Asignar Opciones a Pregunta Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Pregunta Urna Consultar Opciones Urna		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> pregunta, opciones a asignar		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo asignar o quitar a una determinada pregunta el conjunto de opciones que conforman las respuestas que pueden elegir los usuarios para la votación.			
<b>OBSERVACIONES:</b> Al asignar una opción a una pregunta se genera el registro que asocia la pregunta a la opción de respuesta e inicializa el total de la votación para esta opción en cero. Al quitar una opción de respuesta a una pregunta se eliminará el registro de asociación y su total de votación.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor selecciona la opción asignar opciones a pregunta urna.	2.	Permite seleccionar la pregunta a la que se le asignaran las opciones.
3.	El actor selecciona la pregunta deseada.	4.	Despliega todas las opciones de preguntas disponibles indicando cuales ya se encuentran asignadas a la pregunta, permitiendo quitar o asignar mas opciones.
5.	Selecciona las opciones que se desea asignar.	7.	Se pide confirmación para llevar a cabo el proceso.
6.	Da la orden para asignar las opciones a la pregunta (a)	9.	Asigna las opciones a la pregunta actualizando la base de datos con la información suministrada.
8.	El actor confirma su decisión. (a)	10.	Muestra un mensaje indicando que la asignación se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la asignación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Asignación eficiente de las opciones a las preguntas de la urna virtual			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra asignar las opciones de respuesta a las preguntas de la urna virtual de forma veraz y oportuna.			
La información de la base de datos se almacena de forma persistente y consistente.			

- Diagrama Consultar Resultados Urna



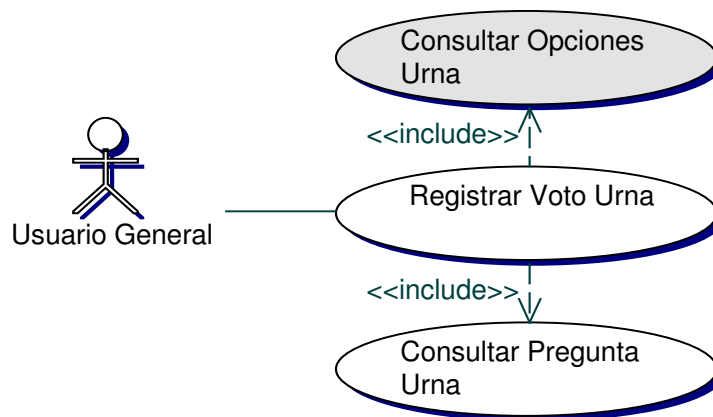
• Iteración Consultar Resultados Urna

<b>IDENTIFICADOR:</b> APURNA1.6		<b>NOMBRE:</b> Consultar Resultados Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Pregunta Urna Consultar Opciones Urna Inicializar Resultados de Votación		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> pregunta a consultar		<b>SALIDA:</b> listado de opciones de la pregunta con su votación	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar el total de votos acumulados para cada opción de una determinada pregunta. Además también permite inicializar el total de votos de las opciones de la pregunta para poder reiniciar las votaciones de los usuarios.			
<b>OBSERVACIONES:</b>			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar resultados urna.	2. Permite seleccionar la pregunta de la que se desea consultar el resultado de la votación	3. Selecciona la pregunta deseada.	4. Despliega el listado de todas las opciones asignadas a la pregunta con el respectivo total de votos para cada una de ellas.
5. El actor selecciona la opción para inicializar los totales de cada opción. (a)	6. Inicializa los totales actualizando la base de datos.		7. Muestra un mensaje indicando que la inicialización se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide no efectuar la inicialización de totales: El proceso de inicialización no se efectúa.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Información oportuna de los resultados obtenidos para las opciones de una pregunta.			

**CRITERIOS DE ACEPTACIÓN**

Se cumple el requerimiento cuando se logra consultar los resultados de la votación por las opciones de una determinada pregunta de la urna virtual de forma veraz y oportuna.  
El resultado de la votación permitirá la toma de decisiones de acuerdo a la opinión de los usuarios sobre un determinado tema.

- Diagrama Registrar Voto Urna



- Iteración Registrar Voto Urna

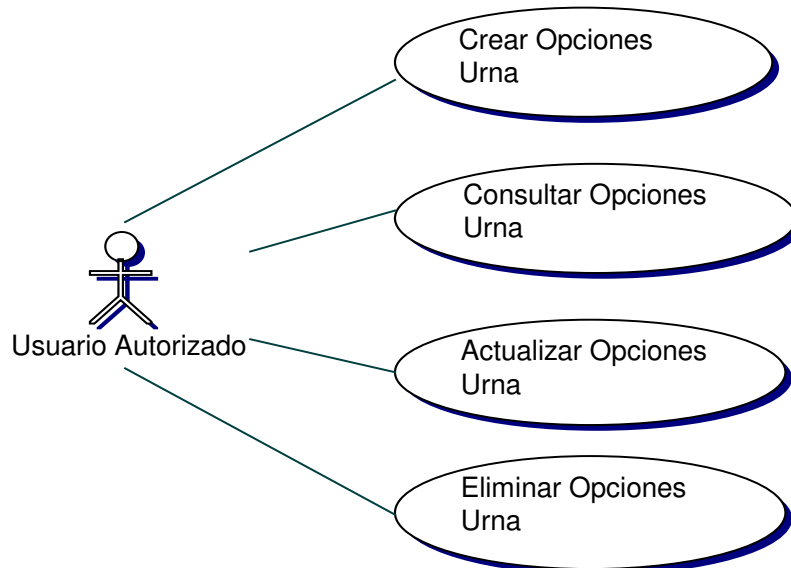
<b>IDENTIFICADOR:</b> APURNA1.7		<b>NOMBRE:</b> Registrar Voto Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Pregunta Urna Consultar Opciones Urna		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario General	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> pregunta, opción seleccionada		<b>SALIDA:</b> voto registrado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar el voto de un usuario por la opción de una determinada pregunta.			
<b>OBSERVACIONES:</b> Cualquier usuario involucrado en el sistema podrá acceder a la votación de la urna virtual.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. EL actor selecciona la opción de la pregunta por la que desea votar.		3. Registra el voto actualizando la base de datos con la información suministrada.	
2. Selecciona la opción para registrar el voto. (a)		4. Muestra un mensaje indicando que la votación se realizó con éxito	

<b>CAMINOS DE EXCEPCION:</b> (a) El actor decide no efectuar la votación: El proceso no se efectúa. • El sistema no responde. El proceso no se efectúa.
Precondición: La información de la base de datos es consistente. Postcondición: Registro de la votación del usuario por una opción de la pregunta de la urna virtual.
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra efectuar la votación de los usuarios de forma persistente. La votación permite obtener la opinión de los usuarios sobre un determinado tema de interés.

## 9.1.11 Administración de Opciones de Urna Virtual

### 9.1.11.1 Primera Iteración - Fachada

- Diagrama

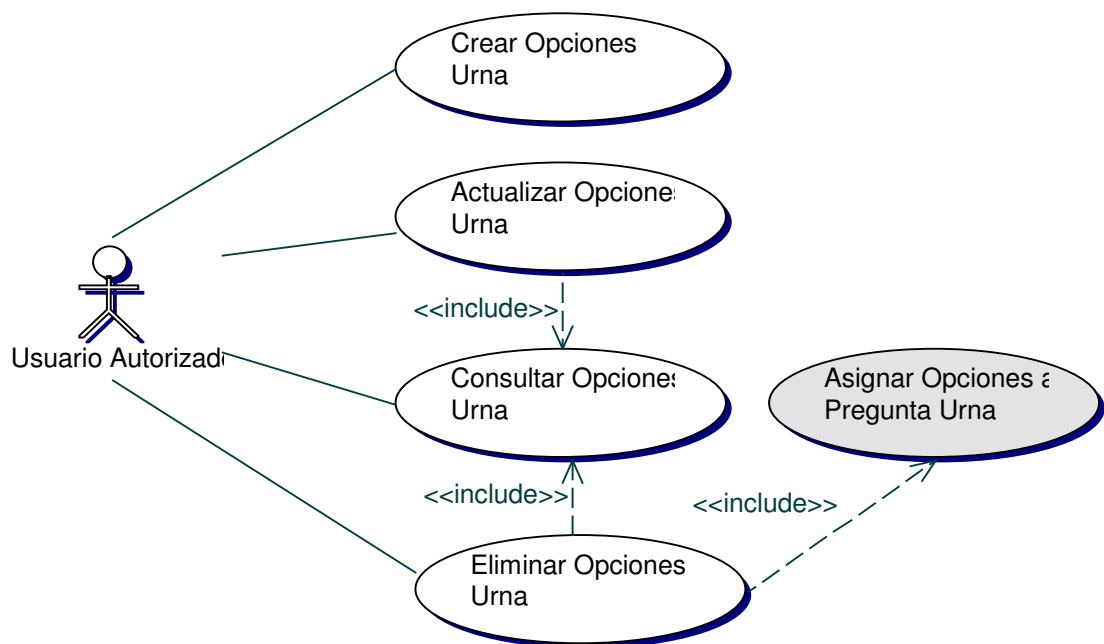


- Iteración

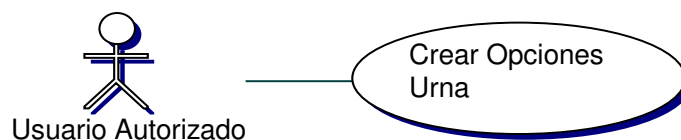
<b>IDENTIFICADOR:</b> AOURNA1		<b>NOMBRE:</b> Administración de Opciones de Urna Virtual
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Opciones Urna Consultar Opciones Urna Actualizar Opciones Urna Eliminar Opciones Urna	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.

<b>ENTRADA:</b> Datos para efectuar el proceso.	<b>SALIDA:</b> Proceso efectuado o cancelado
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las opciones de la urna virtual que servirán de respuestas para las preguntas que se hayan establecido.	
<b>OBSERVACIONES:</b> El usuario autorizado tendrá acceso a la administración de las opciones sin ninguna restricción.	
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.	
Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de las opciones de la urna virtual.	
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de las opciones de la urna virtual. La información en la base de datos se maneja de forma persistente y consistente.	

### 9.1.11.2 Segunda Iteración – Llenado (Diagrama General)



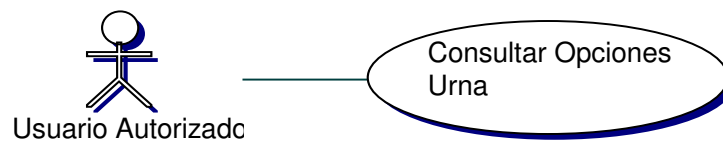
- Diagrama Crear Opciones Urna



- Iteración Crear Opciones Urna

<b>IDENTIFICADOR:</b> AOURNA1.1		<b>NOMBRE:</b> Crear Opciones Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Descripción de la opción.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear una opción de la urna virtual para que pueda ser utilizada como respuesta de una pregunta de la urna virtual.			
<b>OBSERVACIONES:</b>			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1.	EL actor selecciona la opción crear opción.	2.	Solicita la descripción de la opción.
3.	El actor suministra los datos solicitados.	5.	Registra la opción actualizando la base de datos con la información suministrada.
4.	Da la orden para crear la opción (a)	6.	Muestra un mensaje indicando que la creación se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la creación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información en la base de datos es consistente			
Postcondición: Creación eficiente de Opciones			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra crear las opciones de la urna virtual de forma veraz y oportuna para establecer las respuestas necesarias para una pregunta de la urna virtual.			
La información de la base de datos se almacena de forma persistente y consistente.			

- Diagrama Consultar Opciones Urna



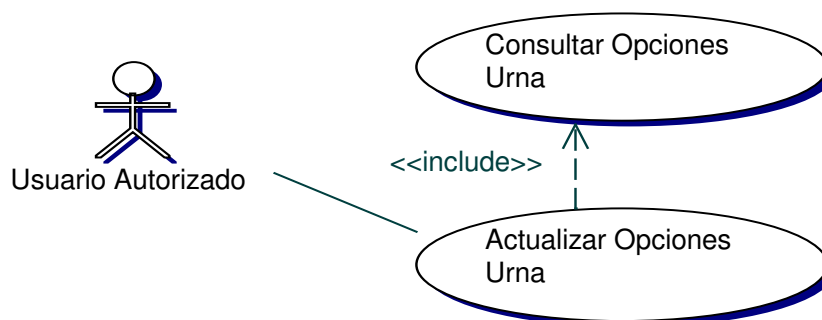
- Iteración Consultar Opciones Urna

<b>IDENTIFICADOR:</b> AOURNA1.2	<b>NOMBRE:</b> Consultar Opciones Urna
------------------------------------	---



<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b>	<b>SALIDA:</b> Listado de Opciones de la Urna Virtual	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las opciones de la Urna Virtual que se hayan creado.		
<b>OBSERVACIONES:</b>		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. El actor selecciona la opción consultar opciones.		2. Despliega un listado de todas las opciones de la urna virtual.
<b>CAMINOS DE EXCEPCION:</b>		
• El sistema no responde. El proceso no se efectúa.		
Precondición: La información de la base de datos es consistente.		
Postcondición: Información oportuna de las opciones existentes.		
<b>CRITERIOS DE ACEPTACIÓN</b>		
Se cumple el requerimiento cuando se logra consultar las opciones de la urna virtual de forma veraz y oportuna.		

- Diagrama Actualizar Opciones Urna

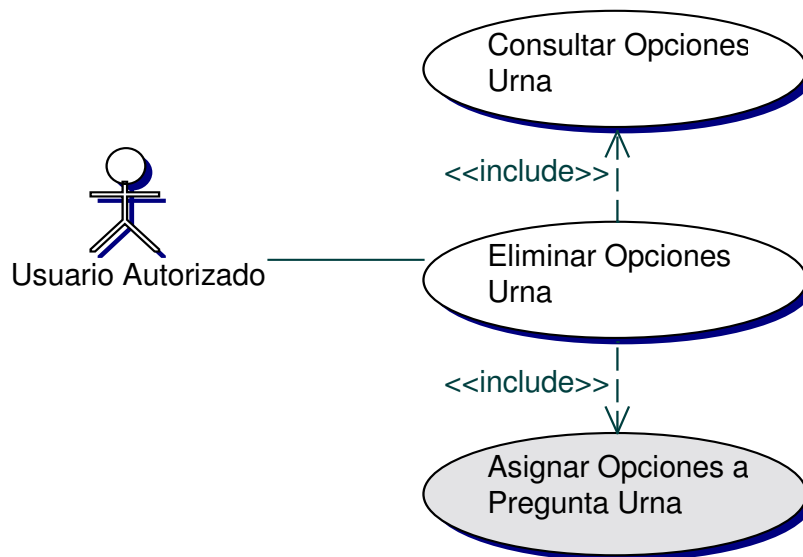


- Iteración Actualizar Opciones Urna

<b>IDENTIFICADOR:</b> AOURNA1.3		<b>NOMBRE:</b> Actualizar Opciones Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Opciones Urna	<b>CRITICO?</b> Si	
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Opción a actualizar, nueva descripción de la opción.	<b>SALIDA:</b> Actualización efectuada o cancelada		

<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar la descripción de una opción de la urna virtual. Se pide confirmación para efectuar el proceso.	
<b>OBSERVACIONES:</b>	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción actualizar opción.	2. Muestra un listado de todas las opciones permitiendo seleccionar la opción a actualizar.
3. Selecciona la opción deseada.	4. Muestra la descripción de la opción permitiendo modificar sus datos.
5. Actualiza los descripción de la opción.	7. Se pide confirmación para llevar a cabo el proceso.
6. Selecciona la opción actualizar opción. (a)	9. Actualiza la opción modificando la base de datos con la información suministrada.
8. El actor confirma su decisión. (a)	10. Muestra un mensaje indicando que la opción se actualizó con éxito
<b>CAMINOS DE EXCEPCION:</b> (a) El actor decide cancelar la actualización: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente. Debe existir al menos una opción creada.	
Postcondición: Actualización eficiente de opciones.	
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra actualizar las opciones de la urna virtual de forma eficiente y oportuna.	

• **Diagrama Eliminar Opciones Urna**



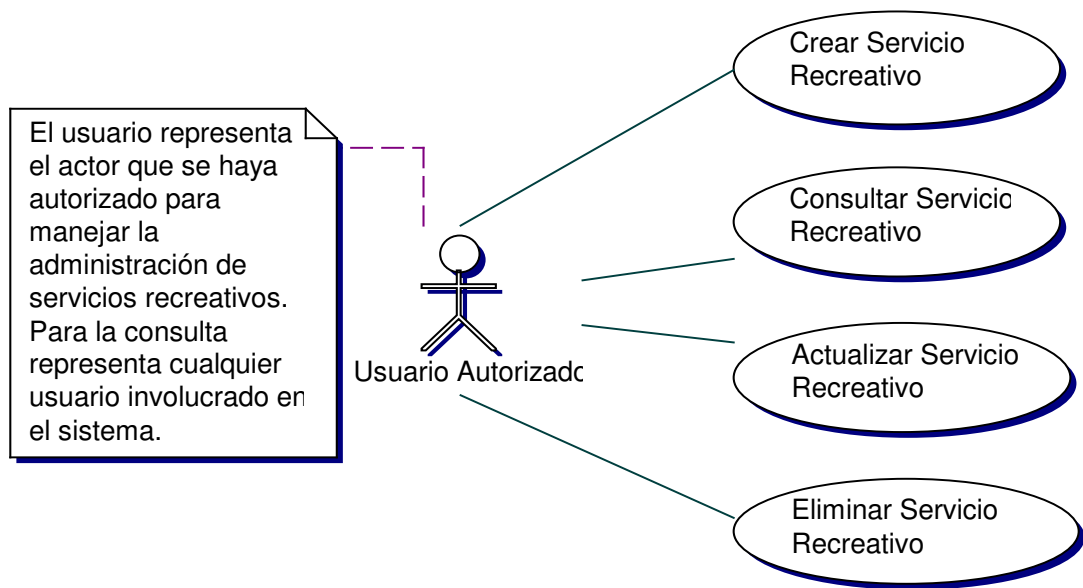
- Iteración Eliminar Opciones Urna

<b>IDENTIFICADOR:</b> ACURNA1.4		<b>NOMBRE:</b> Eliminar Opciones Urna	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Opciones Urna Asignar Opciones a Pregunta Urna		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> opción a eliminar		<b>SALIDA:</b> opción eliminada o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar una determinada opción de la urna virtual. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b> Al eliminar una opción, automáticamente se eliminarán los registros donde se asocia la opción con las preguntas de la urna virtual			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor selecciona la opción eliminar opción.	2.	Permite seleccionar la opción que se desea eliminar.
3.	Selecciona la opción deseada.	5.	Se pide confirmación para llevar a cabo el proceso.
4.	Selecciona la opción eliminar opción. (a)	7.	Elimina la opción modificando la base de datos con la información suministrada.
6.	El actor confirma su decisión. (a)	8.	Muestra un mensaje indicando que la opción se eliminó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la eliminación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>TRIGGERS:</b>			
Eliminar registro de preguntas asociadas a la opción: El sistema eliminará el registro de preguntas que tengan asociada como respuesta la opción a eliminar.			
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos una opción creada.			
<b>Postcondición:</b> Eliminación eficiente de opciones.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar las opciones de la urna virtual de forma consistente.			

## 9.1.12 Administración de Servicios Recreativos

### 9.1.12.1 Primera Iteración - Fachada

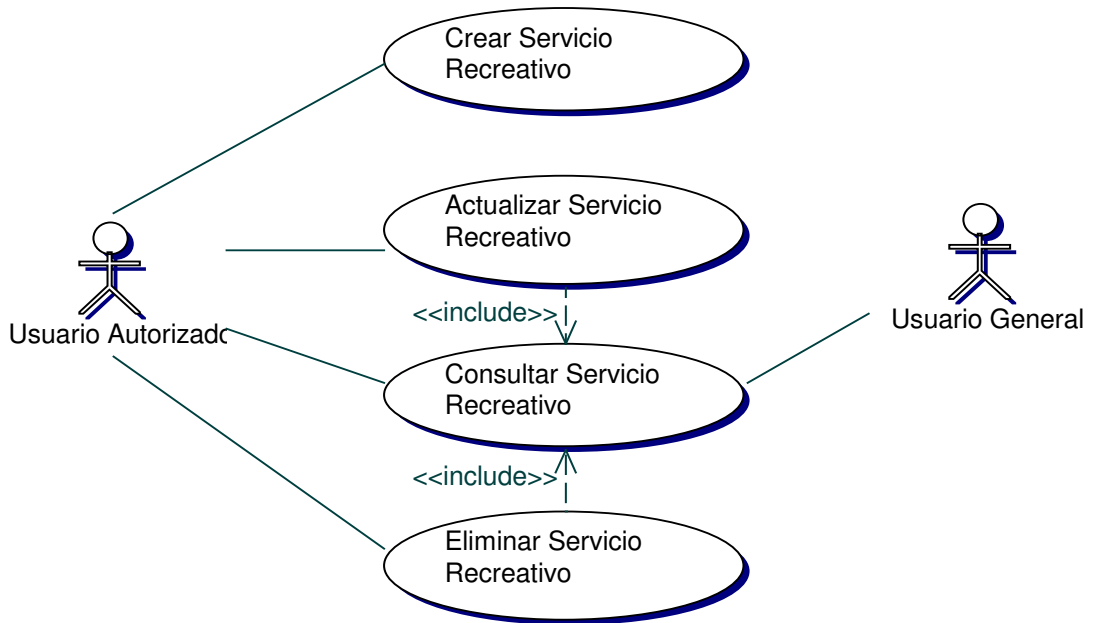
- Diagrama



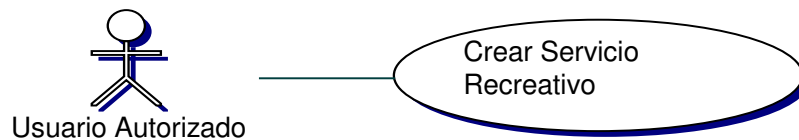
## Iteración

<b>IDENTIFICADOR:</b> ASRECI		<b>NOMBRE:</b> Administración de Servicios Recreativos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Servicios Recreativos Consultar Servicios Recreativos Actualizar Servicios Recreativos Eliminar Servicios Recreativos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los servicios recreativos con los que cuenta la universidad permitiendo mantener informados a los usuarios sobre estos servicios y les puedan dar un buen uso. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y eliminación.			
<b>OBSERVACIONES:</b> El usuario autorizado tendrá acceso a la administración de los servicios recreativos sin ninguna restricción. Los usuarios en general podrán acceder solo a la consulta.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema. Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de los servicios recreativos con los que cuenta la universidad.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de los servicios recreativos. Permite dar a conocer a los usuarios los servicios recreativos disponibles para su utilización. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.12.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Crear Servicio Recreativo

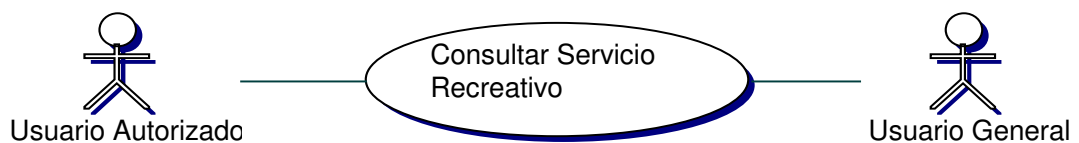


- Iteración Crear Servicio Recreativo

<b>IDENTIFICADOR:</b> ASRECI.1		<b>NOMBRE:</b> Crear Servicio Recreativo	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Titulo, descripción, ruta imagen, nombre imagen.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear un servicio recreativo estableciendo su titulo, una descripción del servicio, seleccionando una imagen que represente el servicio y dándole un nombre a dicha imagen.			
<b>OBSERVACIONES:</b> Todos los servicios recreativos creados podrán ser consultados por los usuarios en general.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción crear servicio recreativo.	2. Solicita el título, la descripción, la ruta de imagen y el nombre de la imagen del servicio recreativo.
3. El actor suministra los datos solicitados.	5. Registra el servicio recreativo actualizando la base de datos con la información suministrada.
4. Da la orden para crear el servicio recreativo (a)	6. Muestra un mensaje indicando que la creación se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor iniciador decide cancelar la creación: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: La información en la base de datos es consistente	
Postcondición: Creación eficiente de Servicios Recreativos.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra crear los servicios recreativos de forma veraz y oportuna.	
La información de la base de datos se almacena de forma persistente y consistente.	

- **Diagrama Consultar Servicio Recreativo**

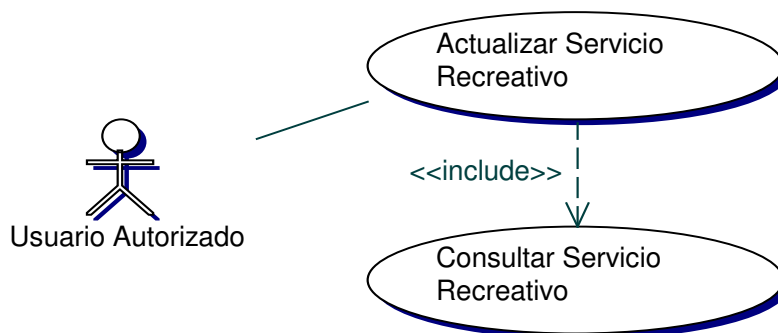


- **Iteración Consultar Servicio Recreativo**

<b>IDENTIFICADOR:</b> ASREC1.2		<b>NOMBRE:</b> Consultar Servicio Recreativo	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado, Usuario General.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Titulo Servicio		<b>SALIDA:</b> Datos Servicios Recreativos	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los servicios recreativos que se hayan creado.			
<b>OBSERVACIONES:</b>			

<b>CURSO BÁSICO DE EVENTOS</b>	
<i>Consulta efectuada por el usuario autorizado</i>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL usuario autorizado selecciona la opción consultar servicios recreativos.	2. Le permite seleccionar el título del servicio recreativo a consultar.
3. El usuario autorizado selecciona el título para su consulta. (a)	4. Despliega el nombre de la imagen y la descripción del servicio.
<i>Consulta efectuada por el usuario general</i>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL usuario general selecciona la opción pública para consultar los servicios recreativos.	2. Despliega todos los servicios recreativos existentes mostrando el título, la imagen y la descripción del servicio recreativo.
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
<i>Precondición: La información de la base de datos es consistente. El usuario ha desplegado la página principal o ha iniciado una sesión en el sistema.</i>	
<i>Postcondición: Información oportuna de los servicios recreativos existentes.</i>	
<b>CRITERIOS DE ACEPTACIÓN</b>	
<i>Se cumple el requerimiento cuando se logra consultar los servicios recreativos de forma veraz y oportuna, permitiendo informar a los usuarios sobre los servicios de los que pueden hacer uso.</i>	

- **Diagrama Actualizar Servicio Recreativo**

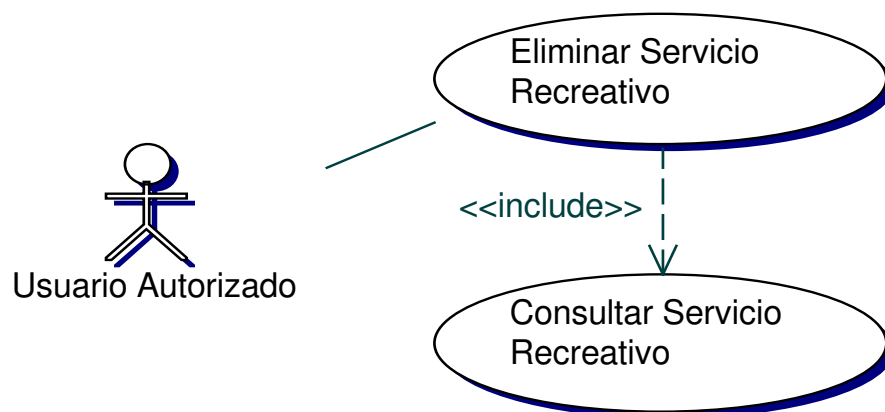


- **Iteración Actualizar Servicio Recreativo**

<b>IDENTIFICADOR:</b> ASREC1.3		<b>NOMBRE:</b> Actualizar Servicio Recreativo	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicio Recreativo		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Servicio Recreativo a actualizar, título, imagen, nombre imagen, descripción.		<b>SALIDA:</b> Actualización efectuada o cancelada	

<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de un servicio recreativo como son su título, imagen, nombre de la imagen y descripción. Se pide confirmación para efectuar el proceso.	
<b>OBSERVACIONES:</b>	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción actualizar servicio recreativo.	2. Permite seleccionar el servicio recreativo que se desea actualizar.
3. Selecciona el servicio recreativo deseado.	4. Muestra el título, ruta de la imagen, nombre de la imagen y la descripción del servicio recreativo seleccionado y permite modificar sus datos.
5. Actualiza los datos deseados	
6. Selecciona la opción actualizar servicio. (a)	7. Se pide confirmación para llevar a cabo el proceso.
8. El actor confirma su decisión. (a)	9. Actualiza el servicio recreativo modificando la base de datos con la información suministrada.
	10. Muestra un mensaje indicando que el servicio se actualizó con éxito
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor decide cancelar la actualización: El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente. Debe existir al menos un servicio recreativo creado.	
Postcondición: Actualización eficiente de servicios recreativos.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra actualizar los servicios recreativos de forma eficiente y oportuna.	

• **Diagrama Eliminar Servicio Recreativo**





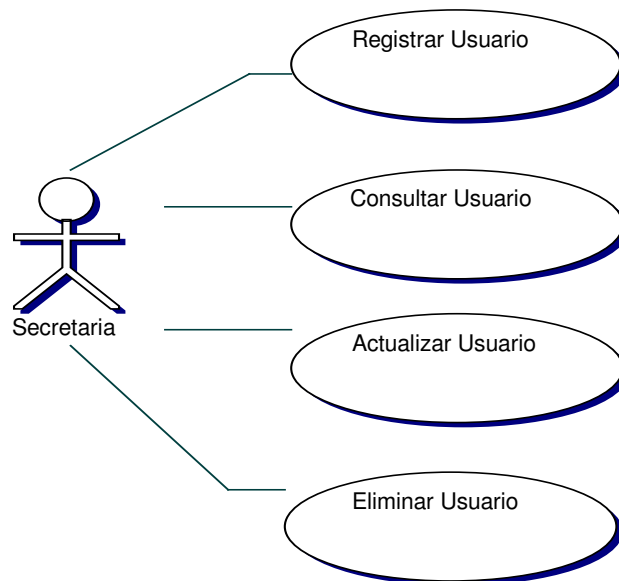
- Iteración Eliminar Servicio Recreativo

<b>IDENTIFICADOR:</b> ASRECI.4		<b>NOMBRE:</b> Eliminar Servicio Recreativo	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicio Recreativo		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Servicio Recreativo a eliminar.		<b>SALIDA:</b> Servicio Recreativo eliminado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un determinado Servicio Recreativo. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b>			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor selecciona la opción eliminar servicio recreativo.	2.	Permite seleccionar el servicio recreativo que se desea eliminar.
3.	Selecciona el servicio recreativo deseado.	4.	Muestra los datos del servicio recreativo seleccionado.
5.	Selecciona la opción eliminar servicio. (a)	6.	Se pide confirmación para llevar a cabo el proceso.
7.	El actor confirma su decisión. (a)	8.	Elimina el servicio recreativo modificando la base de datos con la información suministrada.
		9.	Muestra un mensaje indicando que el servicio se eliminó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la eliminación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente. Debe existir al menos un servicio recreativo creado.			
<b>Postcondición:</b> Eliminación eficiente de Servicios Recreativos.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar los servicios recreativos de forma consistente.			

### 9.1.13 Administración de Datos Personales de Usuarios

#### 9.1.13.1 Primera Iteración - Fachada

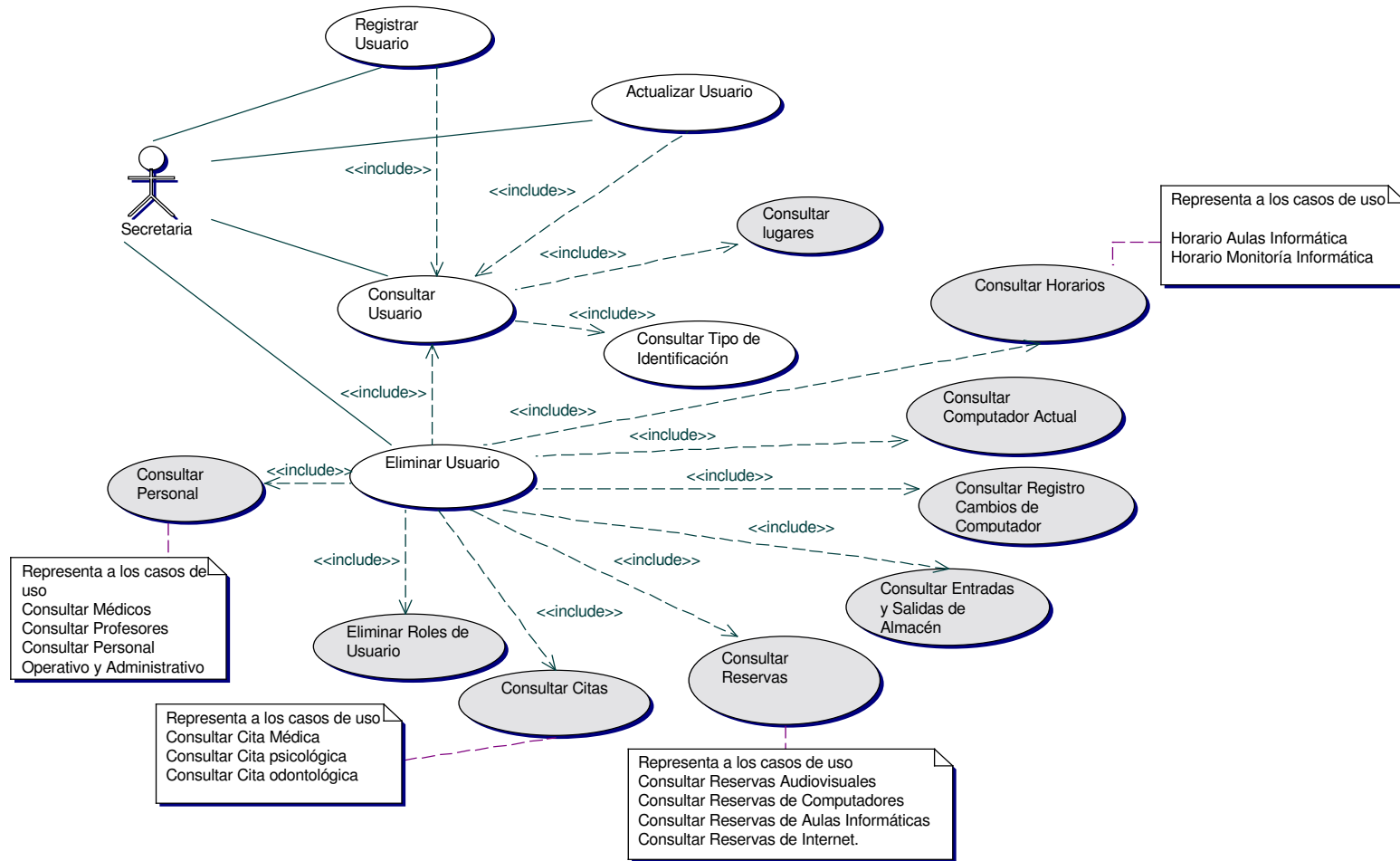
- Diagrama



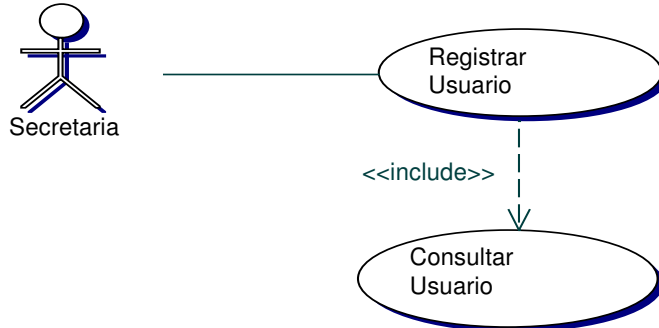
- Iteración

<b>IDENTIFICADOR:</b> ADPU 1		<b>NOMBRE:</b> Administración de Datos de Usuario.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Adicionar Usuario. Consultar Usuario. Actualizar Usuario. Eliminar usuario.		<b>CRITICO?</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los datos de los usuarios de la universidad. Este proceso involucrará las operaciones de creación, consulta, actualización y eliminación.			
<b>OBSERVACIONES:</b> Los usuarios de la universidad corresponden a estudiantes, profesores, personal administrativo y personal operativo.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema. Precondición: El usuario se encuentra correctamente registrado en la base de datos. Postcondición: Administración eficiente de los datos personales del usuario.			
<b>CRITERIOS DE ACEPTACION</b> Permite llevar una administración sistemática de los datos del usuario. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.13.2 Segunda Iteración – Llenado (Diagrama General)



- **Diagrama Registrar Usuarios**

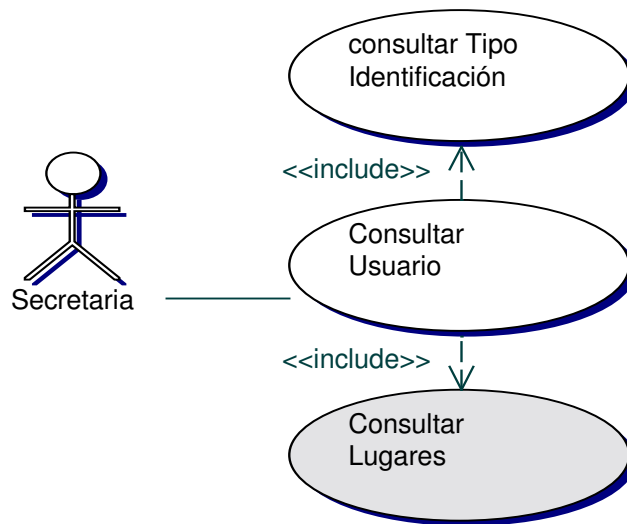


- **Iteración Registrar Usuarios**

<b>IDENTIFICADOR:</b> ADPU 1.1		<b>NOMBRE:</b> Adición Usuario.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Usuario.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, nombres, apellidos, dirección, teléfono, lugar de nacimiento, lugar de residencia, tipo de identificación, lugar de identificación, sexo y estado civil.		<b>SALIDA:</b> Usuario registrado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los usuarios			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Seleccionar la opción adición usuario.		2. Solicita los datos: Identificación, primer nombre, segundo nombre, primer apellido, segundo apellido, dirección, teléfono, lugar de nacimiento, lugar de residencia, tipo de identificación, lugar de identificación, sexo y estado civil.	
3. Ingresar los datos solicitados. <b>(a)</b>			
4. Da la orden para registrar el usuario. <b>(a)</b>		5. Verifica que el usuario no se encuentre registrado <b>(b)</b>	
		6. Muestra un mensaje indicando que el usuario se registró con éxito.	

<b>CAMINOS DE EXCEPCION:</b> (a) El actor decide no registrar el usuario. El sistema cancela el proceso. (b) El usuario ya se encuentra registrado. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.
Precondición: Consistencia en la información de la base de datos. Postcondición: Registro eficiente de los datos del usuario.
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra registrar los datos personales del usuario de forma consistente y persistente.

- **Diagrama Consultar Usuarios**

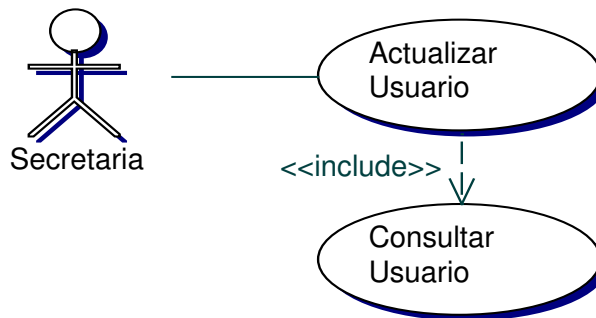


- **Iteración Consultar Usuarios**

<b>IDENTIFICADOR:</b> ADPU 1.2		<b>NOMBRE:</b> Consultar Usuario.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar TipoIdentificación Consultar Lugares.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación.		<b>SALIDA:</b> Datos consultados o proceso cancelado.	
<b>Observación:</b> El requerimiento o método Consultar TipoIdentificación hace referencia al tipo de documento de identificación que tiene el usuario (Cédula, Tarjeta de Identidad etc.)			

<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo consultar los datos personales de los usuarios.</i>	
<b>CURSO BÁSICO DE EVENTOS</b>	
<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>
<i>Consulta de disponibilidad realizada por el Estudiante.</i>	
<ol style="list-style-type: none"> <li>1. <i>Selecciona la opción consultar usuario.</i></li> <li>3. <i>Digita la identificación. (a)</i></li> <li>4. <i>Da la orden para que se muestre los datos del usuario. (a)</i></li> </ol>	<ol style="list-style-type: none"> <li>2. <i>Solicita la identificación.</i></li> <li>5. <i>Se despliega los datos personales del usuario.</i></li> </ol>
<b>CAMINOS DE EXCEPCION:</b>	
<p>(a) <i>El actor decide no consultar los datos. El sistema cancela el proceso.</i></p> <ul style="list-style-type: none"> <li>• <i>El sistema no responde. El proceso no se efectúa</i></li> </ul>	
<i>Precondición: Consistencia en la información de la base de datos.</i>	
<i>Postcondición: Acceso rápido a los datos personales de los usuarios.</i>	
<b>CRITERIOS DE ACEPTACIÓN</b>	
<i>Se cumple el requerimiento cuando se logra consultar de forma oportuna y veraz los datos de los usuarios.</i>	

- **Diagrama Actualizar Usuario**



- **Iteración Actualizar Usuario**

<b>IDENTIFICADOR:</b> ADPU 1.3		<b>NOMBRE:</b> Actualizar Usuario	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación y los datos personales.		<b>SALIDA:</b> Datos de usuario actualizados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo actualizar los datos personales de los usuarios.</i>			

**CURSO BÁSICO DE EVENTOS**

**Acción de los actores**

1. *Selecciona la opción actualizar usuario.*
3. *Digita la identificación. (a)*
4. *Da la orden para que se muestre los datos del usuario. (a)*
  
6. *Realiza los cambios necesarios.*
7. *Da la orden de registrar los cambios realizados.*

**Respuesta del Sistema**

2. *Solicita la identificación.*
  
5. *Se despliega los datos personales del usuario permitiendo realizar los respectivos cambios (b)*
  
8. *Muestra un mensaje indicando que los datos se actualizaron con éxito.*

**CAMINOS DE EXCEPCION:**

- (a) *El actor decide no actualizar los datos. El sistema cancela el proceso.*
- (b) *Identificador no valido. El sistema informa al usuario que el identificador ingresado es erróneo.*
- *El sistema no responde. El proceso no se efectúa*

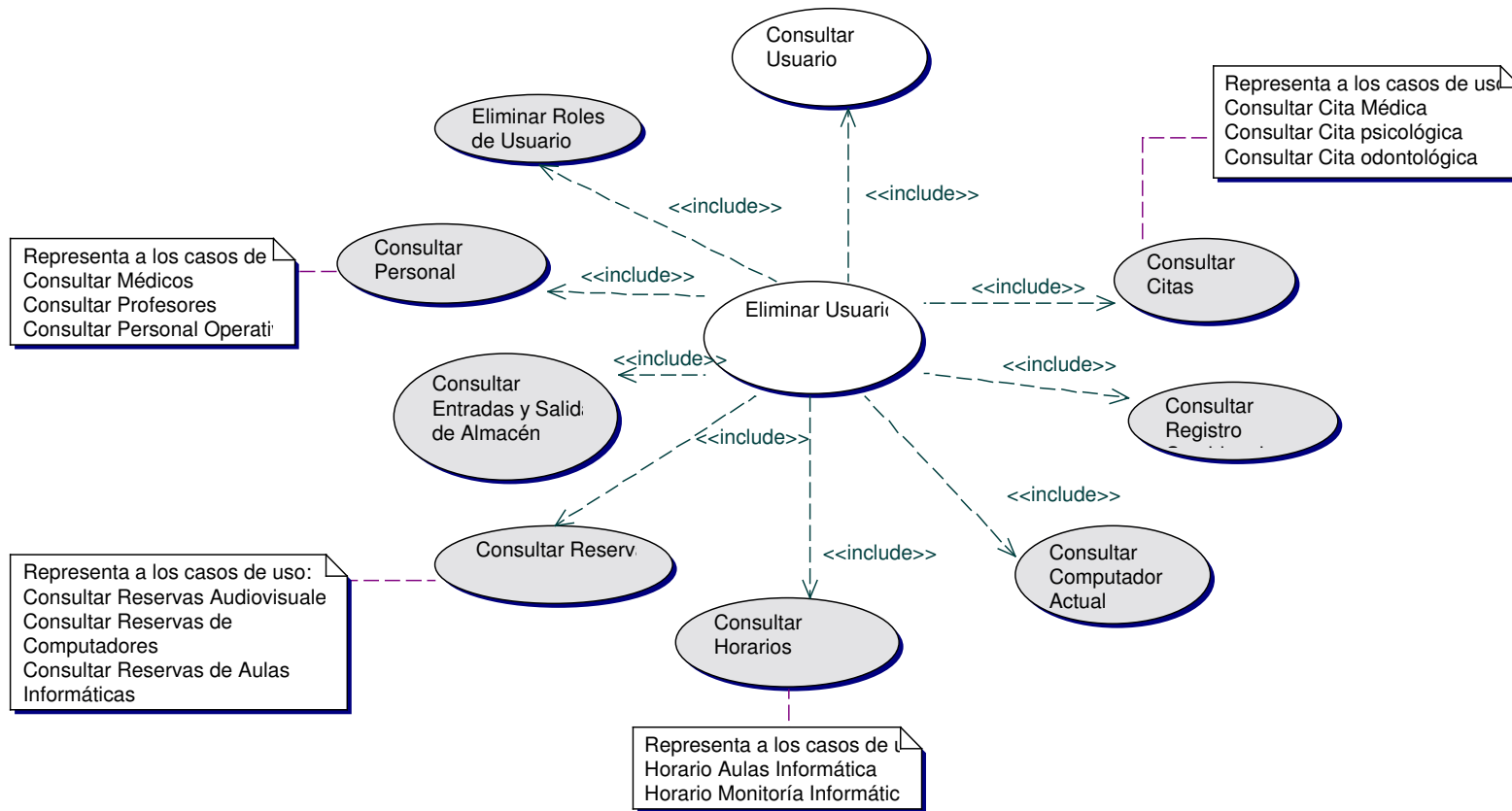
*Precondición: Consistencia en la información de la base de datos.*

*Postcondición: Actualización eficiente de los datos personales de los usuarios.*

**CRITERIOS DE ACEPTACIÓN**

*Se cumple el requerimiento cuando se logra actualizar de forma consistente y persistente los datos personales de los usuarios.*

- Diagrama Eliminar Usuario.





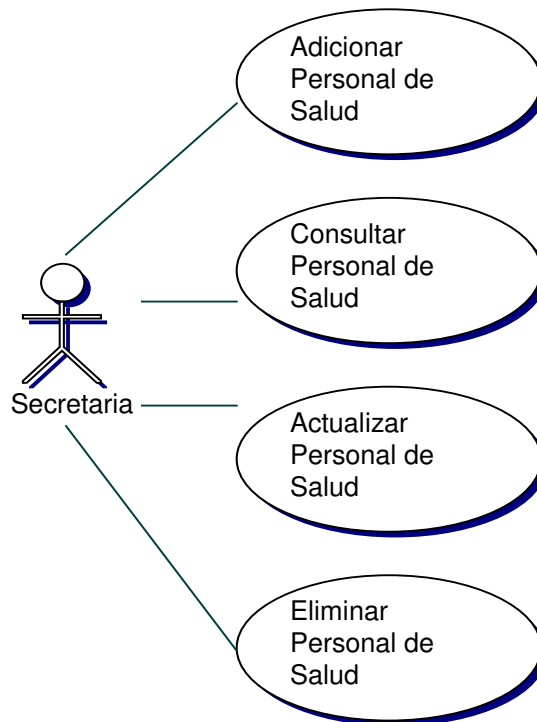
- Iteración Eliminar Usuario

<b>IDENTIFICADOR:</b> ADPU 1.4		<b>NOMBRE:</b> Eliminar Usuario.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Usuario. Consultar Personal. Consultar Entradas y Salidas de Almacén. Consultar Reservas. Consultar Horarios. Consultar Computador Actual. Consultar Registro Cambios de Computador. Consultar Citas. Eliminar Roles de Usuario.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación		<b>SALIDA:</b> Usuario eliminado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un usuario de la Base de Datos.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1.	Selecciona la opción eliminar usuario.	2.	Solicita la identificación.
3.	Digita la identificación. <b>(a)</b>	5.	Despliega los datos personales del usuario.
4.	Da la orden para que se muestre los datos del usuario. <b>(a)</b>	7.	Solicita confirmación para efectuar el proceso.
6.	Da la orden para eliminar el usuario	9.	Verifica que el usuario no esté involucrado en otro proceso. <b>(b)</b>
8.	Confirma la eliminación. <b>(a)</b>	11.	Muestra un mensaje indicando el registro ha sido eliminado con éxito.
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El usuario decide no eliminar el usuario. El sistema cancela el proceso.			
<b>(b)</b> El usuario está involucrado en otro proceso. El sistema cancela el proceso			
• El sistema no responde. El proceso no se efectúa.			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Eliminación eficiente del usuario.			
<b>TRIGGERS:</b> Eliminación de Registro de roles asociados al usuario: Se eliminará automáticamente todos los registros de los roles asociados al usuario.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra eliminar de forma consistente y veraz un usuario.			

### 9.1.14 Administración de Personal de Salud

### 9.1.14.1 Primera Iteración - Fachada

- Diagrama

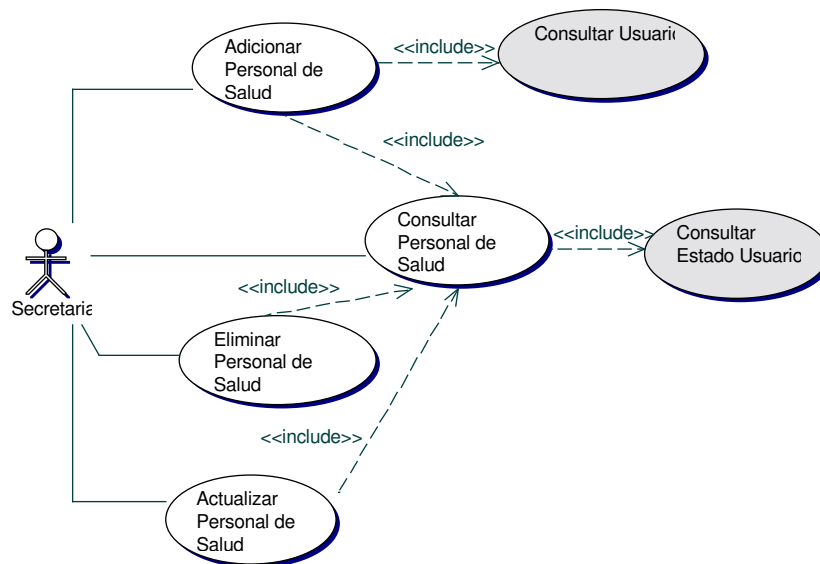


- Iteración

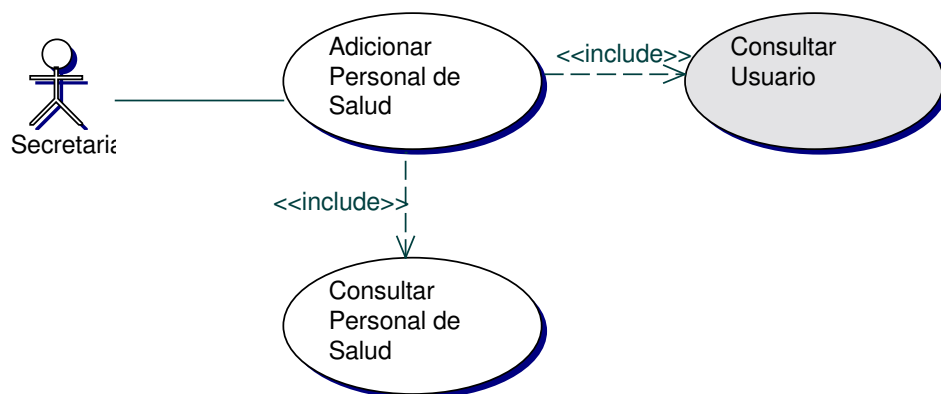
<b>IDENTIFICADOR:</b> APS 1		<b>NOMBRE:</b> Administración de Personal de Salud	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Adicionar Personal de Salud. Consultar Personal de Salud. Actualizar Personal de Salud. Eliminar Personal de Salud.		<b>CRITICO?</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los datos del personal de salud de la universidad. Este proceso involucrará las operaciones de adición, consulta, actualización y eliminación.			

<p><b>OBSERVACIONES:</b> El personal de salud corresponde a médicos, odontólogos y psicólogos.  Los datos que se van a administrar son: Dirección del Consultorio, teléfono del consultorio, Correo Electrónico, Servicio de presta (odontológico odontológico, médico o psicológico) y el estado (activo, inactivo).</p>
<p><b>TRIGGERS:</b>  Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</p>
<p>Precondición: El usuario se encuentra correctamente registrado en la base de datos.  Postcondición: Administración eficiente de los datos profesionales del personal de salud.</p>
<p><b>CRITERIOS DE ACEPTACION</b>  Permite llevar una administración sistemática del personal de salud.  La información en la base de datos se maneja de forma persistente y consistente.</p>

### 9.1.14.2 Segunda Iteración – Llenado (Diagrama General)



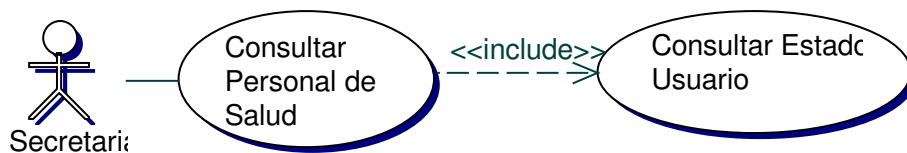
- **Diagrama Adicionar Personal de Salud**



- Iteración Adicionar Personal de Salud

<b>IDENTIFICADOR:</b> APS 1.1		<b>NOMBRE:</b> Adicionar Personal Salud.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Usuario. Consultar Personal de Salud.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Dirección del Consultorio, teléfono del consultorio, Correo Electrónico, Servicio que presta y el estado.		<b>SALIDA:</b> Personal de salud registrado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar el personal de Salud.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. Seleccionar la opción adición personal de salud.		2. Solicita la identificación.	
3. Digita la identificación		4. Verifica que el usuario no se encuentre registrado <b>(b)</b> .	
6. Ingresar los datos solicitados. <b>(a)</b>		5. Solicita los datos del personal de salud.	
7. Da la orden para registrar el usuario. <b>(a)</b>		8. Muestra un mensaje indicando que el usuario se registró con éxito.	
<b>CAMINOS DE EXCEPCION:</b> <b>(a)</b> El actor decide no registrar el personal de salud. El sistema cancela el proceso. <b>(b)</b> El usuario a registrar ya forma parte del personal de salud. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.			
Precondición: Consistencia en la información de la base de datos. Postcondición: Registro eficiente de los datos del personal de salud.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra registrar el personal de salud de forma consistente y persistente.			

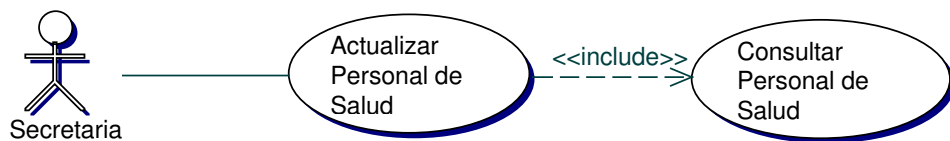
- Diagrama Consultar Personal de Salud



- Iteración Consultar Personal de Salud.

<b>IDENTIFICADOR:</b> APS 1.2		<b>NOMBRE:</b> Consultar Personal de Salud.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Personal de Salud.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación.		<b>SALIDA:</b> Datos consultados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar el personal de salud.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción consultar personal de salud.	2. Permite seleccionar el servicio de salud que ofrece el usuario a consultar.	3. Selecciona el servicio que presta el personal de salud.(a)	4. Se despliega un listado con los datos de todo el personal de salud que presta el servicio seleccionada.
<b>CAMINOS DE EXCEPCION:</b> (a) El actor decide no consultar los datos. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa			
Precondición: Consistencia en la información de la base de datos. Postcondición: Acceso rápido a los datos del personal de salud.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar los datos del personal de salud de forma oportuna y veraz.			

- Diagrama Actualizar Personal de Salud

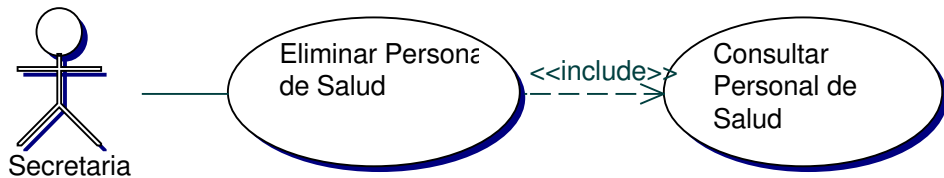


- Iteración Actualizar Personal de Salud

<b>IDENTIFICADOR:</b> APS 1.3		<b>NOMBRE:</b> Actualizar Personal de Salud	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Usuario.		<b>CRITICO?</b> Si

<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Identificación y los datos personales.	<b>SALIDA:</b> Datos de usuario actualizados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos del personal de salud.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. Selecciona la opción actualizar personal de salud.	2. Solicita la identificación.	
3. Digita la identificación. (a)		
4. Da la orden para que se muestre los datos del usuario. (a)	5. Se despliega los datos personales del usuario permitiendo realizar los respectivos cambios (b)	
6. Realiza las modificaciones necesarias.	8. Muestra un mensaje indicando que los datos se actualizaron con éxito.	
7. Da la orden de registrar los cambios realizados.		
<b>CAMINOS DE EXCEPCION:</b>		
(a) El actor decide no actualizar los datos. El sistema cancela el proceso.		
(b) Identificador no valido. El sistema informa al usuario que el identificador ingresado es erróneo. El sistema no responde. El proceso no se efectúa		
Precondición: Consistencia en la información de la base de datos.		
Postcondición: Actualización eficiente de los datos del personal de salud.		
<b>CRITERIOS DE ACEPTACIÓN</b>		
Se cumple el requerimiento cuando se logra actualizar de forma consistente y persistente el personal de salud.		

- **Diagrama Eliminar Personal de Salud**



- **Iteración Eliminar Personal de Salud**

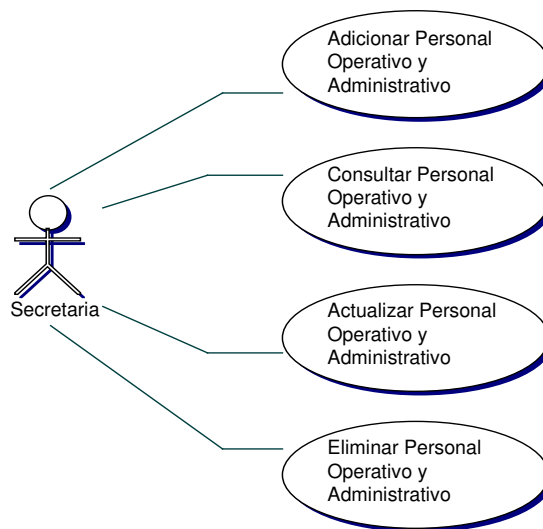
<b>IDENTIFICADOR:</b> APS 1.4		<b>NOMBRE:</b> Eliminar Personal de Salud.
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Personal de Salud.	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.

<b>ENTRADA:</b> <i>Servicio, Identificación</i>	<b>SALIDA:</b> <i>Personal de salud eliminado o proceso cancelado.</i>
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo eliminar un registro del personal de salud.</i>	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores.</b>	<b>Respuesta del Sistema</b>
1. <i>Selecciona la opción eliminar personal de salud.</i>	2. <i>Permite seleccionar el servicio de salud al que pertenece el personal a eliminar.</i>
3. <i>Selecciona el servicio(a)</i>	4. <i>Se despliega un listado del personal que pertenece al servicio de salud seleccionado.</i>
5. <i>Selecciona el personal a eliminar.</i>	
6. <i>Da la orden para eliminar el registro.</i>	7. <i>Pide confirmación para efectuar el proceso.</i>
8. <i>Confirma la realización del proceso</i>	9. <i>Muestra un mensaje indicando que el proceso se efectuó con éxito.</i>
<b>CAMINOS DE EXCEPCION:</b>	
(a) <i>El usuario decide no eliminar el personal de salud. El sistema cancela el proceso.</i>	
<ul style="list-style-type: none"> <li>• <i>El sistema no responde. El proceso no se efectúa</i></li> </ul>	
<i>Precondición: Consistencia en la información de la base de datos.</i>	
<i>Postcondición: Eliminación eficiente del personal de salud.</i>	
<b>CRITERIOS DE ACEPTACIÓN</b>	
<i>Se cumple el requerimiento cuando se logra eliminar el personal de salud de forma oportuna y consistente.</i>	

## 9.1.15 Administración de Personal Administrativo y Operativo

### 9.1.15.1 Primera Iteración - Fachada

- Diagrama

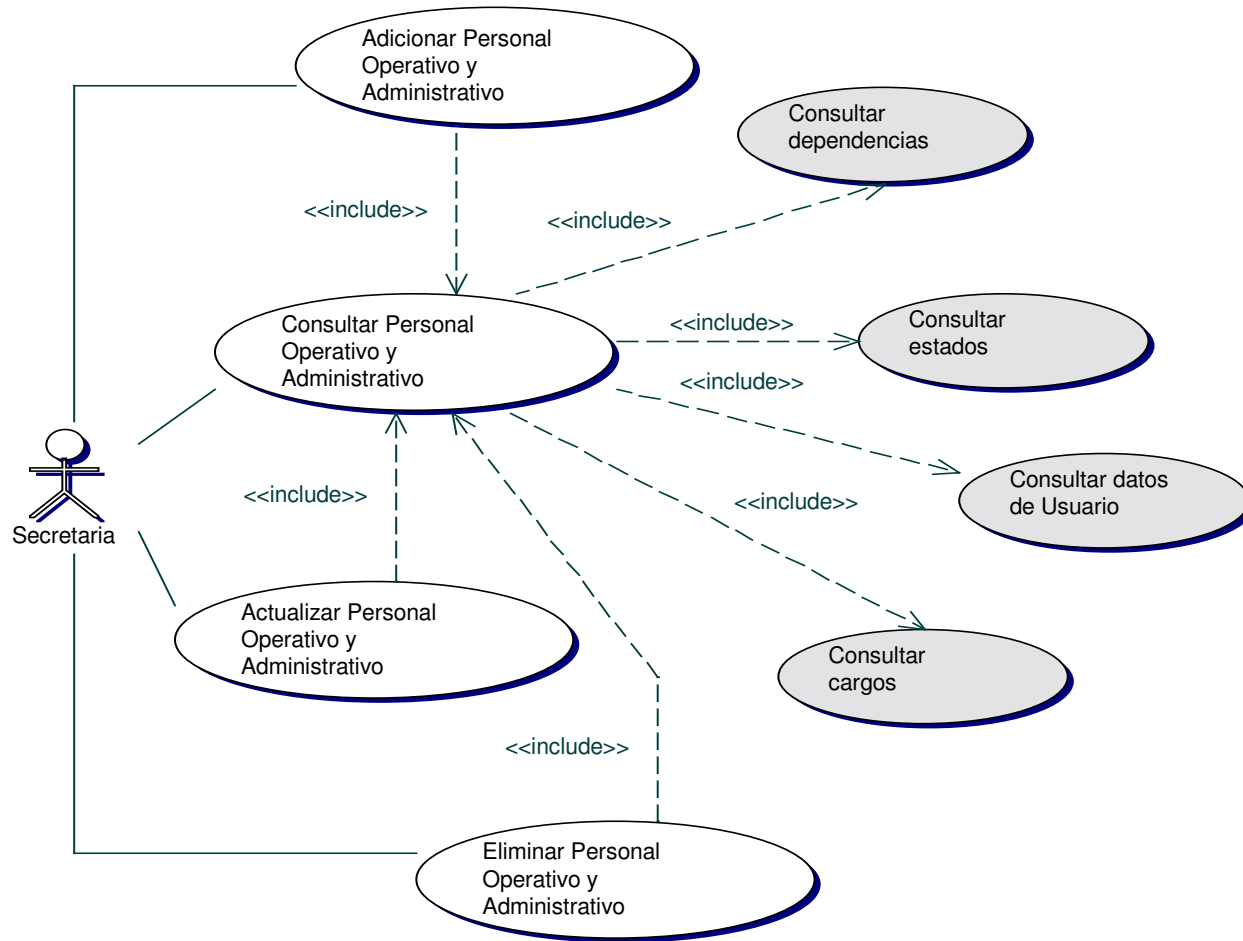


- Iteración

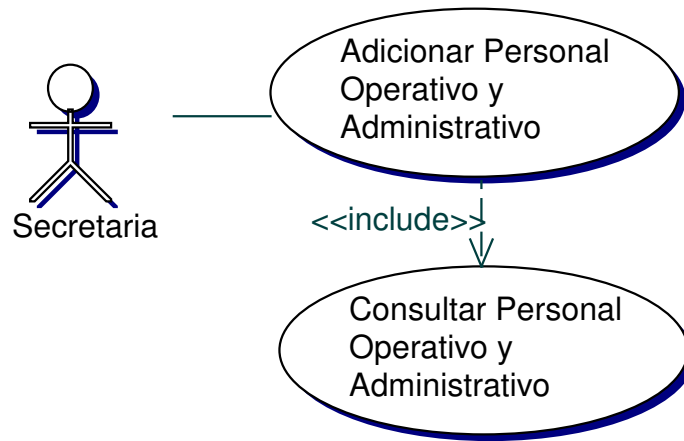
<b>IDENTIFICADOR:</b> APOA 1		<b>NOMBRE:</b> Administración de Personal Administrativo y Operativo	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Adicionar Personal Administrativo y Operativo. Consultar Personal Administrativo y Operativo. Actualizar Personal Administrativo y Operativo. Eliminar Personal Administrativo y Operativo.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los datos del personal Administrativo y operativo de la universidad. Este proceso involucrará las operaciones de adición, consulta, actualización y eliminación.			
<b>OBSERVACIONES:</b> El personal Administrativo y Operativo corresponde a coordinador académico, coordinador general, secretaria, personal de celaduría, personal de aseo etc. Los datos que van a administrar son: Dependencia, Cargo y Estado.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: El usuario se encuentra correctamente registrado en la base de datos. Postcondición: Administración eficiente del personal Operativo y administrativo.			
<b>CRITERIOS DE ACEPTACION</b> Permite llevar una administración sistemática del personal Administrativo y Operativo. La información en la base de datos se maneja de forma persistente y consistente.			



### 9.1.15.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Adicionar Personal Operativo y Administrativo

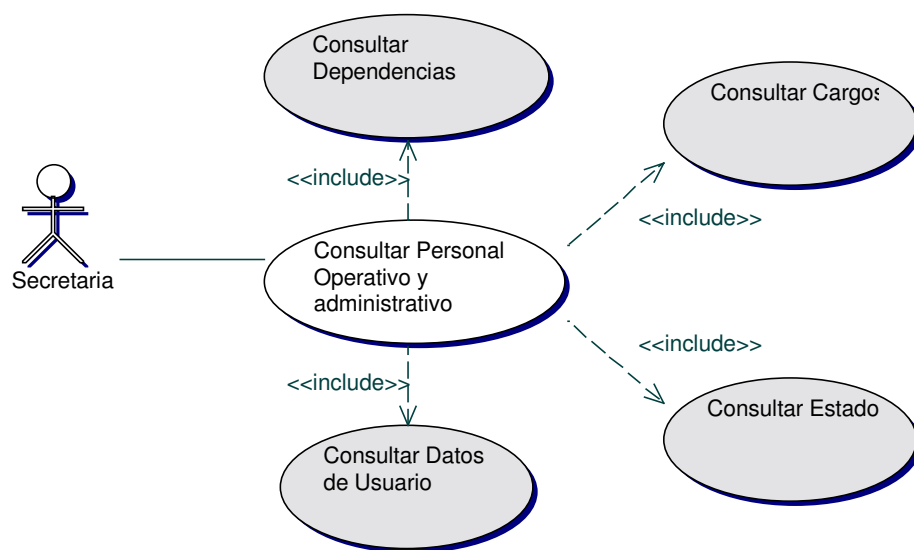


- Iteración Adicionar Personal Operativo y Administrativo

<b>IDENTIFICADOR:</b> APOA 1.1		<b>NOMBRE:</b> Adicionar Personal Operativo y Administrativo	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Personal Operativo y Administrativo.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Dependencia, Cargo y Estado		<b>SALIDA:</b> Personal Operativo y Administrativo registrado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar el personal Operativo y Administrativo.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Seleccionar la opción adición personal Operativo y administrativo.	2.	Solicita la identificación.
3.	Digita la identificación	4.	Verifica que el personal operativo y administrativo no se encuentre registrado (b)
6.	Ingresa los datos solicitados. (a)	5.	Solicita los datos.
7.	Da la orden para registrar el personal operativo y administrativo. (a)	8.	Muestra un mensaje indicando que el registro se efectuó con éxito.

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El actor decide no registrar el personal operativo y administrativo. El sistema cancela el proceso.</p> <p>(b) El usuario a registrar ya forma parte del personal operativo y administrativo. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.</p>
<p>Precondición: Consistencia en la información de la base de datos.</p> <p>Postcondición: Registro eficiente de los datos del personal operativo y administrativo.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra registrar el personal operativo y administrativo de forma consistente y persistente.</p>

• **Diagrama Consultar Personal Operativo y administrativo.**

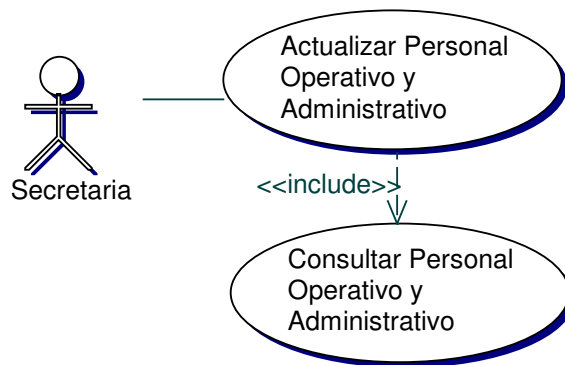


• **Iteración Consultar Personal Operativo y administrativo.**

<b>IDENTIFICADOR:</b> APOA 1.2		<b>NOMBRE:</b> Consultar Personal Operativo y administrativo.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Dependencias Consultar Cargos Consultar estados Consultar datos de usuario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Datos consultados o proceso cancelado.	

<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo consultar el personal operativo y administrativo.</i>
<b>CURSO BÁSICO DE EVENTOS</b> <i>Acción de los actores</i> 1. <i>Selecciona la opción consultar personal operativo y administrativo.</i> <i>Respuesta del Sistema</i> 2. <i>Se despliega un listado con los datos de todo el personal operativo y administrativo</i>
<b>CAMINOS DE EXCEPCIÓN:</b> <i>El sistema no responde. El proceso no se efectúa</i>
<i>Precondición: Consistencia en la información de la base de datos.</i> <i>Postcondición: Acceso rápido a los datos del personal operativo y administrativo.</i>
<b>CRITERIOS DE ACEPTACIÓN</b> <i>Se cumple el requerimiento cuando se logra consultar los datos del personal operativo y administrativo de forma oportuna y veraz.</i>

- **Diagrama Actualizar Personal Operativo y Administrativo**

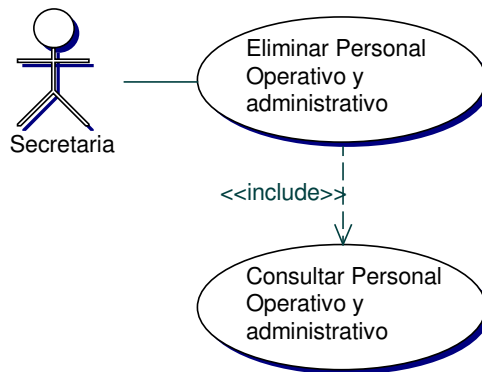


- **Iteración Actualizar Personal Operativo y Administrativo**

<b>IDENTIFICADOR:</b> APOA 1.3		<b>NOMBRE:</b> Actualizar Personal Operativo y administrativo	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Personal Operativo y administrativo		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, dependencia, cargo y estado.		<b>SALIDA:</b> Datos del personal administrativo y operativo actualizados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo actualizar los datos del personal operativo y administrativo</i>			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción actualizar personal administrativo y operativo.	2. Solicita la identificación.
3. Digita la identificación. <b>(a)</b>	
4. Da la orden para que se muestre los datos del usuario. <b>(a)</b>	5. Despliega el nombre completo del usuario permitiendo realizar los respectivos cambios <b>(b)</b>
6. Realiza las respectivas modificaciones.	
7. Da la orden de registrar los cambios realizados.	8. Muestra un mensaje indicando que los datos se actualizaron con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor decide no actualizar los datos. El sistema cancela el proceso.	
<b>(b)</b> Identificador no valido. El sistema informa al actor iniciador que la identificación ingresada es errónea.	
• El sistema no responde. El proceso no se efectúa	
Precondición: Consistencia en la información de la base de datos.	
Postcondición: Actualización eficiente de los datos del personal administrativo y operativo.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra actualizar de forma consistente y persistente los datos del personal operativo y administrativo.	

- **Diagrama Eliminar Personal Operativo y Administrativo**



- **Iteración Eliminar Personal Operativo y Administrativo**

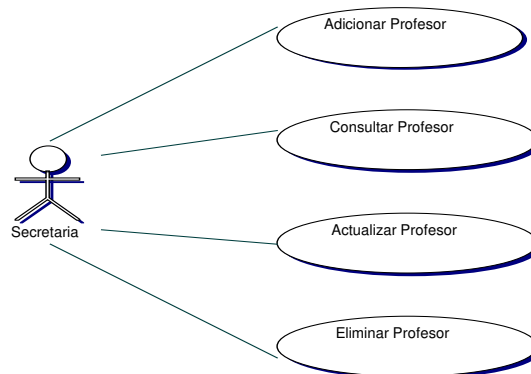
<b>IDENTIFICADOR:</b> APOA 1.4		<b>NOMBRE:</b> Eliminar Personal Operativo y administrativo.
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Personal Operativo y administrativo.	<b>CRITICO?</b> Si

<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Identificación.	<b>SALIDA:</b> Personal operativo y administrativo eliminado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un usuario vinculado al personal operativo y administrativo.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores.</b>		<b>Respuesta del Sistema</b>
1. Selecciona la opción eliminar personal operativo y administrativo.	2. Se despliega un listado con todos los datos del personal administrativo y operativo.	
3. Seleccionar el personal a eliminar.		
4. Da la orden para eliminar el registro.	5. Pide la confirmación para efectuar el proceso. (a)	
6. Confirma la realización del proceso	7. Muestra un mensaje indicando que el proceso se efectuó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>		
(a) El actor decide no eliminar el personal. El sistema cancela el proceso.		
• El sistema no responde. El proceso no se efectúa		
Precondición: Consistencia en la información de la base de datos.		
Postcondición: Eliminación consistente del personal operativo y administrativo		
<b>CRITERIOS DE ACEPTACIÓN</b>		
Se cumple el requerimiento cuando se logra eliminar el personal operativo y administrativo de forma oportuna y consistente.		

## 9.1.16 Administración de Profesores

### 9.1.16.1 Primera Iteración - Fachada

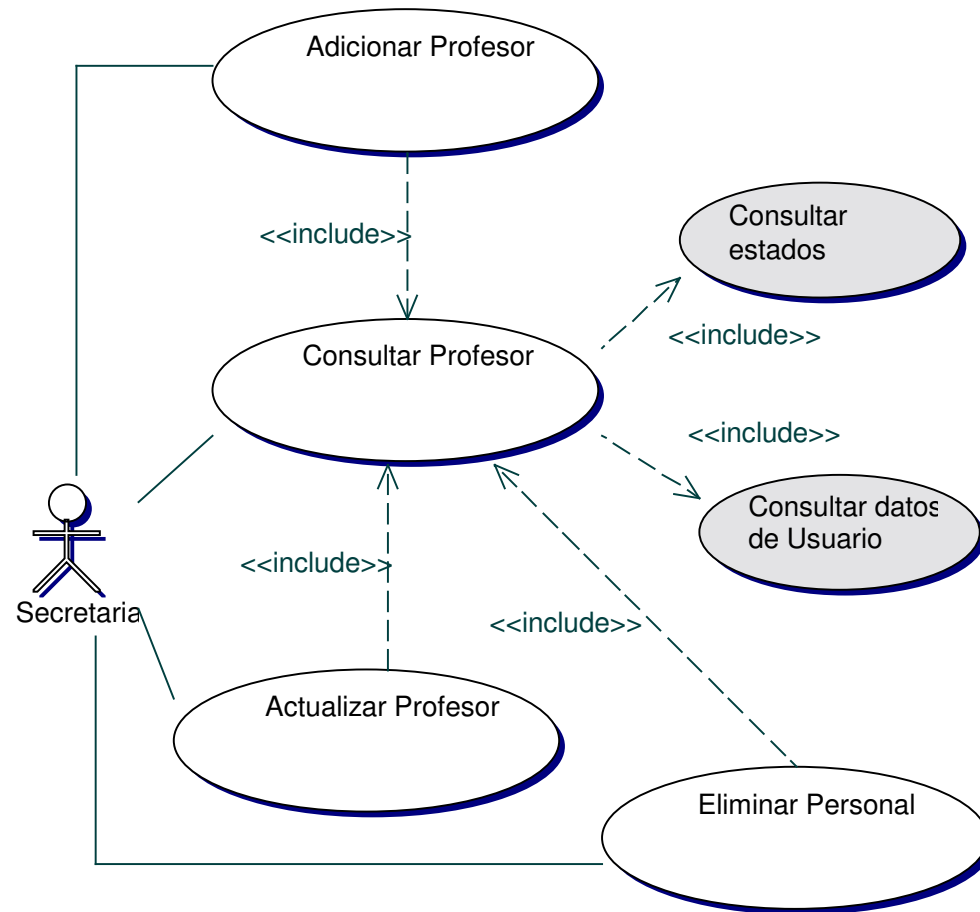
- Diagrama



- Iteración

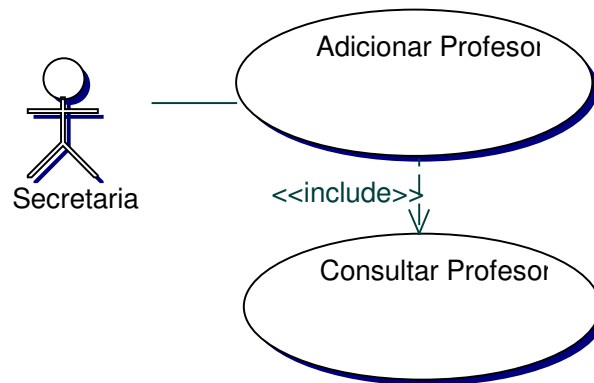
<b>IDENTIFICADOR:</b> AP 1		<b>NOMBRE:</b> Administración de Profesores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Adicionar Profesores Consultar Profesores Actualizar Profesores Eliminar Profesores.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los datos del personal docente de la universidad. Este proceso involucrará las operaciones de adición, consulta, actualización y eliminación.			
<b>OBSERVACIONES:</b> Los datos que van a administrar son: Título, Correo Electrónico y Estado.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: El usuario se encuentra correctamente registrado en la base de datos. Postcondición: Administración eficiente del personal docente.			
<b>CRITERIOS DE ACEPTACION</b> Permite llevar una administración sistemática del personal docente. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.16.2 Segunda Iteración – Llenado (Diagrama General)





- Diagrama Adicionar Profesores



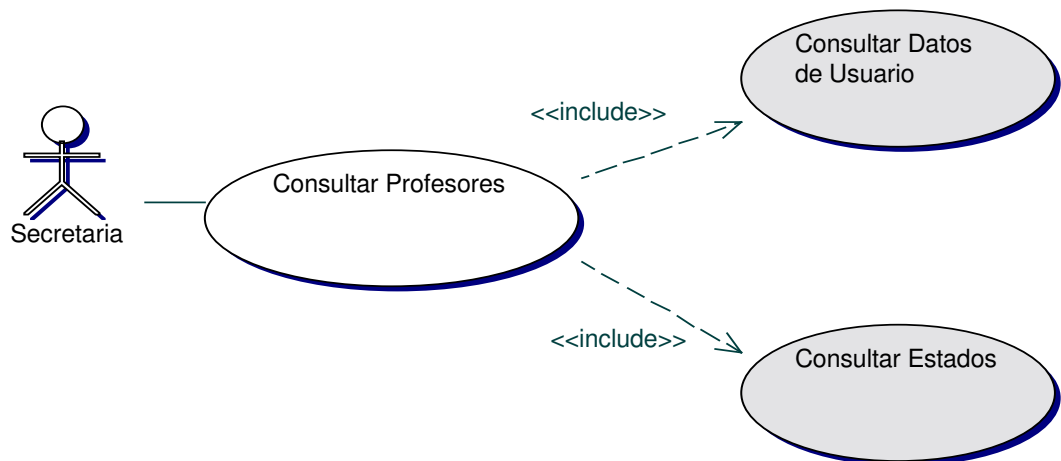
- Iteración Adicionar Profesores

<b>IDENTIFICADOR:</b> A P 1.1		<b>NOMBRE:</b> Adicionar Profesores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Profesores.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Título, Correo electrónico y Estado		<b>SALIDA:</b> Personal docente registrado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar el personal docente.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Seleccionar la opción adición profesor.		2. Solicita la identificación.	
3. Digita la identificación		4. Verifica que el profesor no se encuentre registrado <b>(b)</b> .	
6. Ingresar los datos solicitados. <b>(a)</b>		5. Solicita los datos.	
7. Da la orden para registrar el profesor. <b>(a)</b>		8. Muestra un mensaje indicando que el registro se efectuó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El actor decide no registrar el profesor. El sistema cancela el proceso.			
<b>(b)</b> El profesor a registrar ya se encuentra registrado. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Registro eficiente de los datos del personal docente.			

**CRITERIOS DE ACEPTACIÓN**

Se cumple el requerimiento cuando se logra registrar el personal docente de forma consistente y persistente.

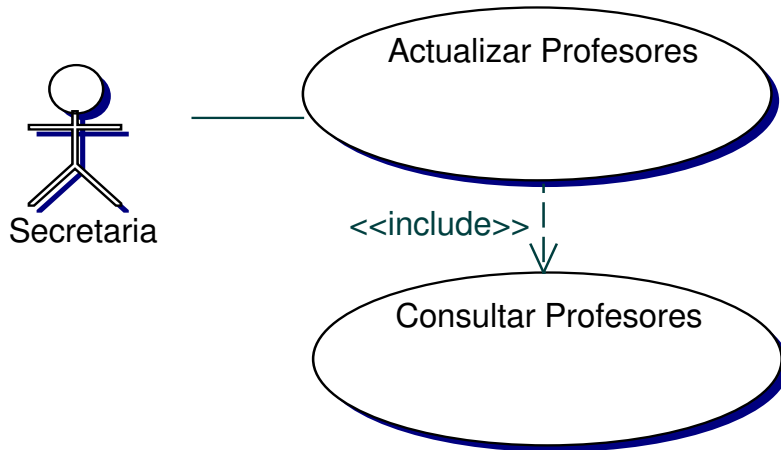
- **Diagrama Consultar Profesores**



- **Iteración Consultar Profesores**

<b>IDENTIFICADOR:</b> AP 1.2		<b>NOMBRE:</b> Consultar Profesores.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar estados Consultar datos de usuario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Datos de los profesores consultados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar el personal docente.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción consultar profesores.		2. Se despliega un listado con los datos de todo el personal docente	
<b>CAMINOS DE EXCEPCION:</b>			
• El sistema no responde. El proceso no se efectúa			
Precondición: Consistencia en la información de la base de datos. Postcondición: Acceso rápido a los datos del personal docente.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los datos del personal docente de forma oportuna y veraz.			

• Diagrama Actualizar Profesores



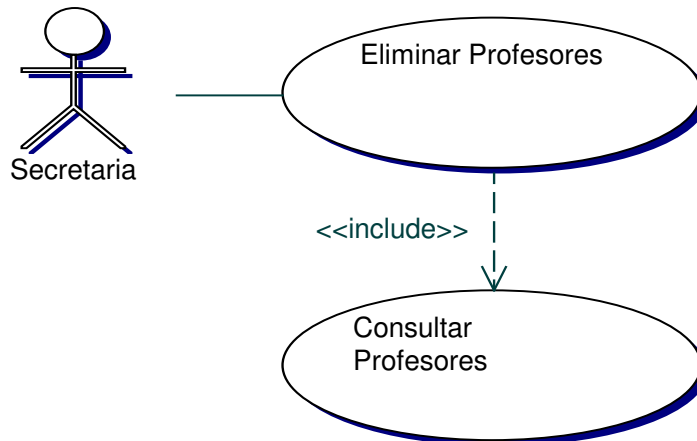
• Iteración Actualizar Profesores

<b>IDENTIFICADOR:</b> A P 1.3		<b>NOMBRE:</b> Actualizar Profesores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Profesores		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, Título, Cargo y Estado		<b>SALIDA:</b> Datos del personal docente actualizados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos del personal docente			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción actualizar profesores		2. Solicita la identificación.	
3. Digita la identificación. (a)			
4. Da la orden para que se muestre los datos del profesor. (a)		5. Despliega los datos del profesor para que se puedan realizar los respectivos cambios (b).	
6. Realiza las respectivas modificaciones.			
7. Da la orden de registrar los cambios realizados.		8. Muestra un mensaje indicando que los datos se actualizaron con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide no actualizar los datos. El sistema cancela el proceso.			
(b) Identificador no valido. El sistema informa al actor iniciador que la identificación ingresada es errónea.			
• El sistema no responde. El proceso no se efectúa			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Actualización eficiente de los datos del personal docente.			

**CRITERIOS DE ACEPTACIÓN**

Se cumple el requerimiento cuando se logra actualizar de forma consistente y persistente los datos del personal docente.

- Diagrama Eliminar Profesores



- Iteración Eliminar Profesores

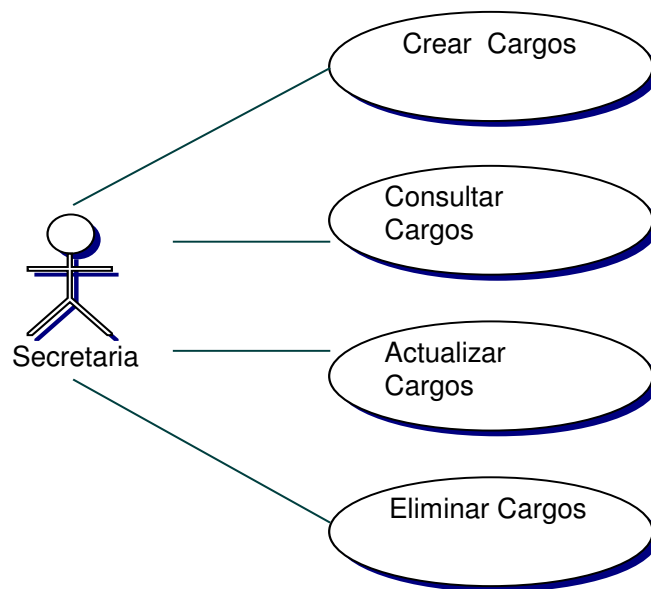
<b>IDENTIFICADOR:</b> AP 1.4		<b>NOMBRE:</b> Eliminar Profesores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Profesores		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación.		<b>SALIDA:</b> Docente eliminado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un docente.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores.</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción eliminar profesor.	2.	Se despliega un listado con todos los datos del personal docente
3.	Seleccionar el personal a eliminar.		
4.	Da la orden para eliminar el registro.		
6.	Confirma la realización del proceso	5.	Pide la confirmación para efectuar el proceso. (a)
		7.	Muestra un mensaje indicando que el proceso se efectuó con éxito.

<b>CAMINOS DE EXCEPCION:</b> (a) El actor decide no eliminar el personal. El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa
Precondición: Consistencia en la información de la base de datos. Postcondición: Eliminación consistente de los docentes.
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra eliminar el personal docente de forma oportuna y consistente.

## 9.1.17 Administración de Cargos

### 9.1.17.1 Primera Iteración - Fachada

- Diagrama

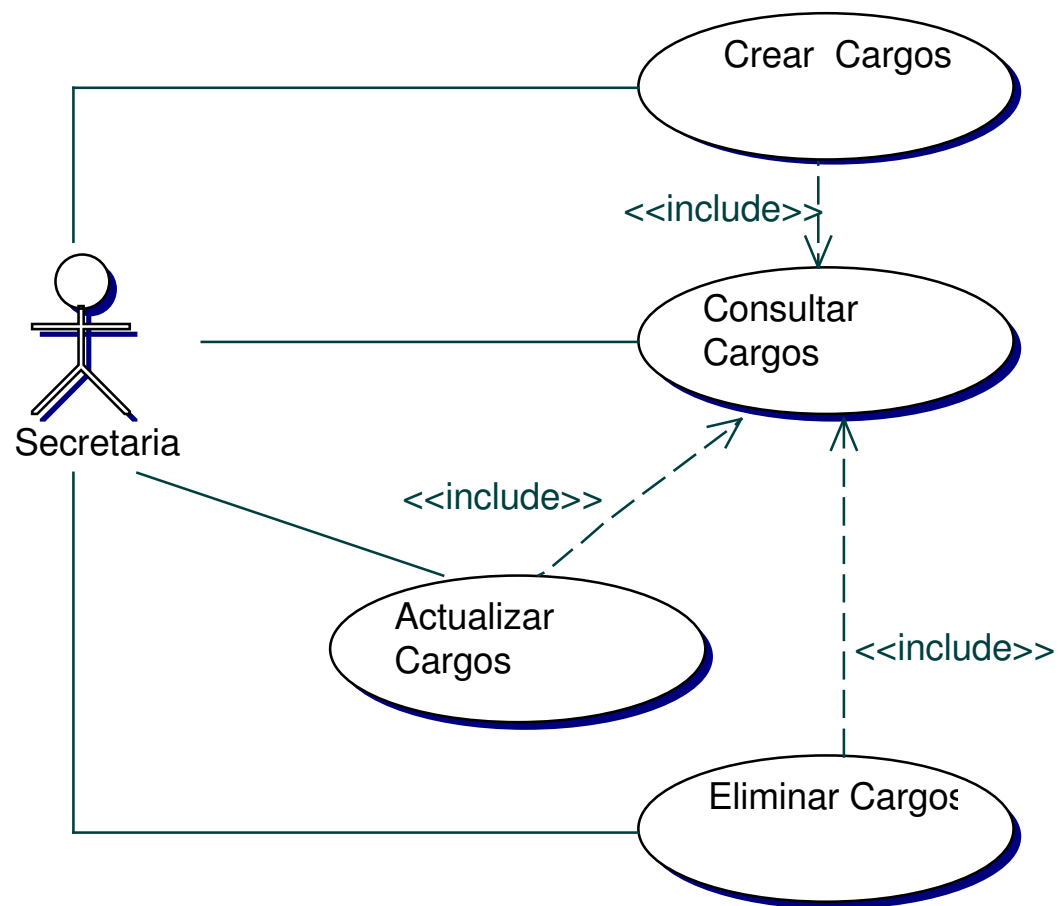


- Iteración

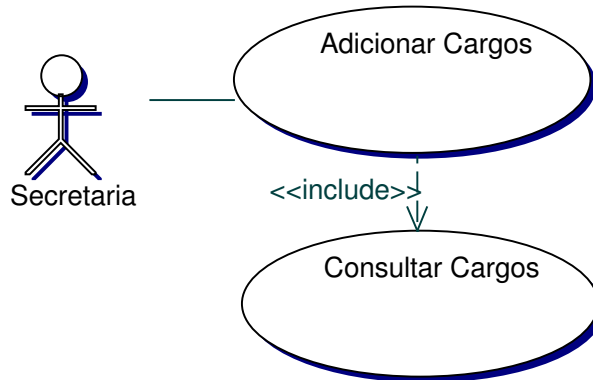
<b>IDENTIFICADOR:</b> AC 1		<b>NOMBRE:</b> Administración de cargos
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Adicionar cargo Consultar cargo Actualizar cargo Eliminar cargo.	<b>CRITICO?</b> Si
<b>PRIORIDAD DE</b>	<b>ACTORES:</b> Actor Iniciador y Ejecutor: Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b>

<b>DESARROLLO:</b> <i>Fachada</i>	<i>Páginas HTML y JSP.</i>
<b>ENTRADA:</b> <i>Datos para efectuar el proceso.</i>	<b>SALIDA:</b> <i>Proceso efectuado o cancelado</i>
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo administrar los cargos laborales de la universidad. Este proceso involucrará las operaciones de adición, consulta, actualización y eliminación.</i>	
<b>OBSERVACIONES:</b> <i>Los datos que van a administrar son: Título, Correo Electrónico y Estado.</i>	
<b>TRIGGERS:</b> <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i>	
<i>Precondición: La información de la base de datos es consistente.</i>	
<i>Postcondición: Administración eficiente de cargos laborales.</i>	
<b>CRITERIOS DE ACEPTACION</b> <i>Permite llevar una administración sistemática de los cargos laborales.</i> <i>La información en la base de datos se maneja de forma persistente y consistente.</i>	

### 9.1.17.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Adicionar Cargos

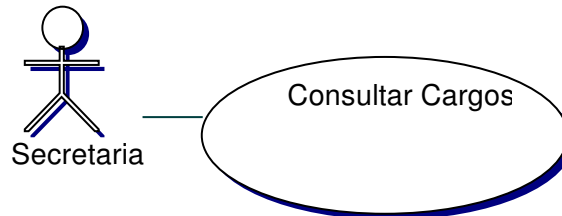


- Iteración Adicionar Cargos

<b>IDENTIFICADOR:</b> A C 1.1		<b>NOMBRE:</b> Adicionar Cargos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Cargos.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Descripción del cargo		<b>SALIDA:</b> Cargos registrados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los cargos laborales.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Seleccionar la opción adición cargos.		2. Solicita la descripción del cargo.	
3. Digita la descripción del cargo.		5. Verifica que el cargo no se encuentre registrado (b).	
4. Da la orden para registrar el profesor. (a)		6. Muestra un mensaje indicando que el registro se efectuó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide no registrar el cargo. El sistema cancela el proceso.			
(b) El cargo ya se encuentra registrado. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Registro eficiente de los cargos laborales.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra registrar los cargos laborales de forma consistente y persistente.			



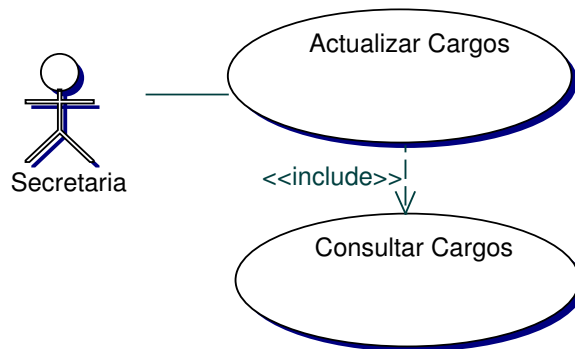
- Diagrama Consultar Cargos



- Iteración Consultar Cargos

<b>IDENTIFICADOR:</b> AC.1.2		<b>NOMBRE:</b> Consultar Cargos.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Cargos consultados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los cargos laborales.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. Selecciona la opción consultar cargos.		2. Se despliega un listado con los cargos laborales que desempeña el personal de la universidad	
<b>CAMINOS DE EXCEPCION:</b> El sistema no responde. El proceso no se efectúa			
Precondición: Consistencia en la información de la base de datos. Postcondición: Acceso rápido a los cargos.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar los cargos de forma oportuna y veraz.			

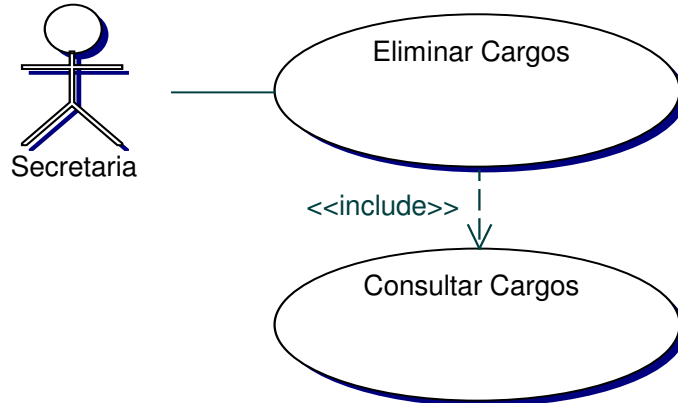
- Diagrama Actualizar Cargos



• Iteración Actualizar Cargos

<b>IDENTIFICADOR:</b> A C 1.3		<b>NOMBRE:</b> Actualizar Cargos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Cargos		<b>CRITICO?</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del cargo, descripción del cargo.		<b>SALIDA:</b> Cargos actualizados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar cargos laborales.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción actualizar cargos		2. Permite seleccionar el cargo a actualizar.	
3. Selecciona el cargo que se va a actualizar.			
(a)			
4. Modifica la descripción del cargo. (a)		6. Verifica que la descripción del cargo no se encuentre registrada (b)	
5. Da la orden para registrar los cambios realizados.		7. Muestra un mensaje indicando que los datos se actualizaron con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide no actualizar el cargo. El sistema cancela el proceso.			
(b) Descripción del cargo ya registrada. El sistema informa al actor iniciador que la descripción ya se encuentra registrada en la Base de datos y se cancela el proceso.			
• El sistema no responde. El proceso no se efectúa			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Actualización eficiente de los cargos laborales.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar de forma consistente y persistente los cargos laborales.			

- Diagrama Eliminar Cargos



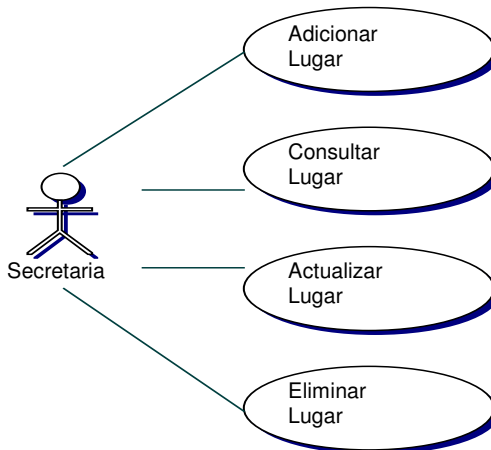
- Iteración Eliminar Cargos

<b>IDENTIFICADOR:</b> A.C. 1.4		<b>NOMBRE:</b> Eliminar Cargos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Cargos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del cargo.		<b>SALIDA:</b> Cargo eliminado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un cargo.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores.</i>		<i>Respuesta del Sistema</i>	
1.	Selecciona la opción eliminar cargo.	2.	Se despliega un listado con todos los cargos laborales.
3.	Seleccionar el cargo a eliminar.		
4.	Da la orden para eliminar el registro.		
6.	Confirma la realización del proceso	5.	Pide la confirmación para efectuar el proceso. (a)
		7.	Muestra un mensaje indicando que el proceso se efectuó con éxito.
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide no eliminar el cargo. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Eliminación consistente de los cargos.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar los cargos de forma oportuna y consistente.			

## 9.1.18 Administración de Lugares

### 9.1.18.1 Primera Iteración - Fachada

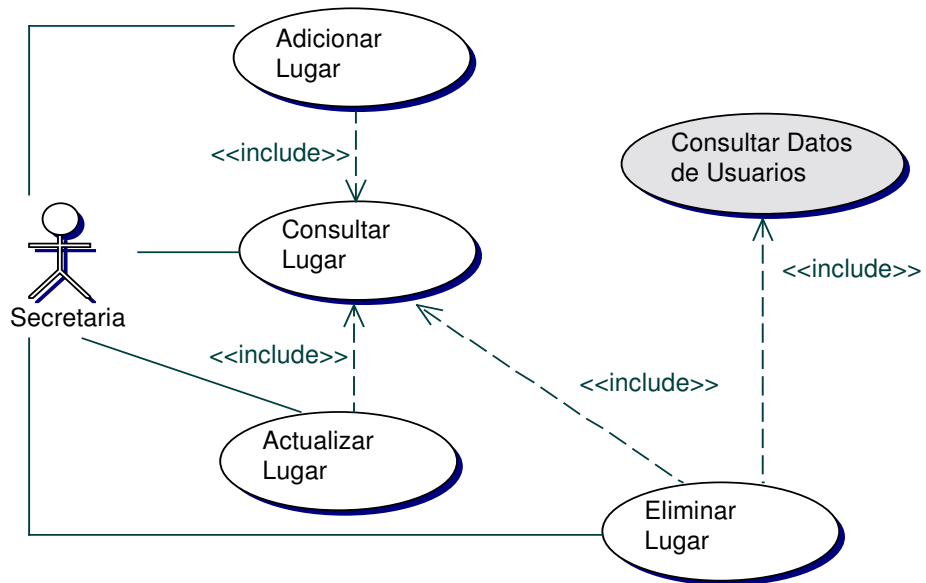
- Diagrama



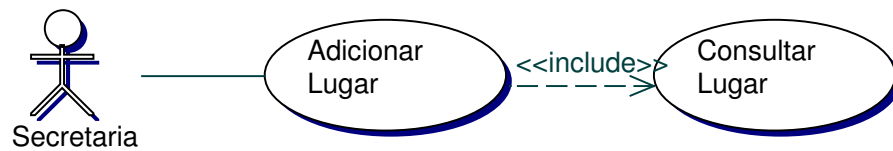
- Iteración

<b>IDENTIFICADOR:</b> <i>AL 1</i>		<b>NOMBRE:</b> <i>Administración de lugar</i>	
<b>TIPO:</b> <i>Necesario</i>	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> <i>Adicionar lugar Consultar lugar Actualizar lugar Eliminar lugar</i>		<b>CRITICO?</b> <i>Si</i>
<b>PRIORIDAD DE DESARROLLO:</b> <i>Fachada</i>	<b>ACTORES:</b> <i>Actor Iniciador y Ejecutor: Secretaria.</i>	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> <i>Páginas HTML y JSP.</i>	
<b>ENTRADA:</b> <i>Datos para efectuar el proceso.</i>		<b>SALIDA:</b> <i>Proceso efectuado o cancelado</i>	
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo administrar los lugares. Este proceso involucrará las operaciones de adición, consulta, actualización y eliminación.</i>			
<b>OBSERVACIONES:</b> <i>Los lugares hace referencia a ciudades, departamentos y países.</i>			
<b>TRIGGERS:</b> <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i>			
<i>Precondición: La información de la base de datos es consistente. Postcondición: Administración eficiente de los lugares.</i>			
<b>CRITERIOS DE ACEPTACION</b> <i>Permite llevar una administración sistemática los lugares. La información en la base de datos se maneja de forma persistente y consistente.</i>			

### 9.1.18.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Adicionar Lugar

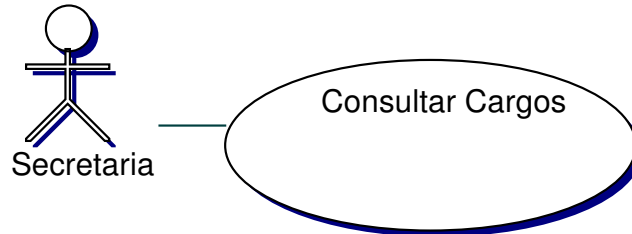


- Iteración Adicionar Lugar

<b>IDENTIFICADOR:</b> A L 1.1		<b>NOMBRE:</b> Adicionar Lugar	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Lugar.		<b>CRITICO?:</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Descripción del Lugar, Raíz.		<b>SALIDA:</b> Lugares registrados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los lugares (Ciudades, departamentos y países).			

<b>OBSERVACION:</b> El dato raíz es un código que indica si el lugar es una ciudad, un departamento o un país. Cuando se registre un país la raíz será igual a cero. Cuando se registre un departamento, la raíz será el código del país al que pertenece. Cuando se registre una ciudad, la raíz será el código del departamento al que pertenece la ciudad.	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Adicionar País</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. Seleccionar la opción adición país.</li> <li>3. Digita la descripción del país.</li> <li>4. Da la orden para registrar el país. <b>(a)</b></li> </ol>	<ol style="list-style-type: none"> <li>2. Solicita la descripción del país.</li> <li>5. Verifica que el país no se encuentre registrado <b>(b)</b>.</li> <li>6. Muestra un mensaje indicando que el registro se efectuó con éxito.</li> </ol>
<b>Adicionar Departamento</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. Seleccionar la opción adición departamento.</li> <li>3. Selecciona el país y Digita la descripción del departamento.</li> <li>4. Da la orden para registrar el departamento. <b>(a)</b></li> </ol>	<ol style="list-style-type: none"> <li>2. Permite elegir el país al que va a pertenecer el departamento a registrar y Solicita la descripción del departamento.</li> <li>5. Verifica que el departamento no se encuentre registrado <b>(b)</b></li> <li>6. Muestra un mensaje indicando que el registro se efectuó con éxito.</li> </ol>
<b>Adicionar Ciudad</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. Seleccionar la opción adición ciudad.</li> <li>3. Selecciona el país y el departamento y Digita la descripción de la ciudad.</li> <li>4. Da la orden para registrar la ciudad. <b>(a)</b></li> </ol>	<ol style="list-style-type: none"> <li>2. Permite elegir el país y el departamento al que va a pertenecer la ciudad a registrar y Solicita la descripción de la ciudad.</li> <li>5. Verifica que la ciudad no se encuentre registrada <b>(b)</b>.</li> <li>6. Muestra un mensaje indicando que el registro se efectuó con éxito.</li> </ol>
<b>CAMINOS DE EXCEPCION:</b> <b>(a)</b> El actor decide no registrar el lugar. El sistema cancela el proceso. <b>(b)</b> El lugar ya se encuentra registrado. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.	
Precondición: Consistencia en la información de la base de datos. Postcondición: Registro eficiente de los lugares.	
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra registrar los lugares de forma consistente y persistente.	

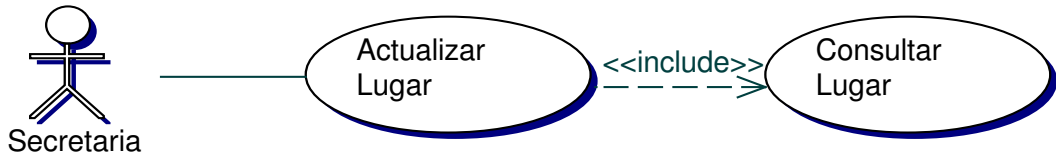
- Diagrama Consultar Lugar



- Iteración Consultar Lugar

<b>IDENTIFICADOR:</b> AL.1.2		<b>NOMBRE:</b> Consultar Lugar.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria,	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del Lugar		<b>SALIDA:</b> Cargos consultados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los Lugares (Pueden ser lugares de Nacimiento, de residencia etc.).			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción consultar Lugares.		2. Se despliega tres listados con los países, departamentos y ciudades. Permitiendo seleccionar el país y el departamento para visualizar las ciudades correspondientes.	
3. Selecciona el país		4. Actualiza el listado de los departamentos y ciudades de acuerdo al país seleccionado, permitiendo seleccionar otro país o departamento.	
5. Selecciona el departamento		6. Actualiza el listado de las ciudades de acuerdo al departamento seleccionado.	
<b>CAMINOS DE EXCEPCION:</b>			
• El sistema no responde. El proceso no se efectúa			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Consulta rápida de los lugares.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los lugares de forma oportuna y veraz.			

- Diagrama Actualizar Lugar



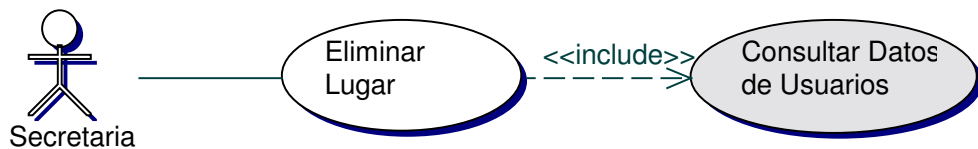
- Iteración Actualizar Lugar

<b>IDENTIFICADOR:</b> A L 1.3		<b>NOMBRE:</b> Actualizar Lugar	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Lugar		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria		<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Código del lugar, descripción del lugar.		<b>SALIDA:</b> Lugares actualizados o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los lugares.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
<b>Actualización de Departamentos</b>			
1. Seleccionar la opción actualización de departamentos.	2. Permite elegir el país al que va a pertenecer el departamento a actualizar y Solicita la nueva descripción del departamento.	3. Selecciona el país, el departamento y Digita la nueva descripción del departamento.	4. Da la orden para actualizar el departamento.
(a)	5. Verifica que el departamento no se encuentre registrado (b)	6. Muestra un mensaje indicando que la actualización se efectuó con éxito.	



<b>Actualización de Países</b>	
1. Seleccionar la opción actualizar país.	2. Permite elegir el país, y Solicita la nueva descripción del país seleccionado.
3. Digita la nueva descripción del país.	
4. Da la orden para registrar el cambio. <b>(a)</b>	5. Verifica que el país no se encuentre registrado <b>(b)</b>
	6. Muestra un mensaje indicando que la actualización se efectuó con éxito.
<b>Actualización de Ciudad</b>	
1. Seleccionar la opción actualizar ciudad.	2. Permite elegir el país y el departamento al que va a pertenecer la ciudad a actualizar y Solicita la nueva descripción de la ciudad.
3. Selecciona el país, el departamento, la ciudad y Digita la nueva descripción de la ciudad.	
4. Da la orden para actualizar la ciudad. <b>(a)</b>	5. Verifica que la ciudad no se encuentre registrada <b>(b)</b>
	6. Muestra un mensaje indicando que la actualización se efectuó con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor decide no actualizar el lugar. El sistema cancela el proceso.	
<b>(b)</b> El lugar ya se encuentra registrado. El sistema informa al actor iniciador que el lugar ya se encuentra registrado en la Base de datos y se cancela el proceso.	
• El sistema no responde. El proceso no se efectúa	
Precondición: Consistencia en la información de la base de datos.	
Postcondición: Actualización eficiente de los lugares.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra actualizar de forma consistente y persistente los lugares.	

## • Diagrama Eliminar Lugares



## • Iteración Eliminar Lugares

<b>IDENTIFICADOR:</b> A.L. 1.4	<b>NOMBRE:</b> Eliminar Lugares
-----------------------------------	------------------------------------

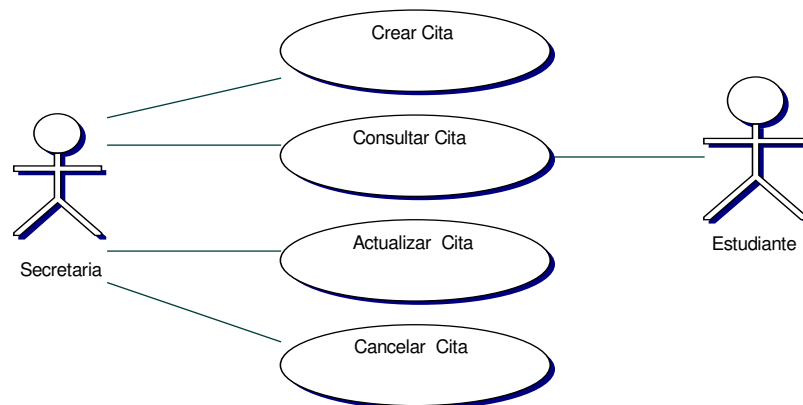
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Lugares Consultar Datos de Usuario	<b>CRITICO?</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Código del lugar	<b>SALIDA:</b> Lugar eliminado o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un lugar.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
<b>Eliminar Países</b>		
1. Seleccionar la opción eliminar país.	2. Muestra un listado con todos los países para que el usuario elija el país a eliminar.	
3. Selecciona el país a eliminar.	5. Pide la confirmación para efectuar el proceso. (a)	
4. Da la orden para eliminar el país. (a)	7. Verifica que el país no tenga departamentos asociados (b).	
6. Confirma la realización del proceso (a)	8. Muestra un mensaje indicando que el registro se eliminó con éxito.	
<b>Eliminar Departamentos</b>		
1. Seleccionar la opción eliminar departamento.	2. Permite elegir el país al que pertenece el departamento a eliminar	
3. Selecciona el país y el departamento a eliminar.	5. Pide la confirmación para efectuar el proceso. (a)	
4. Da la orden para eliminar el departamento. (a)	7. Verifica que el departamento no tenga ciudades asociados (c).	
6. Confirma la realización del proceso (a)	8. Muestra un mensaje indicando que el registro se eliminó con éxito.	
<b>Eliminar Ciudades</b>		
1. Seleccionar la opción eliminar ciudad.	2. Permite elegir el país y el departamento al que pertenece la ciudad a eliminar	
3. Selecciona el país y el departamento a eliminar.	5. Pide la confirmación para efectuar el proceso. (a)	
4. Da la orden para eliminar la ciudad. (a)	7. Verifica que la ciudad no se encuentre registrada en los datos personales de los usuarios (d).	
6. Confirma la realización del proceso (a)	8. Muestra un mensaje indicando que el registro se eliminó con éxito.	

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El actor decide no eliminar el lugar. El sistema cancela el proceso</p> <p>(b) El país tiene departamento asociados: El sistema cancela el proceso</p> <p>(c) El departamento tiene ciudades asociadas: El sistema cancela el proceso</p> <p>(d) La ciudad está utilizándose en los datos personales de los usuarios: El sistema cancela el proceso</p> <p>El sistema no responde. El proceso no se efectúa</p>
<p>Precondición: Consistencia en la información de la base de datos.</p> <p>Postcondición: Eliminación consistente de los lugares.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra eliminar los lugares de forma oportuna y consistente.</p>

## 9.1.19 Administración de Reservas de Citas de Salud (Medicas, Odontologicas, Psicologicas)

### 9.1.19.1 Primera Iteración - Fachada

- Diagrama

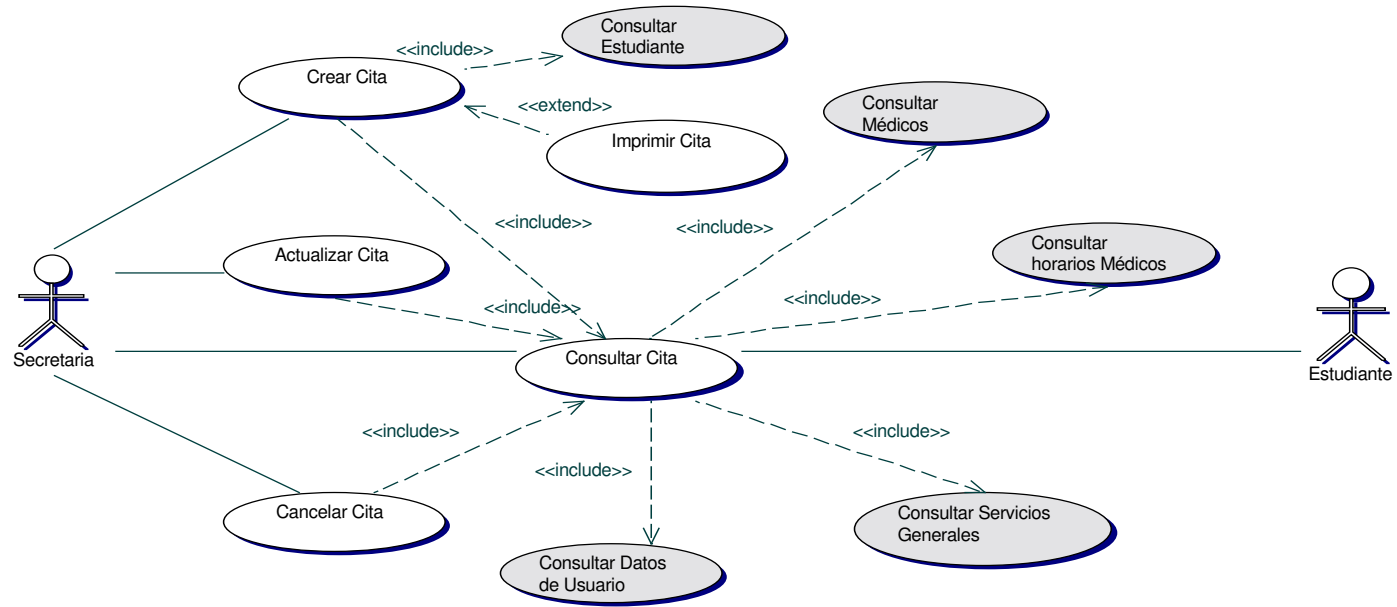


- Iteración

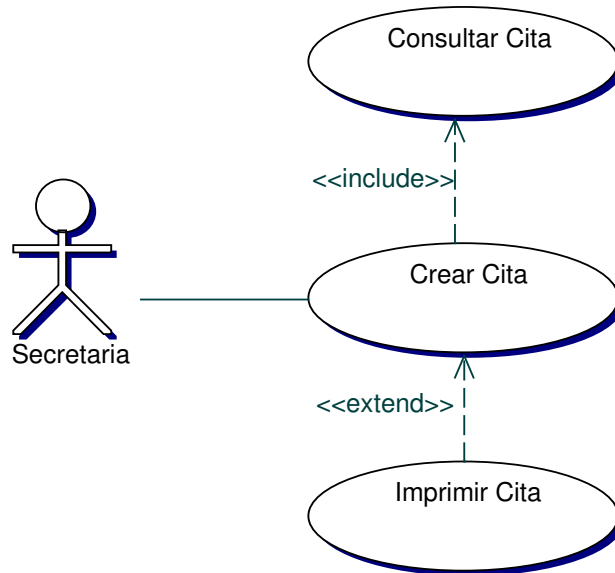
<b>IDENTIFICADOR:</b> ACS1		<b>NOMBRE:</b> Administración Citas.
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Citas. Consultar Citas. Actualizar Citas. Cancelar Citas.	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Actor Iniciador: Estudiante Actor Iniciador: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b>		<b>SALIDA:</b>

<i>Datos para efectuar el proceso.</i>	<i>Proceso efectuado o cancelado</i>
<p><b>DESCRIPCIÓN:</b>  <i>Se satisface este requerimiento permitiendo administrar las citas de salud de los estudiantes de la universidad. Este proceso involucrará las operaciones de creación, consulta, actualización y cancelación.</i></p>	
<p><b>TRIGGERS:</b>  <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i></p>	
<p><i>Precondición: El estudiante se encuentra correctamente registrado en la base de datos.</i>  <i>Postcondición: Administración eficiente de las citas de salud.</i></p>	
<p><b>CRITERIOS DE ACEPTACION</b>  <i>Permite llevar una administración sistemática de las citas de salud.</i>  <i>La información en la base de datos se maneja de forma persistente y consistente.</i></p>	

### 9.1.19.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Crear Citas de Salud ( Médicas, Odontológicas, Psicológicas ).



- Iteración Crear Cita (Médica, Odontológica y Psicológica)

<b>IDENTIFICADOR:</b> ACS1.1		<b>NOMBRE:</b> Crear Cita.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Citas.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Estudiante. Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, Fecha, Franja y Médico.		<b>SALIDA:</b> Cita registrada o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear una cita de salud.			
<b>OBSERVACIONES:</b> Si no dispone del código, la reserva se puede realizar suministrando la identificación del estudiante.			

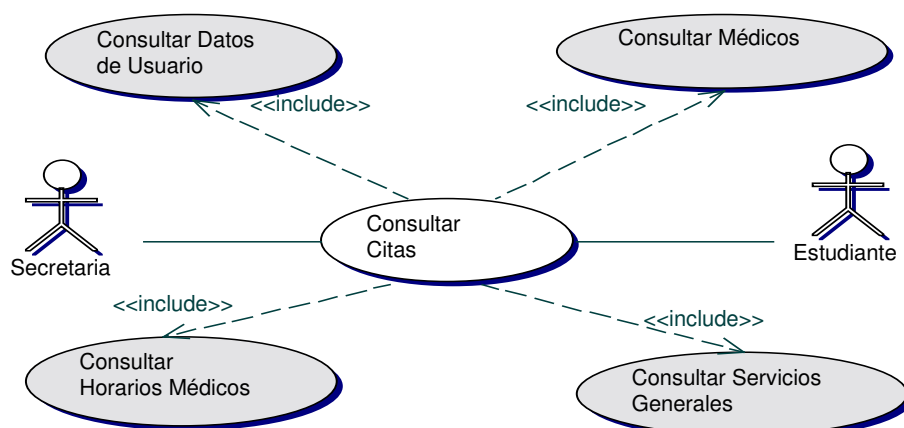
<b>CURSO BÁSICO DE EVENTOS</b>	
<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>
1. <i>Selecciona la opción crear cita.</i>	2. <i>Solicita el código del actor iniciador.</i>
3. <i>Ingresa el dato solicitado. (a)</i>	4. <i>Muestra los datos personales del actor iniciador y las opciones para registrar la cita. (b)</i>
5. <i>Selecciona los datos de la cita.</i>	7. <i>Verifica que el actor iniciador no tenga una cita en la fecha y franja seleccionada. (c)</i>
6. <i>Da la orden para crear la cita. (a)</i>	8. <i>Verifica que el actor iniciador no sobrepase el número de citas diarias permitidas. (d)</i>
	9. <i>Verifica la disponibilidad de citas en la fecha y franja seleccionada. (e)</i>
	10. <i>Registra la cita en la base de datos con la información suministrada.</i>
	11. <i>Muestra un mensaje indicando que la creación de la cita se realizó con éxito, permitiendo imprimir la cita de salud.</i>

**CAMINOS DE EXCEPCION:**  
**(a)** El usuario decide no registrar la cita. El sistema cancela el proceso.  
**(b)** Código no valido. El sistema informa al usuario que el código ingresado es erróneo.  
**(c)** El usuario tiene una cita en la franja y fecha seleccionada. El sistema cancela el proceso.  
**(d)** El usuario sobrepaso el número de citas diarias permitidas. El sistema cancela el proceso.  
**(e)** No hay citas disponibles. El sistema cancela el proceso.  
El sistema no responde. El proceso no se efectúa.

**TRIGGERS:**  
*Verificación Estudiante: Este evento se ejecutará para determinar si el actor iniciador se encuentra registrado como estudiante.*  
*Validar que el usuario no sobre pase el límite de citas permitidas.*  
*Precondición: Consistencia en la información de la base de datos.*  
*Postcondición: Creación eficiente de la cita de salud.*

**CRITERIOS DE ACEPTACIÓN**  
*Se cumple el requerimiento cuando se logra crear de forma consistente y persistente las citas de salud.*

• **Diagrama Consultar Cita (Médica, Odontológica, Psicológica)**



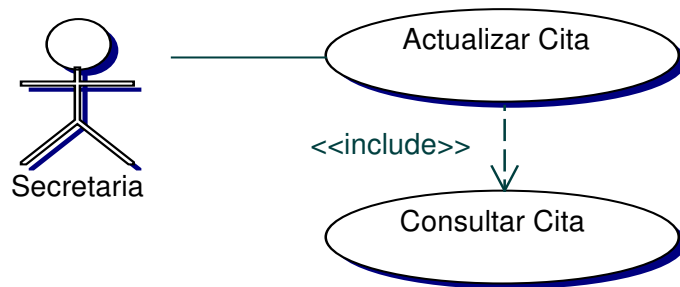
- Iteración Consultar Cita(Médica, Odontológica, Psicológica)

<b>IDENTIFICADOR:</b> ACSI.2		<b>NOMBRE:</b> Consultar Citas.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Estudiantes Consultar Médicos Consultar Horarios Consultar Servicios Generales.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y ejecutor: Secretaria, estudiante.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, identificación.		<b>SALIDA:</b> Datos consultados o proceso cancelado.	
<b>Observación:</b> El requerimiento o método Consultar Horarios hacer referencia al horario médico, odontológico y psicológico			
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las citas (médicas, odontológicas o psicológicas) en la fecha indicada por el usuario. La consulta de reservas puede ser realizada por cualquiera de los actores involucrados en este caso de uso. Se desplegará la información de las reservas existentes y de las franjas disponibles en el día seleccionado por el actor.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
<b>Consulta de disponibilidad realizada por el Estudiante.</b>			
1. Selecciona la opción consultar cita.		2. Solicita la fecha de la cita.	
3. Selecciona la fecha deseada. (a)		5. Se despliega el listado de las citas, mostrando la franja y la disponibilidad en la fecha seleccionada.	
4. Da la orden para que se muestre la disponibilidad.			
<b>Consulta de las citas personales por el Estudiante</b>			
1. Selecciona la opción consultar cita.		2. Solicita la fecha de la cita.	
3. Selecciona la fecha deseada. (a)		5. Despliega un listado con las citas del autor iniciador en la fecha indicada.	
4. Da la orden para que se muestre todas las citas en la fecha seleccionada.			
<b>Consulta de las citas por la secretaria.</b>			
1. Selecciona la opción consultar cita.		2. Solicita la fecha de la cita.	
3. Selecciona la fecha deseada. (a)		5. Despliega un listado con las citas efectuadas en la fecha seleccionada.	
4. Da la orden para que se muestre todas las citas en la fecha seleccionada.			
<b>CAMINOS DE EXCEPCION:</b> (a) El usuario decide no consultar los datos. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa			
<b>TRIGGERS:</b> Verificación Usuario: Este evento se ejecutará para determinar si el actor iniciador se encuentra registrado como usuario y obtener sus datos. Verificación Estudiante: Este evento se ejecutará para determinar si el actor iniciador se encuentra registrado como estudiante. Validar que el usuario no sobre pase el límite de citas permitidas.			



<i>Precondición: Consistencia en la información de la base de datos.</i>
<i>Postcondición: Acceso rápido a los datos de las citas.</i>
<b>CRITERIOS DE ACEPTACIÓN</b>
<i>Se cumple el requerimiento cuando se logra consultar de forma oportuna y veraz las citas de salud.</i>

- **Diagrama Actualizar Cita (Médica, Odontológica y Psicológica)**

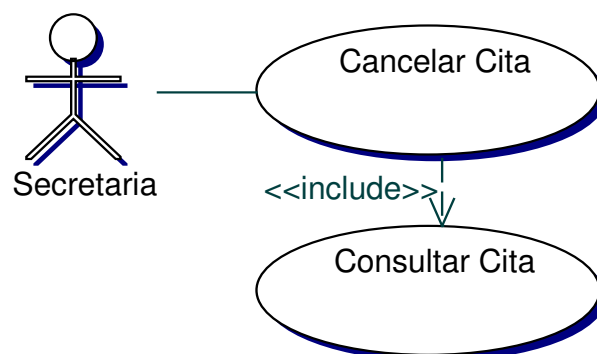


- **Iteración Actualizar Cita (Médica, Odontológica y Psicológica)**

<b>IDENTIFICADOR:</b> ACS1.3		<b>NOMBRE:</b> Actualizar Cita.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Cita.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Estudiante. Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, código de la cita		<b>SALIDA:</b> Cita actualizada o proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar una cita.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>
1. <i>Selecciona la opción actualizar una cita.</i>	2. <i>Solicita la fecha para mostrar el listado de citas.</i>
3. <i>Selecciona la fecha deseada. (a)</i>	4. <i>Muestra el listado de las citas de la fecha seleccionada.</i>
5. <i>Selecciona la cita a actualizar.</i>	7. <i>Muestra los datos actuales de la cita y le permite seleccionar las nuevas opciones (fecha, franja y médico) para la actualización</i>
6. <i>Da la orden para que se muestren las opciones para actualizar la cita. (a)</i>	10. <i>Solicita confirmación para efectuar el proceso.</i>
8. <i>Selecciona los nuevos datos de la cita.</i>	12. <i>Verifica que el actor iniciador no tenga una cita en la fecha y franja seleccionada (b)</i>
9. <i>Da la orden de actualizar.</i>	13. <i>Verifica que el actor iniciador no sobrepase el número de citas diarias permitidas. (c)</i>
11. <i>Confirma la actualización. (a)</i>	14. <i>Verifica la disponibilidad de citas en la fecha y franja seleccionada. (d)</i>
	15. <i>Actualiza la cita en la base de datos con la información suministrada.</i>
	16. <i>Muestra un mensaje indicando que la actualización de la cita se realizó con éxito.</i>
<b>CAMINOS DE EXCEPCION:</b>	
<i>(a) El usuario decide no actualizar los datos. El sistema cancela el proceso.</i>	
<i>(b) El usuario tiene una cita en la franja y fecha seleccionada. El sistema cancela el proceso.</i>	
<i>(c) El usuario sobrepasa el número de citas diarias permitidas. El sistema cancela el proceso.</i>	
<i>(d) No hay citas disponibles. El sistema cancela el proceso.</i>	
<i>El sistema no responde. El proceso no se efectúa</i>	
<b>TRIGGERS:</b>	
<i>Validar que el usuario no sobre pase el límite de citas permitidas.</i>	
<i>Precondición: Consistencia en la información de la base de datos.</i>	
<i>Postcondición: Actualización eficiente de las citas.</i>	
<b>CRITERIOS DE ACEPTACIÓN</b>	
<i>Se cumple el requerimiento cuando se logra actualizar de forma consistente y persistente las citas.</i>	

- **Diagrama Cancelar Cita (Médica, Odontológica y Psicológica)**



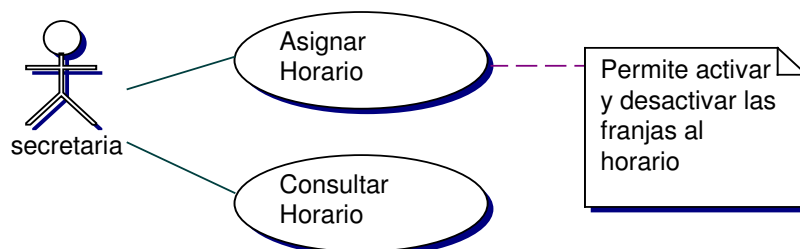
- Iteración Cancelar Cita(Médica, Odontológica y Psicológica)

<b>IDENTIFICADOR:</b> ACSI.4		<b>NOMBRE:</b> Cancelar Cita.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Citas.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Estudiante. Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, código de la cita		<b>SALIDA:</b> Cita cancelada o proceso no efectuado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo cancelar una cita.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. Selecciona la opción cancelar cita.		2. Solicita la fecha de la cita a cancelar.	
3. Selecciona la fecha deseada. (a)		4. Se despliega el listado de todas las citas en la fecha indicada.	
5. Selecciona la cita a cancelar.		7. Pide confirmación para cancelar la cita. (a)	
6. Da la orden para cancelar la cita.		9. Muestra un mensaje indicando que la cancelación de la cita se realizó con éxito.	
8. Confirma la cancelación.			
<b>CAMINOS DE EXCEPCION:</b>			
(a) El usuario decide no cancelar la cita. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Cancelación eficiente de las citas.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra cancelar de forma oportuna y veraz las citas.			

## 9.1.20 Administración de Horarios de Salud (Medico, Odontologico, Psicologico)

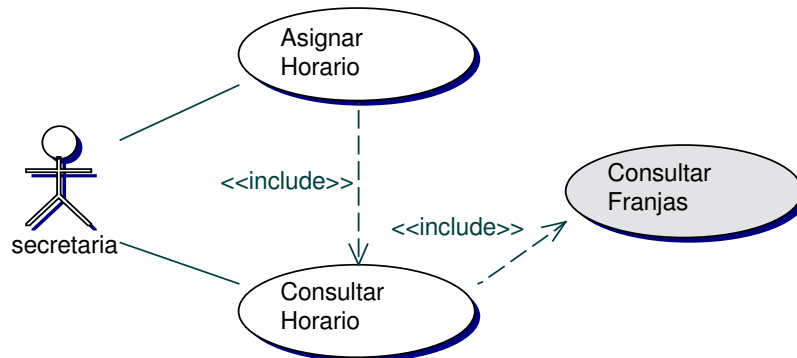
### 9.1.20.1 Primera Iteración - Fachada

- Diagrama

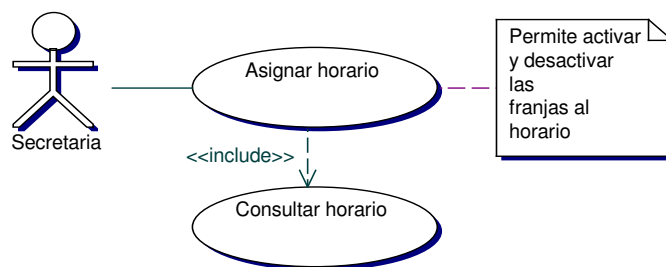


<b>IDENTIFICADOR:</b> AHS1		<b>NOMBRE:</b> Administración Horarios de salud	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Asignación Horario. Consultar Horario.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Actor Iniciador y ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los horarios de salud (horario médico, odontológico y psicológico) este proceso involucrará las operaciones de asignación horario( Por medio de esta operación se podrá asignar las franjas al horario y desactivar las que se encuentran asignadas).			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema. Precondición: El horario se encuentra correctamente registrado en la base de datos. Postcondición: Administración eficiente de los horarios de salud.			
<b>CRITERIOS DE ACEPTACION</b> Permite llevar una administración sistemática de los horarios de salud. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.20.2 Segunda Iteración – Llenado (Diagrama General)



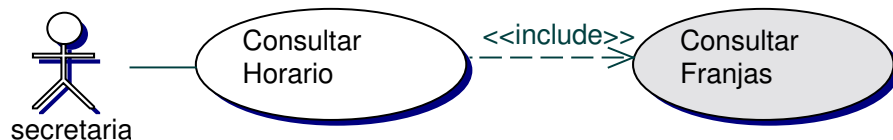
- Diagrama Asignar Horario (Médico, Odontológico, Psicológico).



- Iteración Asignar Horario.

<b>IDENTIFICADOR:</b> AHS1.1		<b>NOMBRE:</b> Asignar Horario.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar horario.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del Día, Código de la franja.		<b>SALIDA:</b> Proceso efectuado o cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo activar o desactivar las franjas al horario del servicio de salud.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Seleccionar la opción Asignar Horario.		2. Despliega un listado de todas las franjas activas y no activas, de acuerdo al día seleccionado.	
3. Elegir el día que se va a asignar o desactivar las franjas.			
4. Activar o desactiva el listado de franjas. (a)		6. Informa al usuario que los cambios realizados al horario podrían afectar a las citas (médicas, odontológicas y psicológicas) realizadas. (a)	
5. Da la orden para actualizar la asignación de las franjas al horario de salud. (a)		7. Muestra un mensaje indicando que el proceso se realizó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El usuario decide cancelar el proceso. El sistema cancela el proceso. El sistema no responde. El proceso no se efectúa.			
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Asegura la Asignación de franjas al horario de forma eficiente.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra activar y desactivar las franjas del horario de salud de forma consistente y persistente			

- Diagrama Consultar Horario (Médico, Odontológico, Psicológico).



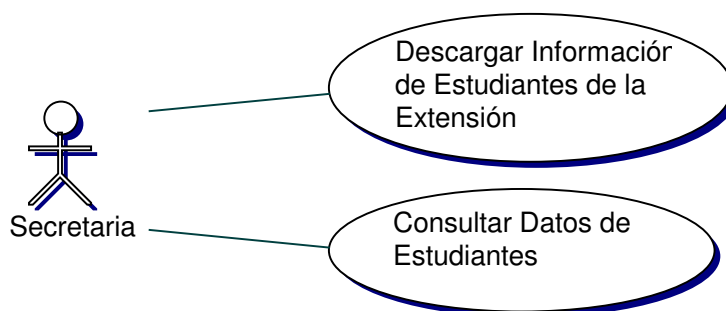
- Iteracion Consultar Horario (Médico, Odontológico, Psicológico).

<b>IDENTIFICADOR:</b> AHSI.2		<b>NOMBRE:</b> Consultar Horario.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar franjas.		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Proceso efectuado o cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar el horario de salud (Médico, odontológico y psicológico).			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. Seleccionar la opción Consultar Horario.		2. Despliega automáticamente el listado de franjas que se prestará el servicio de salud por cada día de la semana.	
Precondición: Consistencia en la información de la base de datos.			
Postcondición: Muestra el horario de servicio de salud de forma eficiente.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se consulta el horario de salud de forma oportuna.			

## 9.1.21 Administración de Estudiantes

### 9.1.21.1 Primera Iteración - Fachada

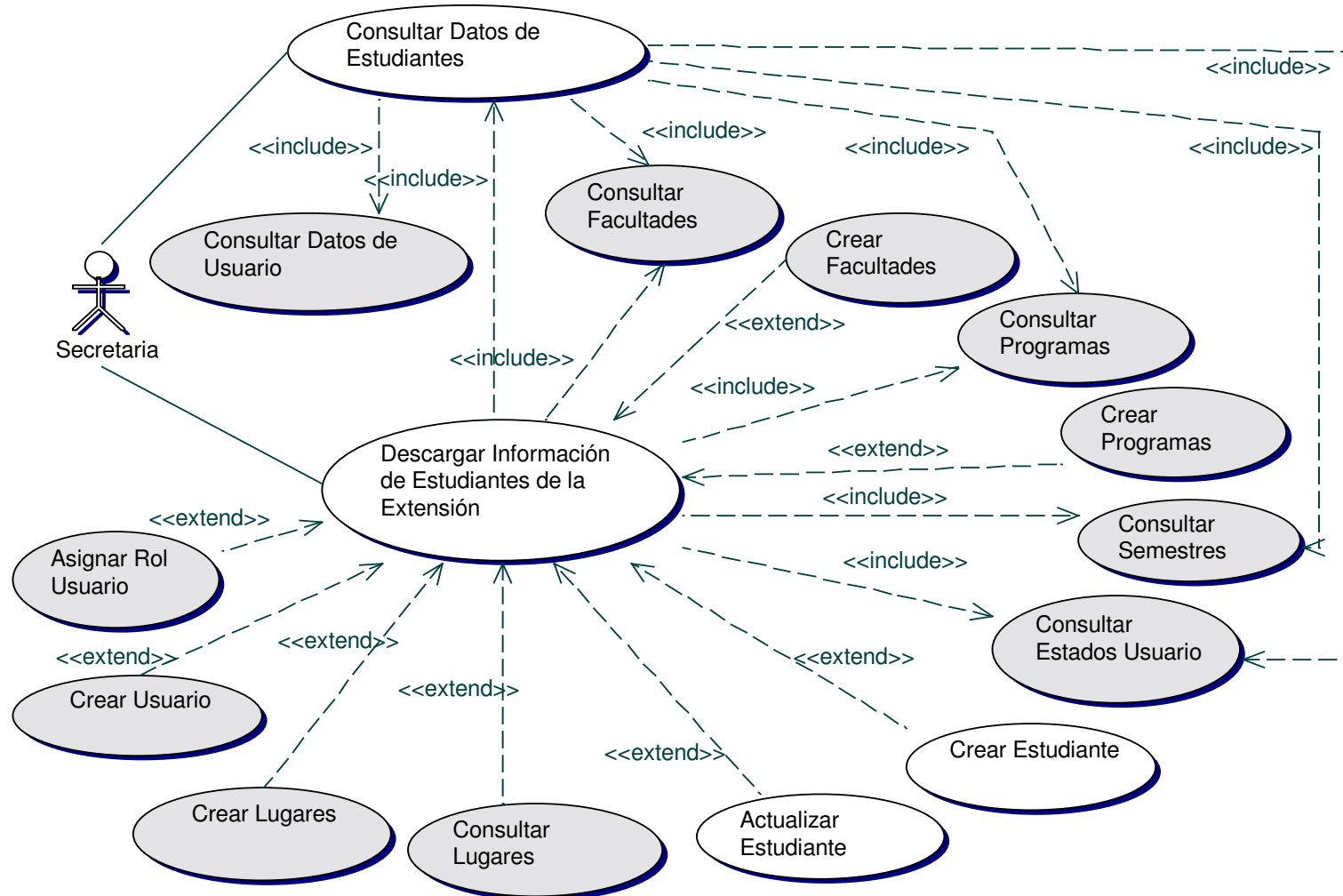
- Diagrama



- Iteración

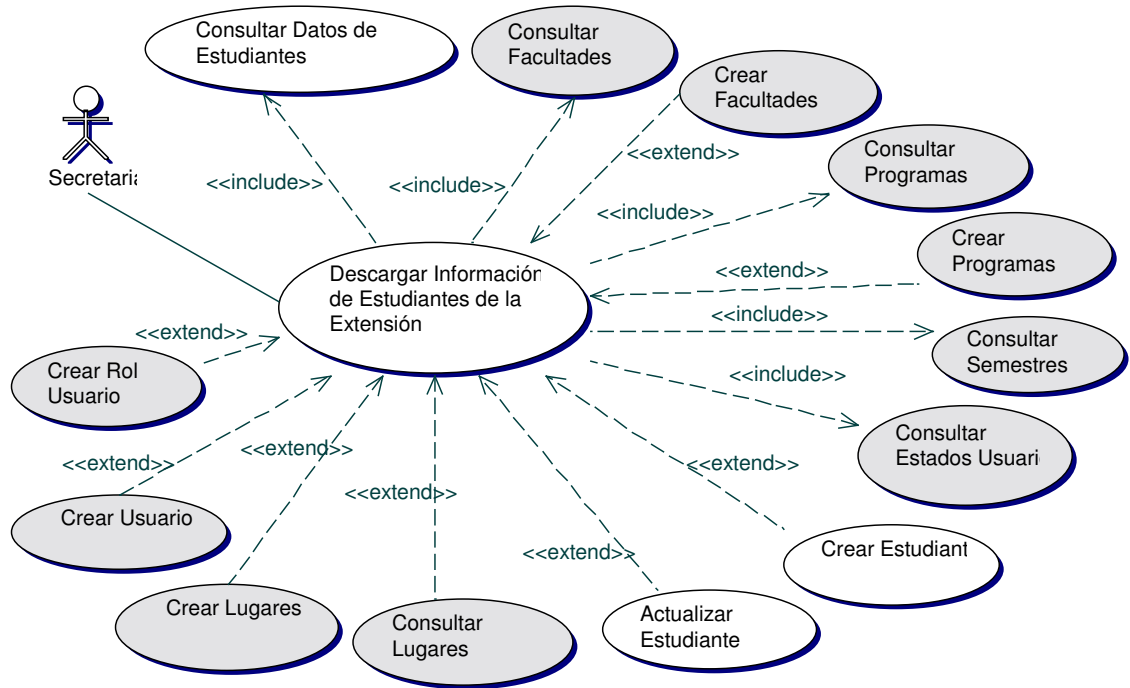
<b>IDENTIFICADOR:</b> AEST1		<b>NOMBRE:</b> Administración de Estudiantes	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Descargar Información de Estudiantes de la Extensión Consultar Datos de Estudiantes		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los datos de los estudiantes de la extensión de acuerdo a la información suministrada desde la sede principal de Pasto. El proceso de administración se desarrollará teniendo en cuenta las operaciones de descarga y consulta.			
<b>OBSERVACIONES:</b> La información de los estudiantes suministrada desde Pasto incluirá datos personales y académicos de los estudiantes. El sistema brindará independencia para el manejo de los datos personales, pero los datos académicos siempre estarán supeditados a la información de Pasto, no podrán ser modificados en la extensión De igual manera el sistema no estará en capacidad de adicionar o eliminar un estudiante de forma independiente, este tipo de operaciones se realizan de forma exclusiva en la sede principal de Pasto.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de los datos de los estudiantes de la extensión.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de los datos de los estudiantes permitiendo acceder a ellos de forma fácil y rápida. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.21.2 Segunda Iteración – Llenado (Diagrama General)





- Diagrama Descargar Información de Estudiantes de la Extensión

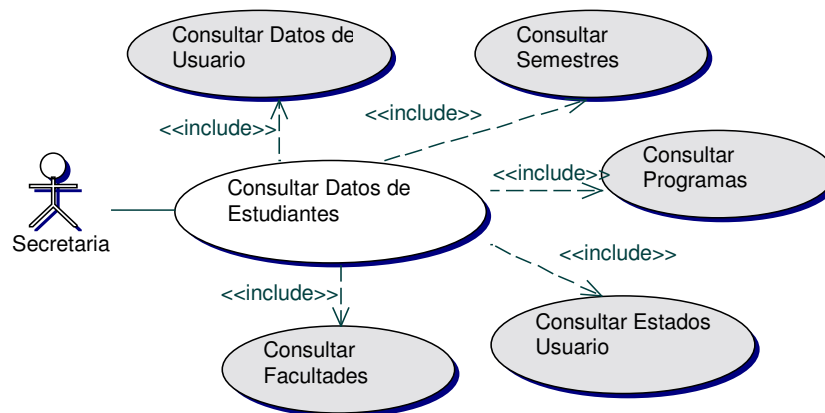


- Iteración Descargar Información de Estudiantes de la Extensión

<b>IDENTIFICADOR:</b> AEST1.1		<b>NOMBRE:</b> Descargar Información de Estudiantes de la Extensión	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de Estudiantes Crear Usuario Crear Lugares Consultar Lugares Crear Facultades Consultar Facultades Consultar Programas Crear Programas Consultar Semestres Consultar Estados Usuario Crear Estudiante Actualizar Estudiante Crear Rol Usuario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretariá		<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b>		<b>SALIDA:</b>	

Archivo de descarga	Proceso efectuado o cancelado						
<p><b>DESCRIPCIÓN:</b>  Se satisface este requerimiento permitiendo descargar la información de los estudiantes de la extensión de acuerdo a los datos suministrados por la sede principal Pasto.  La información suministrada por Pasto será a través de un archivo de Excel que contendrá los datos de todos los estudiantes vinculados con la extensión Ipiales. Los datos serán tanto personales como académicos. El archivo suministrado debe ser editado adecuadamente para que sea compatible con los requerimientos de descarga de la base de datos local. A través del archivo suministrado se podrá ingresar en la base de datos local los datos de los estudiantes de la extensión. Si la información del estudiante que se descarga no existe, se agregará el estudiante como usuario del sistema, y se le asignará automáticamente el rol Estudiante. De igual manera si datos tales como país, departamento, ciudad, facultad o programa aún no se encuentran en la base de datos local serán agregados automáticamente a ella. Si la información del estudiante que se descarga ya existe, el sistema solo actualizará los datos del estudiante que pueden variar con el cambio de semestre, es decir los datos académicos como son: Facultad, programa, semestre y estado.</p>							
<p><b>OBSERVACIONES:</b> El archivo suministrado deberá ubicarse en la misma ruta de la base de datos local.  La descarga deberá realizarse cada semestre para actualizar los datos de los estudiantes antiguos y agregar la información de los posibles estudiantes nuevos.</p>							
<p><b>CURSO BÁSICO DE EVENTOS</b></p> <table border="0"> <thead> <tr> <th style="text-align: center;"><i>Acción de los actores</i></th> <th style="text-align: center;"><i>Respuesta del Sistema</i></th> </tr> </thead> <tbody> <tr> <td>1. EL actor selecciona la opción descargar datos de estudiantes.</td> <td>2. Comienza la descarga de datos a través del archivo que contiene la información.</td> </tr> <tr> <td></td> <td>3. Muestra un mensaje indicando que la descarga se realizó con éxito</td> </tr> </tbody> </table>		<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>	1. EL actor selecciona la opción descargar datos de estudiantes.	2. Comienza la descarga de datos a través del archivo que contiene la información.		3. Muestra un mensaje indicando que la descarga se realizó con éxito
<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>						
1. EL actor selecciona la opción descargar datos de estudiantes.	2. Comienza la descarga de datos a través del archivo que contiene la información.						
	3. Muestra un mensaje indicando que la descarga se realizó con éxito						
<p><b>CAMINOS DE EXCEPCION:</b></p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>							
<p><i>Precondición:</i> La información en la base de datos es consistente  El archivo suministrado contiene la información consistente de los estudiantes</p> <p><i>Postcondición:</i> Descarga eficiente de Estudiantes de la extensión.</p>							
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra descargar la información de los estudiantes de la extensión de forma veraz y oportuna a través de los datos suministrados por la sede principal Pasto.  Fácil actualización de datos académicos de los estudiantes con el cambio de semestre.  La información de la base de datos se almacena de forma persistente y consistente.</p>							

- **Diagrama Consultar Datos de Estudiantes**



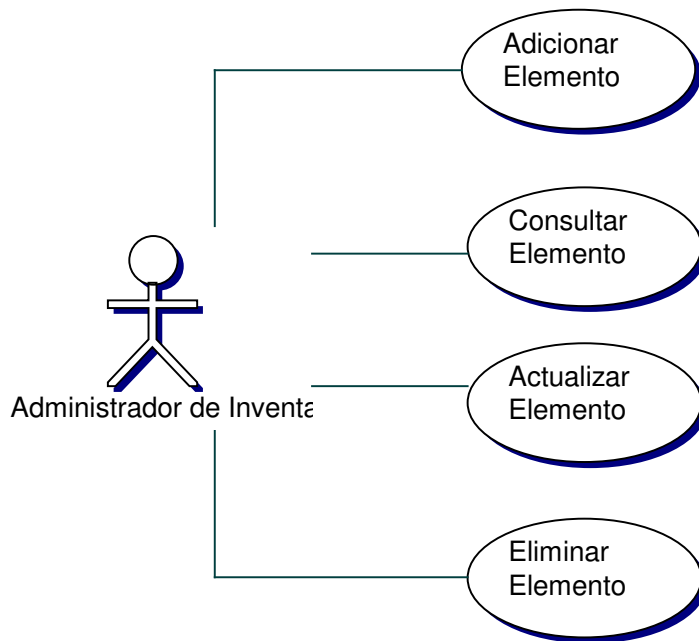
- Iteración Consultar Datos de Estudiantes

<b>IDENTIFICADOR:</b> ARA1.2		<b>NOMBRE:</b> Consultar Datos de Estudiantes	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de usuario Consultar Facultades Consultar Programas Consultar Semestres Consultar Estados Usuarios		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, nombres, programa, semestre, estado.		<b>SALIDA:</b> Datos estudiantes	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los datos de los estudiantes de la extensión Ipiales.			
<b>OBSERVACIONES:</b> La consulta podrá realizarse por el código del estudiante, sus nombres o el programa al que pertenece.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar datos de estudiantes.		2. Despliega un menú presentando las opciones para su consulta (por código, por nombres, por programa).	
3. El actor selecciona la opción para su consulta. (a)		4. Solicita los datos necesarios para efectuar la consulta.	
5. El actor suministra los datos solicitados. (a)		6. Se despliega los datos de los estudiantes.	
<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente. Postcondición: Información oportuna de los datos de los estudiantes de la extensión.			
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar los datos de los estudiantes de forma veraz y oportuna.			

## 9.1.22 Administración de Elementos de Inventario

### 9.1.22.1 Primera Iteración - Fachada

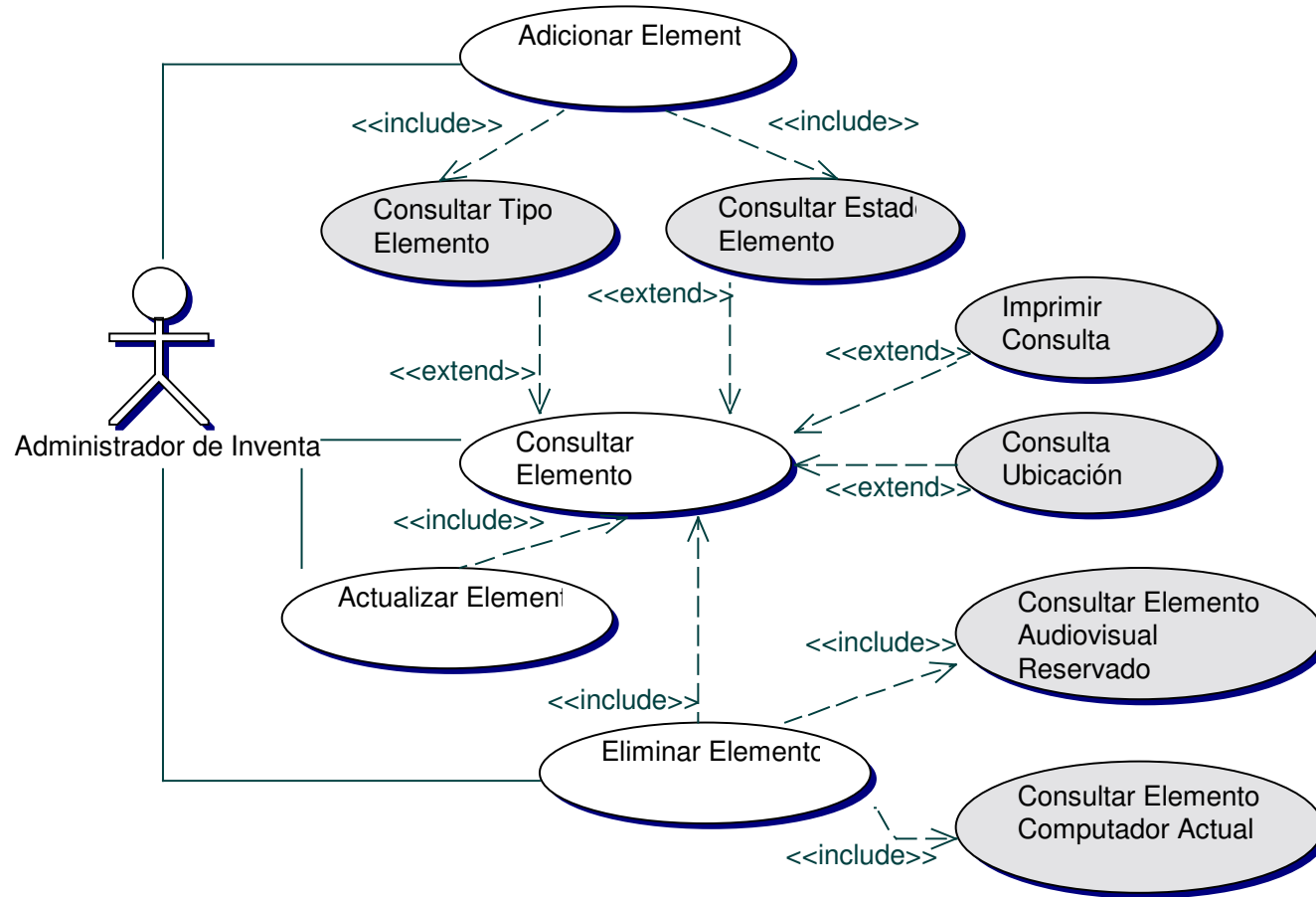
- Diagrama



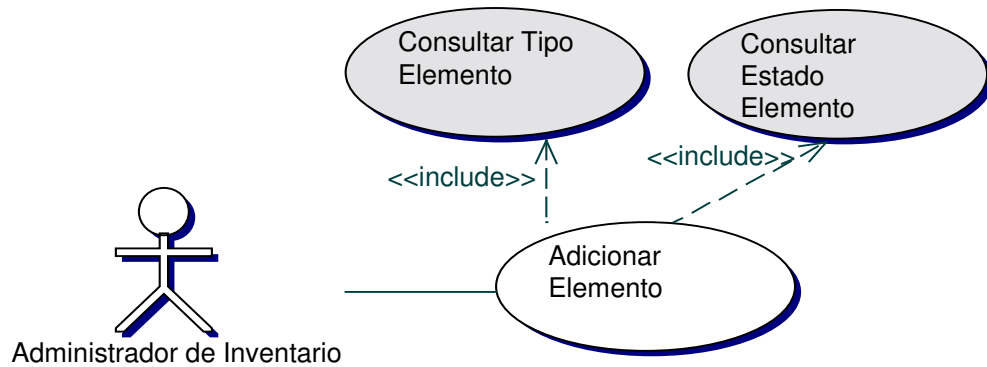
- Iteración

<b>IDENTIFICADOR:</b> <i>I1</i>		<b>NOMBRE:</b> <i>Administración de Inventario</i>	
<b>TIPO:</b> <i>Necesario</i>	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> <i>Adicionar Elemento Consultar Elemento Actualizar Elemento Eliminar Elemento</i>		<b>CRITICO?</b> <i>Si</i>
<b>PRIORIDAD DE DESARROLLO:</b> <i>Fachada</i>	<b>ACTORES:</b> <i>Iniciador y Ejecutor: Administrador de Inventario</i>	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> <i>Páginas HTML y JSP.</i>	
<b>ENTRADA:</b> <i>Datos para efectuar el proceso.</i>		<b>SALIDA:</b> <i>Proceso efectuado o cancelado</i>	
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo administrar los diferentes elementos que lleguen a formar parte del inventario. El proceso de administración se desarrollará teniendo en cuenta las operaciones de adición, consulta, actualización y eliminación.</i>			
<b>OBSERVACIONES:</b> <i>El actor Administrador de Inventario puede acceder al manejo del inventario sin ninguna restricción.</i>			
<b>TRIGGERS:</b> <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i>			
<i>Precondición: Correcta autenticación del actor iniciador.</i>			
<i>Postcondición: Administración eficiente de Inventario.</i>			
<b>CRITERIOS DE ACEPTACIÓN</b> <i>Permite llevar una administración sistemática del Inventario de los elementos existentes. La información en la base de datos se maneja de forma persistente y consistente. Acceso rápido y oportuno a la información actualizada de cada uno de los elementos de inventario.</i>			

9.1.22.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Adicionar Elemento de Inventario



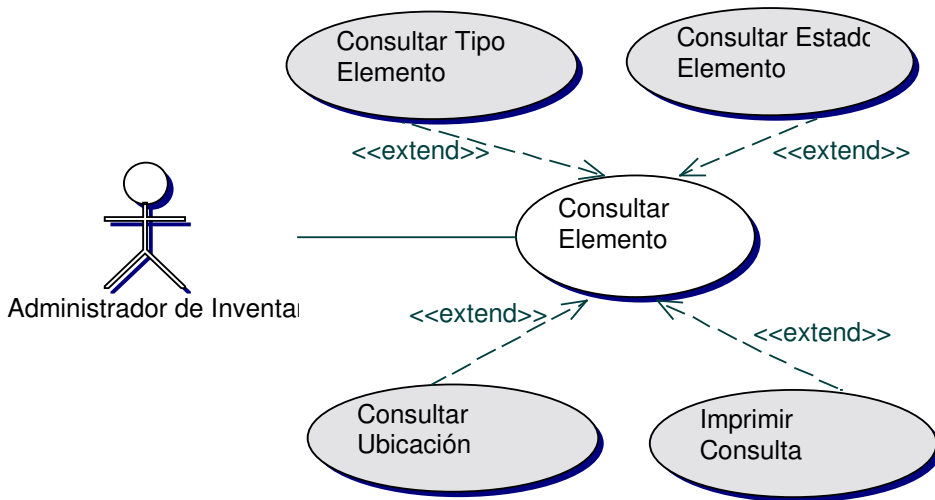
- Iteración Adicionar Elemento de Inventario

<b>IDENTIFICADOR:</b> I1.1		<b>NOMBRE:</b> Adicionar Elementos de Inventario	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Tipo Elemento Consultar Estado Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, descripción, fecha de ingreso, tipo de elemento, marca, modelo, serial, estado, acta de entrega, ubicación, observaciones.		<b>SALIDA:</b> Elemento Registrado o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los diferentes elementos que lleguen a formar parte del inventario. La fecha permitirá identificar cuando ingresa un nuevo elemento al inventario. El tipo de elemento representa un estándar mediante el cual se clasificará los elementos de Inventario. El sistema permitirá seleccionar el tipo de elemento de acuerdo a una lista definida, en caso de que no se presente la clasificación del elemento requerido se podrá establecer un tipo de elemento distinto. El estado permitirá establecer la condición actual de un elemento como es que esté en buen estado, regular, fuera de servicio y dado de baja. La ubicación permitirá determinar el área física en la cual se encuentra asignado un determinado elemento actualmente. Las observaciones permitirán describir los aspectos más relevantes de cada uno de los elementos, como son: capacidad y fallas entre otra.			
<b>OBSERVACIONES:</b> Las observaciones variarán de acuerdo al tipo de elemento y serán descritas según el criterio del administrador. El acta de entrega identifica el número de la resolución por medio de la cual se hace entrega de un determinado elemento. El sistema generará automáticamente un código consecutivo para identificar de manera única cada elemento. De manera predeterminada la ubicación en la que se registrará un elemento cuando ingrese a Inventario será Almacén.			

**CURSO BÁSICO DE EVENTOS**

<p><b>Acción de los actores</b></p> <ol style="list-style-type: none"> <li>1. El actor iniciador selecciona la opción Adicionar Elementos de inventario.</li> <li>3. El actor iniciador ingresa los datos solicitados.</li> <li>4. El actor iniciador selecciona la opción Adicionar elemento. <b>(a)</b></li> </ol>	<p><b>Respuesta del Sistema</b></p> <ol style="list-style-type: none"> <li>2. Solicita la descripción, fecha de ingreso, tipo de elemento, marca, modelo, serial, estado, acta de entrega y las observaciones.</li> <li>5. Adiciona el elemento actualizando la base de datos.</li> <li>6. Muestra un mensaje indicando que el registro se realizó con éxito.</li> </ol>
<p><b>CAMINOS DE EXCEPCION:</b></p> <p><b>(a)</b> El usuario decide no efectuar el proceso: El sistema cancela el proceso</p> <ul style="list-style-type: none"> <li>• El sistema no responde. El proceso no se efectúa.</li> </ul>	
<p>Precondición: Ingreso completo de los datos requeridos.                  Postcondición: Registro eficiente de los elementos de inventario.</p>	
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Almacenamiento de la información en forma ordenada y clasificada.                  Se cumple el requerimiento cuando se logra adicionar un elemento al inventario de forma consistente y persistente.</p>	

• **Diagrama Consultar Elementos de Inventario**

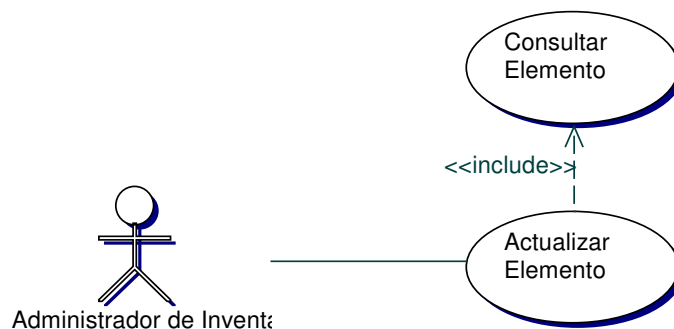


• **Iteración Consultar Elementos de Inventario**

<b>IDENTIFICADOR:</b> I1.2		<b>NOMBRE:</b> Consultar Elementos de Inventario
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Tipo Elemento Consultar Estado Elemento Consultar Ubicación	<b>CRITICO?</b> Si

<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Código, fecha de ingreso, tipo de elemento, estado, acta de entrega, ubicación.		<b>SALIDA:</b> Listado de los elementos consultados
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las existencias de los elementos de inventario de acuerdo a un determinado criterio. La información desplegada incluirá el código, la fecha de ingreso, tipo de elemento, estado, acta de entrega y ubicación		
<b>OBSERVACIONES:</b> Se podrá imprimir la consulta por Acta de Entrega para la presentación de los respectivos informes.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<p style="text-align: center;"><b>Acción de los actores</b></p> <ol style="list-style-type: none"> <li>1. Selecciona la opción Consultar elementos de Inventario</li> <li>3. Selecciona la opción deseada. <b>(a)</b></li> <li>5. El actor iniciador selecciona la opción Consultar Elemento. <b>(a)</b></li> </ol>		<p style="text-align: center;"><b>Respuesta del Sistema</b></p> <ol style="list-style-type: none"> <li>2. Despliega un menú presentando las opciones para su consulta (por código, fecha de ingreso, tipo de elemento, tipo de elemento y estado, acta de entrega, ubicación).</li> <li>4. Solicita la información necesaria de acuerdo al criterio seleccionado.</li> <li>6. Se despliega el resultado de la consulta mostrando la información relacionada al elemento consultado. <b>(b)</b></li> </ol>
<b>CAMINOS DE EXCEPCION:</b>		
<p><b>(a)</b> El actor iniciador decide cancelar la consulta: El sistema cancela el proceso</p> <p><b>(b)</b> La consulta no obtuvo resultado: El sistema muestra un mensaje dando la oportunidad de ingresar nuevamente la información solicitada.</p> <ul style="list-style-type: none"> <li>• El sistema no responde. El proceso no se efectúa.</li> </ul>		
Precondición: La información de la base de datos es consistente.		
Postcondición: Información consolidada de los elementos existentes en inventario.		
<b>CRITERIOS DE ACEPTACIÓN</b>		
Se cumple el requerimiento cuando se logra consultar información de los elementos de inventario de forma veraz y oportuna.		
Los reportes generados permiten a la administración llevar un control del estado y ubicación de los elementos existentes.		
La información generada facilita la confrontación de los datos registrados con las existencias reales. De igual manera permite determinar las necesidades de adquisición de elementos.		

• **Diagrama Actualizar Elemento de Inventario**

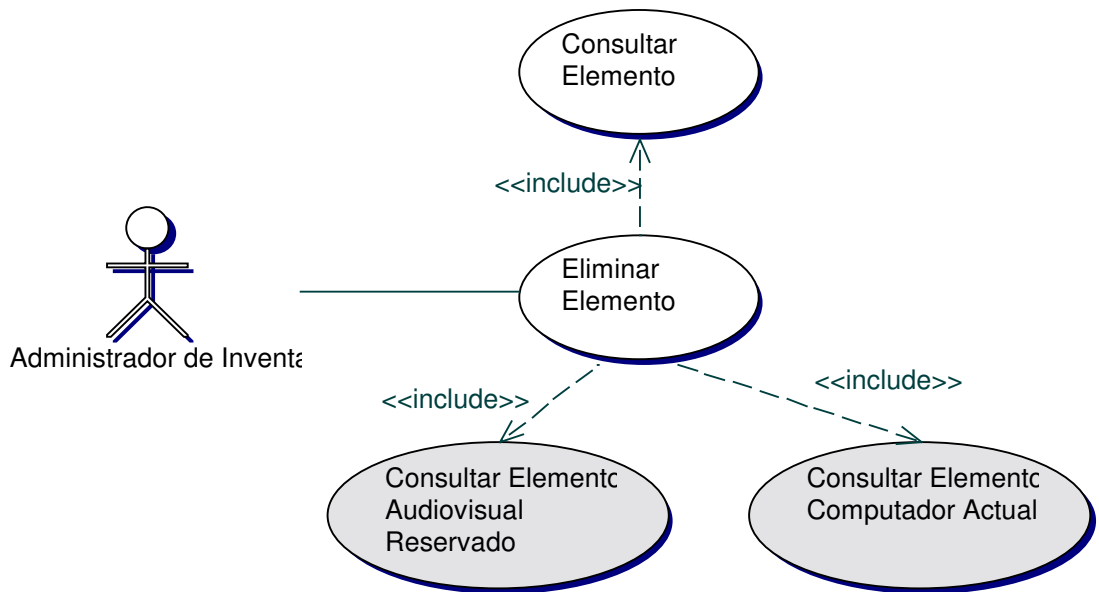




- Iteración Actualizar Elemento de Inventario

<b>IDENTIFICADOR:</b> I1.3		<b>NOMBRE:</b> Actualizar Elementos de Inventario	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, descripción, fecha de ingreso, tipo de elemento, marca, modelo, serial, estado, acta de entrega y observaciones.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de un determinado elemento de inventario. Se desplegará la información correspondiente al elemento permitiendo modificar todos los datos a excepción del código y la ubicación			
<b>OBSERVACIONES:</b> La actualización de la ubicación de un determinado elemento se hará cuando se registre una salida de Almacén.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Actualizar elementos de inventario.	2. Solicita el código del elemento a modificar	3. Suministra la información solicitada. <b>(a)</b>	4. Muestra los datos correspondientes al elemento. <b>(b)</b>
5. Modifica los datos permitidos para dicho elemento.	7. Solicita confirmación.	6. Selecciona la opción actualizar elementos de inventario. <b>(a)</b>	9. Actualiza el elemento modificando la base de datos con la información suministrada
8. Confirma su decisión <b>(a)</b>	10. Muestra un mensaje indicando que los datos se actualizaron con éxito		
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.			
<b>(b)</b> La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> Los procesos de creación se llevan a cabo de forma satisfactoria. La información de la base de datos es consistente.			
<b>Postcondición:</b> Eficiente actualización de los elementos de inventario.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar la información de los elementos de inventario de forma veraz y oportuna.			
El control del estado de los elementos permite determinar el uso eficiente de los mismos.			
La información de la base de datos se almacena de forma persistente y consistente.			

- Diagrama Eliminar Elemento de Inventario



- Iteración Eliminar Elemento de Inventario

<b>IDENTIFICADOR:</b> I1.4		<b>NOMBRE:</b> Eliminar Elementos de Inventario	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Elemento Consultar Elemento Audiovisual Reservado Consultar Elemento Computador Actual		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del elemento.		<b>SALIDA:</b> Eliminación efectuada o cancelada	
<b>DESCRIPCION:</b>  Se satisface este requerimiento permitiendo eliminar un determinado elemento de inventario. El actor Administrador de Inventario como responsable del manejo del inventario podrá eliminar un determinado elemento debido a posibles errores en los datos ingresados siempre y cuando no se encuentre involucrado en otro proceso. Se pide confirmación para efectuar el proceso.			

## **CURSO BÁSICO DE EVENTOS**

### ***Acción de los actores***

1. *Selecciona la opción Eliminar elementos de inventario.*
3. *Suministra la información solicitada. (a)*
5. *Selecciona la opción eliminar elemento. (a)*
8. *Confirma su decisión. (a)*

### ***Respuesta del Sistema***

2. *Solicita el código del elemento a eliminar*
4. *Muestra los datos correspondientes al elemento. (b)*
6. *Se pide confirmación para llevar a cabo el proceso.*
7. *Verifica que el elemento a eliminar no este involucrado en otros procesos. (c)*
9. *Elimina el elemento actualizando la base de datos.*
10. *Muestra un mensaje indicando que el elemento se eliminó con éxito.*

### **CAMINOS DE EXCEPCION:**

- (a) *El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.*
- (b) *La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.*
- (c) *Elemento involucrado en otro proceso: El sistema cancela la eliminación del elemento, informando al usuario del error.*
  - *El sistema no responde. El proceso no se efectúa.*

*Precondición: Los procesos de creación y actualización se llevan a cabo de forma satisfactoria  
La información de la base de datos es consistente.  
El elemento a eliminar no debe estar involucrado en otros procesos.*

*Postcondición: Eliminación satisfactoria de los elementos de inventario.*

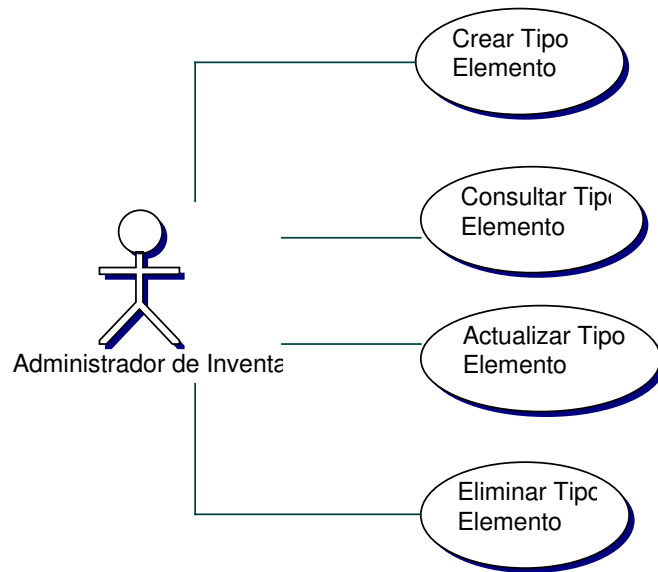
### **CRITERIOS DE ACEPTACIÓN**

*Se cumple el requerimiento cuando se logra eliminar los elementos de Inventario de forma consistente.  
Se garantiza la persistencia y consistencia de la información, ya que los elementos eliminados no afectarán el normal funcionamiento de los otros*

## **9.1.23 Administración Tipos Elementos**

### **9.1.23.1 Primera Iteración - Fachada**

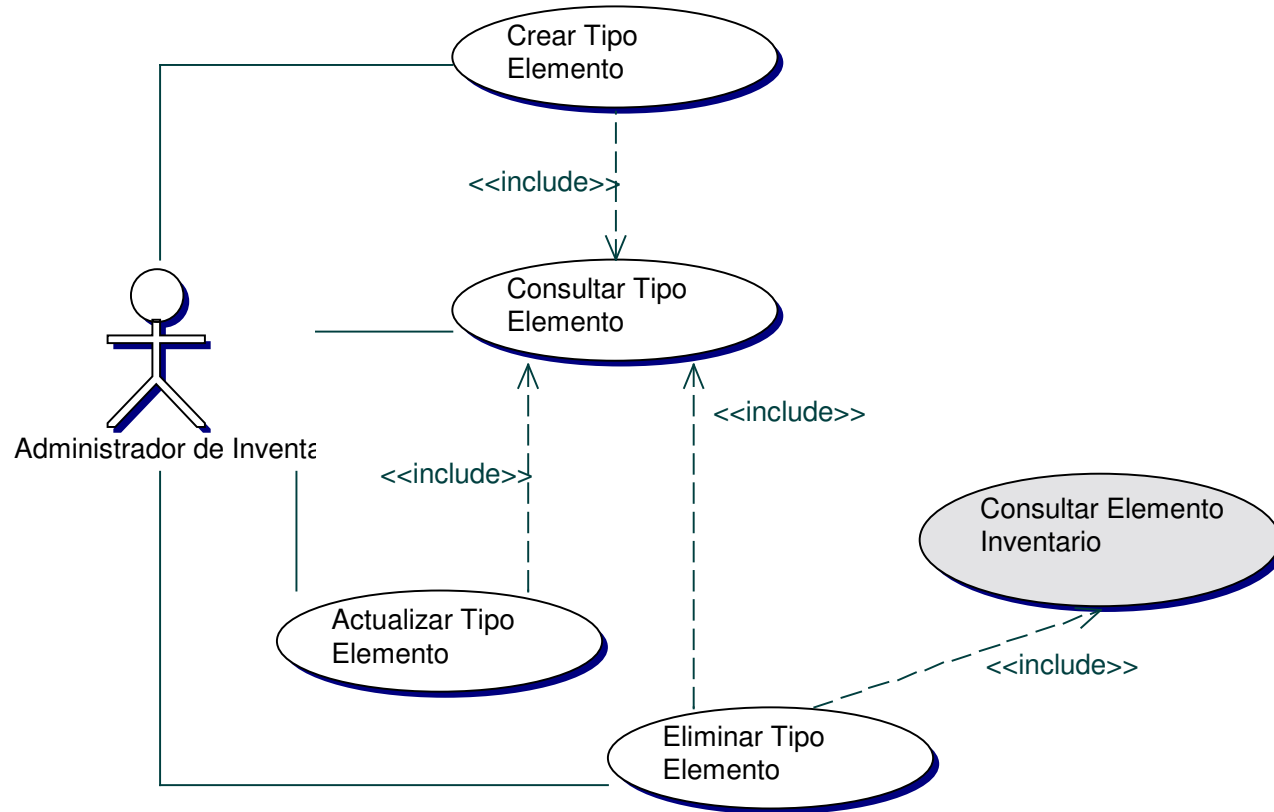
- **Diagrama**



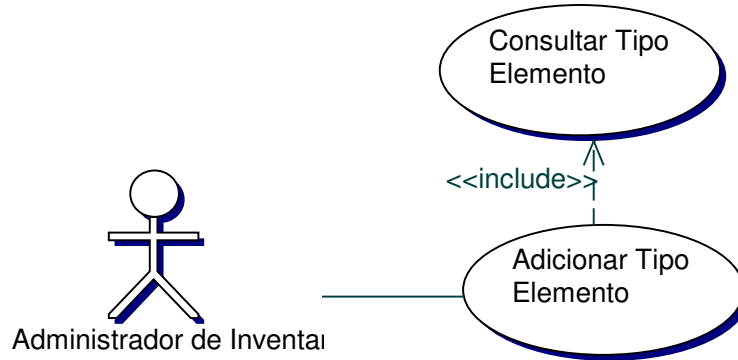
- Iteración

<b>IDENTIFICADOR:</b> ATE1		<b>NOMBRE:</b> Administración de Tipos de Elementos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Tipos de Elementos Consultar Tipos de Elementos Actualizar Tipos de Elementos Eliminar Tipos de Elementos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los diferentes Tipos de Elementos necesarios para asignar a los elementos de inventario. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y eliminación.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: Correcta autenticación del actor iniciador. Postcondición: Administración eficiente de los Tipos de Elementos.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de los Tipos de Elementos existentes. La información en la base de datos se maneja de forma persistente y consistente. Acceso rápido y oportuno a la información actualizada de cada uno de los tipos de elementos.			

### 9.1.23.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Adicionar Tipos de Elementos



- Iteración Adicionar Elemento de Inventario

<b>IDENTIFICADOR:</b> ATE1.1		<b>NOMBRE:</b> Adicionar Tipo Elemento	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Tipo Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, descripción del tipo de elemento.		<b>SALIDA:</b> Tipo de Elemento Registrado o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los diferentes Tipos de Elementos necesarios para adicionar un elemento de Inventario.			
<b>OBSERVACIONES:</b> El sistema generará automáticamente un código consecutivo para identificar de manera única cada tipo de elemento.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	El actor iniciador selecciona la opción Crear Tipo Elemento.	2.	Solicita la descripción.
3.	El actor iniciador ingresa los datos solicitados.	5.	Verifica que la descripción digitada no se repita. <b>(b)</b>
4.	El actor iniciador selecciona la opción Crear Tipo Elemento. <b>(a)</b>	6.	Crea el tipo de elemento actualizando la base de datos.
		7.	Muestra un mensaje indicando que el registro se realizó con éxito.

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no efectuar el proceso: El sistema cancela el proceso</p> <p>(b) La descripción suministrada ya existe: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: Ingreso completo de los datos requeridos.</p> <p>Postcondición: Registro eficiente de los tipos de elementos.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Almacenamiento de la información en forma ordenada y clasificada.</p> <p>Se cumple el requerimiento cuando se logra adicionar un tipo de elemento de forma consistente y persistente.</p>

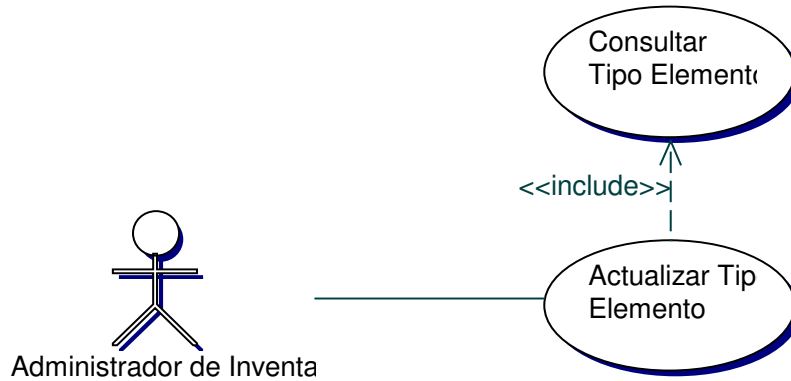
• Diagrama Consultar Tipos de Elementos



• Iteración Consultar Tipos de Elementos

<b>IDENTIFICADOR:</b> II.2		<b>NOMBRE:</b> Consultar Tipos de Elementos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Listado de los tipos de elementos	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los diferentes tipos de elementos existentes. La información desplegada incluirá el código y la descripción del tipo de elemento.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Consultar Tipo Elemento (a)		2. Se despliega el resultado de la consulta mostrando la información relacionada a todos los tipos de elementos existentes	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Información consolidada de los tipos de elementos existentes.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar información de los tipos de elementos de forma veraz y oportuna.			

- Diagrama Actualizar Tipos de Elementos



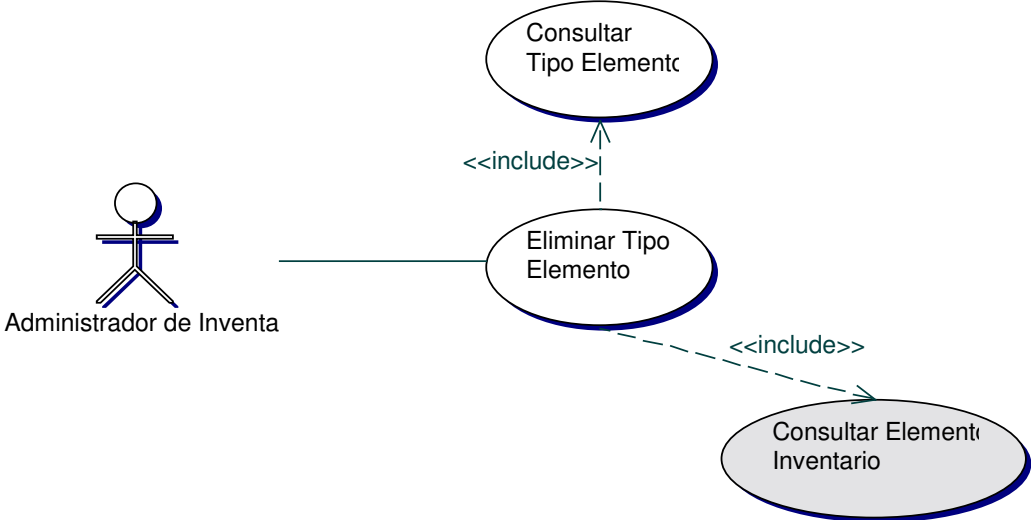
- Iteración Actualizar Elemento de Inventario

<b>IDENTIFICADOR:</b> ATE1.3		<b>NOMBRE:</b> Actualizar Tipo de Elementos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Tipo Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código y descripción.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar la descripción de un determinado tipo de elemento.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción Actualizar elementos de inventario.	2.	Despliega una pantalla mostrando todos los tipos de elementos existentes
3.	Selecciona un tipo de elemento.		
4.	Modifica la descripción de dicho elemento.		
5.	Selecciona la opción actualizar elementos de inventario. <b>(a)</b>	6.	Se pide confirmación para llevar a cabo el proceso.
7.	Confirma su decisión. <b>(a)</b>	8.	Verifica que la descripción digitada no se repita. <b>(b)</b>
		9.	Actualiza el elemento modificando la base de datos con la información suministrada.
		10.	Muestra un mensaje indicando que los datos se actualizaron con éxito



<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.</p> <p>(b) La descripción suministrada ya existe: El sistema muestra un mensaje de error y cancela el proceso.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: Los procesos de creación se llevan a cabo de forma satisfactoria. La información de la base de datos es consistente.</p> <p>Postcondición: Eficiente actualización de los tipos de elementos existentes.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra actualizar la información de los tipos de elementos de forma veraz y oportuna..</p> <p>La información de la base de datos se almacena de forma persistente y consistente.</p>

• Diagrama Eliminar Tipos de Elementos



• Iteración Eliminar Tipos de Elementos

<b>IDENTIFICADOR:</b> ATE1.4		<b>NOMBRE:</b> Eliminar Tipos de Elementos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Tipo Elemento Consultar Elemento de Inventario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del elemento.		<b>SALIDA:</b> Eliminación efectuada o cancelada	
<b>DESCRIPCION:</b> Se satisface este requerimiento permitiendo eliminar un determinado tipo de elemento. Se pide confirmación para efectuar el proceso.			

**CURSO BÁSICO DE EVENTOS**

**Acción de los actores**

1. Selecciona la opción Eliminar tipo elementos.
3. Selecciona la descripción del elemento que desea eliminar.
4. Selecciona la opción eliminar tipo elemento.  
**(a)**
6. Confirma su decisión. **(a)**

**Respuesta del Sistema**

2. Muestra un listado de todos los tipos de elementos existentes.
5. Se pide confirmación para llevar a cabo el proceso.
7. Verifica que el tipo de elemento a eliminar no esté asignado a un elemento de inventario.  
**(b)**
8. Elimina el tipo elemento actualizando la base de datos.
9. Muestra un mensaje indicando que el tipo elemento se eliminó con éxito.

**CAMINOS DE EXCEPCION:**

- (a)** El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.
- (b)** Tipo Elemento involucrado en otro proceso: El sistema cancela la eliminación del tipo elemento, informando al usuario del error.
  - El sistema no responde. El proceso no se efectúa.

Precondición: Los procesos de creación y actualización se llevan a cabo de forma satisfactoria  
La información de la base de datos es consistente.  
El tipo elemento a eliminar no debe estar involucrado en otros procesos.

Postcondición: Eliminación satisfactoria de los elementos de inventario.

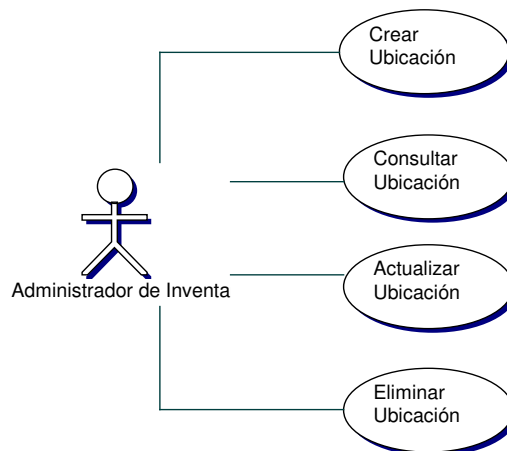
**CRITERIOS DE ACEPTACIÓN**

Se cumple el requerimiento cuando se logra eliminar los tipos elementos existentes de forma consistente.  
Se garantiza la persistencia y consistencia de la información, ya que los tipos de elementos eliminados no afectarán el normal funcionamiento del sistema.

## 9.1.24 Administración Ubicaciones

### 9.1.24.1 Primera Iteración - Fachada

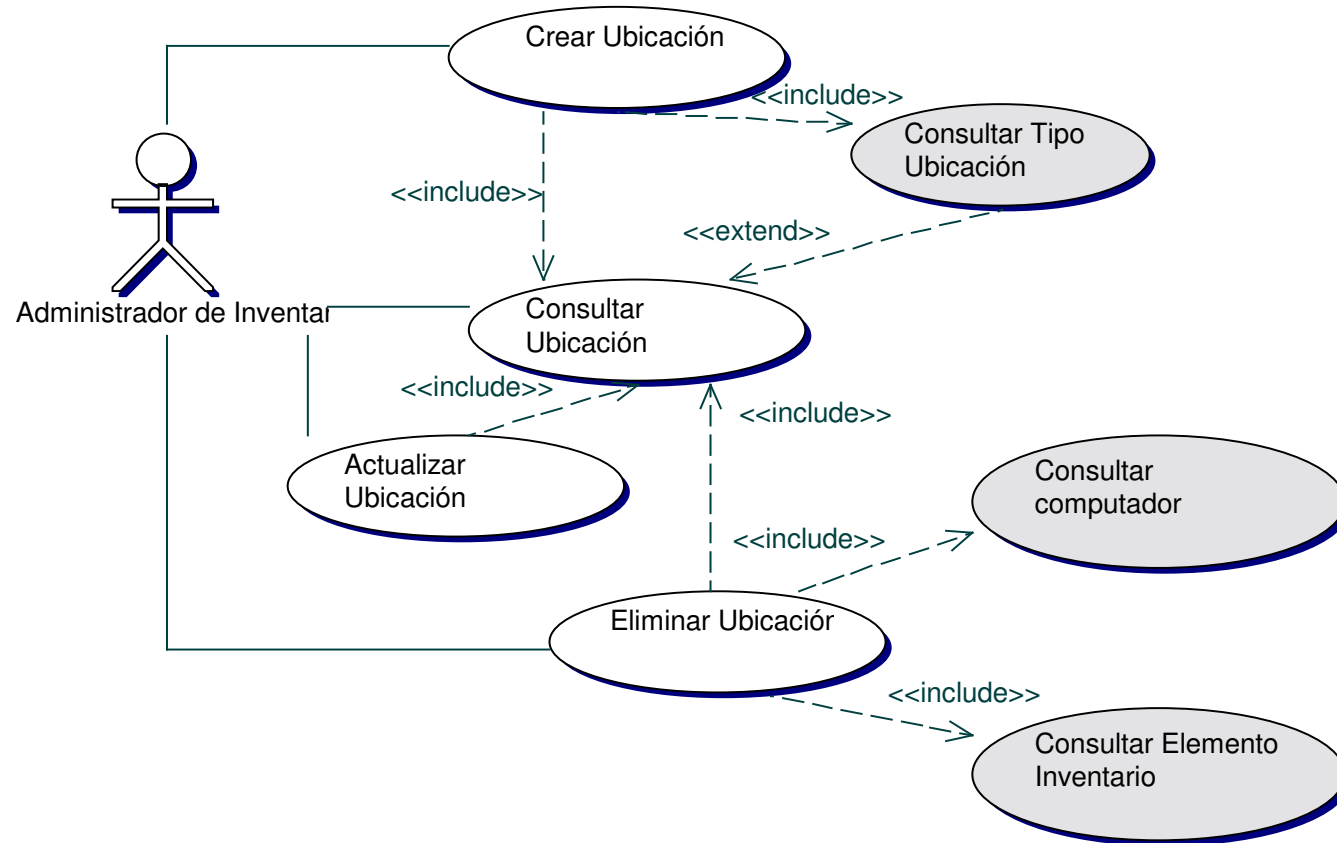
- Diagrama



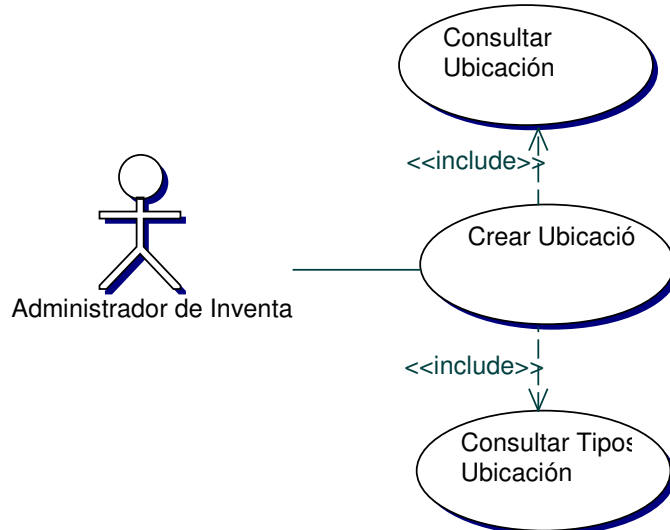
- Iteración

<b>IDENTIFICADOR:</b> AU1		<b>NOMBRE:</b> Administración de Ubicaciones	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Ubicación Consultar Tipo Ubicación Consultar Ubicación Actualizar Ubicación Eliminar Ubicación		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las diferentes ubicaciones o áreas físicas con las que se cuenta. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y eliminación.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: Correcta autenticación del actor iniciador. Postcondición: Administración eficiente de las ubicaciones.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de las ubicaciones existentes. La información en la base de datos se maneja de forma persistente y consistente. Acceso rápido y oportuno a la información actualizada de cada una de los tipos ubicaciones.			

### 9.1.24.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Crear Ubicación

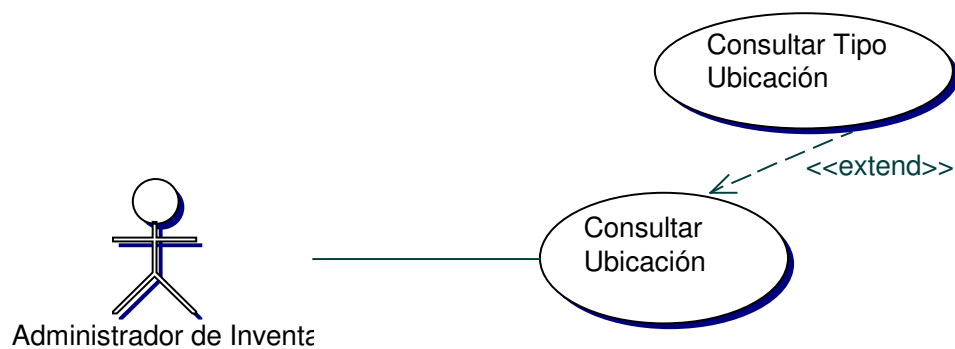


- Iteración Crear Ubicación

<b>IDENTIFICADOR:</b> AU.1		<b>NOMBRE:</b> Crear Ubicación	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Tipo Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, nombre, descripción y tipo de ubicación		<b>SALIDA:</b> Ubicación Registrada o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar las ubicaciones o áreas físicas existentes.			
<b>OBSERVACIONES:</b> El sistema generará automáticamente un código consecutivo para identificar de manera única cada ubicación.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	El actor iniciador selecciona la opción Crear Ubicación.	2.	Despliega una pantalla solicitando el nombre y la descripción, permite seleccionar el tipo de ubicación que se desea crear.
3.	Ingresa los datos solicitados.	5.	Verifica que el nombre digitado no se repita. (b)
4.	Selecciona la opción Crear Ubicación. (a)	6.	Crea la ubicación actualizando la base de datos.
		7.	Muestra un mensaje indicando que la creación se realizó con éxito.

<p><b>CAMINOS DE EXCEPCION:</b>  <b>(a)</b> El usuario decide no efectuar el proceso: El sistema cancela el proceso</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: Ingreso completo de los datos requeridos.  Postcondición: Registro eficiente de las ubicaciones.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b>  Almacenamiento de la información en forma ordenada y clasificada.  Se cumple el requerimiento cuando se logra adicionar una ubicación de forma consistente y persistente.</p>

• **Diagrama Consultar Ubicaciones**

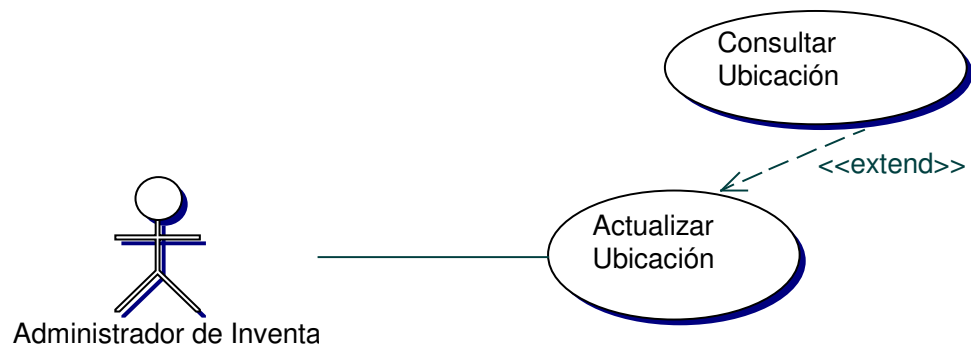


• **Iteración Consultar Ubicaciones**

<b>IDENTIFICADOR:</b> AU1.2		<b>NOMBRE:</b> Consultar Ubicaciones	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Tipo Ubicación		<b>CRITICO?</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Nombre ubicación, tipo ubicación		<b>SALIDA:</b> Listado de las ubicaciones existentes	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar las diferentes ubicaciones. La información desplegada incluirá el nombre y la descripción de las ubicaciones.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción Consultar ubicación	2.	Solicita el criterio para la consulta
3.	Selecciona el criterio de consulta deseado.		
4.	Selecciona la opción consultar ubicación. <b>(a)</b>	5.	Despliega una pantalla mostrando un listado de las ubicaciones existentes.

<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso • El sistema no responde. El proceso no se efectúa.
Precondición: La información de la base de datos es consistente. Postcondición: Información consolidada de las ubicaciones existentes.
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar información de las ubicaciones de forma veraz y oportuna.

• **Diagrama Actualizar Ubicaciones**



• **Iteración Actualizar Ubicaciones**

<b>IDENTIFICADOR:</b> AU1.3		<b>NOMBRE:</b> Actualizar Ubicación	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Ubicación		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, nombre, descripción y tipo ubicación.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar el nombre, la descripción y el tipo de ubicación.			

**CURSO BÁSICO DE EVENTOS**

<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>
1. <i>Selecciona la opción Actualizar ubicación.</i>	2. <i>Muestra un listado con los nombres de las ubicaciones existentes.</i>
3. <i>Selecciona el nombre de la ubicación a actualizar.</i>	4. <i>Muestra los datos actuales de la ubicación seleccionada, permitiendo su correspondiente modificación.</i>
5. <i>Modifica los datos permitidos.</i>	7. <i>Se pide confirmación para llevar a cabo el proceso.</i>
6. <i>Selecciona la opción actualizar ubicación. (a)</i>	9. <i>Verifica que el nombre no se repita. (b)</i>
8. <i>Confirma su decisión. (a)</i>	10. <i>Actualiza la ubicación modificando la base de datos con la información suministrada.</i>
	11. <i>Muestra un mensaje indicando que los datos se actualizaron con éxito</i>

**CAMINOS DE EXCEPCION:**

- (a) *El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.*
- (b) *El dato suministrado ya existe: El sistema muestra un mensaje de error y cancela el proceso.*
- *El sistema no responde. El proceso no se efectúa.*

*Precondición: Los procesos de creación se llevan a cabo de forma satisfactoria.  
La información de la base de datos es consistente.*

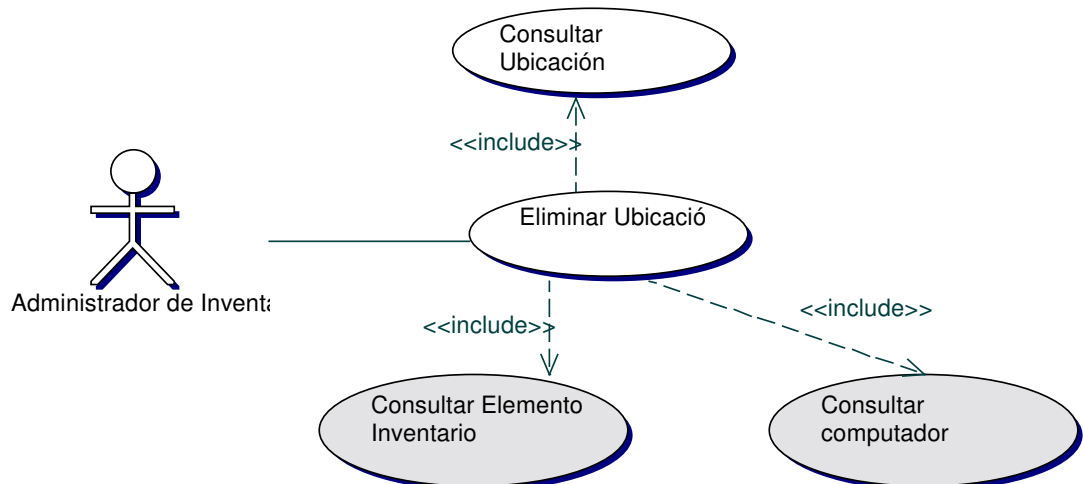
*Postcondición: Eficiente actualización de las ubicaciones existentes.*

**CRITERIOS DE ACEPTACIÓN**

*Se cumple el requerimiento cuando se logra actualizar la información de las ubicaciones existentes de forma veraz y oportuna..*

*La información de la base de datos se almacena de forma persistente y consistente.*

• **Diagrama Eliminar Ubicaciones**





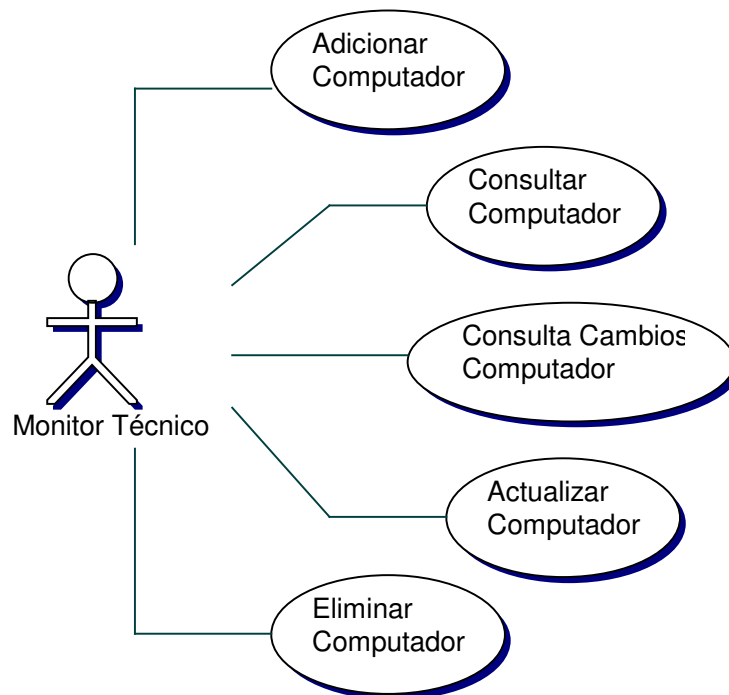
- Iteración Eliminar Ubicaciones

<b>IDENTIFICADOR:</b> AU1.4		<b>NOMBRE:</b> Eliminar Ubicación	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Ubicación Consultar Computador Consulta Elemento Inventario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, nombre de la ubicación.		<b>SALIDA:</b> Eliminación efectuada o cancelada	
<b>DESCRIPCION:</b> Se satisface este requerimiento permitiendo eliminar una determinada ubicación. Se pide confirmación para efectuar el proceso.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<p style="text-align: center;"><b>Acción de los actores</b></p> <ol style="list-style-type: none"> <li>1. Selecciona la opción Eliminar ubicación.</li> <li>3. Selecciona el nombre de la ubicación que desea eliminar.</li> <li>4. Selecciona la opción eliminar ubicación. <b>(a)</b></li> <li>6. Confirma su decisión. <b>(a)</b></li> </ol>		<p style="text-align: center;"><b>Respuesta del Sistema</b></p> <ol style="list-style-type: none"> <li>2. Muestra un listado de todas las ubicaciones de existentes.</li> <li>5. Se pide confirmación para llevar a cabo el proceso.</li> <li>7. Verifica que la ubicación a eliminar no esté involucrada en otro proceso. <b>(b)</b></li> <li>8. Elimina la ubicación actualizando la base de datos.</li> <li>9. Muestra un mensaje indicando que el tipo elemento se eliminó con éxito.</li> </ol>	
<b>CAMINOS DE EXCEPCION:</b>			
<p><b>(a)</b> El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.</p> <p><b>(b)</b> Ubicación involucrada en otro proceso: El sistema cancela la eliminación de la ubicación, informando al usuario del error.</p> <ul style="list-style-type: none"> <li>• El sistema no responde. El proceso no se efectúa.</li> </ul>			
<p><b>Precondición:</b> Los procesos de creación y actualización se llevan a cabo de forma satisfactoria La información de la base de datos es consistente. La ubicación a eliminar no debe estar involucrado en otros procesos.</p> <p><b>Postcondición:</b> Eliminación satisfactoria de las ubicaciones.</p>			
<b>CRITERIOS DE ACEPTACIÓN</b>			
<p>Se cumple el requerimiento cuando se logra eliminar las ubicaciones existentes de forma consistente. Se garantiza la persistencia y consistencia de la información, ya que las ubicaciones eliminados no afectarán el normal funcionamiento del sistema.</p>			

## 9.1.25 Administración De Computadores

### 9.1.25.1 Primera Iteración - Fachada

- Diagrama

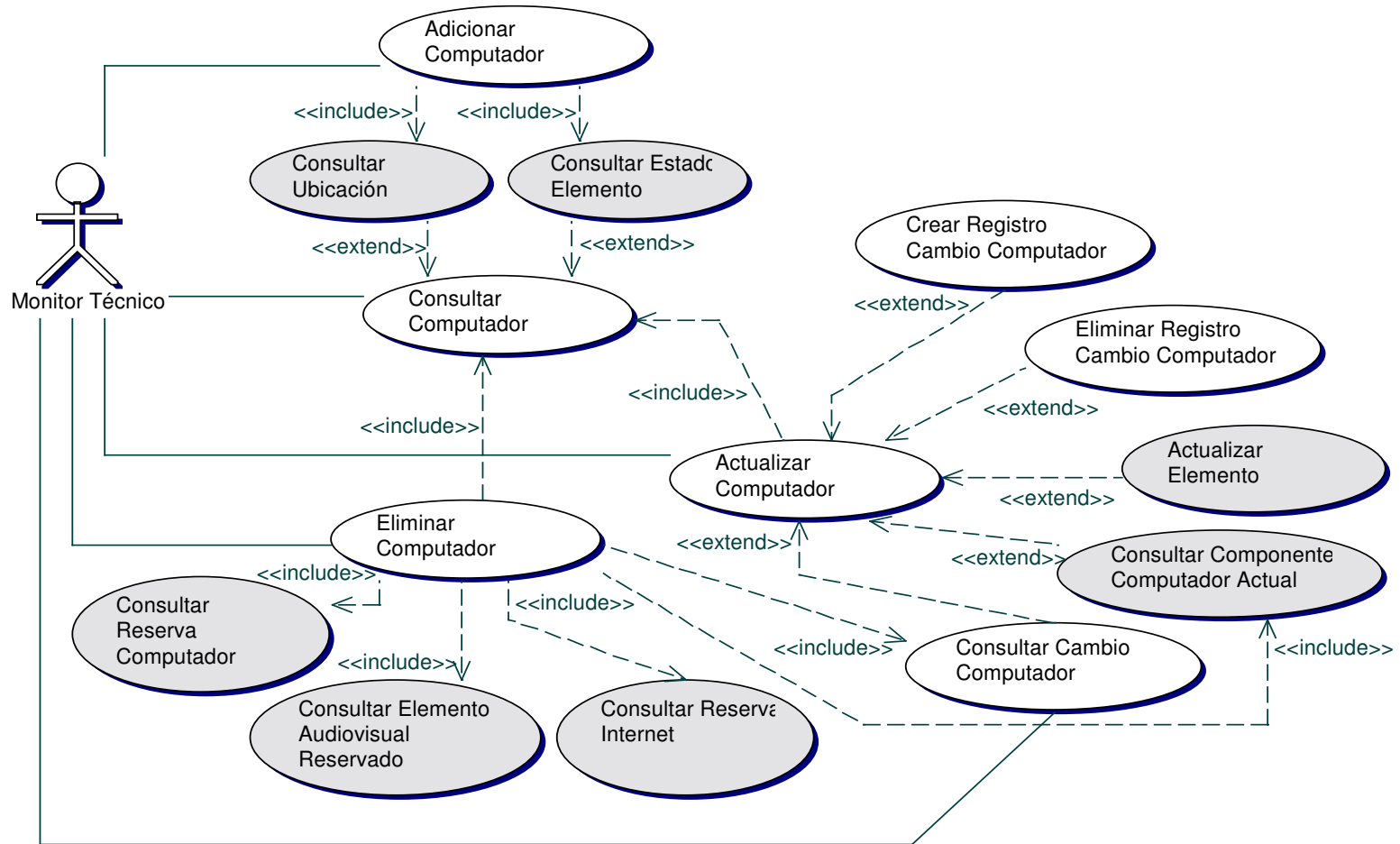


- Iteración

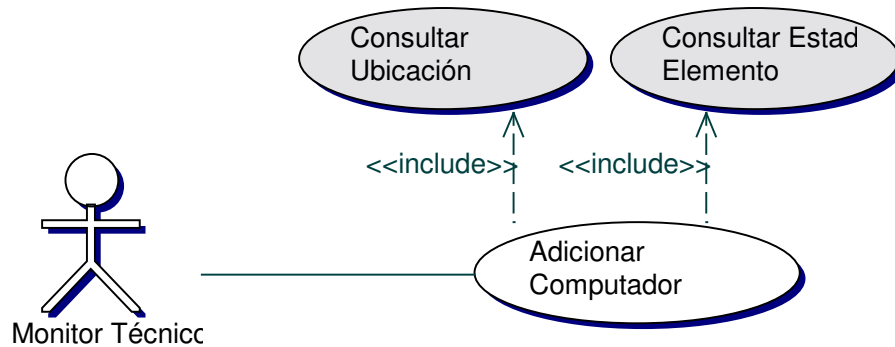
<b>IDENTIFICADOR:</b> AC1		<b>NOMBRE:</b> Administración de Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Adicionar Computador Consultar Computador Actualizar Computador Eliminar Computador Consultar Estado Elemento Consultar Ubicación Crear Registro Cambio Computador Consultar Componente Computador Actual Actualizar Elemento Consultar Cambios Computador Eliminar Cambios por Computador Consultar Reservas de Computadores Consultar Elemento Audiovisual Reservado Consultar Reserva de Internet		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los computadores. De igual manera se puede llevar un			

<i>historial de los cambios de estado o ubicación que se efectúen a un determinado computador.. El proceso de administración se desarrollará teniendo en cuenta las operaciones de adición, consulta, actualización y eliminación.</i>
<b>OBSERVACIONES:</b> <i>El actor Monitor Técnico del Aula de Informática puede acceder a la administración de computadores sin ninguna restricción.</i>
<b>TRIGGERS:</b> <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i>
<i>Precondición: Correcta autenticación del actor iniciador.</i> <i>Postcondición: Administración eficiente de Computadores.</i>
<b>CRITERIOS DE ACEPTACIÓN</b> <i>Permite llevar una administración sistemática de los computadores.</i> <i>La información en la base de datos se maneja de forma persistente y consistente.</i> <i>Acceso rápido y oportuno a la información actualizada de cada uno los computadores.</i>

### 9.1.25.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Adicionar Computador

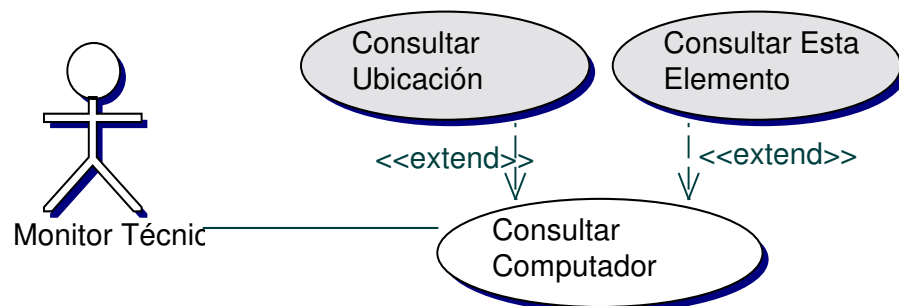


- Iteración Adicionar Computador

<b>IDENTIFICADOR:</b> ACI.1		<b>NOMBRE:</b> Adicionar Elementos de Inventario	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Ubicación Consultar Estado Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, fecha de ingreso, acta de entrega, estado, ubicación, función y observaciones.		<b>SALIDA:</b> Computador Registrado o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los computadores que lleguen a formar parte de las existencias. La fecha permitirá identificar cuando ingresa un nuevo computador. El estado permitirá establecer la condición actual de un computador como es que esté en buen estado, regular, fuera de servicio y dado de baja. La ubicación permitirá determinar el área física en la cual se encuentra asignado un computador actualmente. Las observaciones permitirán describir las principales características de un computador.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción Adicionar computador.	2.	Solicita el código, fecha de ingreso, acta de entrega, estado, ubicación, función y observaciones.
3.	Ingresa los datos solicitados.	5.	Verifica que el código digitado no se repita.
4.	Selecciona la opción Adicionar computador. <b>(a)</b>	6.	Adiciona el computador actualizando la base de datos.
		7.	Muestra un mensaje indicando que el registro se realizó con éxito.

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no efectuar el proceso: El sistema cancela el proceso</p> <p>(b) El dato suministrado ya existe: El sistema muestra un mensaje de error y cancela el proceso.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: Ingreso completo de los datos requeridos.</p> <p>Postcondición: Registro eficiente de los diferentes computadores.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Almacenamiento de la información en forma ordenada y clasificada.</p> <p>Se cumple el requerimiento cuando se logra adicionar un computador de forma consistente y persistente.</p>

• **Diagrama Consultar Computadores**



• **Iteración Consultar Computadores**

<b>IDENTIFICADOR:</b> AC1.2		<b>NOMBRE:</b> Consultar Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Estado Elemento Consultar Ubicación		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, fecha de ingreso, acta de entrega, estado, ubicación y función.		<b>SALIDA:</b> Listado de los computadores consultados	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los computadores existentes de acuerdo a un determinado criterio. La información desplegada incluirá el código, la fecha de ingreso, acta de entrega, estado, ubicación, función y observaciones.			
<b>OBSERVACIONES:</b> El estado hace referencia al estado general del computador, los componentes a el asociados manejan su propio estado. Se podrá imprimir la consulta por Código computador, esta incluirá además de los datos generales los datos de los componentes que actualmente tenga asignado.			

**CURSO BÁSICO DE EVENTOS**

<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción Consultar computador	2. Despliega un menú presentando las opciones para su consulta (por código, fecha de ingreso, acta de entrega, estado, ubicación y función).
3. Selecciona la opción deseada. <b>(a)</b>	4. Solicita la información necesaria de acuerdo al criterio seleccionado.
5. Ingresa la información solicitada	7. Se despliega el resultado de la consulta mostrando la información relacionada al computador consultado. <b>(b)</b>
6. Selecciona la opción Consultar computador <b>(a)</b>	

**CAMINOS DE EXCEPCION:**

- (a)** El actor iniciador decide cancelar la consulta: El sistema cancela el proceso
- (b)** La consulta no obtuvo resultado: El sistema muestra un mensaje dando la oportunidad de ingresar nuevamente la información solicitada.
- El sistema no responde. El proceso no se efectúa.

Precondición: La información de la base de datos es consistente.

Postcondición: Información consolidada de los computadores existentes.

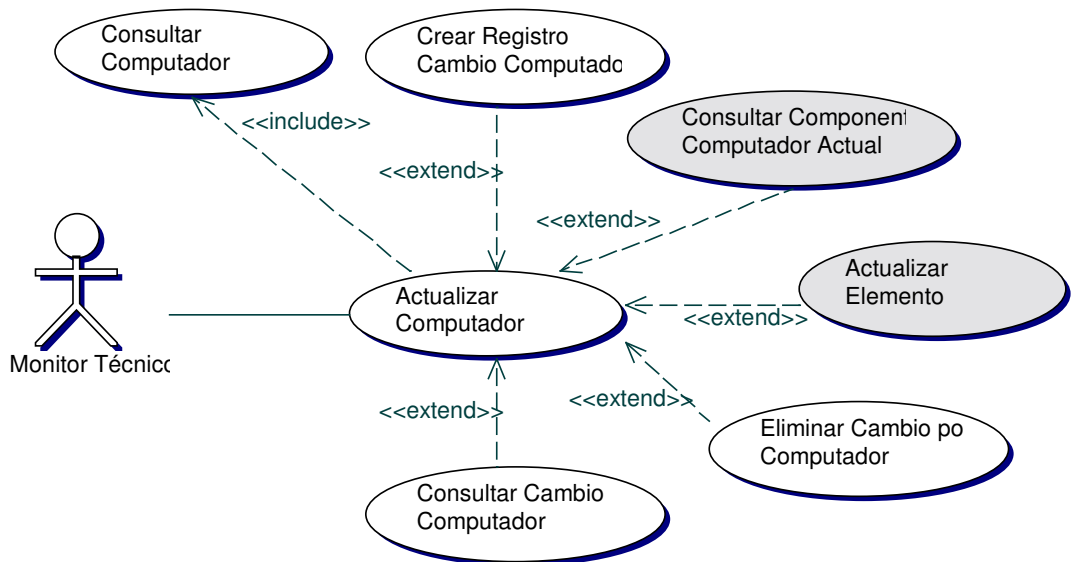
**CRITERIOS DE ACEPTACIÓN**

Se cumple el requerimiento cuando se logra consultar información de los computadores existentes de forma veraz y oportuna.

Los reportes generados permiten a la administración llevar un control del estado y ubicación de los computadores existentes.

La información generada facilita la confrontación de los datos registrados con las existencias reales. De igual manera permite determinar las necesidades de adquisición de computadores.

• **Diagrama Actualizar Computadores**



- Iteración Actualizar Computadores

<b>IDENTIFICADOR:</b> AC1.3		<b>NOMBRE:</b> Actualizar Computadores	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Registro Cambio Componente Consultar Componente Computador Actual Actualizar Elemento Consultar Cambios Computador Eliminar Registro Cambio Computador Consultar Computador		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha de ingreso, acta de entrega, estado, ubicación, función y observaciones.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de un determinado computador. Se desplegará la información correspondiente al computador permitiendo modificar todos los datos a excepción del código			
<b>OBSERVACIONES:</b> La actualización de la ubicación de un determinado computador hará que se actualice automáticamente la ubicación de cada uno de sus componentes.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Actualizar computador.	2. Solicita el código del computador a modificar.	3. Suministra la información solicitada. (a)	4. Muestra los datos correspondientes al computador. (b)
5. Modifica los datos permitidos para dicho computador.	7. Solicita confirmación.	6. Selecciona la opción actualizar computador. (a)	9. Verifica si se actualizó la ubicación o el estado. (c)
8. Confirma su decisión (a)	10. Crea un registro del cambio efectuado al computador.		11. Actualiza el computador modificando la base de datos con la información suministrada.
	11. Actualiza el computador modificando la base de datos con la información suministrada.		12. Verifica si se actualizó la ubicación. (d)
	12. Verifica si se actualizó la ubicación. (d)		13. Actualiza la ubicación de cada uno de los componentes con la nueva información seleccionada.
	13. Actualiza la ubicación de cada uno de los componentes con la nueva información seleccionada.		14. Muestra un mensaje indicando que los datos se actualizaron con éxito
	14. Muestra un mensaje indicando que los datos se actualizaron con éxito		



**CAMINOS DE EXCEPCION:**

- (a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.
  - (b) La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.
  - (c) Se actualizan los datos a excepción del estado y la ubicación: El sistema actualiza el computador modificando la base de datos con la información suministrada y muestra un mensaje indicando que los datos se actualizaron con éxito.
  - (d) Se actualiza todo a excepción de la ubicación: El sistema termina el proceso y muestra un mensaje indicando que los datos se actualizaron con éxito.
- El sistema no responde. El proceso no se efectúa.

**TRIGGERS:**

Crear un registro del cambio efectuado al computador: Este evento se ejecutará para llevar un registro de los cambios que se puedan realizar a un computador, identificar a la persona responsable, el estado inicial, el estado actual, la ubicación inicial, la ubicación actual y algunas observaciones pertinentes al cambio efectuado.

Actualizar la ubicación de los componentes asignados al computador: Si el usuario actualiza la ubicación del computador se actualizará la ubicación de cada uno de los componentes asignados al computador en cuestión.

Eliminar el cambio efectuado: si la actualización del computador no se puede llevar a cabo, se eliminará automáticamente el registro del cambio efectuado.

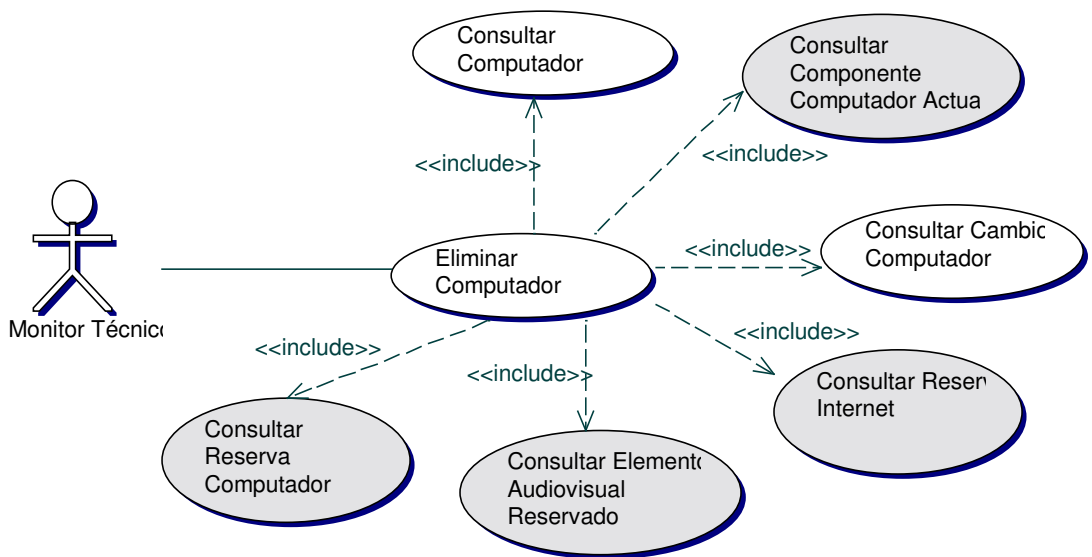
Precondición: Los procesos de creación se llevan a cabo de forma satisfactoria.  
La información de la base de datos es consistente.

Postcondición: Eficiente actualización de los computadores.

**CRITERIOS DE ACEPTACIÓN**

Se cumple el requerimiento cuando se logra actualizar la información de los computadores de forma veraz y oportuna.  
Permite llevar un historial de los cambios efectuado a un determinado computador.  
La información de la base de datos se almacena de forma persistente y consistente.

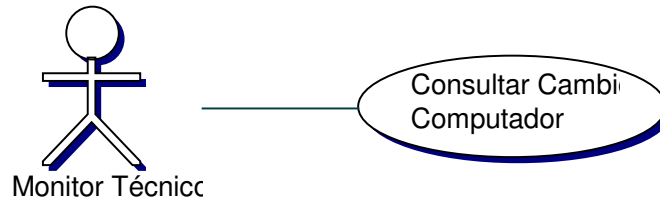
• **Diagrama Eliminar Computadores**



- Iteración Eliminar Computadores

<b>IDENTIFICADOR:</b> AC1.4		<b>NOMBRE:</b> Eliminar Computadores.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Computador Consultar Reserva Computador Consultar Elemento Audiovisual Reservado Consultar Reserva Internet Consulta Cambio Computador Consultar Componente Computador Actual		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del computador.		<b>SALIDA:</b> Eliminación efectuada o cancelada	
<b>DESCRIPCION:</b> Se satisface este requerimiento permitiendo eliminar un determinado computador. El actor Monitor Técnico del Aula de Informática como responsable de la administración de computadores podrá eliminar un determinado computador debido a posibles errores en los datos ingresados siempre y cuando no se encuentre involucrado en otro proceso. Se pide confirmación para efectuar el proceso.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Eliminar computador.		2. Solicita el código del computador.	
3. Suministra la información solicitada.		4. Muestra los datos correspondientes al computador. <b>(b)</b>	
5. Selecciona la opción eliminar computador. <b>(a)</b>		6. Se pide confirmación para llevar a cabo el proceso.	
7. Confirma su decisión. <b>(a)</b>		8. Verifica que el computador a eliminar no este involucrado en otros procesos. <b>(c)</b>	
		9. Elimina el elemento actualizando la base de datos.	
		10. Muestra un mensaje indicando que el computador se eliminó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.			
<b>(b)</b> La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.			
<b>(c)</b> Elemento involucrado en otro proceso: El sistema cancela la eliminación del elemento, informando al usuario del error.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> Los procesos de creación y actualización se llevan a cabo de forma satisfactoria La información de la base de datos es consistente. El computador a eliminar no debe estar involucrado en otros procesos.			
<b>Postcondición:</b> Eliminación satisfactoria de los computadores.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar los computadores de forma consistente. Se garantiza la persistencia y consistencia de la información, ya que los computadores eliminados no afectarán el normal funcionamiento de los otros			

- Diagrama Consultar Cambios Computador



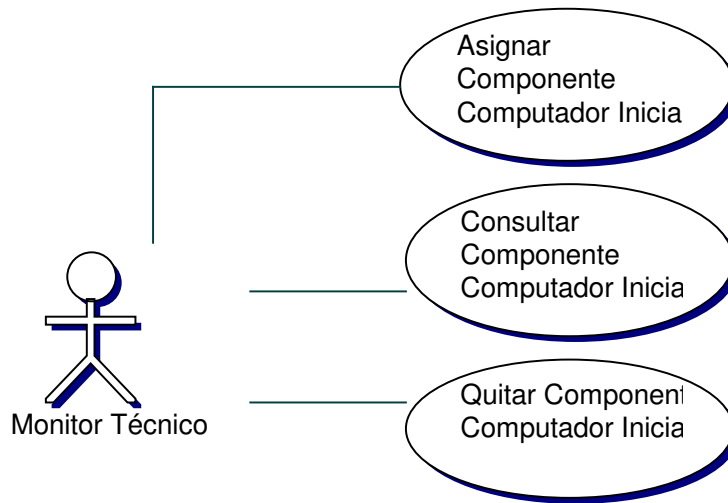
- Iteración Consultar Cambios Computador

<b>IDENTIFICADOR:</b> AC1.5		<b>NOMBRE:</b> Consultar Cambios Computador.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha de ingreso, código del computador, código del elemento		<b>SALIDA:</b> Listado de los cambios efectuados	
<b>DESCRIPCION:</b> Se satisface este requerimiento permitiendo consultar los cambios efectuados a determinado un computador, en una fecha dada o de un determinado elemento de inventario.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<p style="text-align: center;"><b>Acción de los actores</b></p> <ol style="list-style-type: none"> <li>1. Selecciona la opción Consultar cambio computador</li> <li>3. Selecciona la opción deseada. <b>(a)</b></li> <li>5. Ingresa la información solicitada</li> <li>6. Selecciona la opción Mostrar <b>(a)</b></li> </ol>		<p style="text-align: center;"><b>Respuesta del Sistema</b></p> <ol style="list-style-type: none"> <li>2. Despliega un menú presentando las opciones para su consulta (por código de computador, fecha de ingreso, código elemento inventario).</li> <li>4. Solicita la información necesaria de acuerdo al criterio seleccionado.</li> <li>7. Se despliega el resultado de la consulta mostrando la información. <b>(b)</b></li> </ol>	
<b>CAMINOS DE EXCEPCION:</b>			
<p><b>(a)</b> El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.</p> <p><b>(b)</b> La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.</p> <ul style="list-style-type: none"> <li>• El sistema no responde. El proceso no se efectúa.</li> </ul>			
<p><b>Precondición:</b> El proceso de creación y actualización se llevan a cabo de forma satisfactoria La información de la base de datos es consistente.</p> <p><b>Postcondición:</b> Información consolidada de la consulta efectuada.</p>			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los cambios efectuados a un determinado computador de forma consistente.			

## 9.1.26 Administración de Componentes Computador Inicial

### 9.1.26.1 Primera Iteración - Fachada

- Diagrama

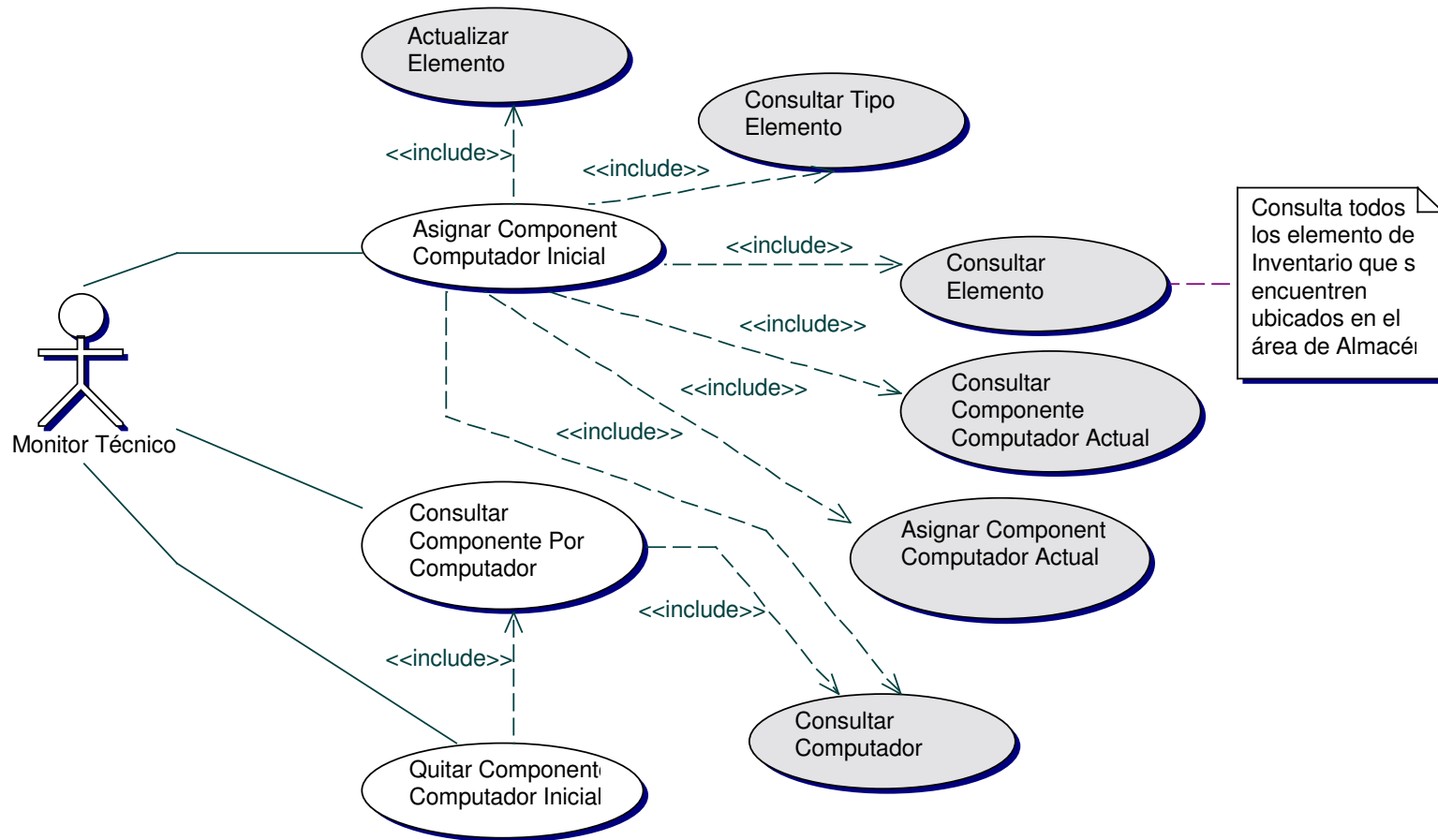


- Iteración

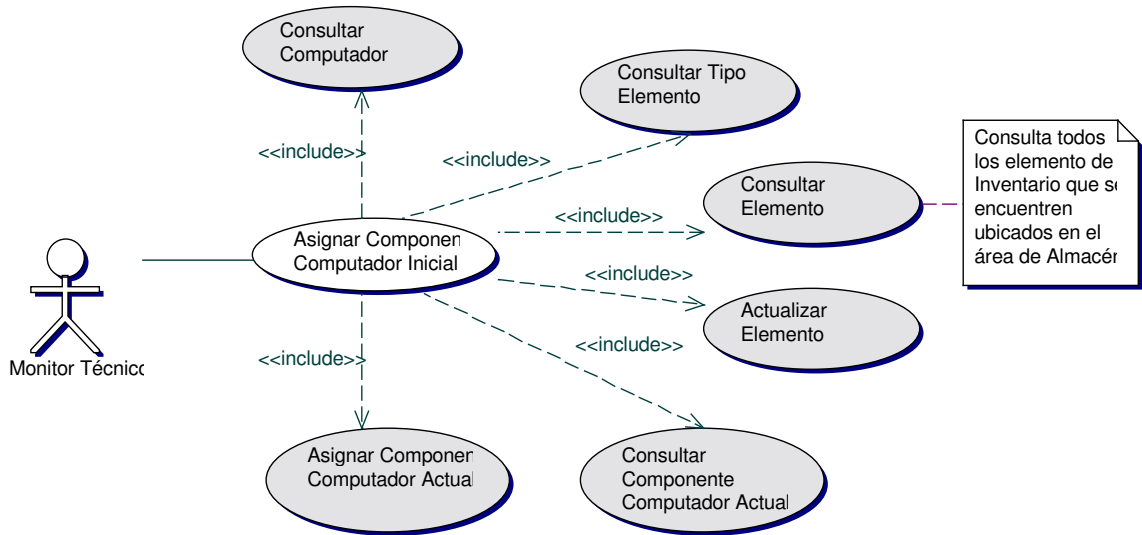
<b>IDENTIFICADOR:</b> ACCI1		<b>NOMBRE:</b> Administración de Componentes Computador Inicial	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Asignar Componente Computador Inicial Consultar Componente Por Computador Inicial Quitar Componente Computador Inicial Consultar Computador Consultar Tipo Elemento Consultar Elemento Consultar Elemento Computador Actual Asignar Componente Computador Actual Actualizar Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los computadores con sus respectivos componentes internos y externos. El proceso de administración se desarrollará teniendo en cuenta las operaciones de asignación, consulta y eliminación.			

<p><b>OBSERVACIONES:</b> El actor Monitor Técnico del Aula de Informática puede acceder a la administración de componentes iniciales de un computador sin ninguna restricción.</p>
<p><b>TRIGGERS:</b>  <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i></p>
<p><i>Precondición: Correcta autenticación del actor iniciador.</i>  <i>Postcondición: Administración eficiente de los componentes actuales de un computador.</i></p>
<p><b>CRITERIOS DE ACEPTACIÓN</b>  <i>Permite llevar una administración sistemática de los componentes iniciales de un computador.</i>  <i>La información en la base de datos se maneja de forma persistente y consistente.</i>  <i>Acceso rápido y oportuno a la información actualizada de cada uno los componentes de un computador.</i></p>

### 9.1.26.2 Segunda Iteración – Llenado (Diagrama General)



- **Diagrama Asignar Componente Computador Inicial**



- **Iteración Asignar Componente Computador Inicial**

<b>IDENTIFICADOR:</b> ACCII.1		<b>NOMBRE:</b> Asignar Componente Computador Inicial	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Computador Consultar Tipo Elemento Consultar Elemento Consultar Elemento Computador Actual Asignar Componente Computador Actual Actualizar Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del computador, código del elemento.		<b>SALIDA:</b> Componente Asignado o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar de forma detallada un computador con cada uno de sus componentes. Los componentes iniciales de un computador podrán ser elegidos entre aquellos elementos que se encuentren en la dependencia de almacén.			
<b>OBSERVACIONES:</b> La información de un computador y sus componentes se almacenará para identificar la conformación original de un computador y de forma paralela se generará un registro idéntico a través del cual se podrá manejar la conformación actual del computador. La ubicación que se le asigne a los componentes de un computador dependerá de la ubicación que tenga el computador, esto se actualizará automáticamente en inventario.			

**CURSO BÁSICO DE EVENTOS**

**Acción de los actores**

1. Selecciona la opción Asignar componentes iniciales.
3. Ingresa el dato solicitado.
4. Selecciona la opción Mostrar. (a)
  
8. Selecciona el elemento a asignar.
9. Selecciona la opción Asignar componente. (a)

**Respuesta del Sistema**

2. Solicita el código del computador al cual se le asignarán los componentes iniciales.
5. Verifica que la información suministrada sea correcta. (b)
6. Consulta los tipos de elementos disponibles. (c)
7. Despliega una pantalla presentado los elementos disponibles para la asignación.
10. Actualiza la ubicación del componente de acuerdo con la ubicación en la que se encuentre el computador.
11. Asigna el componente al computador actualizando la base de datos.
12. Genera el mismo registro en los componentes actuales del computador.
13. Muestra un mensaje indicando que el registro se realizó con éxito.

**CAMINOS DE EXCEPCION:**

- (a) El usuario decide no efectuar el proceso: El sistema cancela el proceso
- (b) La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.
- (c) No existen tipos de elementos: El sistema cancela el proceso
  - El sistema no responde. El proceso no se efectúa.

**TRIGGERS:**

Asignar componente computador actual: Permite crear un registro idéntico en componentes computador actual.  
 Actualización de ubicación: Este evento se ejecutará para actualizar automáticamente la ubicación de un determinado elemento de un computador en el inventario..

Precondición: Disponibilidad de componentes para llevar a cabo el registro.

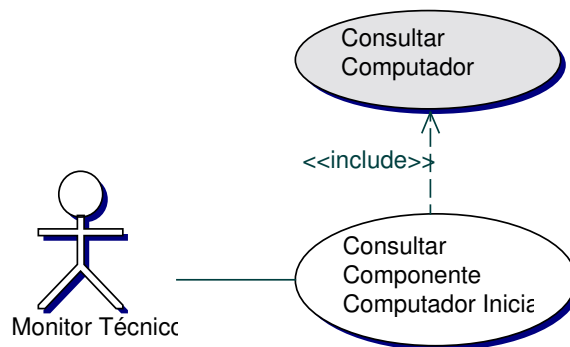
Postcondición: Registro eficiente de un computador y sus componentes.

**CRITERIOS DE ACEPTACIÓN**

El registro de los componentes de un computador permite guardar la información correspondiente a la configuración inicial de un computador la cual se podrá utilizar para la confrontación con la configuración actual.

Registro eficiente y consistente de un computador y sus componentes iniciales.

• **Diagrama Consultar Componente Inicial Por Computador**

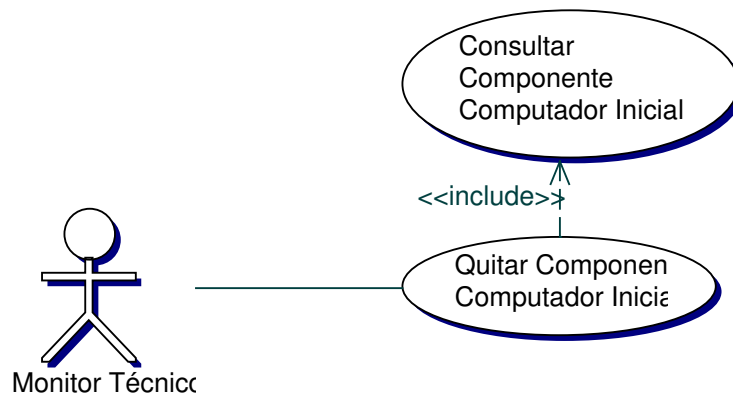




- Iteración Consultar Componente Computador Inicial

<b>IDENTIFICADOR:</b> ACCI1.2		<b>NOMBRE:</b> Consultar Componente Computador Inicial	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Computador		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del computador.		<b>SALIDA:</b> Listado de los componentes iniciales.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo realizar la consulta de la configuración inicial de un determinado computador.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Consultar componente inicial por computador.		2. Solicita el código del computador.	
3. Ingresa la información solicitada		5. Verifica que la información suministrada sea correcta. (b)	
4. Selecciona la opción Mostrar (a)		6. Despliega una pantalla presentando los componentes iniciales del computador.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso			
(b) La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente.			
<b>Postcondición:</b> Información consolidada de los componentes iniciales de un computador.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar de forma consolidada la información de los componentes iniciales de un computador.			

- Diagrama Quitar Componente Computador Inicial



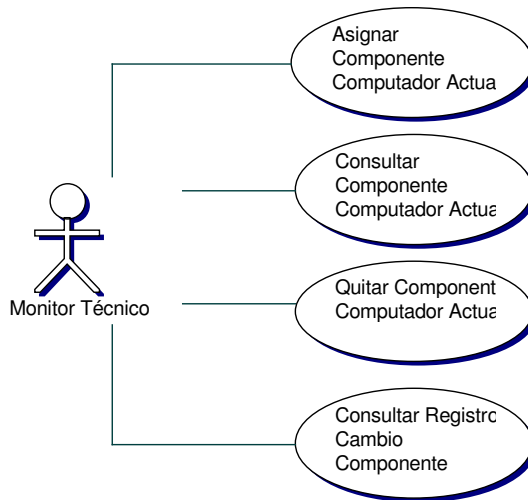
- Iteración Quitar Componentes Computador Inicial

<b>IDENTIFICADOR:</b> ACCI1.4		<b>NOMBRE:</b> Quitar Componentes Computador Inicial.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Computador Consultar Componente Computador Inicial		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del computador, código del elemento.		<b>SALIDA:</b> Componente eliminado proceso cancelado.	
<b>DESCRIPCION:</b> Se satisface este requerimiento permitiendo quitar un elemento inicial asignado a un computador.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Eliminar componentes computador inicial.		2. Solicita el código del computador.	
3. Suministra la información solicitada. (a)		5. Muestra los componentes asignados al computador. (b)	
4. Selecciona la opción Mostrar.		8. Se pide confirmación para llevar a cabo el proceso.	
6. Selecciona el componente a eliminar.		10. Elimina el componente actualizando la base de datos.	
7. Selecciona la opción eliminar componente. (a)		11. Muestra un mensaje indicando que el proceso se realizó con éxito.	
9. Confirma su decisión. (a)			
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.			
(b) La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: Los procesos de creación y actualización se llevan a cabo de forma satisfactoria La información de la base de datos es consistente			
Postcondición: Eliminación satisfactoria de los componentes iniciales de un computador.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar los componentes iniciales de un computador de forma consistente.			
Se garantiza la persistencia y consistencia de la información, ya que los componentes eliminados no afectarán el normal funcionamiento de los computadores.			

## 9.1.27 Administración de Componentes Computador Actual

### 9.1.27.1 Primera Iteración - Fachada

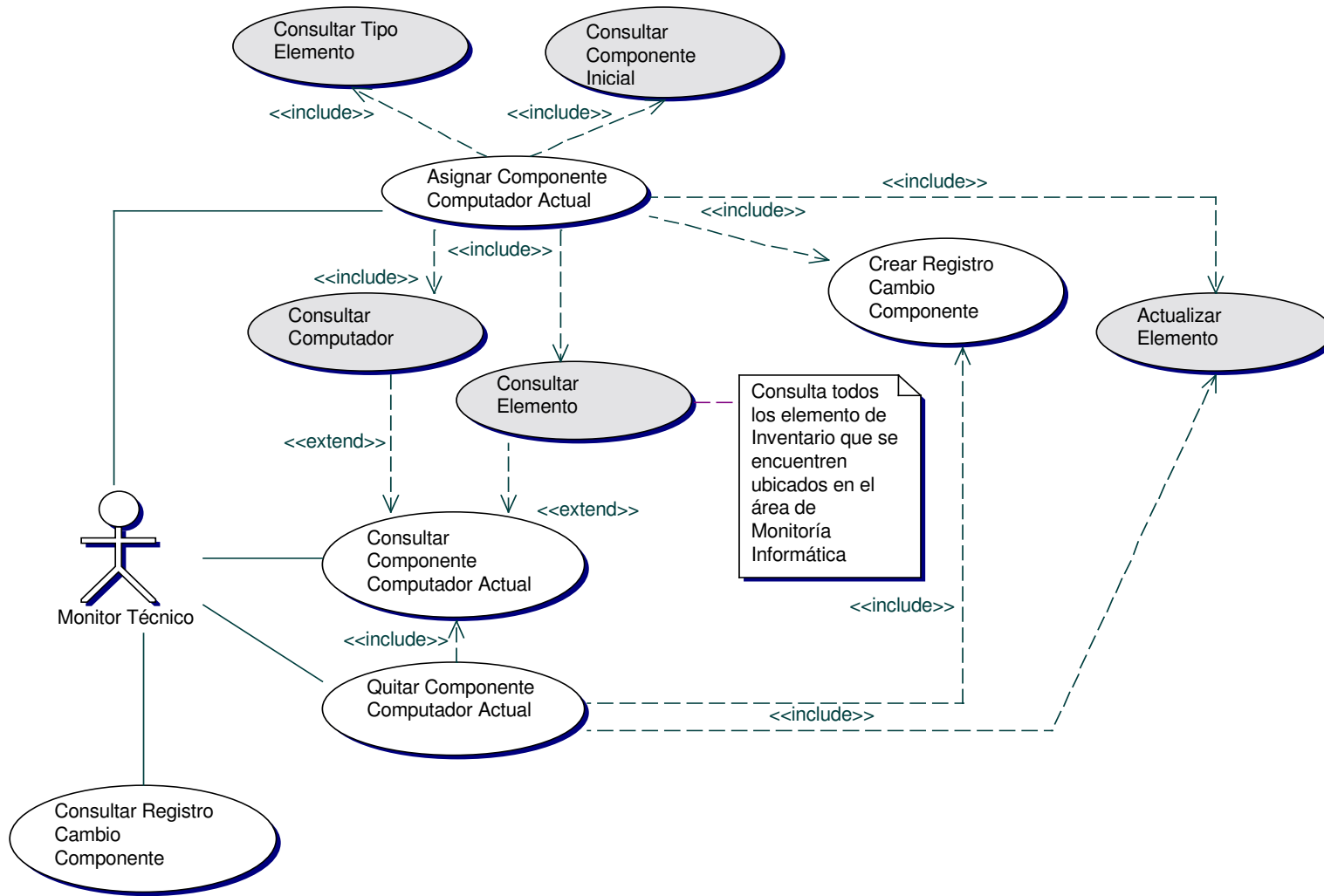
- Diagrama



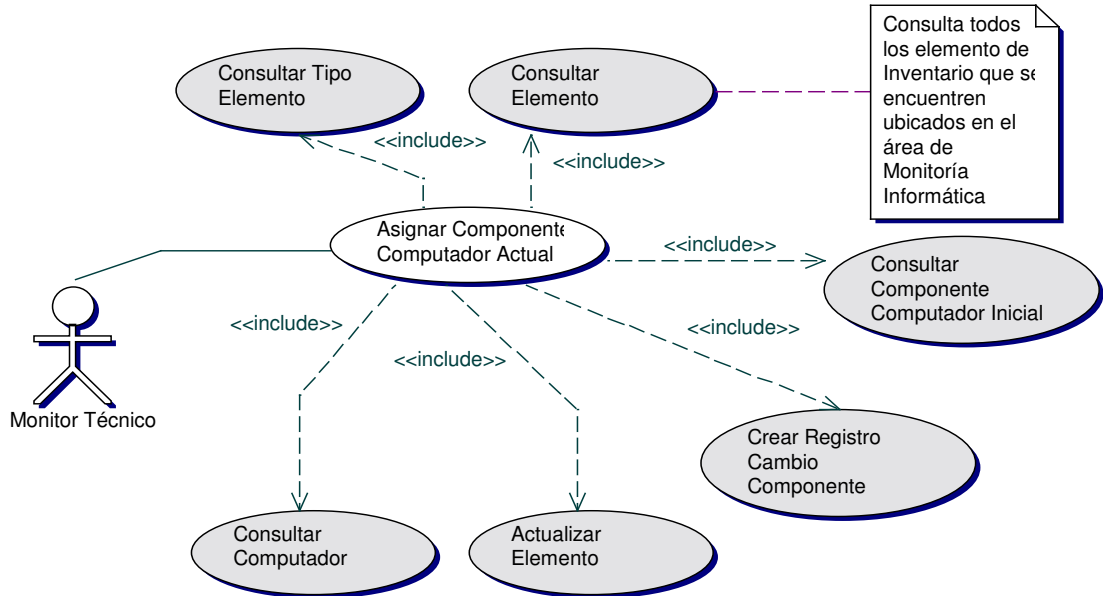
- Iteración

<b>IDENTIFICADOR:</b> ACCA1		<b>NOMBRE:</b> Administración de Componentes Computador Actual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Asignar Componente Computador Actual Consultar Componente Computador Actual Quitar Componente Computador Actual Consultar Registro Cambio Componente Consultar Computador Consultar Tipo Elemento Crear Registro Cambio Componente Consultar Elemento Consultar Componente Computador Inicial Actualizar Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los componentes internos y externos con los que cuenta actualmente un determinado computador. El proceso de administración se desarrollará teniendo en cuenta las operaciones de asignación, consulta, y eliminación.			
<b>OBSERVACIONES:</b> El actor Monitor Técnico del Aula de Informática puede acceder a la administración de componentes actuales de un computador sin ninguna restricción.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema. Precondición: Correcta autenticación del actor iniciador. Postcondición: Administración eficiente de los componentes actuales de un computador.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de los componentes actuales de un computador. La información en la base de datos se maneja de forma persistente y consistente. Acceso rápido y oportuno a la información actualizada de cada uno componentes actuales de un computador.			

### 9.1.27.2 Segunda Iteración – Llenado (Diagrama General)



- **Diagrama Asignar Componente Computador Actual**

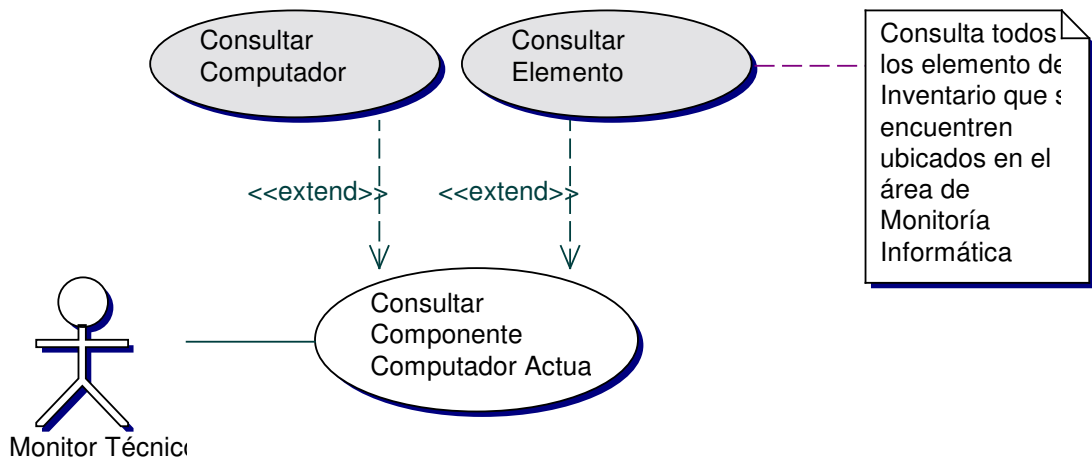


- **Iteración Asignar Componente Computador Actual**

<b>IDENTIFICADOR:</b> ACCA1.1		<b>NOMBRE:</b> Asignar Componente Computador Actual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Computador Consultar Componente Computador Inicial Consultar Tipo Elemento Consultar Elemento Actualizar Elemento Crear Registro Cambio Componente		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del computador, código del elemento, observaciones.		<b>SALIDA:</b> Componente asignado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar de forma detallada la conformación actual de un determinado computador. Los componentes actuales de un computador podrán ser elegidos entre aquellos elementos que se encuentren en la dependencia de monitoría informática. Las observaciones a registrar dependen del criterio del administrador y podrán ser omitidas si así se lo requiere.			

<p><b>OBSERVACIONES:</b> Para poder disponer de un elemento en la dependencia de monitoría informática este deberá ser previamente solicitado a la dependencia de almacén.</p> <p>La ubicación en la que se encuentre actualmente el computador será tomada para actualizar la ubicación de cada uno de los componentes actuales que se le asignen. Al asignar un componente actual se generará automáticamente un registro del cambio del componente.</p>																					
<p><b>CURSO BÁSICO DE EVENTOS</b></p> <table border="0"> <thead> <tr> <th style="text-align: center;"><b>Acción de los actores</b></th> <th style="text-align: center;"><b>Respuesta del Sistema</b></th> </tr> </thead> <tbody> <tr> <td>1. Selecciona la opción Asignar componentes actuales.</td> <td>2. Solicita el código del computador al cual se le asignarán los componentes.</td> </tr> <tr> <td>3. Ingresa el dato solicitado.</td> <td>5. Verifica que la información suministrada sea correcta. <b>(b)</b></td> </tr> <tr> <td>4. Selecciona la opción Mostrar. <b>(a)</b></td> <td>6. Consulta los componentes iniciales del computador. <b>(c)</b></td> </tr> <tr> <td></td> <td>7. Consulta los tipos de elementos. <b>(d)</b></td> </tr> <tr> <td>9. Selecciona el elemento a asignar.</td> <td>8. Despliega una pantalla presentando los elementos disponibles para la asignación.</td> </tr> <tr> <td>10. Selecciona la opción Asignar componente. <b>(a)</b></td> <td>11. Asigna el componente al computador actualizando la base de datos.</td> </tr> <tr> <td>13. Ingresa las observaciones solicitadas.</td> <td>12. Solicita las observaciones de la asignación.</td> </tr> <tr> <td></td> <td>14. Genera un registro del cambio actualizando la base de datos.</td> </tr> <tr> <td></td> <td>15. Muestra un mensaje indicando que el registro se realizó con éxito.</td> </tr> </tbody> </table>		<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>	1. Selecciona la opción Asignar componentes actuales.	2. Solicita el código del computador al cual se le asignarán los componentes.	3. Ingresa el dato solicitado.	5. Verifica que la información suministrada sea correcta. <b>(b)</b>	4. Selecciona la opción Mostrar. <b>(a)</b>	6. Consulta los componentes iniciales del computador. <b>(c)</b>		7. Consulta los tipos de elementos. <b>(d)</b>	9. Selecciona el elemento a asignar.	8. Despliega una pantalla presentando los elementos disponibles para la asignación.	10. Selecciona la opción Asignar componente. <b>(a)</b>	11. Asigna el componente al computador actualizando la base de datos.	13. Ingresa las observaciones solicitadas.	12. Solicita las observaciones de la asignación.		14. Genera un registro del cambio actualizando la base de datos.		15. Muestra un mensaje indicando que el registro se realizó con éxito.
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>																				
1. Selecciona la opción Asignar componentes actuales.	2. Solicita el código del computador al cual se le asignarán los componentes.																				
3. Ingresa el dato solicitado.	5. Verifica que la información suministrada sea correcta. <b>(b)</b>																				
4. Selecciona la opción Mostrar. <b>(a)</b>	6. Consulta los componentes iniciales del computador. <b>(c)</b>																				
	7. Consulta los tipos de elementos. <b>(d)</b>																				
9. Selecciona el elemento a asignar.	8. Despliega una pantalla presentando los elementos disponibles para la asignación.																				
10. Selecciona la opción Asignar componente. <b>(a)</b>	11. Asigna el componente al computador actualizando la base de datos.																				
13. Ingresa las observaciones solicitadas.	12. Solicita las observaciones de la asignación.																				
	14. Genera un registro del cambio actualizando la base de datos.																				
	15. Muestra un mensaje indicando que el registro se realizó con éxito.																				
<p><b>CAMINOS DE EXCEPCION:</b></p> <p><b>(a)</b> El usuario decide no efectuar el proceso: El sistema cancela el proceso</p> <p><b>(b)</b> La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.</p> <p><b>(c)</b> El computador no tiene asignados los componentes iniciales: El sistema cancela el proceso.</p> <p><b>(d)</b> No existen tipo de elementos: El sistema cancela el proceso.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>																					
<p><b>TRIGGERS:</b></p> <p>Actualización de ubicación: Este evento se ejecutará para actualizar automáticamente la ubicación de un determinado elemento de un computador en el inventario.</p> <p>Guardar un registro del cambio de componente efectuado: Este evento se ejecutará para crear un registro de los diferentes cambios que se hagan a un determinado computador.</p>																					
<p>Precondición: Disponibilidad de componentes en el área de monitoría Informática para llevar a cabo el registro. El computador al cual se le va asignar componentes actuales debe estar registrado con sus componentes iniciales.</p> <p>Postcondición: Registro eficiente de un computador y sus componentes actuales.</p>																					
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>El registro de los componentes actuales de un computador permite generar un historial de todos los cambios que sufre un computador a lo largo de su vida útil.</p> <p>Registro eficiente y consistente de un computador y sus componentes actuales.</p>																					

- **Diagrama Consultar Componente Computador Actual**

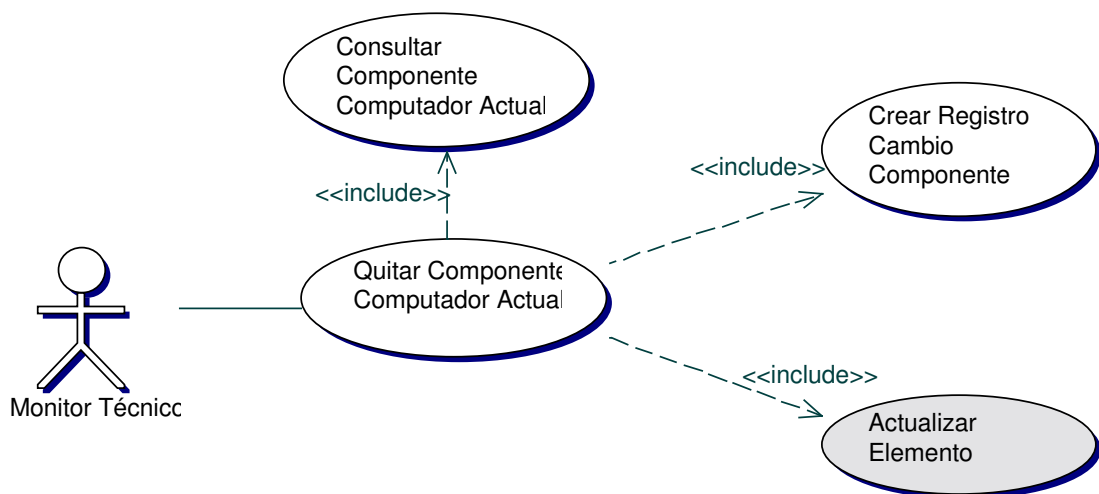


- Iteración Consultar Componente Computador Actual

<b>IDENTIFICADOR:</b> ACCA1.2		<b>NOMBRE:</b> Consultar Componente Computador Actual	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Computador Consultar Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del computador, código del elemento.		<b>SALIDA:</b> Listado de los componentes iniciales.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo realizar la consulta de la configuración actual de un determinado computador o la información de donde se encuentra un determinado elemento..			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción Consultar componente computador actual.	2.	Despliega un menú presentando las opciones para su consulta (código computador, código de elemento).
3.	Selecciona la opción deseada.	4.	Solicita la información necesaria de acuerdo al criterio seleccionado.
5.	Ingresa la información solicitada.	7.	Verifica que la información suministrada sea correcta. <b>(b)</b>
6.	Selecciona la opción <b>Mostrar (a)</b>	8.	Despliega una pantalla presentando los componentes actuales del computador.

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso</p> <p>(b) La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: La información de la base de datos es consistente.</p> <p>Postcondición: Información consolidada de los componentes actuales de un computador.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Se cumple el requerimiento cuando se logra consultar de forma consolidada la información de los componentes actuales de un computador.</p>

• **Diagrama Quitar Componente Computador Actual**



• **Iteración Quitar Componentes Computador Actual**

<b>IDENTIFICADOR:</b> ACCA1.4		<b>NOMBRE:</b> Quitar Componentes Computador Actual.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Componente Computador Actual Crear Registro Cambio Componente Actualizar Elemento		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del computador, código del elemento, observaciones		<b>SALIDA:</b> Componente eliminado proceso cancelado.	
<b>DESCRIPCION:</b> Se satisface este requerimiento permitiendo quitar un elemento asignado a un computador. Las observaciones a registrar dependen del criterio del administrador y podrán ser omitidas si así se lo requiere.			



### CURSO BÁSICO DE EVENTOS

<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. <i>Selecciona la opción Eliminar componentes computador inicial.</i>	2. <i>Solicita el código del computador.</i>
3. <i>Suministra la información solicitada. (a)</i>	
4. <i>Selecciona la opción Mostrar.</i>	5. <i>Muestra los componentes asignados al computador. (b)</i>
6. <i>Selecciona el componente a eliminar.</i>	
7. <i>Selecciona la opción eliminar componente. (a)</i>	8. <i>Se pide confirmación para llevar a cabo el proceso.</i>
9. <i>Confirma su decisión. (a)</i>	10. <i>Elimina el componente actualizando la base de datos.</i>
	11. <i>Actualiza la ubicación del elemento eliminado a la dependencia de monitoría informática.</i>
	12. <i>Solicita las observaciones de la eliminación.</i>
13. <i>Suministra la información solicitada. (a)</i>	14. <i>Actualiza la base de datos creando un registro del cambio de componente.</i>
	15. <i>Muestra un mensaje indicando que el proceso se realizó con éxito.</i>

**CAMINOS DE EXCEPCION:**

- (a) *El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.*
- (b) *La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.*

- *El sistema no responde. El proceso no se efectúa.*

**TRIGGERS:**

*Actualización de ubicación: Este evento se ejecutará para actualizar automáticamente la ubicación de un determinado elemento de un computador en el inventario.*

*Guardar un registro del cambio de componente efectuado: Este evento se ejecutará para crear un registro de los diferentes cambios que se hagan a un determinado computador.*

*Precondición: Los procesos de creación y actualización se llevan a cabo de forma satisfactoria  
La información de la base de datos es consistente*

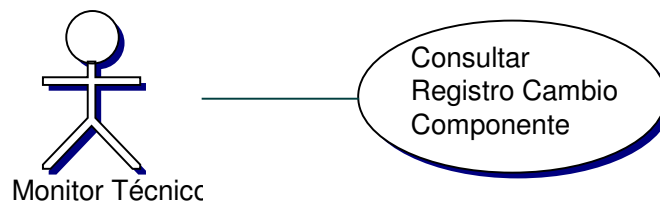
*Postcondición: Eliminación satisfactoria de los componentes actuales de un computador.  
Información consolidada de los cambios efectuados sobre un computador a lo largo de su vida útil.*

**CRITERIOS DE ACEPTACIÓN**

*Se cumple el requerimiento cuando se logra eliminar los componentes actuales de un computador de forma consistente.*

*Se garantiza la persistencia y consistencia de la información, ya que los componentes eliminados no afectarán el normal funcionamiento de los computadores.*

### • Diagrama Consultar Registro Cambio Componente



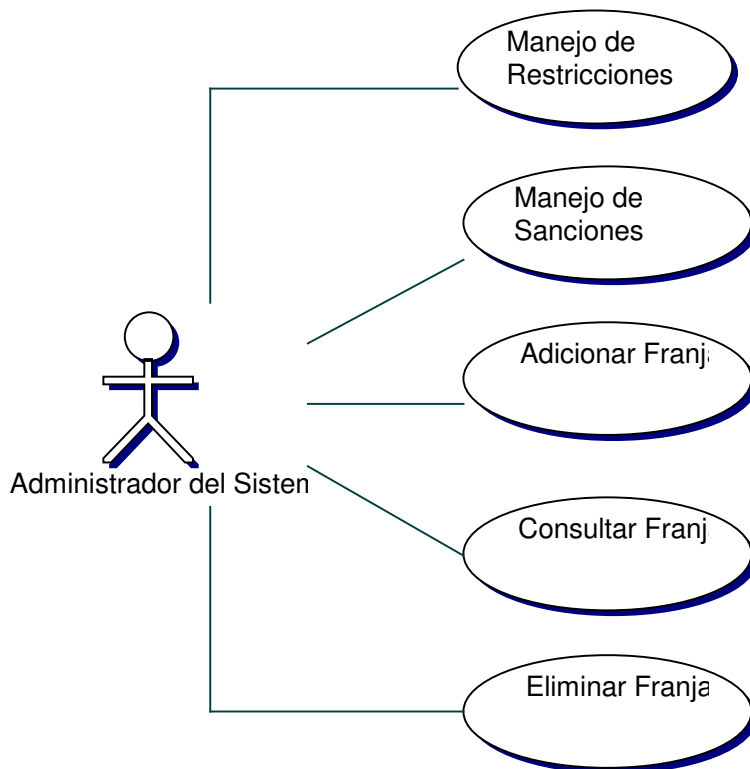
- Iteración Consultar Registro Cambio Componente

<b>IDENTIFICADOR:</b> ACCA1.5		<b>NOMBRE:</b> Consultar Registro Cambio Componente.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Fecha, código del computador, código del elemento		<b>SALIDA:</b> Listado de los cambios efectuados	
<b>DESCRIPCION:</b> Se satisface este requerimiento permitiendo consultar los cambios de componentes efectuados a determinado un computador, en una fecha dada o de un determinado elemento de inventario.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Consultar registro cambio componente.		2. Despliega un menú presentando las opciones para su consulta (por código de computador, fecha, código elemento).	
3. Selecciona la opción deseada. <b>(a)</b>		4. Solicita la información necesaria de acuerdo al criterio seleccionado.	
5. Ingresar la información solicitada		7. Se despliega el resultado de la consulta mostrando la información. <b>(b)</b>	
6. Selecciona la opción Mostrar <b>(a)</b>			
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.			
<b>(b)</b> La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: El proceso de creación y eliminación se llevan a cabo de forma satisfactoria La información de la base de datos es consistente.			
Postcondición: Información consolidada de la consulta efectuada.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los cambios de componentes efectuados de forma consistente.			

## 9.1.28 Administración de Servicios Informaticos

### 9.1.28.1 Primera Iteración - Fachada

- Diagrama

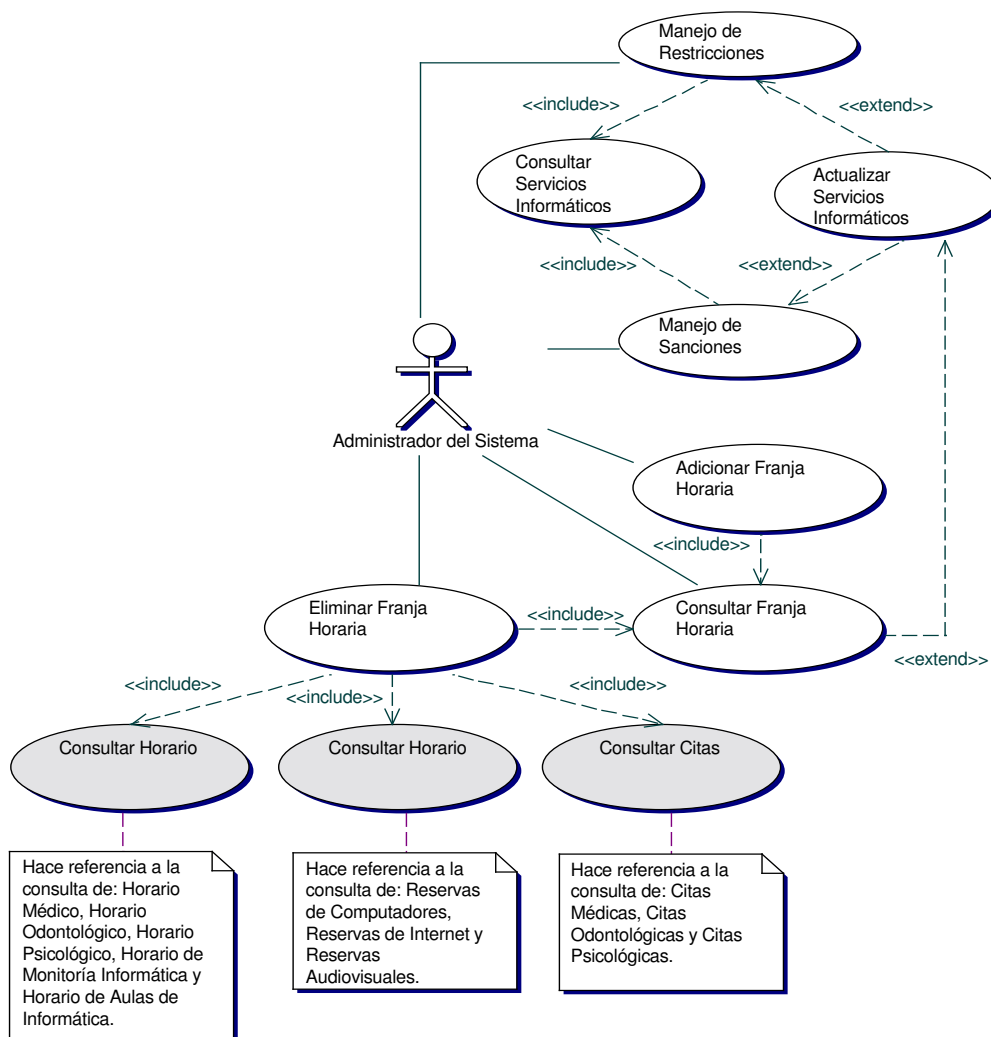


- Iteración

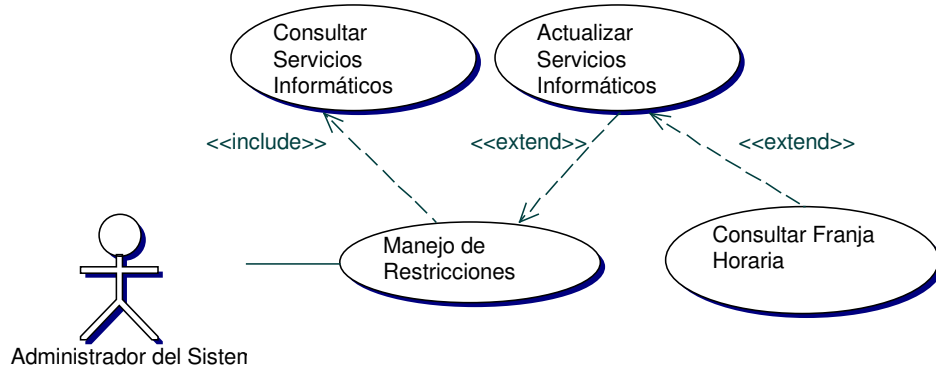
<b>IDENTIFICADOR:</b> ASI1		<b>NOMBRE:</b> Administración de Servicios Informáticos	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicios Informáticos Actualizar Servicios Informáticos Adicionar Franja Horaria Consultar Franja Horaria Eliminar Franja Horaria Consultar Horario Consultar Reservas Consultar citas		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo manejar las restricciones y las sanciones que se determinan para el uso de los servicios de internet, computadores y audiovisuales. De igual manera permite administrar las franjas horarias establecidas para la prestación de dichos servicios. El proceso de administración se desarrollará teniendo en cuenta las operaciones de adición, consulta, actualización y eliminación.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una			

<i>respuesta del sistema.</i>
<i>Precondición: Correcta autenticación del actor iniciador.</i>
<i>Postcondición: Administración eficiente de los servicios informáticos.</i>
<b>CRITERIOS DE ACEPTACIÓN</b>
<i>La información en la base de datos se maneja de forma persistente y consistente.</i>
<i>Acceso rápido y oportuno a la información actualizada de cada una de las restricciones, sanciones y franjas horarias establecidas.</i>

### 9.1.28.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Manejo de Restricciones

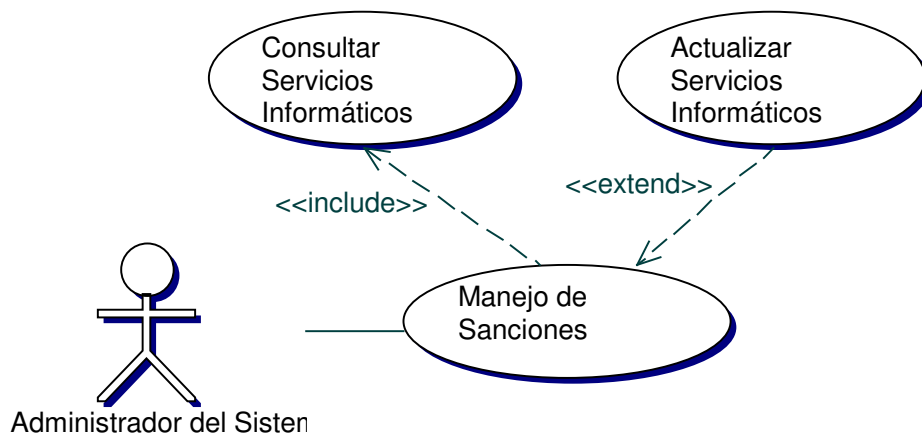


- Iteración Manejo de Restricciones

<b>IDENTIFICADOR:</b> ASI1.1		<b>NOMBRE:</b> Manejo de Restricciones	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicios Informáticos Actualizar Servicios Informáticos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del servicio, periodo de anticipación, número de reservas.		<b>SALIDA:</b> Actualización efectuada o proceso cancelado	
<b>DESCRIPCIÓN:</b> El manejo de restricciones permite una administración flexible en cuanto a las posibles eventualidades y requerimientos que surjan con el tiempo. La flexibilidad se verá reflejada en la capacidad de establecer periodos de anticipación en días para realizar las reservas y número de reservas permitidas para un usuario en un mismo día.			
<b>OBSERVACIONES:</b> Los servicios informáticos establecidos son: Reservas de Computadores, Reservas de Internet y Reservas Audiovisuales. Dentro de este requerimiento se cumple tanto con la consulta como con la actualización de las restricciones.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Restricciones		2. Despliega una pantalla con las restricciones establecidas actualmente, permitiendo modificar sus valores.	
3. Modifica los datos de acuerdo a su criterio.		5. Verifica que el número de reservas ingresado no sobrepase el número de franjas horarias activas. (b)	
4. Selecciona la opción Actualizar. (a)		6. Actualiza la base de datos con los nuevos valores ingresados	
		7. Muestra un mensaje indicando que la actualización se realizó con éxito.	

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no efectuar la actualización: El sistema muestra la consulta y cancela el proceso de actualización.</p> <p>(b) La información suministrada es incorrecta: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la información solicitada..</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: Registro eficiente de los diferentes servicios informáticos.</p> <p>Postcondición: Información consolidada y actualizada de las restricciones establecidas para cada uno de los servicios informáticos..</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Actualización y/o consulta eficiente y consistente de las restricciones establecidas.</p>

• **Diagrama Manejo de Sanciones**

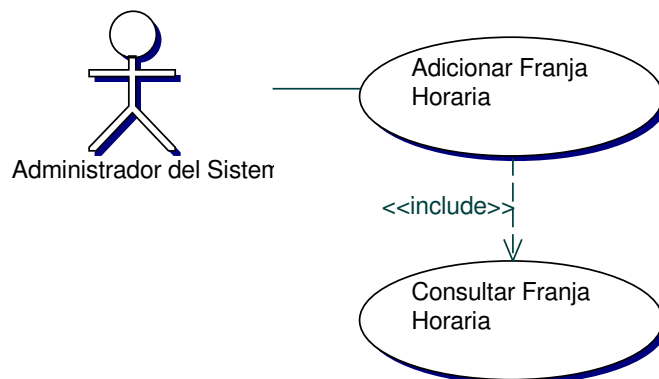


• **Iteración Manejo de Sanciones**

<b>IDENTIFICADOR:</b> ASI1.2		<b>NOMBRE:</b> Manejo de Sanciones	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicios Informáticos Actualizar Servicios Informáticos		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del servicio, número de faltas, periodo de sanción.		<b>SALIDA:</b> Actualización efectuada o proceso cancelado	
<b>DESCRIPCIÓN:</b> El manejo de sanciones permite una administración flexible en cuanto a las posibles eventualidades y requerimientos que surjan con el tiempo. La flexibilidad se verá reflejada en la capacidad de establecer el número de faltas con el que un usuario podría entrar en estado de sanción y el número de días con el cual se podría amonestar a un usuario por faltas incurridas en el uso de los servicios informáticos..			

<p><b>OBSERVACIONES:</b> Los servicios informáticos establecidos son: Reservas de Computadores, Reservas de Internet y Reservas Audiovisuales. Dentro de este requerimiento se cumple tanto con la consulta como con la actualización de las sanciones.</p>									
<p><b>CURSO BÁSICO DE EVENTOS</b></p> <table border="0"> <tr> <td style="text-align: center;"><b>Acción de los actores</b></td> <td style="text-align: center;"><b>Respuesta del Sistema</b></td> </tr> <tr> <td>1. Selecciona la opción Sanciones.</td> <td>2. Despliega una pantalla con las sanciones establecidas actualmente, permitiendo modificar sus valores.</td> </tr> <tr> <td>3. Modifica los datos de acuerdo a su criterio.</td> <td>5. Actualiza la base de datos con los nuevos valores ingresados</td> </tr> <tr> <td>4. Selecciona la opción Actualizar. (a)</td> <td>6. Muestra un mensaje indicando que la actualización se realizó con éxito.</td> </tr> </table>		<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>	1. Selecciona la opción Sanciones.	2. Despliega una pantalla con las sanciones establecidas actualmente, permitiendo modificar sus valores.	3. Modifica los datos de acuerdo a su criterio.	5. Actualiza la base de datos con los nuevos valores ingresados	4. Selecciona la opción Actualizar. (a)	6. Muestra un mensaje indicando que la actualización se realizó con éxito.
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>								
1. Selecciona la opción Sanciones.	2. Despliega una pantalla con las sanciones establecidas actualmente, permitiendo modificar sus valores.								
3. Modifica los datos de acuerdo a su criterio.	5. Actualiza la base de datos con los nuevos valores ingresados								
4. Selecciona la opción Actualizar. (a)	6. Muestra un mensaje indicando que la actualización se realizó con éxito.								
<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no efectuar la actualización: El sistema muestra la consulta y cancela el proceso de actualización.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>									
<p>Precondición: Registro eficiente de los diferentes servicios informáticos. Postcondición: Información consolidada y actualizada de las sanciones establecidas para cada uno de los servicios informáticos..</p>									
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Actualización y/o consulta eficiente y consistente de las sanciones establecidas.</p>									

• **Diagrama Adicionar Franja Horaria**

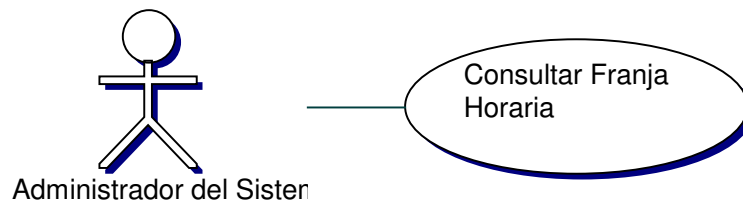


• **Iteración Adicionar Franja Horaria**

<b>IDENTIFICADOR:</b> ASI1.3		<b>NOMBRE:</b> Adicionar Franja Horaria
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consulta Franja Horaria	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.

<b>ENTRADA:</b> Código de la franja.	<b>SALIDA:</b> Franja adicionada o proceso cancelado
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo adicionar una determinada franja horaria.	
<b>OBSERVACIONES:</b> Se establecen 24 franjas de una hora cada una las cuales pueden tener un estado activo o inactivo, por lo tanto este proceso permite cambiar a estado activo una determinada franja.	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción Adicionar franja horaria.	2. Despliega una pantalla presentando las franjas activas y las posibles franjas que pueden ser adicionadas. <b>(b)</b>
3. Selecciona la franja a adicionar.	5. Adiciona la franja horaria seleccionada actualizando la base de datos.
4. Selecciona la opción Adicionar. <b>(a)</b>	6. Muestra un mensaje indicando que el registro se realizó con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El usuario decide no efectuar el proceso: El sistema cancela el proceso	
<b>(b)</b> No hay franjas horarias disponibles El sistema cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: Disponibilidad de franjas horarias para adicionar.	
Postcondición: Adición eficiente de una franja horaria.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Adición eficiente y consistente de una determinada franja horaria.	

- **Diagrama Consultar Franja Horaria**



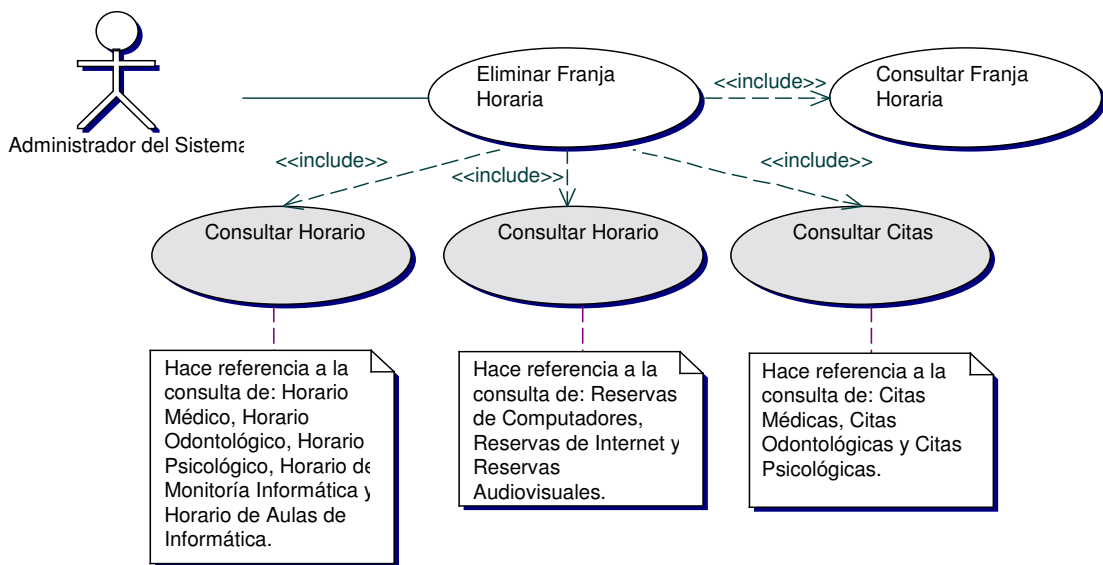
- **Iteración Consultar Franja Horaria**

<b>IDENTIFICADOR:</b> ASI1.4		<b>NOMBRE:</b> Consultar Franja Horaria	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Listado de las franjas activas.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo realizar la consulta de las franjas activas.			



<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. <i>Selecciona la opción Consultar franjas horarias. (a)</i>	2. <i>Despliega una pantalla presentando las franjas actualmente activas. (b)</i>
<b>CAMINOS DE EXCEPCION:</b>	
<i>(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso</i>	
<i>(b) No hay franjas activas: El sistema informa al usuario y cancela el proceso.</i>	
<ul style="list-style-type: none"> <li><i>El sistema no responde. El proceso no se efectúa.</i></li> </ul>	
<i>Precondición: La información de la base de datos es consistente.</i>	
<i>Postcondición: Información consolidada de las franjas activas.</i>	
<b>CRITERIOS DE ACEPTACIÓN</b>	
<i>Se cumple el requerimiento cuando se logra consultar de forma consolidada la información de las franjas horarias activas.</i>	

• **Diagrama Eliminar Franja Horaria**



• **Iteración Eliminar Franja Horaria**

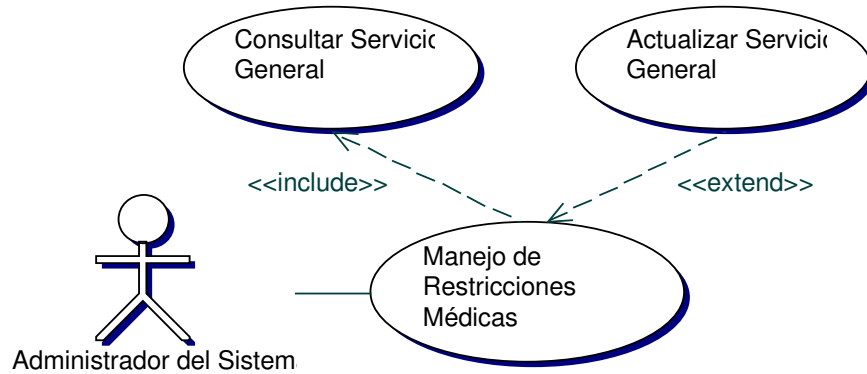
<b>IDENTIFICADOR:</b> ASI1.5	<b>NOMBRE:</b> Eliminar Franja Horaria.
---------------------------------	--

<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Franja Horaria Consultar Horario Médico Consultar Horario Odontológico Consultar Horario Psicológico Consultar Reservas de Internet Consultar Reservas de Computadores Consultar Reservas Audiovisuales Consultar Citas Médicas Consultar Citas Odontológicas Consultar Citas Psicológicas	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Código de la franja	<b>SALIDA:</b> Franja eliminada proceso cancelado.	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar una determinada franja horaria.		
<b>OBSERVACIONES:</b> Se establecen 24 franjas de una hora cada una las cuales pueden tener un estado activo o inactivo, por lo tanto este proceso permite cambiar a estado inactivo una determinada franja.		
<b>CURSO BÁSICO DE EVENTOS</b>		
<p style="text-align: center;"><b>Acción de los actores</b></p> <ol style="list-style-type: none"> <li>1. Selecciona la opción Eliminar franja horaria.</li> <li>3. Selecciona la franja a eliminar.</li> <li>4. Selecciona la opción Eliminar. (a)</li> </ol>		<p style="text-align: center;"><b>Respuesta del Sistema</b></p> <ol style="list-style-type: none"> <li>2. Despliega una pantalla presentando las franjas activas que pueden ser eliminadas.(b)</li> <li>5. Verifica que la franja a eliminar no este involucrada en otro proceso.(c)</li> <li>6. Elimina la franja horaria seleccionada actualizando la base de datos.</li> <li>7. Muestra un mensaje indicando que el proceso se realizó con éxito.</li> </ol>
<b>CAMINOS DE EXCEPCION:</b>		
(a) El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.		
(b) No hay franjas activas: El sistema informa al usuario y cancela el proceso.		
(c) La franja a eliminar está involucrada en otro proceso: El sistema muestra un mensaje de error dando la oportunidad de seleccionar otra franja.		
• El sistema no responde. El proceso no se efectúa.		
<b>Precondición:</b> El proceso de adición se llevan a cabo de forma satisfactoria La información de la base de datos es consistente		
<b>Postcondición:</b> Eliminación satisfactoria de una determinada franja.		
<b>CRITERIOS DE ACEPTACIÓN</b>		
Se cumple el requerimiento cuando se logra eliminar una determinada franja horaria. Se garantiza la persistencia y consistencia de la información, ya que la franja eliminada no afectará ningún otro proceso.		

## 9.1.29 Manejo de Restricciones Medicas

### 9.1.29.1 Primera Iteración - Fachada

- Diagrama Manejo de Restricciones Médicas



- Iteración Manejo de Restricciones Médicas

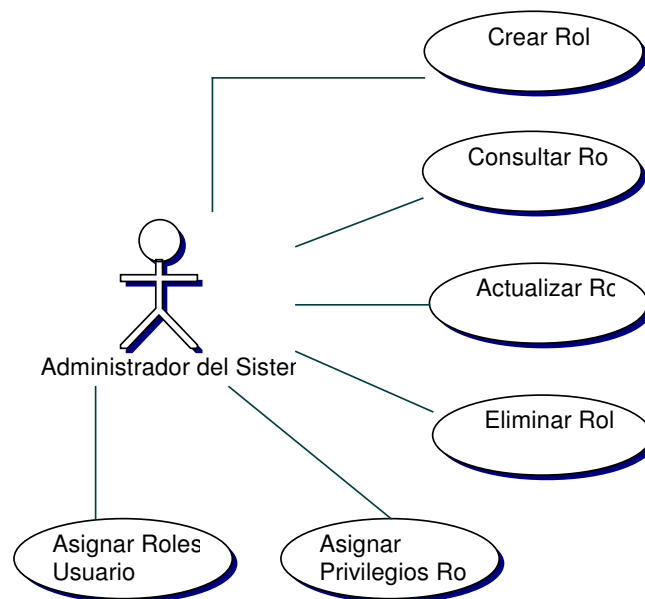
<b>IDENTIFICADOR:</b> MRM1		<b>NOMBRE:</b> Manejo de Restricciones Médicas	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Servicio General Actualizar Servicio General		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del servicio médico, periodo de anticipación, número de citas, número de pacientes.		<b>SALIDA:</b> Actualización efectuada o proceso cancelado	
<b>DESCRIPCIÓN:</b> El manejo de restricciones médicas permite una administración flexible en cuanto a las posibles eventualidades y requerimientos que surjan con el tiempo. La flexibilidad se verá reflejada en la capacidad de establecer periodos de anticipación en días para solicitar una cita médica, número de citas que puede solicitar un usuario en un mismo día y número de pacientes por franja que puede atender un médico.			
<b>OBSERVACIONES:</b> Los servicios médicos o generales establecidos son: Servicio Médico general, Servicio Odontólogo y Servicio Psicológico. Dentro de este requerimiento se cumple tanto con la consulta como con la actualización de las Restricciones médicas.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Restricciones Médicas		2. Despliega una pantalla con las restricciones médicas establecidas actualmente, permitiendo modificar sus valores.	
3. Modifica los datos de acuerdo a su criterio.		5. Actualiza la base de datos con los nuevos valores ingresados	
4. Selecciona la opción Actualizar. (a)		6. Muestra un mensaje indicando que la actualización se realizó con éxito.	

<p><b>CAMINOS DE EXCEPCION:</b></p> <p>(a) El usuario decide no efectuar la actualización: El sistema muestra la consulta y cancela el proceso de actualización.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: Registro eficiente de los de los diferentes servicios generales.</p> <p>Postcondición: Información consolidada y actualizada de las restricciones médicas establecidas para cada uno de los servicios generales.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b></p> <p>Actualización y/o consulta eficiente y consistente de las restricciones médicas.</p>

## 9.1.30 Administración De Roles

### 9.1.30.1 Primera Iteración - Fachada

- Diagrama

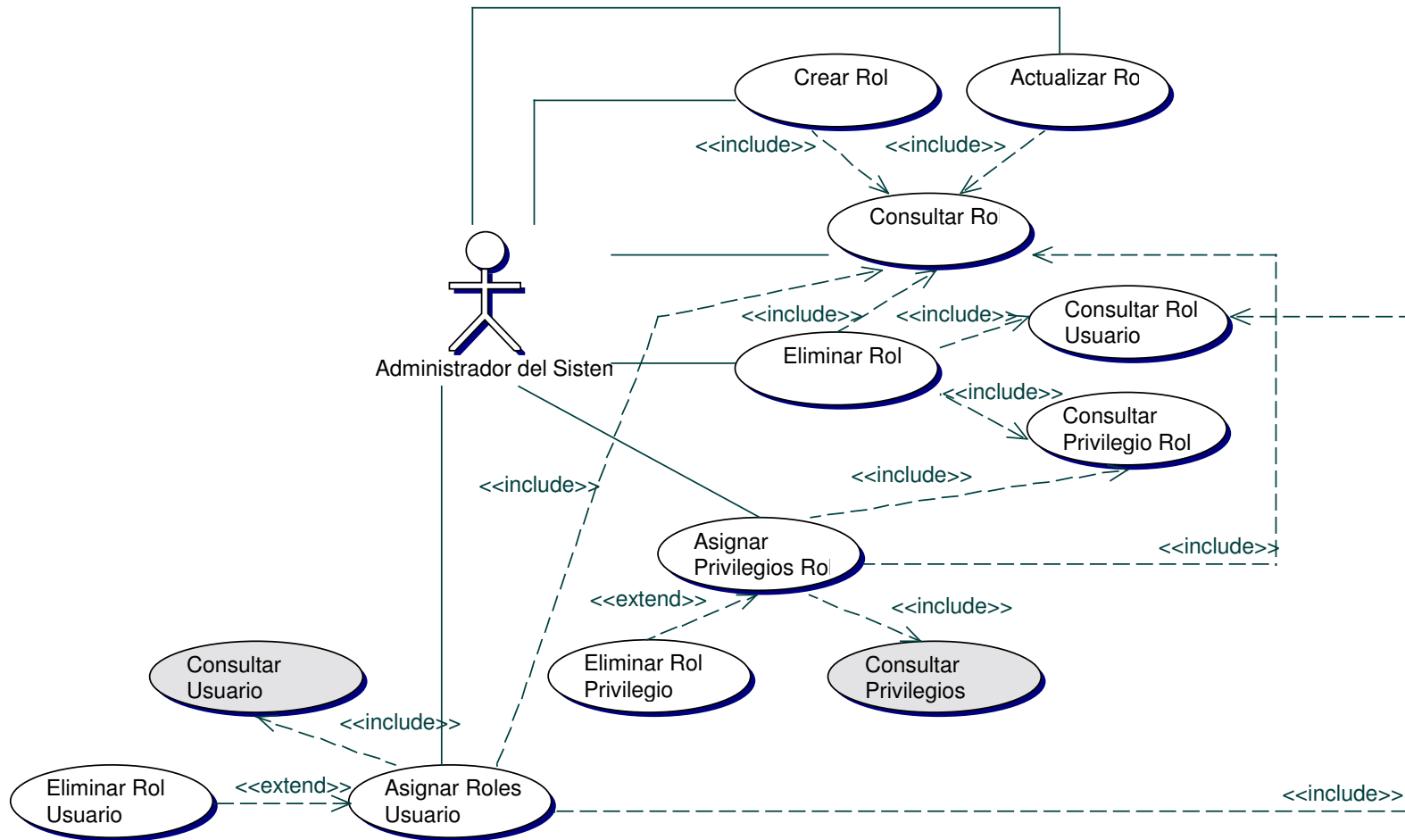


- Iteración

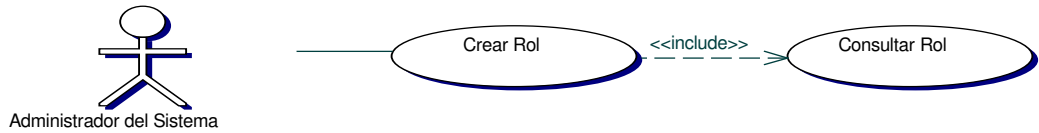
<b>IDENTIFICADOR:</b> ARI		<b>NOMBRE:</b> Administración de Roles
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Roles Consultar Roles Actualizar Roles Eliminar Roles	<b>CRITICO?</b> Si

	<i>Asignar Roles Usuario  Asignar Privilegios Rol  Consultar Rol Usuario  Consultar Privilegio Rol  Consultar Privilegios  Eliminar Rol Privilegio  Consultar Rol  Eliminar Rol Usuario</i>	
<b>PRIORIDAD DE DESARROLLO:</b> <i>Fachada</i>	<b>ACTORES:</b> <i>Iniciador y Ejecutor: Administrador del Sistema</i>	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> <i>Páginas HTML y JSP.</i>
<b>ENTRADA:</b> <i>Datos para efectuar el proceso.</i>	<b>SALIDA:</b> <i>Proceso efectuado o cancelado</i>	
<b>DESCRIPCIÓN:</b> <i>Se satisface este requerimiento permitiendo administrar los roles necesarios para una excelente gestión del sistema. De igual manera permite asignar, consultar o eliminar los privilegios de un determinado rol como también asignar, consultar o eliminar los roles de un determinado usuario. El proceso de administración de roles se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y eliminación.</i>		
<b>OBSERVACION:</b> <i>El rol correspondiente a Administrador y Estudiante no podrán ser eliminado ni actualizados.</i>		
<b>TRIGGERS:</b> <i>Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.</i>		
<i>Precondición: Correcta autenticación del actor iniciador.  Postcondición: Administración eficiente de Roles.</i>		
<b>CRITERIOS DE ACEPTACIÓN</b> <i>La información en la base de datos se maneja de forma persistente y consistente.  Acceso rápido y oportuno a la información actualizada de cada uno los roles.</i>		

### 9.1.30.2 Segunda Iteración – Llenado (Diagrama General)



• Diagrama Crear Rol



• Iteración Crear Rol

<b>IDENTIFICADOR:</b> AR1.1		<b>NOMBRE:</b> Crear Rol	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rol		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, descripción.		<b>SALIDA:</b> Rol Registrado o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear los roles necesarios para la buena gestión del sistema. El código será generado automáticamente, no pueden existir descripción duplicadas.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción Crear rol.	2.	Solicita la descripción.
3.	Ingresa la información solicitada.	5.	Verifica que la descripción digitado no se repita. <b>(b)</b>
4.	Selecciona la opción Crear. <b>(a)</b>	6.	Crea el rol actualizando la base de datos.
		7.	Muestra un mensaje indicando que el registro se realizó con éxito.
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El usuario decide no efectuar el proceso: El sistema cancela el proceso			
<b>(b)</b> El dato suministrado ya existe: El sistema muestra un mensaje de error y cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: Ingreso completo de los datos requeridos.			
Postcondición: Registro eficiente de los diferentes roles.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Almacenamiento de la información en forma ordenada y clasificada.			
Se cumple el requerimiento cuando se logra crear un rol de forma consistente y persistente.			

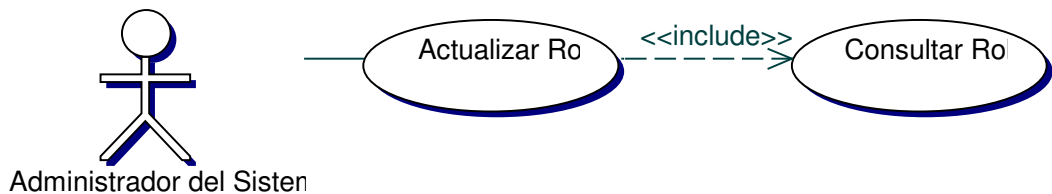
• Diagrama Consultar Rol



- Iteración Consultar Rol

<b>IDENTIFICADOR:</b> AR1.2		<b>NOMBRE:</b> Consultar Rol	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Listado de los roles existentes	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los roles existentes. La información desplegada incluirá únicamente la descripción.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. Selecciona la opción Consultar roles. (a)		2. Se despliega el resultado de la consulta.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Información consolidada de los roles existentes.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los roles existentes de forma veraz y oportuna.			

- Diagrama Actualizar Roles



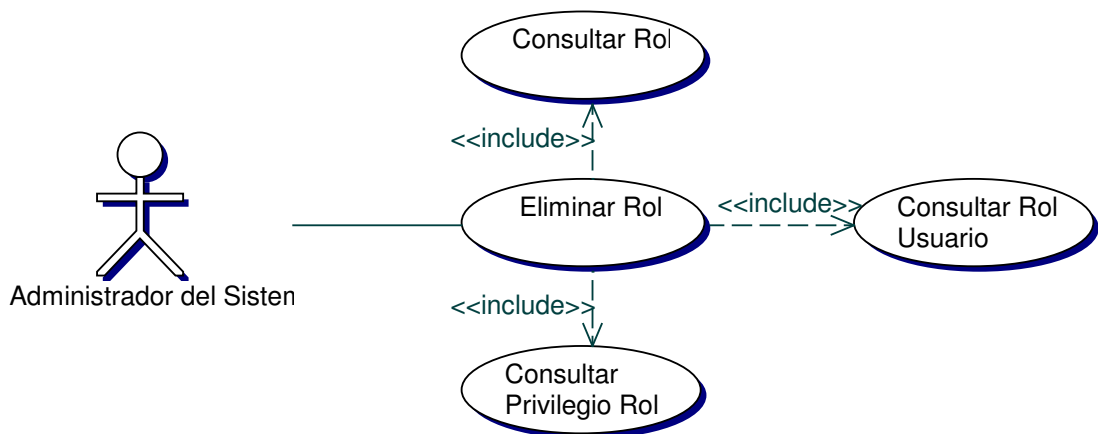
- Iteración Actualizar Roles

<b>IDENTIFICADOR:</b> AR1.3		<b>NOMBRE:</b> Actualizar Rol	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rol		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b>	



Código, descripción.	Actualización efectuada o cancelada
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar la descripción de un determinado rol.	
<b>OBSERVACIONES:</b> Se desplegará la información correspondiente a todos los roles existentes excepto el rol Administrador y Estudiante los cuales no podrán ser modificados.	
<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción Actualizar rol.	2. Despliega una pantalla mostrando todos los roles existentes y solicitando la nueva descripción
3. Suministra la información solicitada.	5. Solicita confirmación.
4. Selecciona la opción actualizar rol.(a)	7. Verifica que la descripción suministrada no se repita. (b)
6. Confirma su decisión (a)	8. Actualiza el rol modificando la base de datos con la información suministrada.
	9. Muestra un mensaje indicando que el proceso se realizó con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.	
(b) La información suministrada es incorrecta: El sistema muestra un mensaje de error y cancela el proceso.	
<ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>	
Precondición: Los procesos de creación se llevan a cabo de forma satisfactoria. La información de la base de datos es consistente.	
Postcondición: Eficiente actualización de los roles.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra actualizar la descripción de un determinado rol de forma veraz y oportuna.	
La información de la base de datos se almacena de forma persistente y consistente.	

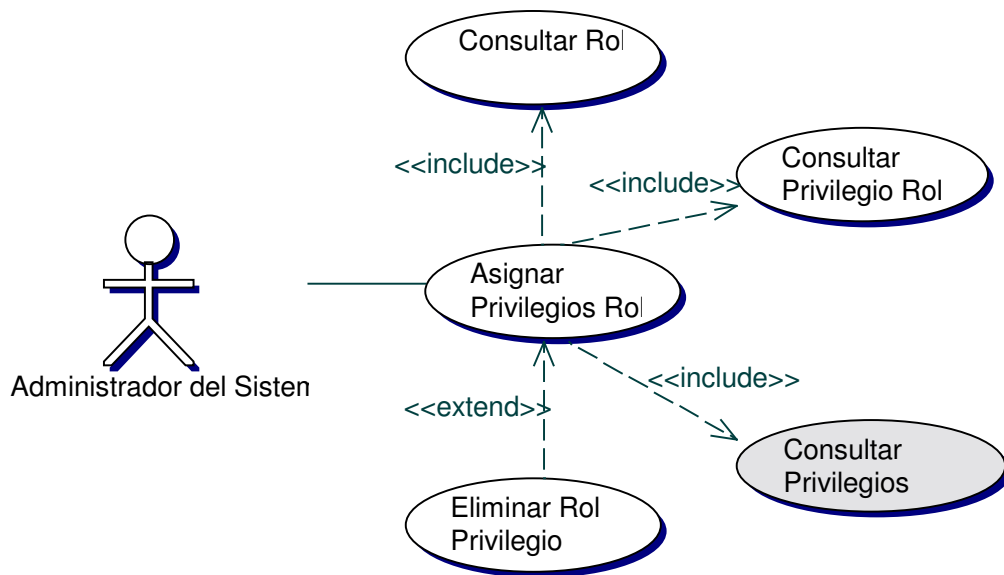
• **Diagrama Eliminar Rol**



- Iteración Eliminar Rol

<b>IDENTIFICADOR:</b> ARI.4		<b>NOMBRE:</b> Eliminar Rol.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rol Consultar Rol Usuario Consultar Privilegio Rol		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del rol.		<b>SALIDA:</b> Eliminación efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un determinado rol. Se pide confirmación para efectuar el proceso.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción Eliminar rol.	2.	Muestra los roles existentes, permitiendo seleccionar el que se desea eliminar. <b>(b)</b>
3.	Selecciona el rol a eliminar.	5.	Se pide confirmación para llevar a cabo el proceso.
4.	Selecciona la opción eliminar rol. <b>(a)</b>	7.	Verifica que el rol a eliminar no este involucrado en otros procesos. <b>(c)</b>
6.	Confirma su decisión. <b>(a)</b>	8.	Elimina el rol actualizando la base de datos.
		9.	Muestra un mensaje indicando que el rol se eliminó con éxito.
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.			
<b>(b)</b> No existen roles: El sistema muestra un mensaje de error y cancela el proceso.			
<b>(c)</b> Rol involucrado en otro proceso: El sistema cancela la eliminación del rol, informando al usuario del error.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> Los procesos de creación y actualización se llevan a cabo de forma satisfactoria La información de la base de datos es consistente. El rol a eliminar no debe estar involucrado en otros procesos.			
<b>Postcondición:</b> Eliminación satisfactoria de los roles.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar los roles de forma consistente. Se garantiza la persistencia y consistencia de la información, ya que los roles eliminados no afectarán el normal funcionamiento del sistema.			

- Diagrama Asignar Privilegios Rol

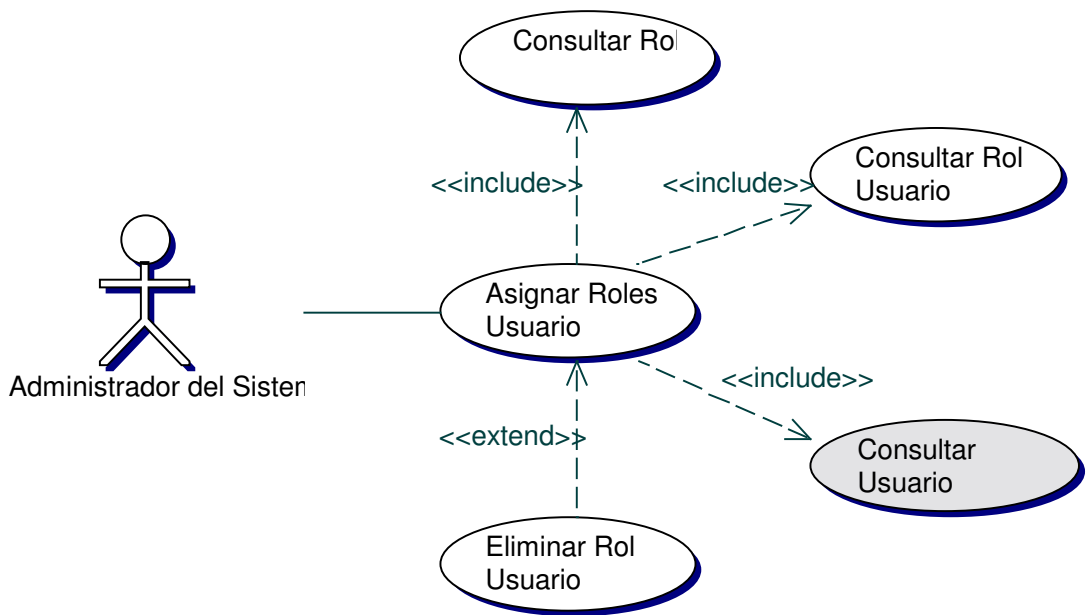


• Iteración Asignar Privilegios Rol

<b>IDENTIFICADOR:</b> ARI.5		<b>NOMBRE:</b> Asignar Privilegios Rol.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rol Consultar Privilegio Rol Consultar Privilegios Eliminar Rol Privilegio		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del rol, código del privilegio		<b>SALIDA:</b> Privilegio asignado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo asignar, consultar o eliminar los privilegios de un determinado rol.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	Selecciona la opción Asignar privilegios rol.	2.	Despliega una pantalla presentando los roles existentes.
3.	Selecciona el rol al que desea asignar un privilegio.	4.	Muestra los privilegios existentes, indicando cuales tiene asignado actualmente el rol seleccionado.
5.	Selecciona el privilegio.	7.	Verifica si ese privilegio se debe asignar o quitar y actualiza la base de datos con la información suministrada.
6.	Selecciona la opción Asignar (a)	8.	Muestra un mensaje indicando que el proceso se realizó con éxito.

<p><b>CAMINOS DE EXCEPCION:</b>  <b>(a)</b> El actor iniciador decide cancelar la asignación: El sistema cancela el proceso.</p> <ul style="list-style-type: none"> <li>El sistema no responde. El proceso no se efectúa.</li> </ul>
<p>Precondición: El proceso de creación y actualización de roles y privilegios se llevan a cabo de forma satisfactoria  La información de la base de datos es consistente.</p> <p>Postcondición: Información consolidada de los privilegios asignados a un rol.</p>
<p><b>CRITERIOS DE ACEPTACIÓN</b>  Se cumple el requerimiento cuando se logra consultar, asignar o quitar privilegios a un determinado rol de forma consistente.</p>

• **Diagrama Asignar Roles Usuario**



• **Iteración Asignar Roles Usuario**

<b>IDENTIFICADOR:</b> AR1.6		<b>NOMBRE:</b> Asignar Roles Usuario.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rol Consultar Rol Usuario Consultar Usuario Eliminar Rol Usuario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación del usuario, Código del rol.		<b>SALIDA:</b> Rol asignado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo asignar, consultar o eliminar los roles de un determinado usuario.			

### CURSO BÁSICO DE EVENTOS

#### Acción de los actores

1. Selecciona la opción Asignar rol.
3. Suministra la información solicitada.
4. Selecciona la opción Mostrar. (a)
7. Selecciona el rol que desea asignar.
8. Confirma su decisión. (a)

#### Respuesta del Sistema

2. Solicita la identificación del usuario al que se le asignarán roles.
5. Verifica que sea un usuario válido.
6. Despliega una pantalla mostrando todos los roles existentes, indicando aquellos que tenga asignado actualmente el usuario. (b)
8. Se pide confirmación para llevar a cabo el proceso.
9. Verifica si ese rol se debe asignar o quitar y actualiza la base de datos con la información suministrada.
10. Muestra un mensaje indicando que el proceso se realizó con éxito.

#### CAMINOS DE EXCEPCIÓN:

- (a) El actor iniciador decide cancelar la asignación: El sistema cancela el proceso.
- (b) La información suministrada es incorrecta: El sistema informa el error dando la oportunidad de ingresar nuevamente la información.
- El sistema no responde. El proceso no se efectúa.

Precondición: El proceso de creación y actualización de roles y usuarios se llevan a cabo de forma satisfactoria  
La información de la base de datos es consistente.

Postcondición: Información consolidada de los roles asignados a un usuario.

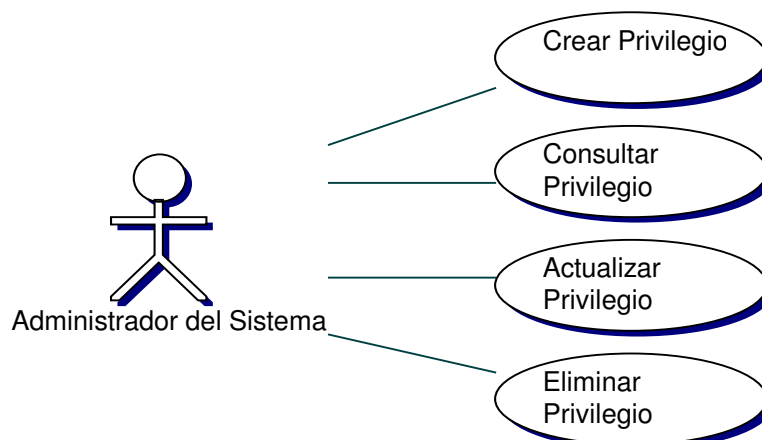
#### CRITERIOS DE ACEPTACIÓN

Se cumple el requerimiento cuando se logra consultar, asignar o quitar roles a un determinado usuario de forma consistente.

## 9.1.31 Administración de Privilegios

### 9.1.31.1 Primera Iteración - Fachada

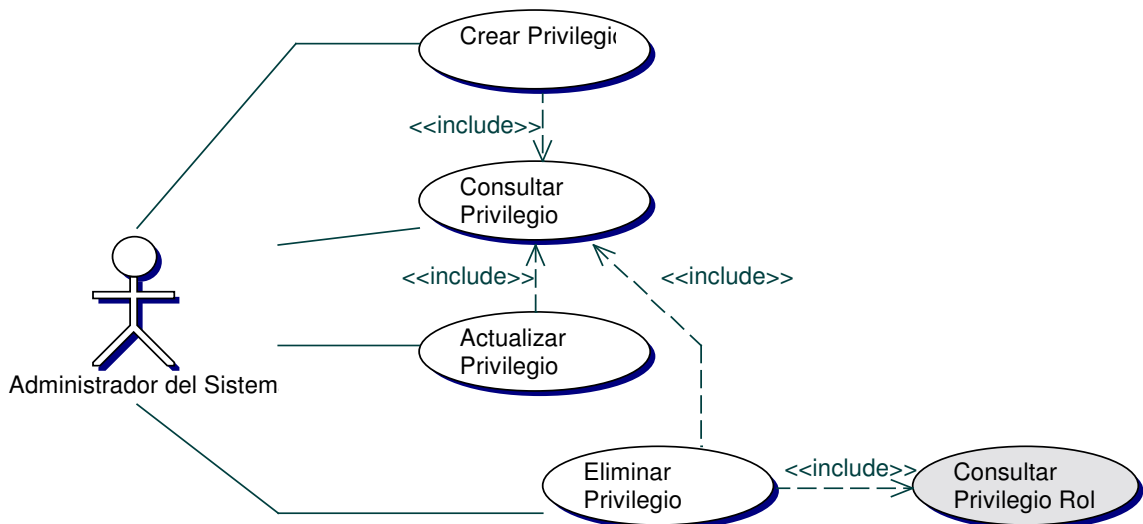
#### • Diagrama



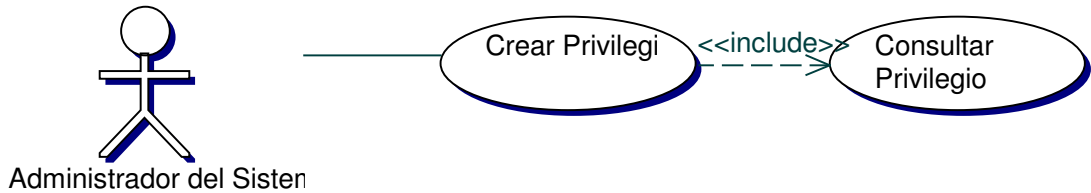
- Iteración

<b>IDENTIFICADOR:</b> API		<b>NOMBRE:</b> Administración de Privilegios	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Crear Privilegio Consultar Privilegio Actualizar Privilegio Eliminar Privilegio Consultar Privilegio Rol		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los privilegios necesarios para un excelente y seguro uso del sistema. El proceso de administración de privilegios se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y eliminación.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: Correcta autenticación del actor iniciador. Postcondición: Administración eficiente de los Privilegios.			
<b>CRITERIOS DE ACEPTACIÓN</b> La información en la base de datos se maneja de forma persistente y consistente. Acceso rápido y oportuno a la información actualizada de cada uno los privilegios.			

### 9.1.31.2 Segunda Iteración – Llenado (Diagrama General)



• Diagrama Crear Privilegio



• Iteración Crear Privilegio

<b>IDENTIFICADOR:</b> API.1		<b>NOMBRE:</b> Crear Privilegio	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Privilegio		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, descripción, url.		<b>SALIDA:</b> Privilegio Registrado o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo crear los privilegios necesarios para la buena gestión del sistema. El código será generado automáticamente, no pueden existir descripción duplicadas, la url hace referencia a la dirección en la cual se encuentra el privilegio a asignar.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Crear privilegio.		2. Despliega una pantalla con los privilegios existentes, permitiendo ingresar la descripción y la url del nuevo privilegio.	
3. Ingresa la información solicitada.		5. Verifica que la descripción y url digitadas no se repitan. (b)	
4. Selecciona la opción Crear. (a)		6. Crea el privilegio actualizando la base de datos.	
		7. Muestra un mensaje indicando que el registro se realizó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El usuario decide no efectuar el proceso: El sistema cancela el proceso			
(b) El dato suministrado ya existe: El sistema muestra un mensaje de error y cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: Ingreso completo de los datos requeridos.			
Postcondición: Registro eficiente de los diferentes privilegios.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Almacenamiento de la información en forma ordenada y clasificada.			
Se cumple el requerimiento cuando se logra crear un privilegio de forma consistente y persistente.			

- Diagrama Consultar Privilegio



- Iteración Consultar Privilegio

<b>IDENTIFICADOR:</b> AP1.2		<b>NOMBRE:</b> Consultar Privilegio	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Listado de los privilegios existentes	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los privilegios existentes. La información desplegada incluirá únicamente la descripción y la url.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. Selecciona la opción Consultar privilegios. (a)		2. Se despliega el resultado de la consulta.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso			
• El sistema no responde. El proceso no se efectúa.			
<i>Precondición: La información de la base de datos es consistente.</i>			
<i>Postcondición: Información consolidada de los privilegios existentes.</i>			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los privilegios existentes de forma veraz y oportuna.			

- Diagrama Actualizar Privilegio

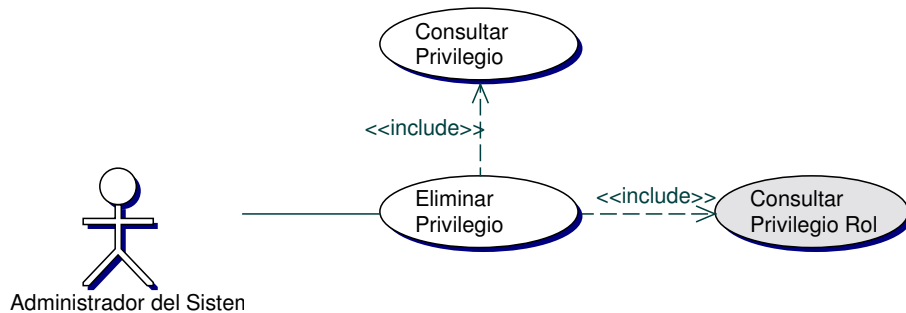




- Iteración Actualizar Privilegio

<b>IDENTIFICADOR:</b> AP1.3		<b>NOMBRE:</b> Actualizar Privilegio	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Privilegio		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código, descripción, url.		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar la descripción y la url de un determinado privilegio.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Actualizar privilegio.		2. Despliega una pantalla mostrando todos los privilegios existentes, permitiendo ingresar la nueva descripción y url.	
3. Suministra la información solicitada.		5. Solicita confirmación.	
4. Selecciona la opción actualizar.		7. Verifica que la descripción y url suministrada no se repita. (b)	
6. Confirma su decisión (a)		8. Actualiza el privilegio modificando la base de datos con la información suministrada.	
		9. Muestra un mensaje indicando que el proceso se realizó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.			
(b) La información suministrada es incorrecta: El sistema muestra un mensaje de error y cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: Los procesos de creación se llevan a cabo de forma satisfactoria. La información de la base de datos es consistente.			
Postcondición: Eficiente actualización de los privilegios.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar la descripción y url de un determinado privilegio de forma veraz y oportuna.			
La información de la base de datos se almacena de forma persistente y consistente.			

- Diagrama Eliminar Privilegio



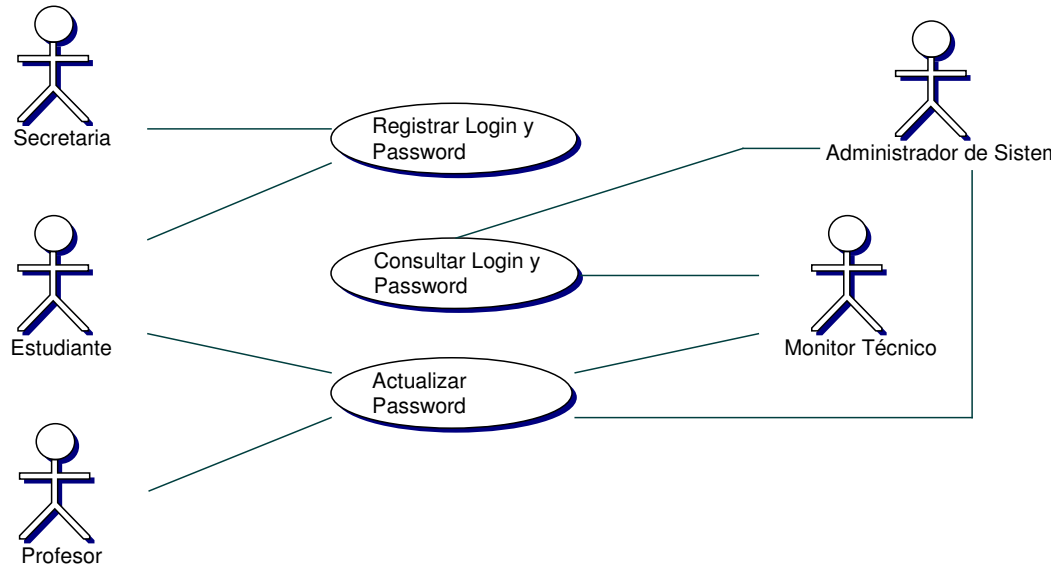
- Iteración Eliminar Privilegio

<b>IDENTIFICADOR:</b> AP1.4		<b>NOMBRE:</b> Eliminar Privilegio.	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Privilegio Consultar Privilegio Rol		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del Privilegio.		<b>SALIDA:</b> Eliminación efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un determinado privilegio. Se pide confirmación para efectuar el proceso.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Eliminar privilegio.		2. Muestra los privilegios existentes, permitiendo seleccionar el que se desea eliminar. <b>(b)</b>	
3. Selecciona el privilegio a eliminar.		5. Se pide confirmación para llevar a cabo el proceso.	
4. Selecciona la opción eliminar privilegio.		7. Verifica que el privilegio a eliminar no este involucrado en otros procesos. <b>(c)</b>	
6. Confirma su decisión. <b>(a)</b>		8. Elimina el privilegio actualizando la base de datos.	
		9. Muestra un mensaje indicando que el privilegio se eliminó con éxito.	
<b>CAMINOS DE EXCEPCION:</b>			
<b>(a)</b> El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso.			
<b>(b)</b> No existen privilegios: El sistema muestra un mensaje de error y cancela el proceso.			
<b>(c)</b> Privilegio involucrado en otro proceso: El sistema cancela la eliminación del privilegio, informando al usuario del error.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> Los procesos de creación y actualización se llevan a cabo de forma satisfactoria La información de la base de datos es consistente. El privilegio a eliminar no debe estar involucrado en otros procesos.			
<b>Postcondición:</b> Eliminación satisfactoria de los privilegios.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar los privilegios de forma consistente. Se garantiza la persistencia y consistencia de la información, ya que los privilegios eliminados no afectarán el normal funcionamiento del sistema.			

## 9.1.32 Administración Login Password

### 9.1.32.1 Primera Iteración - Fachada

• Diagrama



• Iteración

<b>IDENTIFICADOR:</b> ALP1		<b>NOMBRE:</b> Administración de Login y Password	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Registrar Login y Password Consultar Login y Password Actualizar Password Consultar Usuario Consultar Estudiante Consultar Empleado Consultar Profesor Consultar Médico		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema, Profesor, Monitor Técnico del Aula de Informática, Secretaria, Estudiante	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los datos correspondientes al login y password de los usuarios que se registran en el sistema. El proceso de administración se desarrollará teniendo en cuenta las operaciones de registro, consulta, y actualización.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
<b>Precondición:</b> Correcta autenticación del actor iniciador. El actor debe encontrarse registrado en la base de datos como usuario del sistema.			

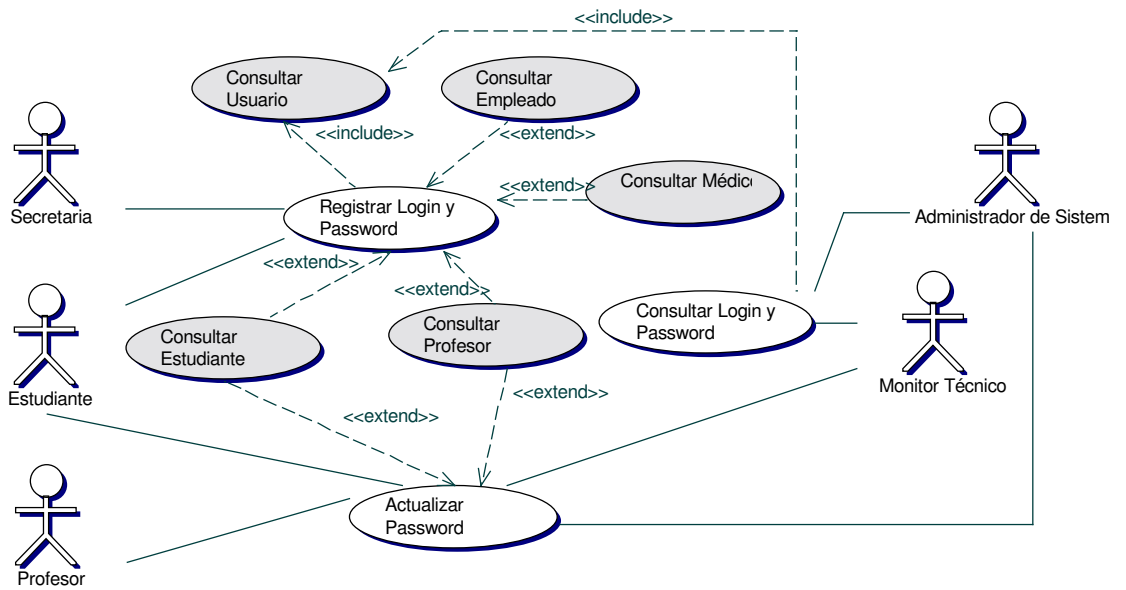
Postcondición: Administración eficiente de login y password.

**CRITERIOS DE ACEPTACIÓN**

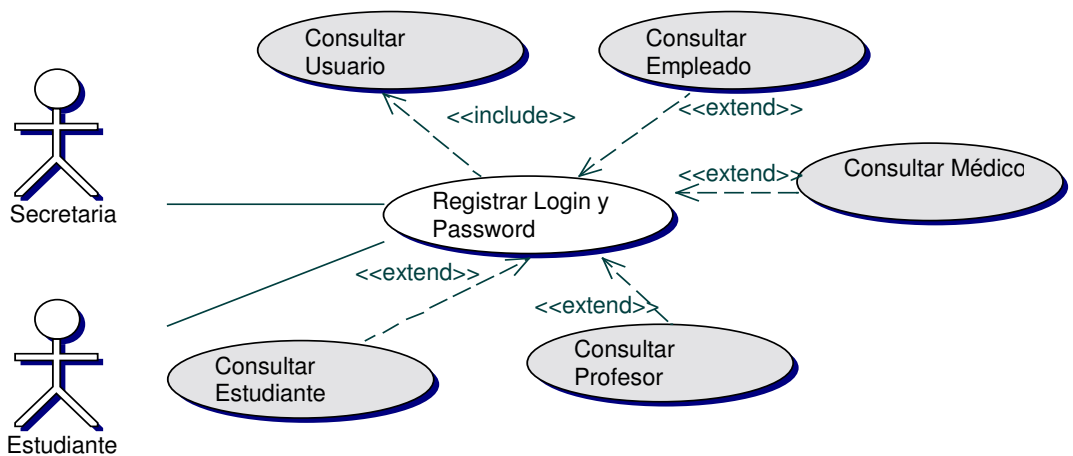
Permite llevar una administración sistemática de los datos de autenticación de usuarios del Sistema.

La información en la base de datos se maneja de forma persistente y consistente.

**9.1.32.2 Segunda Iteración – Llenado (Diagrama General)**



• **Diagrama Registrar Login y Password**

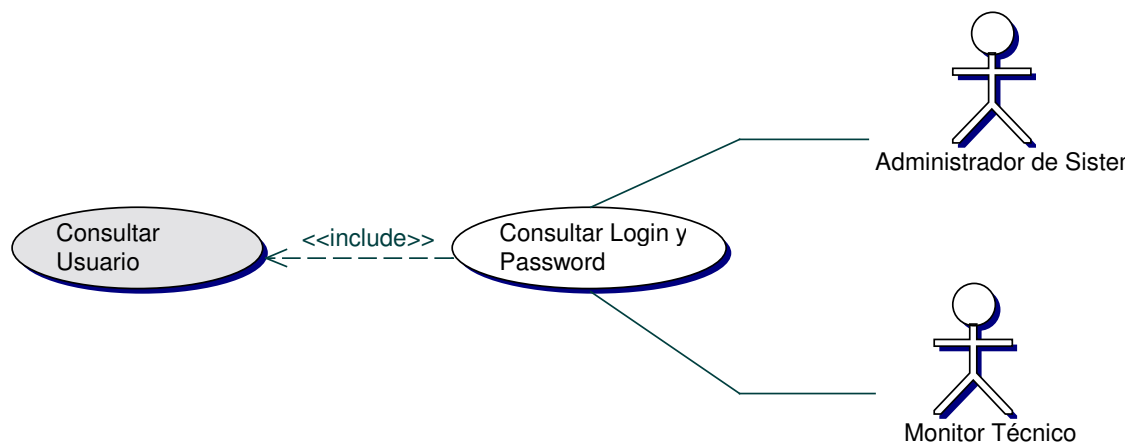


- Iteración Registrar Login y Password

<b>IDENTIFICADOR:</b> AU.1		<b>NOMBRE:</b> Registrar Login y Password	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Usuario Consultar Estudiante Consultar Empleado Consultar Profesor Consultar Médico		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria, Estudiante	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, login, password y confirmación		<b>SALIDA:</b> Datos Registrados o Proceso Cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los datos de autenticación de los usuarios para que puedan hacer uso de los diferentes servicios ofrecidos por el sistema. El registro se hace a través de un identificador que permitirá verificar si el usuario se encuentra registrado en la base de datos de usuarios de la universidad y de esta forma acceder a determinar su login y su password.			
<b>OBSERVACIONES:</b> El actor podrá registrarse una sola vez como usuario del sistema.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Usuario: Secretaria</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Registrar Login y Password.		2. Solicita la identificación del usuario a registrar.	
3. Ingresa la información solicitada.		5. Verifica que sea un usuario vinculado a la universidad, además debe ser del personal operativo, profesor o médico. (b)	
4. Selecciona la opción Mostrar. (a)		6. Despliega una pantalla mostrando el nombre completo y solicita los datos necesarios.	
7. Suministra la información solicitada.		9. Verifica que el login digitado no se repita. (c)	
8. Selecciona la opción Aceptar. (a)		10. Registra los datos actualizando la base de datos.	
		11. Muestra un mensaje indicando que el registro se realizó con éxito.	
<b>Usuario: Estudiante</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Registrar Login y Password.		2. Solicita la identificación, el login, el password y la confirmación del password del usuario a registrar.	
3. Ingresa la información solicitada.		5. Verifica que sea un usuario vinculado a la universidad, además debe ser estudiante. (b)	
4. Selecciona la opción Aceptar. (a)		6. Verifica que el login digitado no se repita. (c)	
		7. Registra los datos actualizando la base de datos.	
		8. Muestra un mensaje indicando que el registro se realizó con éxito.	

<b>CAMINOS DE EXCEPCION:</b> (a) El usuario decide no efectuar el proceso: El sistema cancela el proceso (b) La información suministrada es incorrecta: EL sistema informa al usuario y cancela el proceso (c) Login duplicado: El sistema le informa al usuario y cancela el proceso • El sistema no responde. El proceso no se efectúa.
Precondición: Ingreso completo de los datos requeridos. Consistencia en la información de la base de datos.
Postcondición: Registro eficiente de los usuarios que acceden al sistema.
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra registrar de forma consistente y persistente los datos de autenticación de los diferentes usuarios para acceder al sistema.

• **Diagrama Consultar Login y Password**



• **Iteración Consultar Login y Password**

<b>IDENTIFICADOR:</b> ALP1.2		<b>NOMBRE:</b> Consultar Login y Password	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Usuario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador de Inventario y Monitor Técnico del Aula de Informática	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación		<b>SALIDA:</b> Consulta efectuada o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los datos correspondientes al login y password de un determinado usuario a través de su identificador. Los usuarios podrán solicitar la consulta de su login y password al monitor en caso de ser estudiantes o al administrador del sistema en caso de ser del personal operativo, docentes o médicos.			

**CURSO BÁSICO DE EVENTOS**

**Usuario: Administrador del Sistema**

- Acción de los actores**
1. Selecciona la opción Consultar Login y Password
  3. Suministra la información solicitada.
  4. Selecciona la opción consultar. **(a)**

- Respuesta del Sistema**
2. Solicita la identificación del usuario al que se consultará su login y password
  5. Despliega una pantalla mostrando los datos de autenticación correspondientes al usuario. **(b)**

**Usuario: Monitor Técnico del Aula de Informática**

- Acción de los actores**
1. Selecciona la opción Consultar Login y Password
  3. Suministra la información solicitada.
  4. Selecciona la opción consultar. **(a)**

- Respuesta del Sistema**
2. Solicita la identificación del usuario al que se consultará su login y password
  5. Verifica que el usuario sea un estudiante **(c)**
  6. Despliega una pantalla mostrando los datos de autenticación correspondientes al usuario. **(b)**

**CAMINOS DE EXCEPCION:**

- (a)** El actor iniciador decide cancelar la consulta: El sistema cancela el proceso
- (b)** El identificador suministrado es incorrecto: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente el código.
- (c)** El usuario Solicitante no es estudiante: El sistema informa al usuario y cancela el proceso de consulta.
- El sistema no responde. El proceso no se efectúa.

**Precondición:** Consistencia en la información de la base de datos.

El actor debe estar registrado como usuario del sistema.

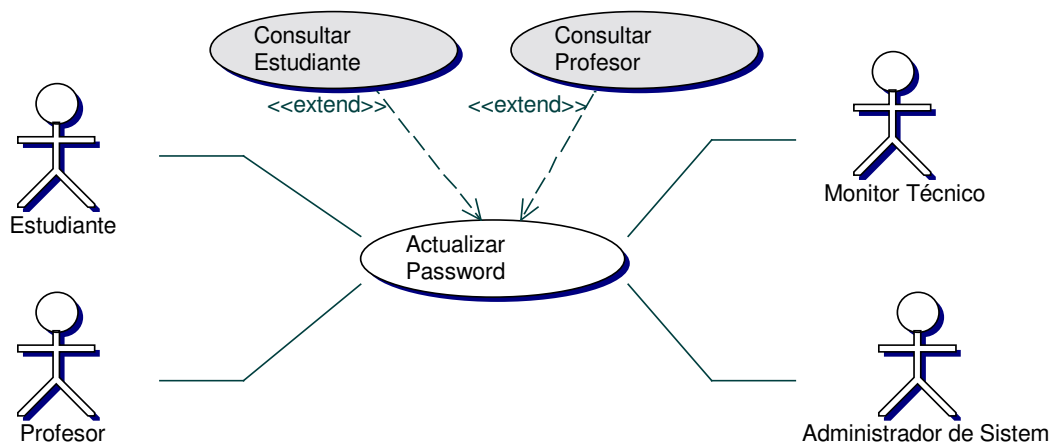
**Postcondición:** Información de los datos correspondientes al login y password del usuario.

**CRITERIOS DE ACEPTACIÓN**

Se cumple el requerimiento cuando se logra consultar de forma eficiente y oportuna los datos de autenticación de los usuarios que acceden al sistema.

Fácil recuperación de los datos de login y password de un determinado usuario, en caso de olvido.

• **Diagrama Actualizar Password**



- Iteración Actualizar Password

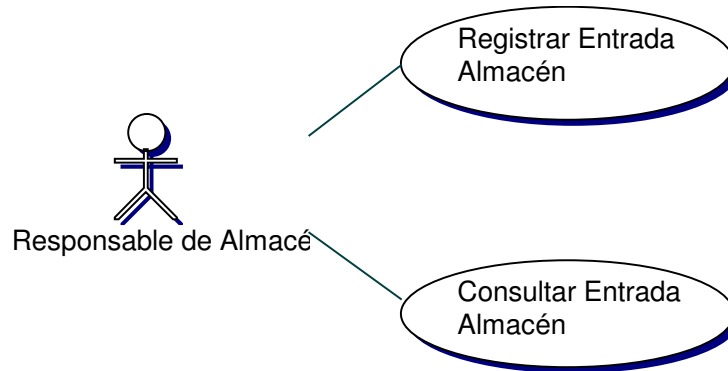
<b>IDENTIFICADOR:</b> ALP1.3		<b>NOMBRE:</b> Actualizar Password	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Profesor Consultar Estudiante		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Administrador del Sistema, Profesor, Monitor Técnico del Aula de Informática, Estudiante	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Identificación, password actual, nuevo password		<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar el password de un determinado usuario. Para efectuar el proceso se solicita el password actual, el nuevo password y la confirmación del mismo.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. Selecciona la opción Actualizar login y password.		2. Solicita la identificación del usuario al se va a actualizar los datos.	
3. Suministra la información solicitada.		5. Verifica que sea usuario válido.	
4. Selecciona la opción mostrar. (a)		6. Muestra los datos como son nombre completo y login. Solicita el password actual, el nuevo password y su correspondiente confirmación. (b)	
7. Ingresar la información solicitada. (a)		8. Modificando la base de datos con la información suministrada.	
		9. Muestra un mensaje indicando que los datos de autenticación se actualizaron con éxito	
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso.			
(b) No es un usuario válido: El sistema muestra un mensaje de error dando la oportunidad de ingresar nuevamente la identificación.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> El actor debe estar registrado como usuario del sistema. La información de la base de datos es consistente.			
<b>Postcondición:</b> Actualización del password del usuario.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra actualizar de forma eficiente y segura el password de un usuario. La información de la base de datos se almacena de forma persistente y consistente.			

### 9.1.33 Administración de Entradas de Almacén

#### 9.1.33.1 Primera Iteración - Fachada



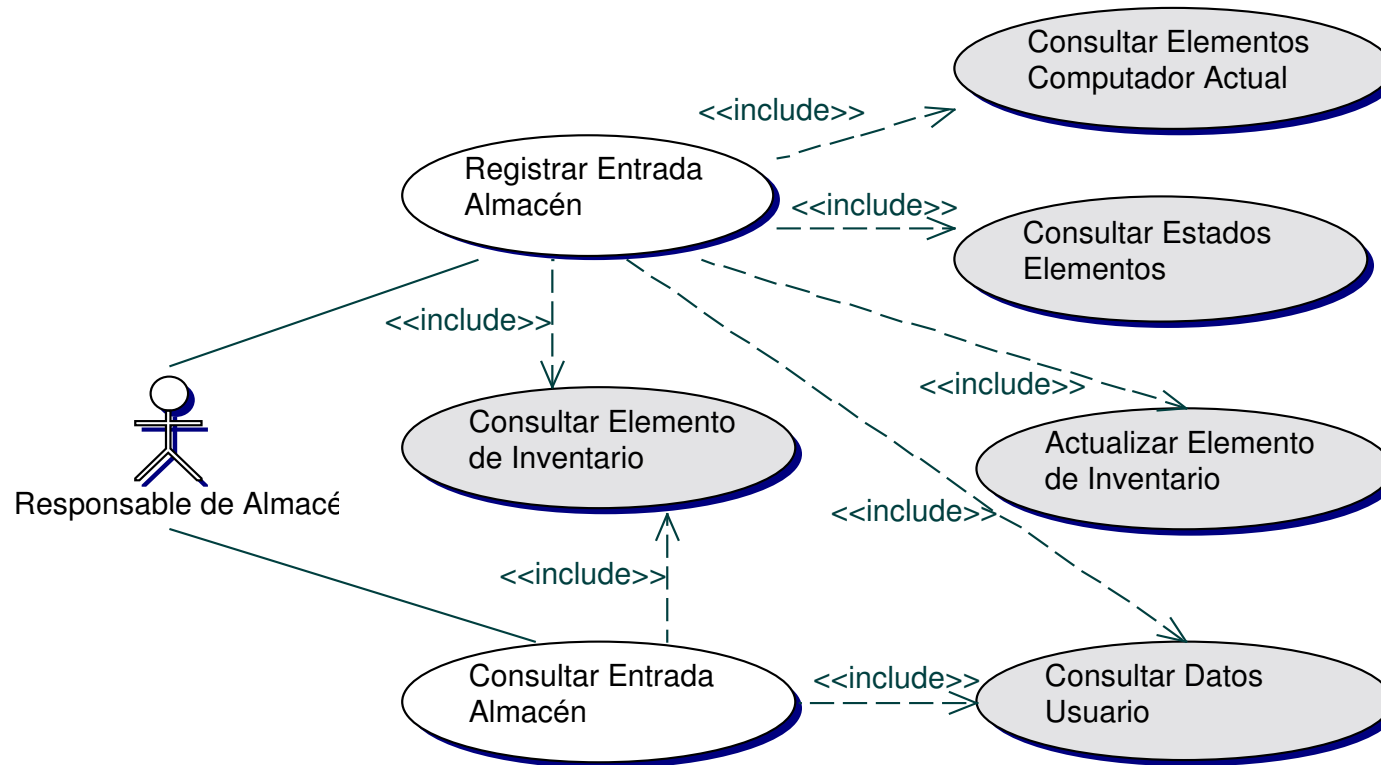
- Diagrama



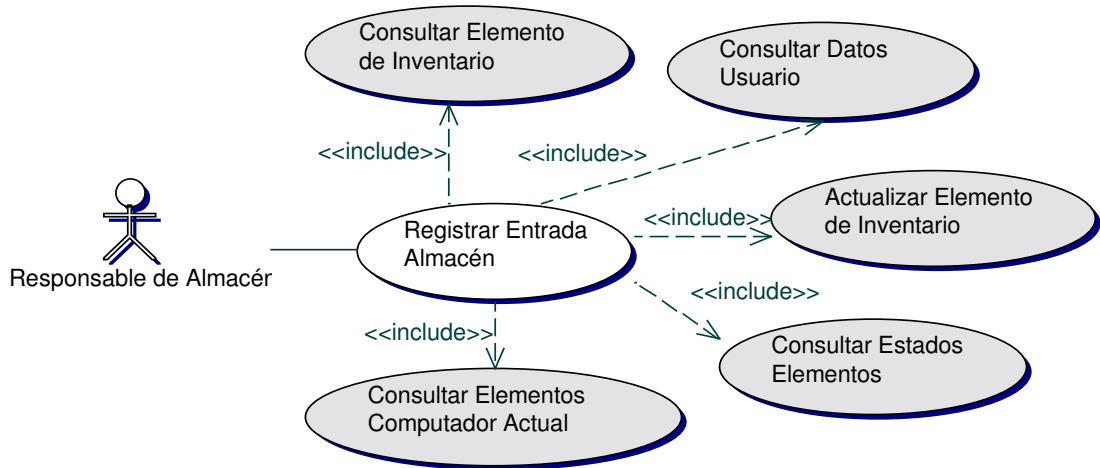
- Iteración

<b>IDENTIFICADOR:</b> AEALM1		<b>NOMBRE:</b> Administración de Entradas de Almacén	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Registrar Entrada Almacén Consultar Entrada Almacén		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador: Responsable de Almacén, Usuario que entrega. Ejecutor: Responsable de Almacén.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las entradas de elementos a la sección de almacén permitiendo llevar un control de los elementos bajo la responsabilidad del actor ejecutor. El proceso de administración se desarrollará teniendo en cuenta las operaciones de registro y consulta.			
<b>OBSERVACIONES:</b>			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema. Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de las entradas de elementos de almacén.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de las entradas de los elementos a la sección de almacén. La información en la base de datos se maneja de forma persistente y consistente.			

### 9.1.33.2 Segunda Iteración – Llenado (Diagrama General)



• **Diagrama Registrar Entrada Almacén**

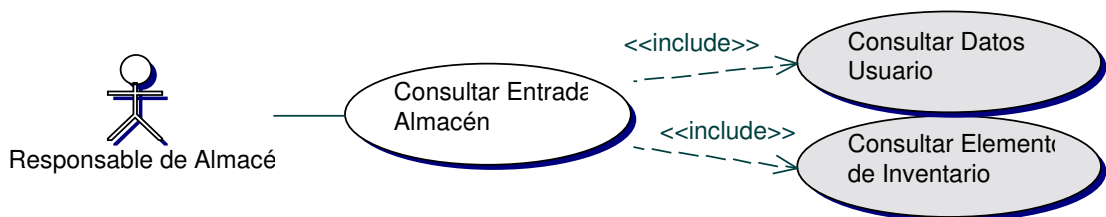


• **Iteración Registrar Entrada Almacén**

<b>IDENTIFICADOR:</b> AEALM1.1		<b>NOMBRE:</b> Registrar Entrada Almacén	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Elemento de Inventario Consultar Datos Usuario Actualizar Elemento de Inventario Consultar Estados Elementos Consultar Elementos Computador Actual		<b>CRITICO?</b> Sí
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Usuario que Entrega Ejecutor: Responsable de Almacén	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código Elemento, fecha, estado, identificación entrega, identificación recibe.		<b>SALIDA:</b> Registro efectuado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar la entrada de un elemento a la sección de almacén. Al ingresar un elemento se registrará la entrada con la fecha actual, se determinará el estado de entrada del elemento, el usuario responsable de almacén que recibe el elemento y el usuario que hace entrega del elemento.			
<b>OBSERVACIONES:</b> El usuario que recibe el elemento se obtendrá automáticamente de acuerdo a los datos del usuario que tiene abierta la sesión en el sistema. Al ingresar un elemento a Almacén automáticamente se actualizará el estado del elemento en inventario de acuerdo al estado de entrada del elemento seleccionado al efectuar el proceso. De igual manera se actualizará la ubicación del elemento a la sección de Almacén. Para poder efectuar la entrada del elemento, este no debe encontrarse asignado a un computador ni encontrarse en la ubicación de almacén.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor iniciador hace entrega al actor ejecutor de un elemento para su ingreso a almacén.	
2. El actor ejecutor selecciona la opción registrar entrada.	3. Solicita el código del elemento a ingresar.
4. El actor ejecutor digita el código del elemento. <b>(a)</b>	5. Despliega los datos del elemento como son: descripción, marca, serial, ubicación actual, estado actual, fecha de entrada y solicita el estado de entrada del elemento y la identificación del usuario que entrega.
6. El actor ejecutor suministra los datos solicitados.	
7. El actor ejecutor da la orden para registrar la entrada. <b>(a)</b>	8. Verifica que el usuario que entrega sea un usuario valido de la universidad. <b>(b)</b>
	9. Verifica que el elemento no se encuentre asignado a un computador. <b>(c)</b>
	10. Verifica que el elemento no se encuentre en la ubicación de Almacén. <b>(d)</b>
	11. Registra la entrada actualizando la base de datos con la información suministrada.
	12. Muestra un mensaje indicando que el registro se realizó con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor iniciador decide cancelar el registro de entrada: El sistema cancela el proceso.	
<b>(b)</b> El Usuario que entrega no es un usuario valido de la Universidad: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
<b>(c)</b> El elemento se encuentra asignado como un componente de un computador: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
<b>(d)</b> El elemento ya se encuentra en la ubicación de almacén: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: El elemento de entrada no debe encontrarse asignado a un computador. El elemento debe encontrarse en una ubicación diferente de Almacén.	
Postcondición: Registro eficiente de entrada de elementos a Almacén.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra registrar la entrada de elementos a la sección de Almacén de forma veraz y oportuna.	
Se lleva un control eficiente de los elementos de entrada, su estado y las personas responsables de hacer el registro.	

• **Diagrama Consultar Entrada Almacén**



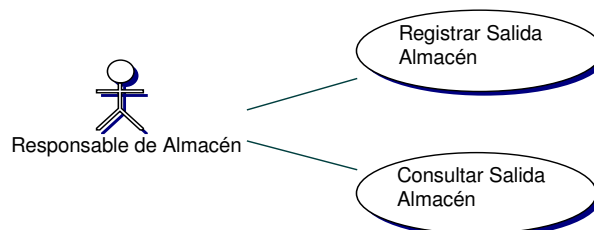
- Iteración Consultar Entrada Almacén

<b>IDENTIFICADOR:</b> AEALM1.2		<b>NOMBRE:</b> Consultar Entrada Almacén	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos de usuario Consultar Elemento de Inventario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Responsable de Almacén	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código elemento, fecha entrada.		<b>SALIDA:</b> Listado de Entradas de Elementos	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los datos de los registros de entrada de elementos a la sección de almacén.			
<b>OBSERVACIONES:</b> La consulta podrá realizarse por el código del elemento o la fecha de entrada.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar entradas de Almacén.	2. Despliega un menú presentando las opciones para su consulta (por código y por fecha.		
3. El actor selecciona la opción para su consulta. (a)	4. Solicita los datos necesarios para efectuar la consulta.		
5. El actor suministra los datos solicitados. (a)	6. Se despliega los datos de los registros de entrada.		
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la consulta: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>Precondición:</b> La información de la base de datos es consistente.			
<b>Postcondición:</b> Información oportuna de los registros de entrada de elementos a la sección de almacén.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los registros de entrada de elementos a la sección de almacén de forma veraz y oportuna, permitiendo verificar el manejo de elementos de los usuarios involucrados en el proceso.			

### 9.1.34 Administración de Salidas de Almacén

#### 9.1.34.1 Primera Iteración - Fachada

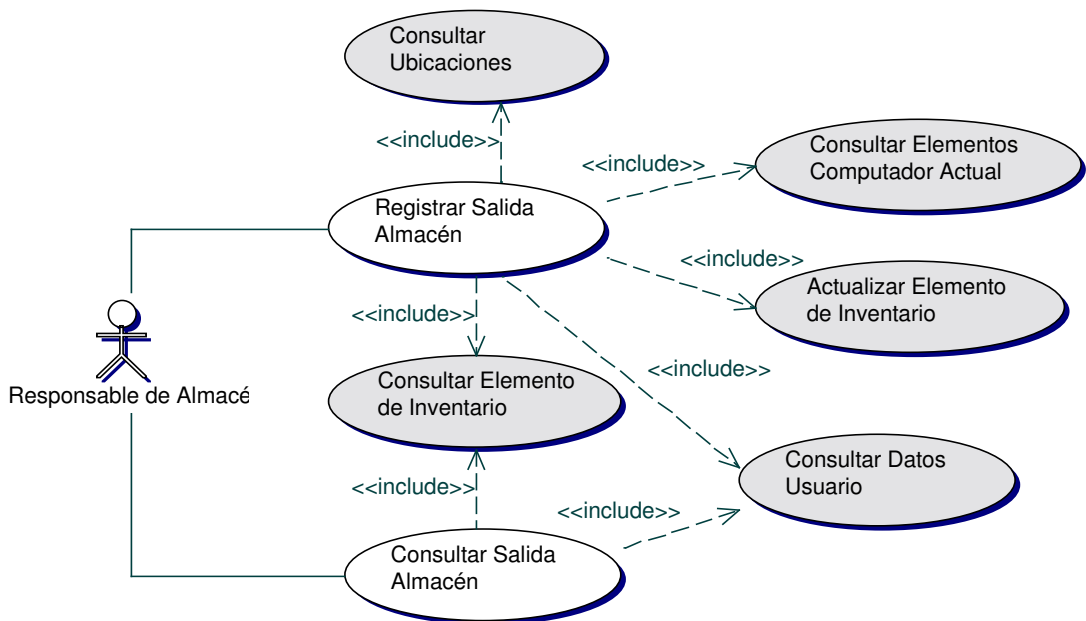
- Diagrama



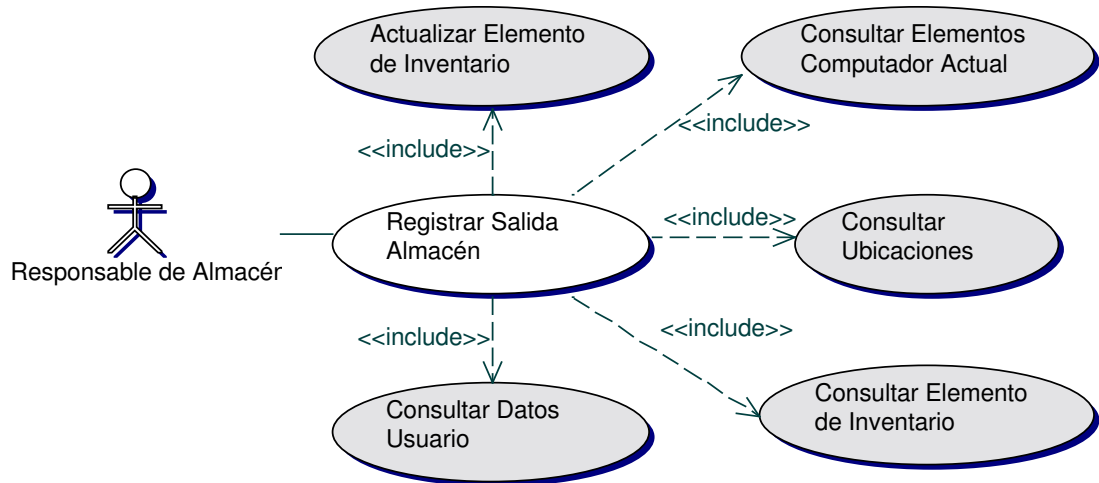
- Iteración

<b>IDENTIFICADOR:</b> ASALM1		<b>NOMBRE:</b> Administración de Salidas de Almacén
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Registrar Salida Almacén Consultar Salida Almacén	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador: Responsable de Almacén, Usuario que recibe. Ejecutor: Responsable de Almacén.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar las salidas de elementos de la sección de almacén llevando un control de a quien se hace entrega de un determinado elemento. El proceso de administración se desarrollará teniendo en cuenta las operaciones de registro y consulta.		
<b>OBSERVACIONES:</b>		
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.		
Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de las salidas de elementos de Almacén.		
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de las salidas de los elementos de la sección de almacén controlando los responsables de la entrega y recepción de elementos. La información en la base de datos se maneja de forma persistente y consistente.		

### 9.1.34.2 Segunda Iteración – Llenado (Diagrama General)



- **Diagrama Registrar Salida Almacén**



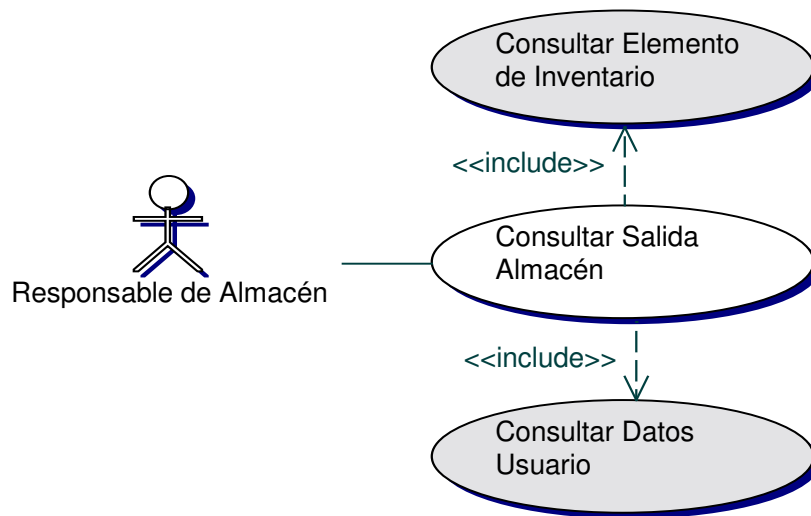
- **Iteración Registrar Salida Almacén**

<b>IDENTIFICADOR:</b> ASALM1.1		<b>NOMBRE:</b> Registrar Salida Almacén	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Elemento de Inventario Consultar Datos Usuario Actualizar Elemento de Inventario Consultar Ubicaciones Consultar Elementos Computador Actual		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador: Usuario que Recibe Ejecutor: Responsable de Almacén	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código Elemento, fecha, ubicación destino, identificación entrega, identificación recibe.		<b>SALIDA:</b> Registro efectuado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar la salida de un elemento de la sección de almacén. La salida de un elemento se registrará con la fecha actual y el estado actual del elemento, el usuario responsable de almacén que entrega el elemento y el usuario que lo recibe.			
<b>OBSERVACIONES:</b> El usuario que entrega el elemento se obtendrá automáticamente de acuerdo a los datos del usuario que tiene abierta la sesión en el sistema. La salida del elemento actualizará la ubicación del elemento de acuerdo al dato seleccionado en el registro de salida. Para poder efectuar la salida del elemento, este no debe encontrarse asignado a un computador y encontrarse en la ubicación de almacén. Por seguridad para efectuar el registro de salida se solicitará el password del usuario quien recibe el elemento, ya que él se hará responsable del elemento que se le entrega.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor iniciador solicita al actor ejecutor la salida de un elemento de almacén.	
2. El actor ejecutor selecciona la opción registrar salida.	3. Solicita el código del elemento.
4. El actor ejecutor digita el código del elemento. <b>(a)</b>	5. Despliega los datos del elemento como son: descripción, marca, serial, ubicación actual, estado de salida, fecha de salida y solicita la nueva ubicación del elemento, la identificación del usuario que recibe y su password.
6. El actor ejecutor suministra los datos solicitados.	
7. El actor ejecutor da la orden para registrar la salida. <b>(a)</b>	8. Verifica que el usuario que recibe sea un usuario valido de la universidad. <b>(b)</b>
	9. Verifica que el password digitado sea correcto. <b>(c)</b>
	9. Verifica que el elemento no se encuentre asignado a un computador. <b>(d)</b>
	10. Verifica que el elemento se encuentre en la ubicación de Almacén. <b>(e)</b>
	11. Registra la entrada actualizando la base de datos con la información suministrada.
	12. Muestra un mensaje indicando que el registro se realizó con éxito.
<b>CAMINOS DE EXCEPCION:</b>	
<b>(a)</b> El actor iniciador decide cancelar el registro de entrada: El sistema cancela el proceso.	
<b>(b)</b> El Usuario que recibe no es un usuario valido de la Universidad: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
<b>(c)</b> El password del usuario que recibe es incorrecto: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
<b>(d)</b> El elemento se encuentra asignado como un componente de un computador: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
<b>(e)</b> El elemento no se encuentra en la ubicación de almacén: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: El elemento de entrada no debe encontrarse asignado a un computador. El elemento debe encontrarse en la ubicación de Almacén.	
Postcondición: Registro eficiente de salida de elementos de Almacén.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra registrar la salida de elementos de la sección de Almacén de forma veraz y oportuna.	
Se lleva un control eficiente de los elementos de salida, su nueva ubicación y las personas responsables de hacer el registro.	

- **Diagrama Consultar Salida Almacén**





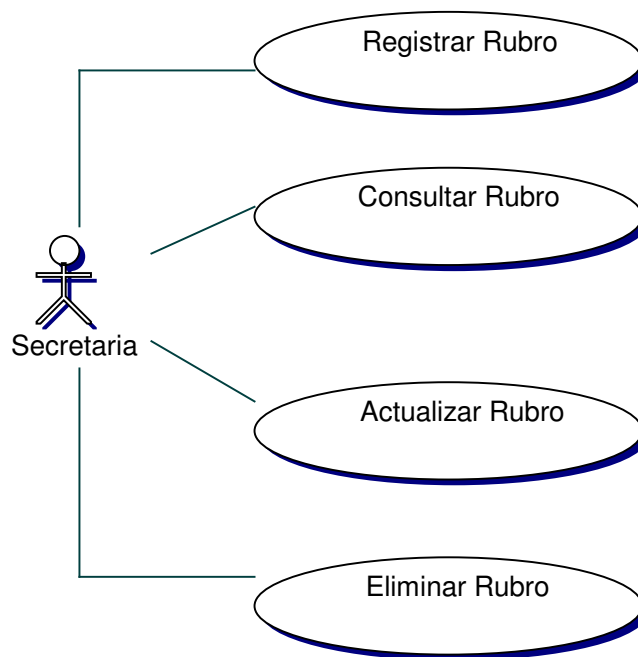
• Iteración Consultar Salida Almacén

<b>IDENTIFICADOR:</b> ASALM1.2		<b>NOMBRE:</b> Consultar Salida Almacén	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Datos Usuario Consultar Elemento de Inventario		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Responsable de Almacén	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código elemento, fecha salida.		<b>SALIDA:</b> Listado de Salidas de Elementos	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los datos de los registros de salida de elementos de la sección de almacén.			
<b>OBSERVACIONES:</b> La consulta podrá realizarse por el código del elemento o la fecha de salida.			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1. EL actor selecciona la opción consultar salidas de Almacén.	2. Despliega un menú presentando las opciones para su consulta (por código y por fecha.	3. El actor selecciona la opción para su consulta. (a)	4. Solicita los datos necesarios para efectuar la consulta.
5. El actor suministra los datos solicitados. (a)	6. Se despliega los datos de los registros de salida.		
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor decide cancelar la consulta: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Información oportuna de los registros de salida de elementos de la sección de almacén.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar los registros de salida de elementos de la sección de almacén de forma veraz y oportuna, permitiendo verificar el manejo de elementos de los usuarios involucrados en el proceso.			

### 9.1.35 Administración de Rubros

#### 9.1.35.1 Primera Iteración - Fachada

- Diagrama



- Iteración

<b>IDENTIFICADOR:</b> ARI		<b>NOMBRE:</b> Administración de Rubros	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Registrar Rubro Consultar Rubro Actualizar Rubro Eliminar Rubro		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los rubros manejados en caja menor. El proceso de administración se desarrollará teniendo en cuenta las operaciones de creación, consulta, actualización y cancelación.			
<b>TRIGGERS:</b>			

*Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.*

*Precondición: El actor ha iniciado una sesión en la aplicación.*

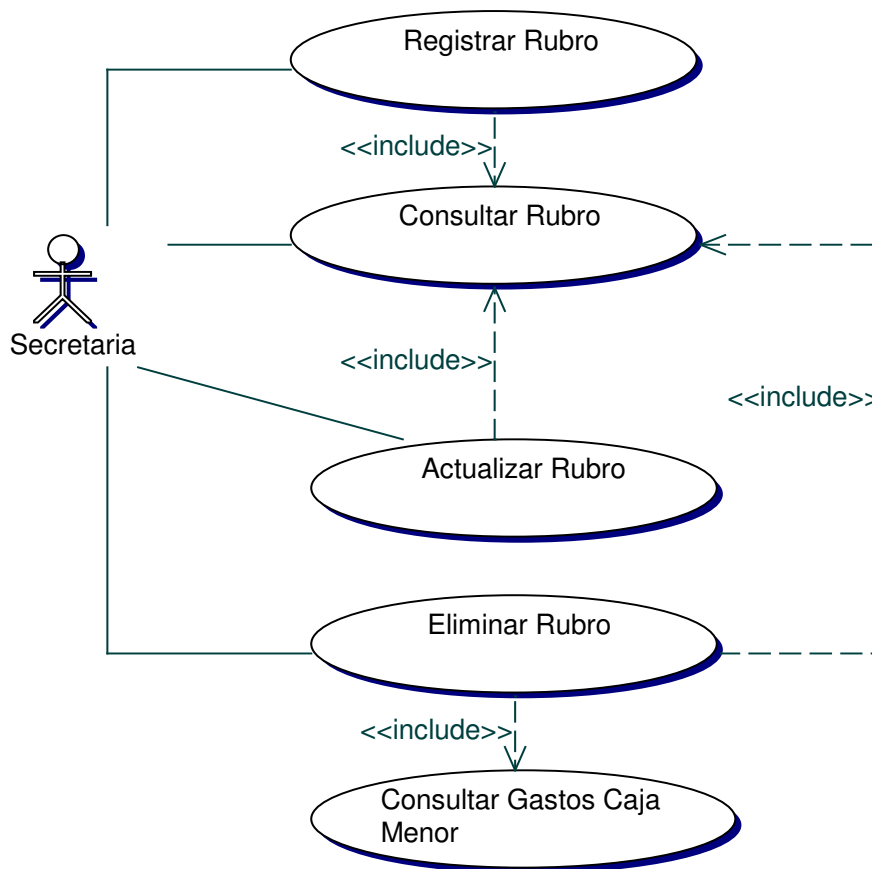
*Postcondición: Administración eficiente de los rubros.*

**CRITERIOS DE ACEPTACIÓN**

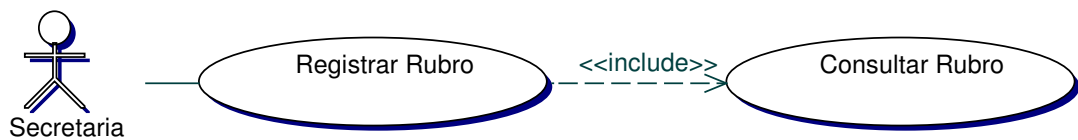
*Permite llevar un control y una administración sistemática de los rubros manejados en caja menor.*

*La información en la base de datos se maneja de forma persistente y consistente.*

### 9.1.35.2 Segunda Iteración – Llenado (Diagrama General)



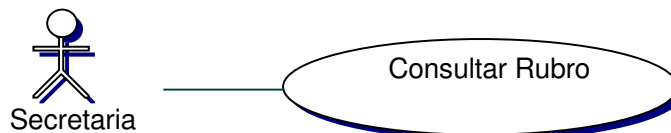
- **Diagrama Registrar Rubro**



- Iteración Registrar Rubro

<b>IDENTIFICADOR:</b> ARI.1		<b>NOMBRE:</b> Registrar Rubro	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rubro		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del Rubro y Descripción del Rubro		<b>SALIDA:</b> Rubro registrado o proceso no efectuado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar los rubros manejados en caja menor.			
<b>CURSO BÁSICO DE EVENTOS</b> <i>Creación efectuada por el actor iniciador</i>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. El actor iniciador selecciona la opción registrar rubros.	2. Solicita el código y la descripción del rubro		
3. Suministra los datos solicitados			
4. El actor iniciador da la orden para que se registre el rubro. (a)	5. Verifica que el rubro no se encuentre registrado(b)		
	6. Registra el rubro actualizando la base de datos con la información suministrada.		
	7. Muestra un mensaje indicando que el registro se realizó con éxito.		
<b>CAMINOS DE EXCEPCION:</b>			
(a) El usuario decide no registrar el rubro: El sistema cancela el proceso			
(b) El rubro se encuentra registrado: El sistema le indica este suceso a través de un mensaje y cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Registro eficiente de los rubros manejado en caja menor..			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra registrar un rubro que será manejado en los gastos de caja menor. La información de la base de datos se almacena de forma persistente y consistente.			

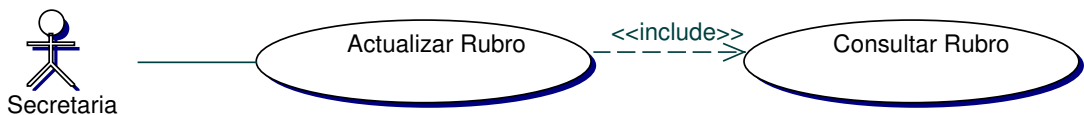
- Diagrama Consultar Rubro



- Iteración Consultar Rubro

<b>IDENTIFICADOR:</b> ARI.2		<b>NOMBRE:</b> Consultar Rubro	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b>		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b>		<b>SALIDA:</b> Listado Rubros existentes	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los rubros manejados en caja menor.			
<b>CURSO BÁSICO DE EVENTOS</b> <i>Consulta efectuada por el monitor</i>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1. EL actor iniciador selecciona la opción consultar Rubros.		2. Despliega un listado con todos los rubros existentes.	
<b>CAMINOS DE EXCEPCION:</b>			
• El sistema no responde. El proceso no se efectúa.			
<i>Precondición: La información de la base de datos es consistente.</i>			
<i>Postcondición: Información oportuna de los rubros manejados en caja menor..</i>			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar la información de los rubros existentes.			

- Diagrama Actualizar Rubro

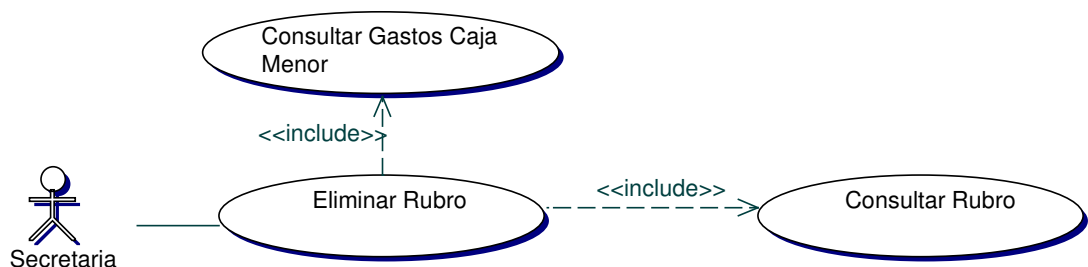


- Iteración Actualizar Rubro

<b>IDENTIFICADOR:</b> ARI.3		<b>NOMBRE:</b> Actualizar Rubro	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rubro		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Código del rubro y Descripción del rubro		<b>SALIDA:</b> Actualización efectuada o cancelada	

<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar un determinado rubro. Se pide confirmación para efectuar el proceso.	
<b>CURSO BÁSICO DE EVENTOS</b> <b>Actualización efectuada por el actor monitor</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. El actor selecciona la opción actualizar rubro	2. Permite seleccionar el rubro a actualizar.
3. Modifica la descripción del rubro(a)	5. Pide que se confirme la ejecución del proceso
4. Da la orden para actualizar los datos (a)	7. Verifica que la nueva descripción del rubro no se encuentre registrada (b).
6. El actor monitor confirma su decisión. (a)	8. Actualiza la información en la base de datos con los datos suministrados.
	9. Despliega un mensaje informando que la actualización se realizó con éxito.
<b>CAMINOS DE EXCEPCION:</b> (a) El actor iniciador decide cancelar la actualización: El sistema cancela el proceso. (b) La descripción del rubro ya se encuentra registrada: El sistema le indica este suceso a través de un mensaje y cancela el proceso. • El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente. Postcondición: Actualización oportuna de los rubros manejados en caja menor	
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra actualizar la información de los rubros existentes. La información de la base de datos se almacena de forma persistente y consistente.	

## • Diagrama Eliminar Rubro



## • Iteración Eliminar Rubro

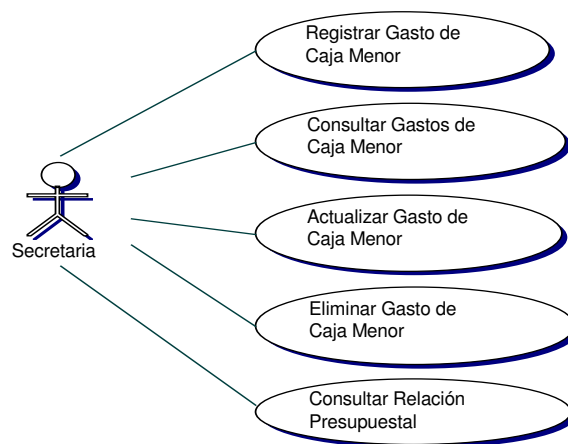
<b>IDENTIFICADOR:</b> AR1.4		<b>NOMBRE:</b> Eliminar Rubros	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rubros Consultar Gastos Caja Menor		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	

<b>ENTRADA:</b> Código del rubro	<b>SALIDA:</b> Proceso efectuado o cancelado
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un determinado rubro. Se pide confirmación para efectuar el proceso.	
<b>CURSO BÁSICO DE EVENTOS</b> <i>Eliminación efectuada por el actor monitor</i>	
<i>Acción de los actores</i>	<i>Respuesta del Sistema</i>
1. El actor selecciona la opción Eliminar Rubro	2. Muestra un listado de los rubros existentes
3. Selecciona el rubro a eliminar	5. Pide confirmación para llevar a cabo el proceso.
4. Da la orden para eliminar el rubro seleccionado. <b>(a)</b>	7. Verifica que el rubro no se encuentre registrado en los gastos de caja menor. <b>(b)</b>
6. Confirma su decisión. <b>(a)</b>	8. Eliminar el rubro actualizando la base de datos.
	9. Muestra un mensaje indicando que el rubro se eliminó con éxito.
<b>CAMINOS DE EXCEPCION:</b> <b>(a)</b> El actor iniciador decide cancelar la eliminación: El sistema cancela el proceso. <b>(b)</b> El rubro se encuentra registrado en los gastos de caja menor. • El sistema no responde. El proceso no se efectúa.	
Precondición: La información de la base de datos es consistente. Postcondición: Eliminación eficiente de los rubros manejados en caja menor.	
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra eliminar los rubros manejados en caja menor de forma consistente.	

## 9.1.36 Administración de Gastos de Caja Menor

### 9.1.36.1 Primera Iteración - Fachada

- Diagrama

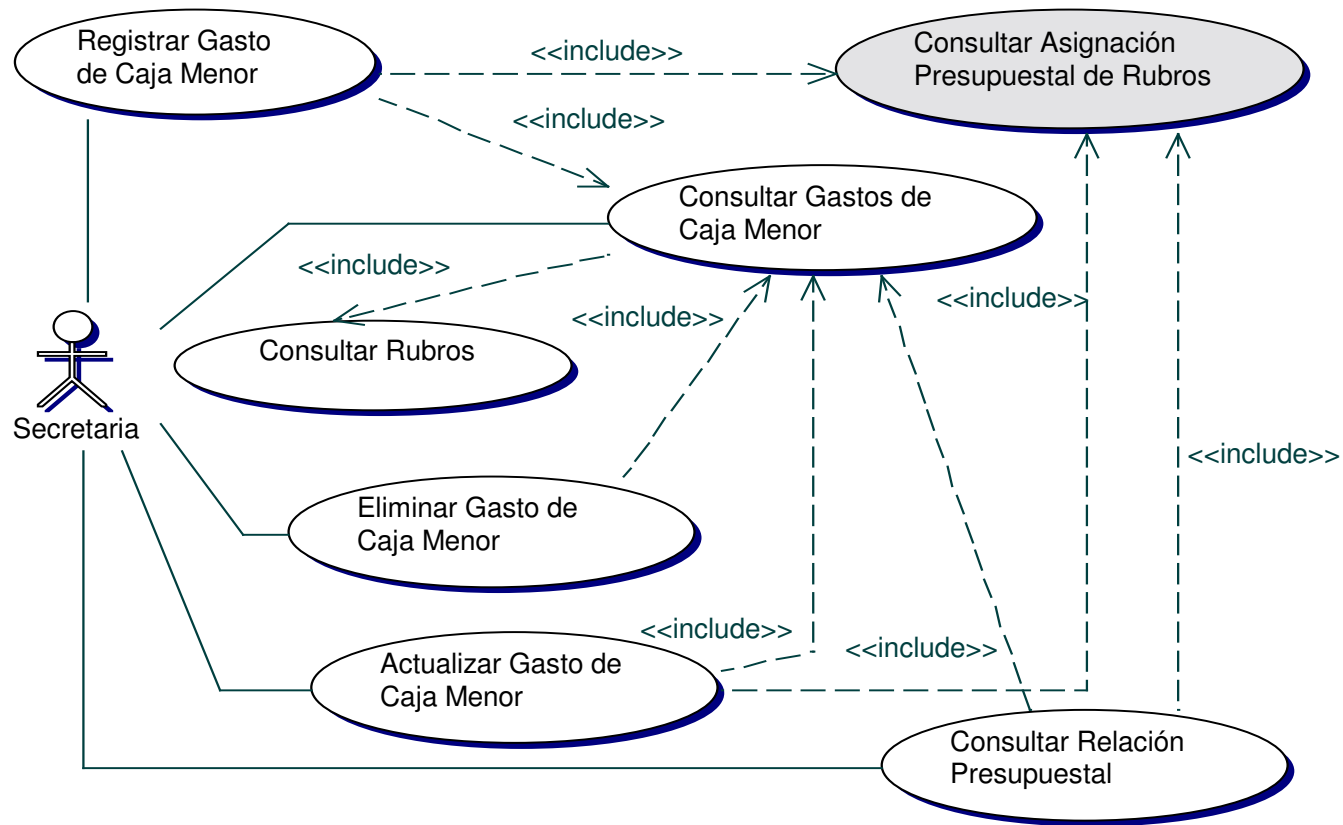


- Iteración

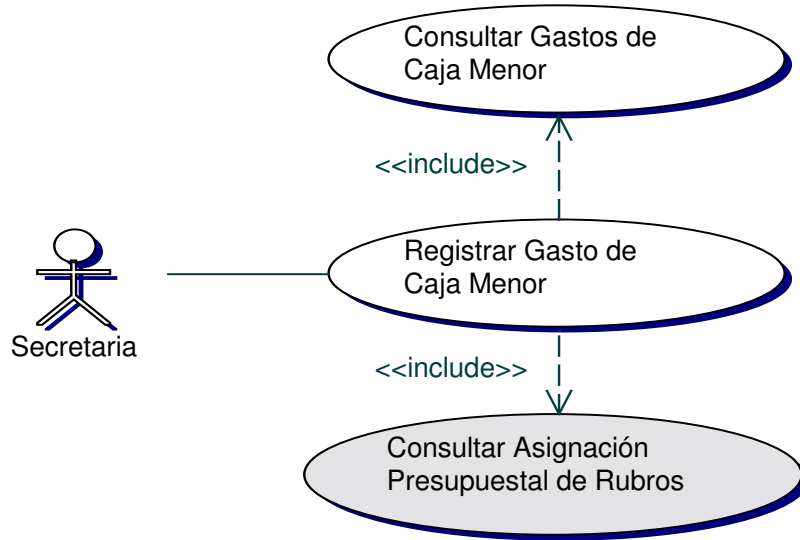
<b>IDENTIFICADOR:</b> AGCM1		<b>NOMBRE:</b> Administración de Gastos de Caja Menor	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Registrar Gasto de Caja Menor Consultar Gastos de Caja Menor Actualizar Gasto de Caja Menor Eliminar Gasto de Caja Menor Consultar Relación Presupuestal		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Fachada	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Datos para efectuar el proceso.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo administrar los gastos de caja menor efectuados en la Universidad. El proceso de administración se desarrollará teniendo en cuenta las operaciones de registro, consulta, actualización y eliminación.			
<b>OBSERVACIONES:</b> La administración de los gastos se hará de acuerdo a un determinado año y mes.			
<b>TRIGGERS:</b> Exceso en el tiempo de respuesta: Este evento se ejecutará para determinar el tiempo límite de espera para una respuesta del sistema.			
Precondición: El actor ha iniciado una sesión en la aplicación. Postcondición: Administración eficiente de los gastos de caja menor.			
<b>CRITERIOS DE ACEPTACIÓN</b> Permite llevar una administración sistemática de los gastos de caja menor llevando un control del presupuesto existente. Permite acceder de forma rápida y oportuna a los datos de gastos de caja menor registrados. La información en la base de datos se maneja de forma persistente y consistente.			



### 9.1.36.2 Segunda Iteración – Llenado (Diagrama General)



- Diagrama Registrar Gasto de Caja Menor

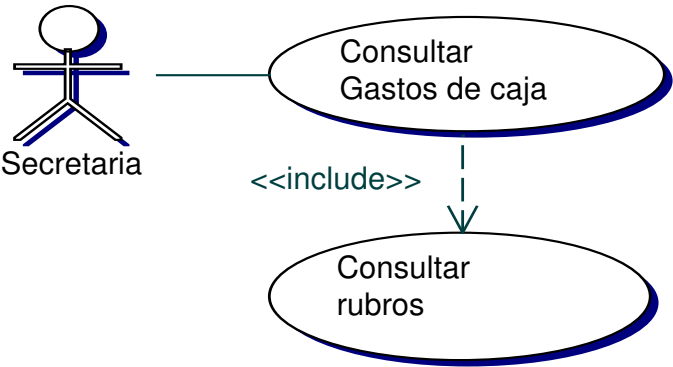


- Iteración Registrar Gasto de Caja Menor

<b>IDENTIFICADOR:</b> AGCM1.1		<b>NOMBRE:</b> Registrar Gasto de Caja Menor	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Gastos de Caja Menor Consultar Asignación Presupuestal de Rubros		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Usuario Autorizado	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Mes, año, fecha, detalle, valor, rubro.		<b>SALIDA:</b> Proceso efectuado o cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo registrar un gasto de caja menor efectuado para un determinado mes y año.			
<b>OBSERVACIONES:</b> Para poder registrar un gasto de caja menor, la suma de todos los gastos de un rubro en ese mes y año no debe sobrepasar el valor del presupuesto que se le haya asignado.			

<b>CURSO BÁSICO DE EVENTOS</b>	
<b>Acción de los actores</b>	<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción registrar gasto de caja menor.	2. Solicita el mes, año, fecha, detalle, valor y rubro del gasto.
3. El actor suministra los datos solicitados.	5. Verifica que el total de gastos efectuado en el mes no sobrepase el presupuesto asignado al rubro. (b)
4. Da la orden para registrar el gasto (a)	6. Registra el gasto actualizando la base de datos con la información suministrada.
	7. Muestra un mensaje indicando que el registro se realizó con éxito
<b>CAMINOS DE EXCEPCION:</b>	
(a) El actor iniciador decide cancelar la creación: El sistema cancela el proceso.	
(b) El total de gastos sobrepasa el presupuesto asignado: El sistema informa el suceso a través de un mensaje y cancela el proceso.	
• El sistema no responde. El proceso no se efectúa.	
Precondición: Debe existir una asignación presupuestal del rubro involucrado en el gasto.	
Postcondición: Registro eficiente de gastos de caja menor.	
<b>CRITERIOS DE ACEPTACIÓN</b>	
Se cumple el requerimiento cuando se logra registrar los gastos de caja menor de forma veraz y oportuna.	
Se lleva un control adecuado de los gastos efectuados de acuerdo al presupuesto asignado a cada rubro.	
La información de la base de datos se almacena de forma persistente y consistente.	

• **Diagrama Consultar Gastos de Caja Menor**

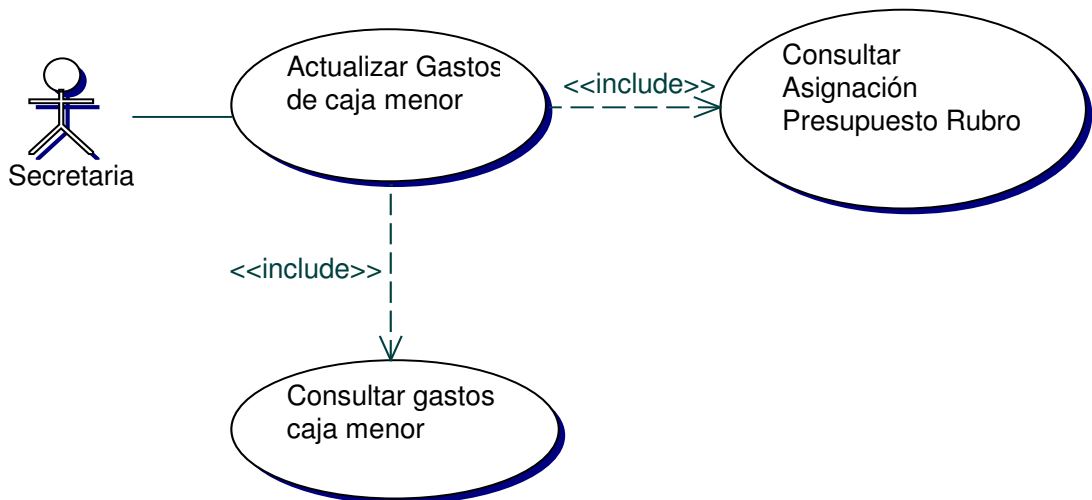


• **Iteración Consultar Gastos de Caja Menor**

<b>IDENTIFICADOR:</b> AGCM1.2		<b>NOMBRE:</b> Consultar Gastos de Caja Menor
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Rubros	<b>CRITICO?:</b> Si

<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Mes, año, rubro	<b>SALIDA:</b> Listado de Gastos	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar los gastos de caja menor efectuados..		
<b>OBSERVACIONES:</b>		
<b>CURSO BÁSICO DE EVENTOS</b> <i>Consulta efectuada por el usuario autorizado</i>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. La secretaria selecciona la opción consultar gastos.	2. Solicita el mes, año y rubro del gasto a consultar.	
3. El usuario suministra los datos solicitados. (a)	4. Despliega el listado de gastos.	
<b>CAMINOS DE EXCEPCION:</b> (b) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso. • El sistema no responde. El proceso no se efectúa.		
Precondición: La información de la base de datos es consistente. Debe haber una asignación presupuestal para poder registrar rubros y consultarlos.		
Postcondición: Información veraz y oportuna de los gastos de caja menor realizados.		
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra consultar los gastos de caja menor de forma veraz y oportuna. Se lleva un control eficiente de los gastos efectuados.		

• **Diagrama Actualizar Gastos de Caja Menor**

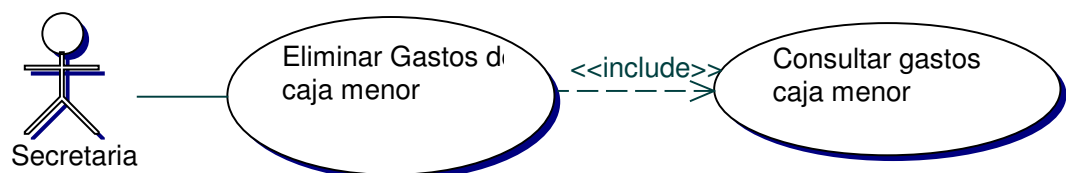


• **Iteración Actualizar Gastos de Caja Menor**

<b>IDENTIFICADOR:</b> AGCM1.3	<b>NOMBRE:</b> Actualizar Gastos de Caja Menor
----------------------------------	---

<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Gastos Caja Menor Consultar Asignación Presupuesto Rubro	<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.
<b>ENTRADA:</b> Detalle Gasto, fecha, valor, rubro.	<b>SALIDA:</b> Actualización efectuada o cancelada	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo actualizar los datos de un gasto de caja menor. Se pide confirmación para efectuar el proceso.		
<b>OBSERVACIONES:</b> El gasto no debe sobrepasar el valor de presupuesto asignado al rubro		
<b>CURSO BÁSICO DE EVENTOS</b>		
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>
1. EL actor selecciona la opción actualizar gasto caja menor.		2. Permite seleccionar la fecha, el año y el rubro del gasto a actualizar.
3. Selecciona los datos.		4. Muestra un listado de los gastos existentes.
5. Selecciona el gasto a actualizar		6. Muestra los datos del gasto y las opciones de actualización.
7. Actualiza los datos deseados		
8. Selecciona la opción actualizar novedad. (a)		8. Se pide confirmación para llevar a cabo el proceso.
8. El actor confirma su decisión. (a)		9. Verifica que valor del gasto no sobrepase el presupuesto asignado al rubro. (b)
		10. Actualiza el gasto modificando la base de datos con la información suministrada.
		11. Muestra un mensaje indicando que el gasto se actualizó con éxito
<b>CAMINOS DE EXCEPCION:</b>		
(b) El actor decide cancelar la actualización: El sistema cancela el proceso.		
(c) El valor del gasto es mayor al presupuesto existente: El sistema cancela el proceso, informando el suceso.		
• El sistema no responde. El proceso no se efectúa.		
<b>TRIGGERS:</b> Generar registro de actualización: Este evento se ejecutará para generar un registro de auditoria que almacene los datos iniciales del gasto, los datos actualizados del gasto, la fecha y el responsable de la actualización.		
<b>CRITERIOS DE ACEPTACIÓN</b> Se cumple el requerimiento cuando se logra actualizar los gastos de forma eficiente y oportuna. Se controla adecuadamente la utilización de los recursos financieros.		

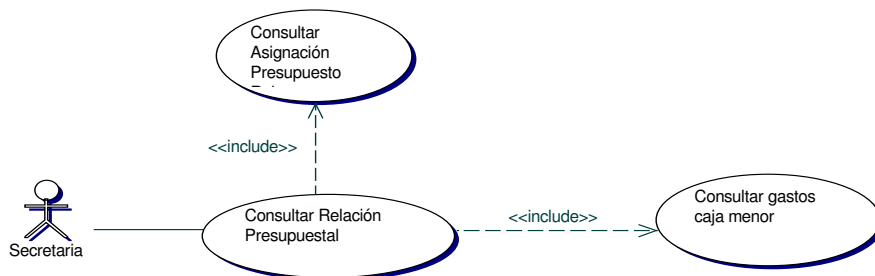
- **Diagrama Eliminar Gastos de Caja Menor**



- Iteración Eliminar Gastos de Caja Menor

<b>IDENTIFICADOR:</b> AGCM1.4		<b>NOMBRE:</b> Eliminar Gastos de Caja Menor	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Gastos de Caja Menor		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria.	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Gasto a eliminar		<b>SALIDA:</b> Gasto eliminado o proceso cancelado	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo eliminar un determinado gasto de caja menor. Se pide confirmación para efectuar el proceso.			
<b>OBSERVACIONES:</b>			
<b>CURSO BÁSICO DE EVENTOS</b>			
<b>Acción de los actores</b>		<b>Respuesta del Sistema</b>	
1.	EL actor selecciona la opción eliminar gasto de caja menor.	2.	Permite seleccionar la fecha, el año y el rubro del gasto a actualizar.
3.	Selecciona los datos.	4.	Muestra un listado de los gastos existentes.
5.	Selecciona el gasto a eliminar	7.	Se pide confirmación para llevar a cabo el proceso.
6.	Selecciona la opción eliminar gasto. (a)	9.	Elimina el gasto modificando la base de datos con la información suministrada.
8.	El actor confirma su decisión. (a)	10.	Muestra un mensaje indicando que el gasto se eliminó con éxito
<b>CAMINOS DE EXCEPCION:</b>			
(b) El actor decide cancelar la eliminación: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
<b>TRIGGERS:</b>			
Generar registro de eliminación: Este evento se ejecutará para generar un registro de auditoria que almacene los datos del gasto eliminado, la fecha y el responsable de la eliminación.			
Precondición: La información de la base de datos es consistente. Debe existir al menos un gasto creado.			
Postcondición: Eliminación eficiente de gastos.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra eliminar los gastos de caja menor de forma consistente.			

- Diagrama Consultar Relación Presupuestal

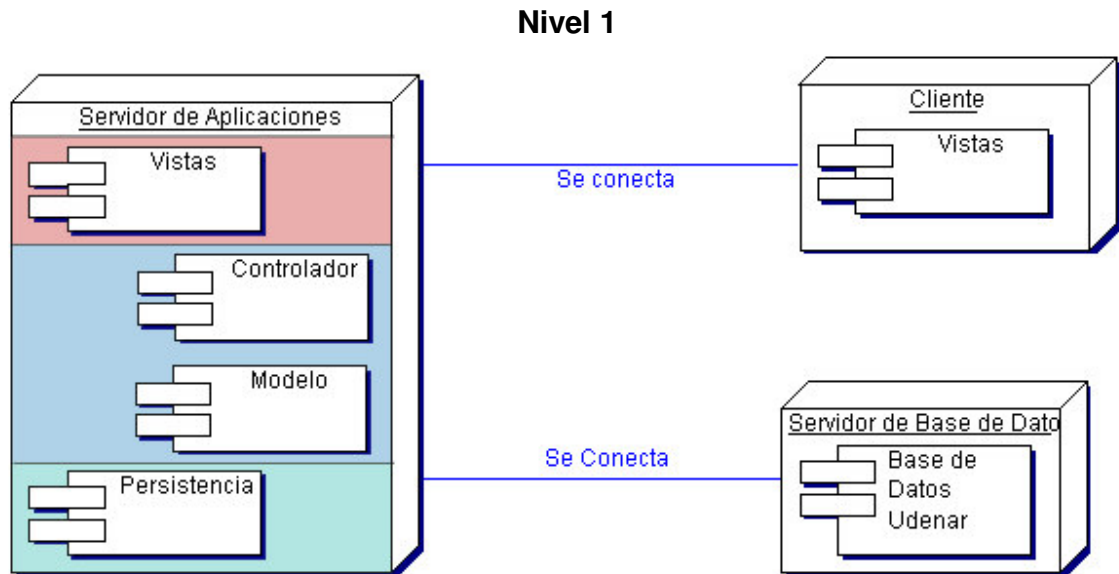


- Iteración Consultar Relación Presupuestal

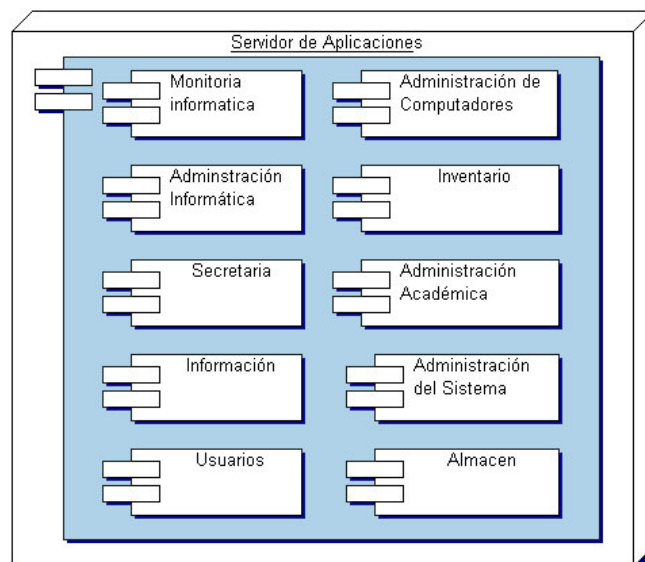
<b>IDENTIFICADOR:</b> AGCM1.5		<b>NOMBRE:</b> Consultar Relación Presupuestal	
<b>TIPO:</b> Necesario	<b>REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA:</b> Consultar Asignación Presupuestal Rubro Consultar Gastos de Caja Menor		<b>CRITICO?</b> Si
<b>PRIORIDAD DE DESARROLLO:</b> Llenado	<b>ACTORES:</b> Iniciador y Ejecutor: Secretaria	<b>DOCUMENTOS DE VISUALIZACION ASOCIADOS:</b> Páginas HTML y JSP.	
<b>ENTRADA:</b> Mes, año.		<b>SALIDA:</b> relación presupuestal	
<b>DESCRIPCIÓN:</b> Se satisface este requerimiento permitiendo consultar la relación presupuestal.			
<b>OBSERVACIONES:</b>			
<b>CURSO BÁSICO DE EVENTOS</b>			
<i>Consulta efectuada por el usuario autorizado</i>			
<i>Acción de los actores</i>		<i>Respuesta del Sistema</i>	
1.	La secretaria selecciona la opción consultar relación presupuestal.	2.	Solicita el mes y el año para la consulta.
3.	El usuario suministra los datos solicitados. (a)	4.	Despliega el total de gastos por cada rubro, el valor presupuestal asignado por cada rubro y el saldo.
<b>CAMINOS DE EXCEPCION:</b>			
(a) El actor iniciador decide cancelar la consulta: El sistema cancela el proceso.			
• El sistema no responde. El proceso no se efectúa.			
Precondición: La información de la base de datos es consistente.			
Postcondición: Información veraz y oportuna de los gastos de caja menor, presupuesto y saldo respectivo.			
<b>CRITERIOS DE ACEPTACIÓN</b>			
Se cumple el requerimiento cuando se logra consultar a relación presupuestal de forma veraz y oportuna.			
Se lleva un control eficiente de los recursos financieros existentes.			

## 10. DISEÑO

### 10.1 DIAGRAMA DE ARQUITECTURA



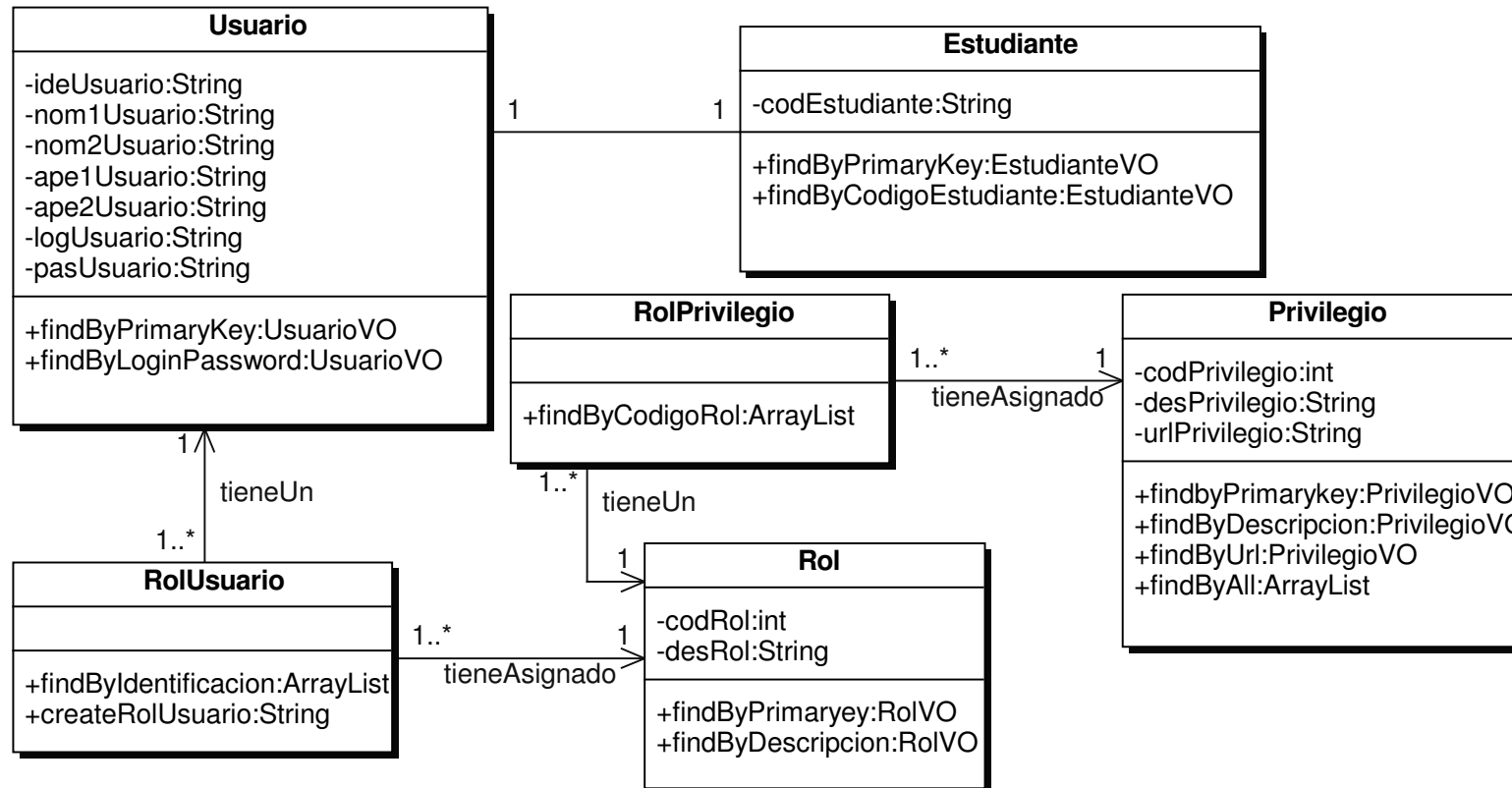
**Nivel 2**



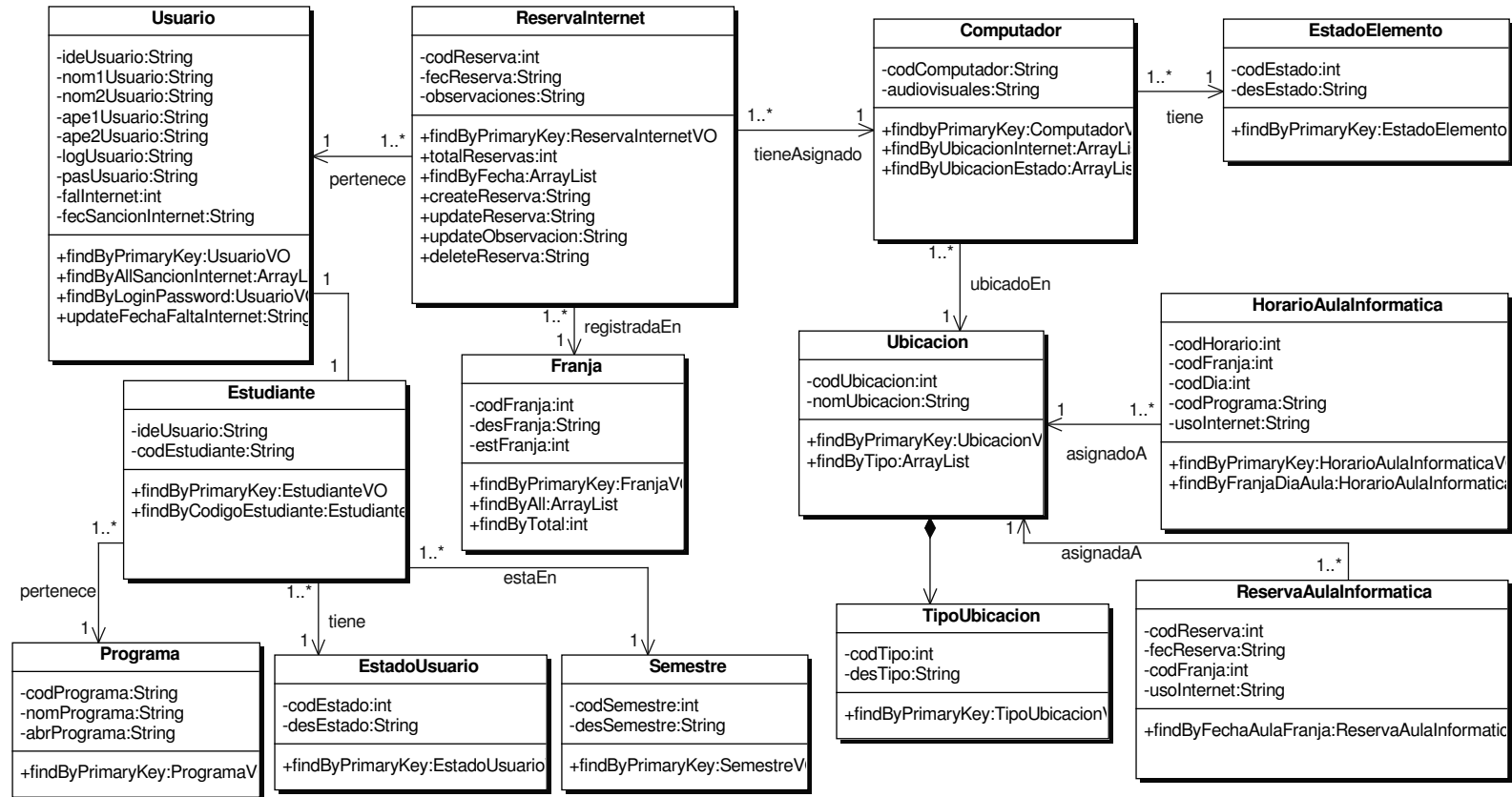


## 10.2 DIAGRAMAS DE CLASES

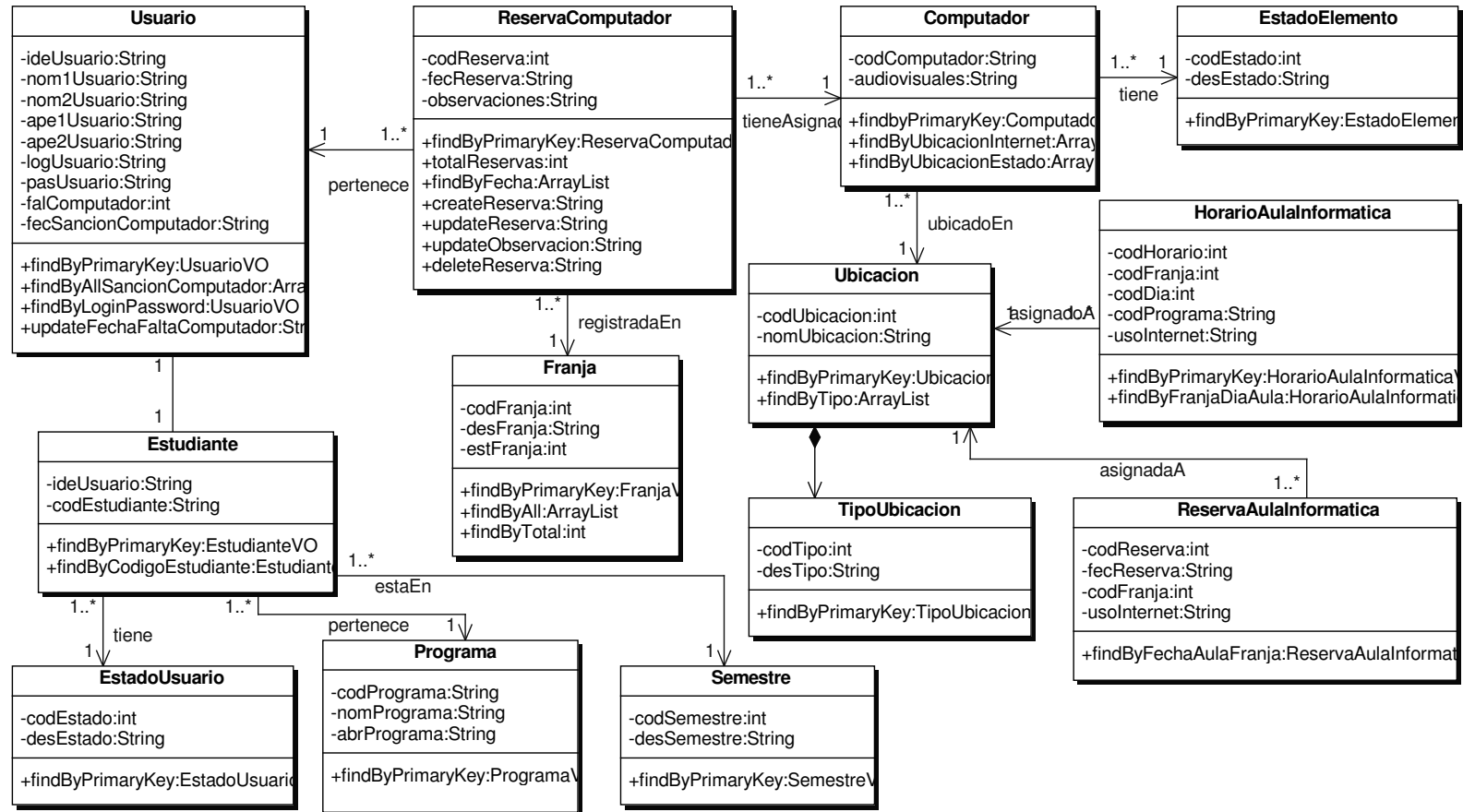
### 10.2.1 Control de Acceso al Sistema



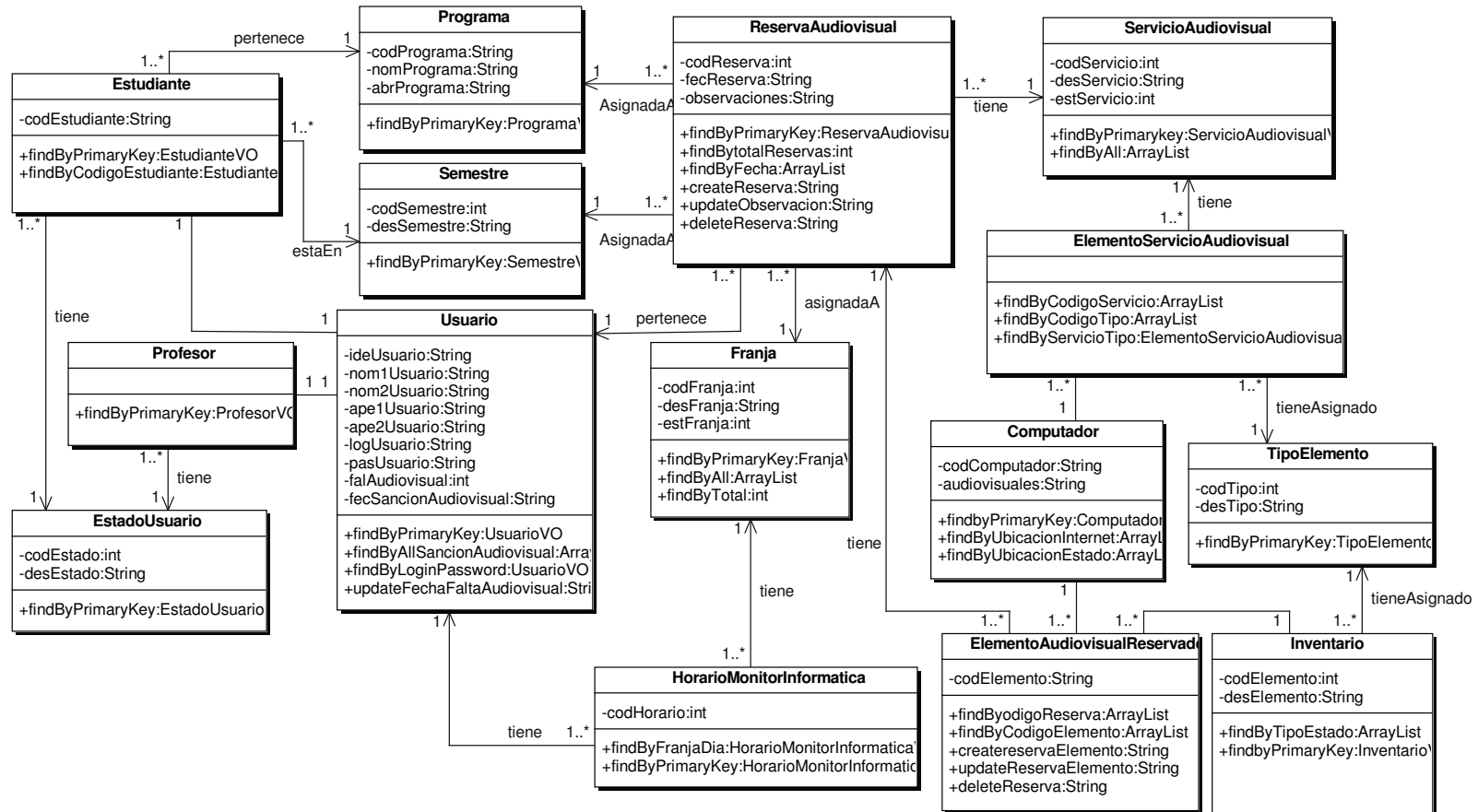
## 10.2.2 Administración de Reservas de Internet



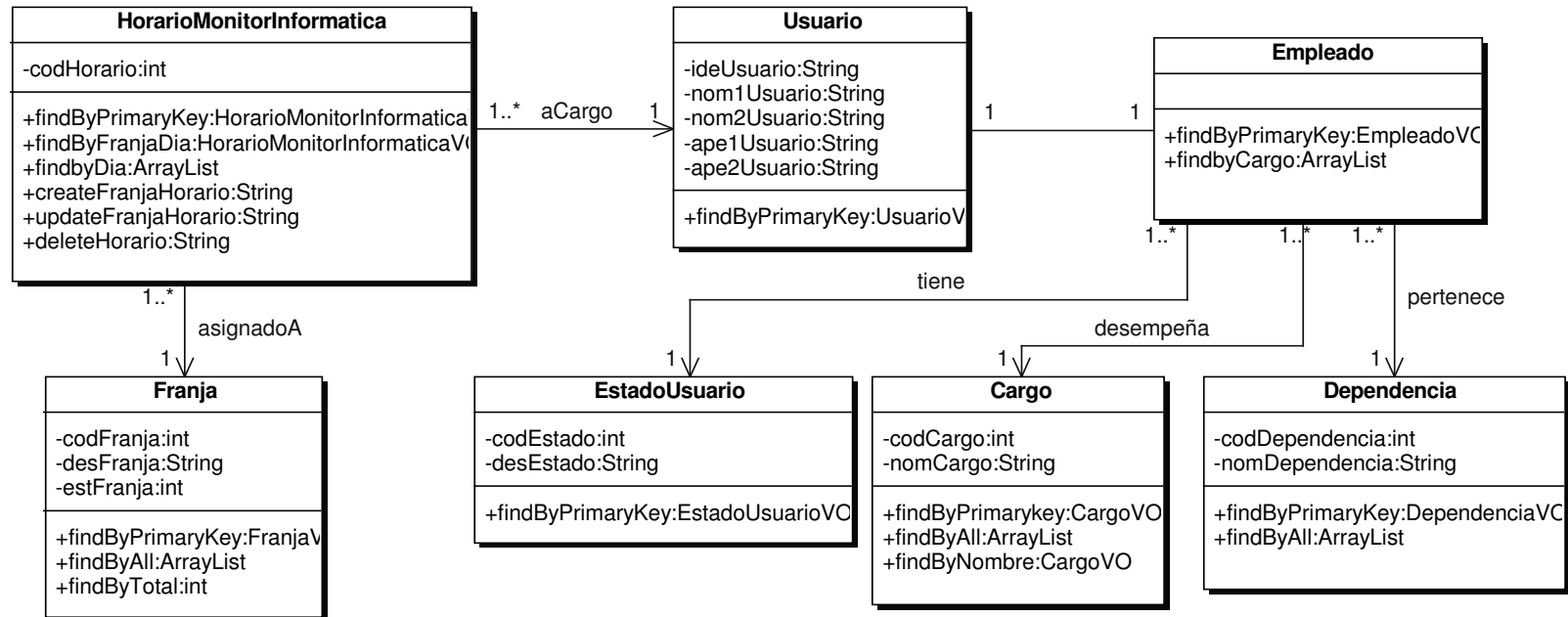
### 10.2.3 Administración de Reservas de Computadores



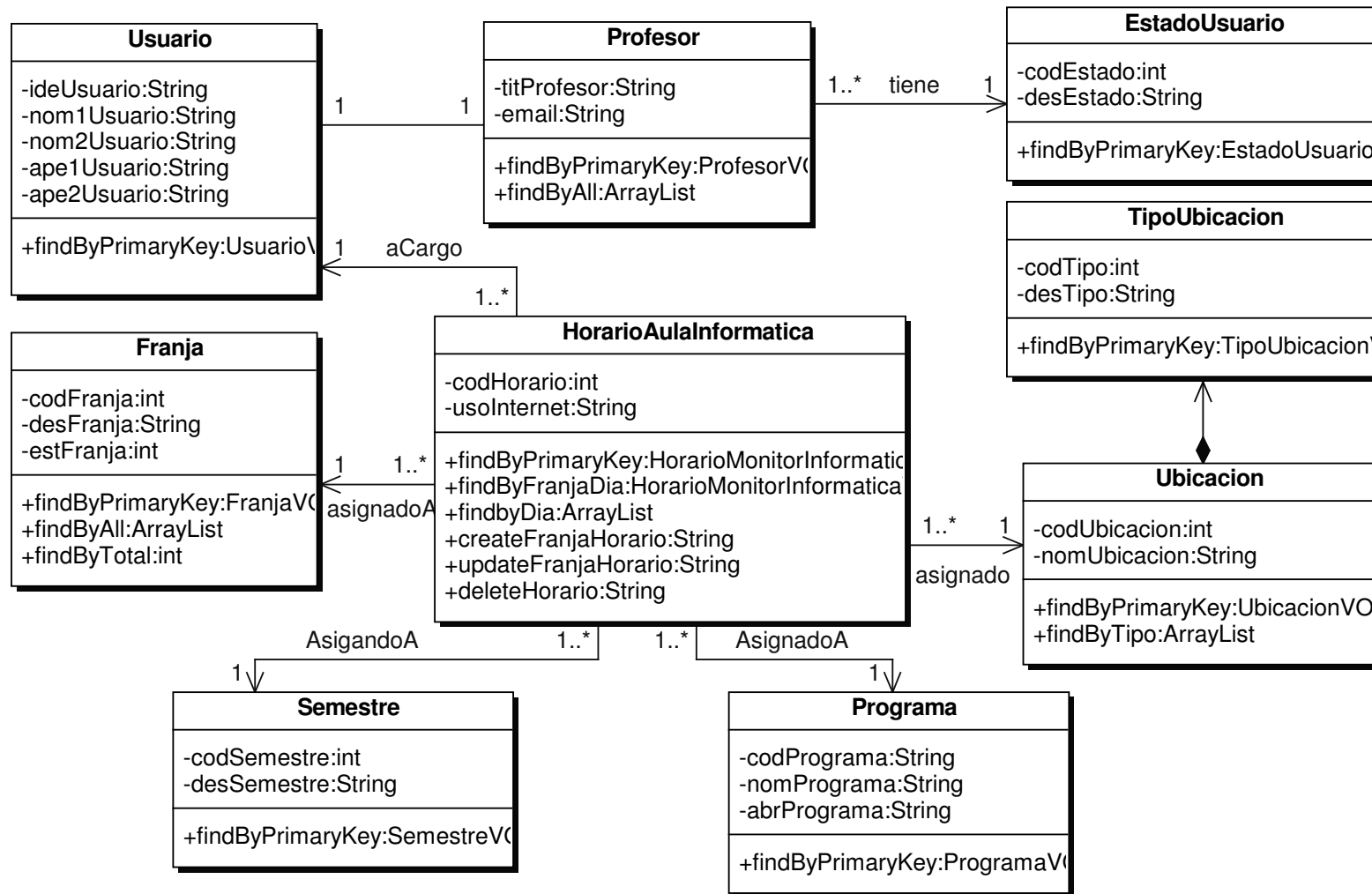
## 10.2.4 Administración de Reservas Audiovisuales



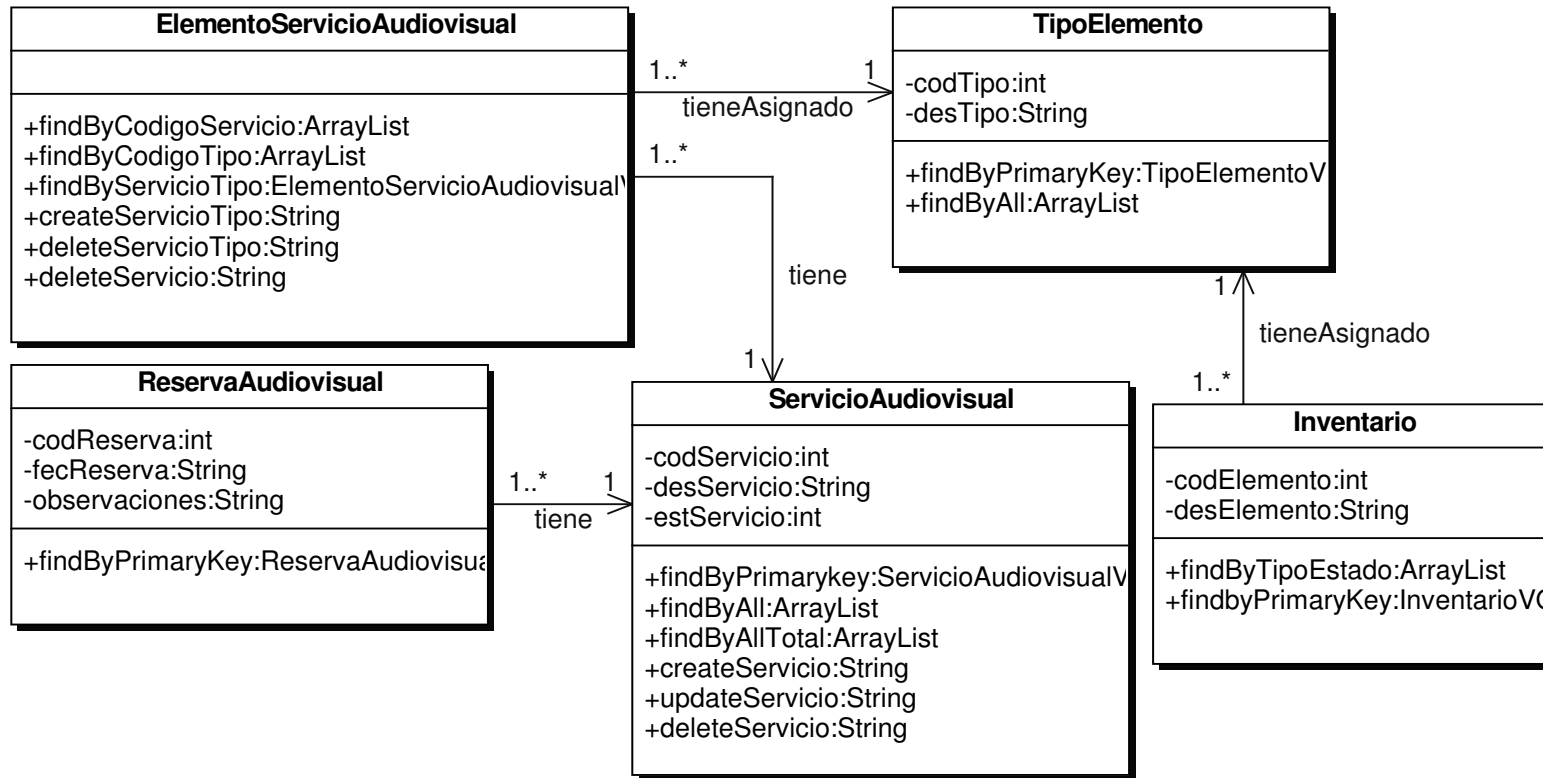
## 10.2.5 Administración de Horarios de Monitoría Informática



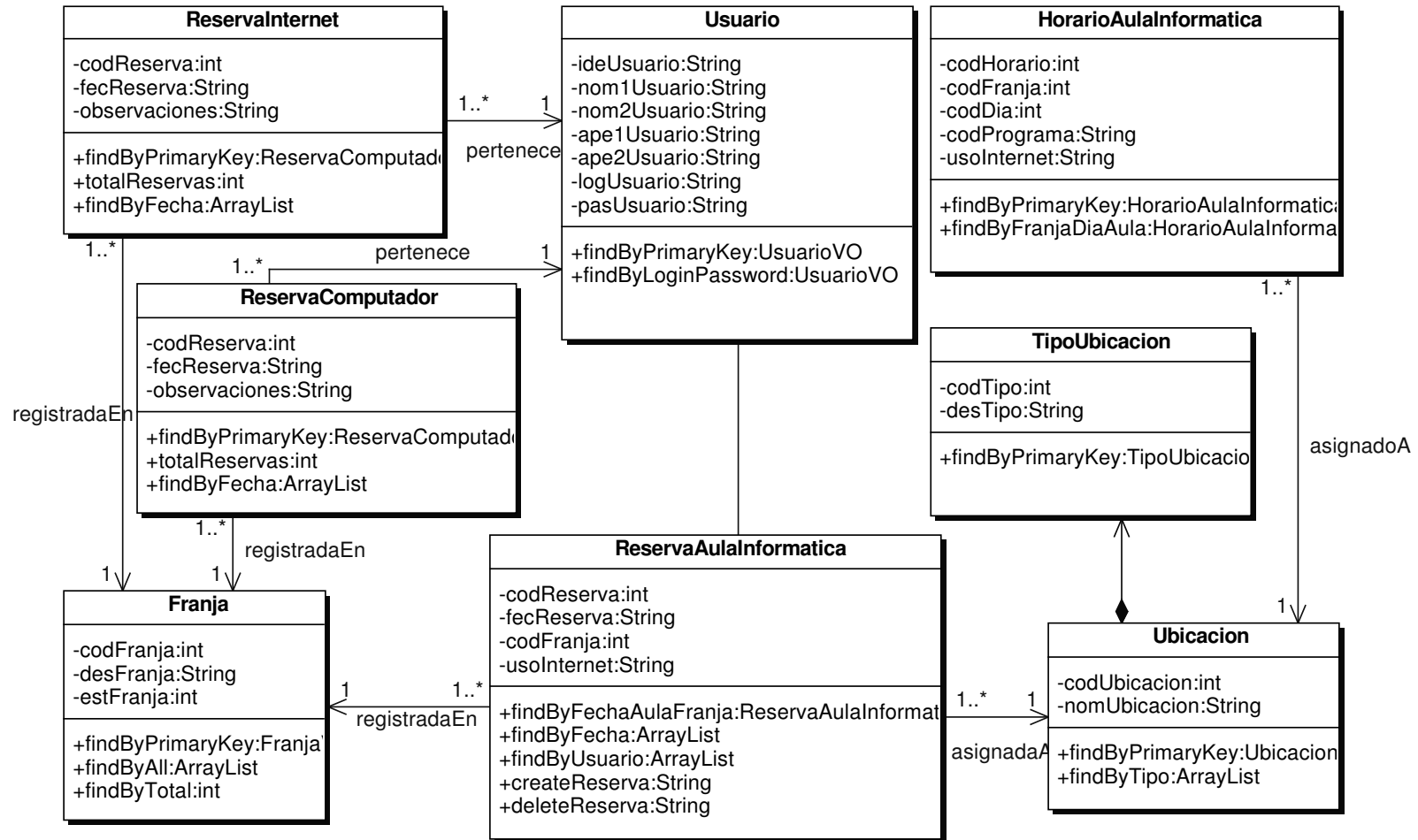
## 10.2.6 Administración de Horarios de Aulas de Informática



### 10.2.7 Administración de Servicios Audiovisuales

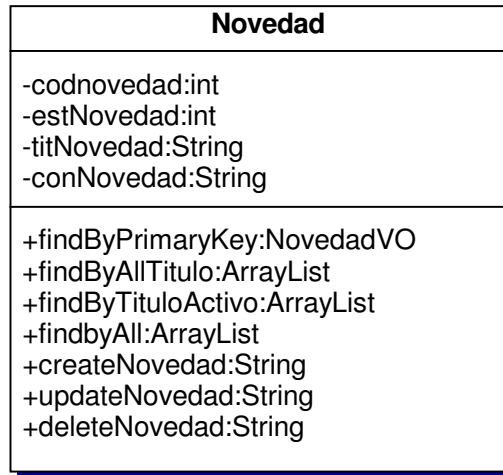


## 10.2.8 Administración de Reservas de Aulas de Informatica

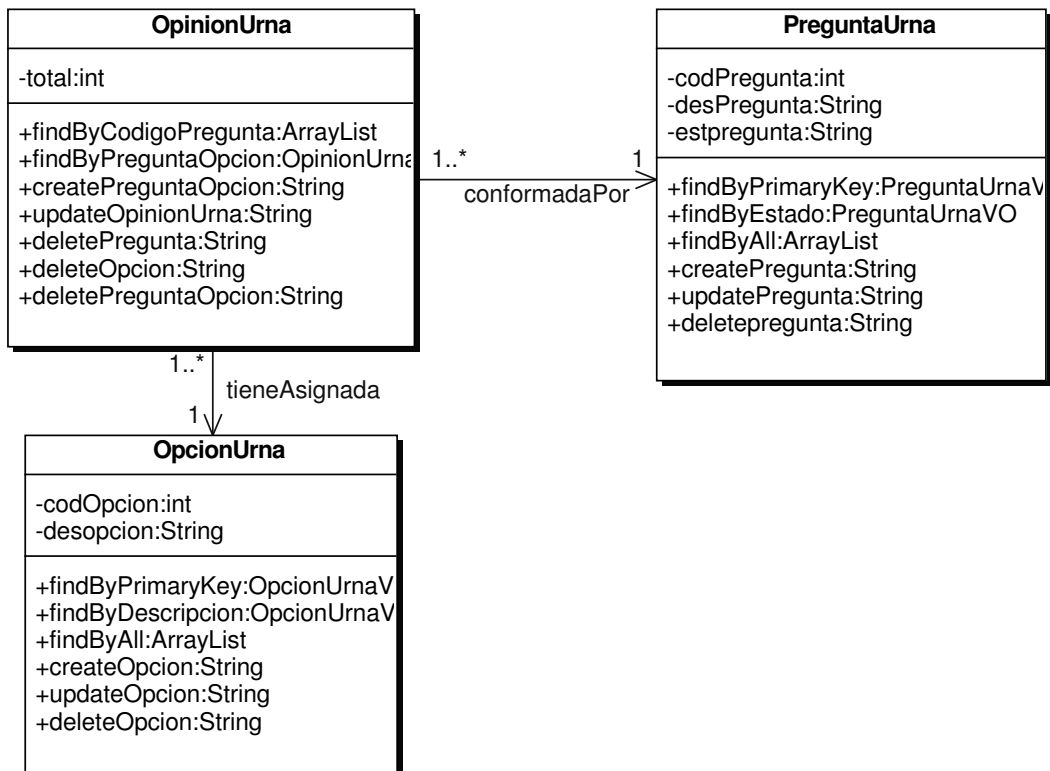




### 10.2.9 Administración de Novedades



### 10.2.10 Administración de Preguntas de Urna Virtual



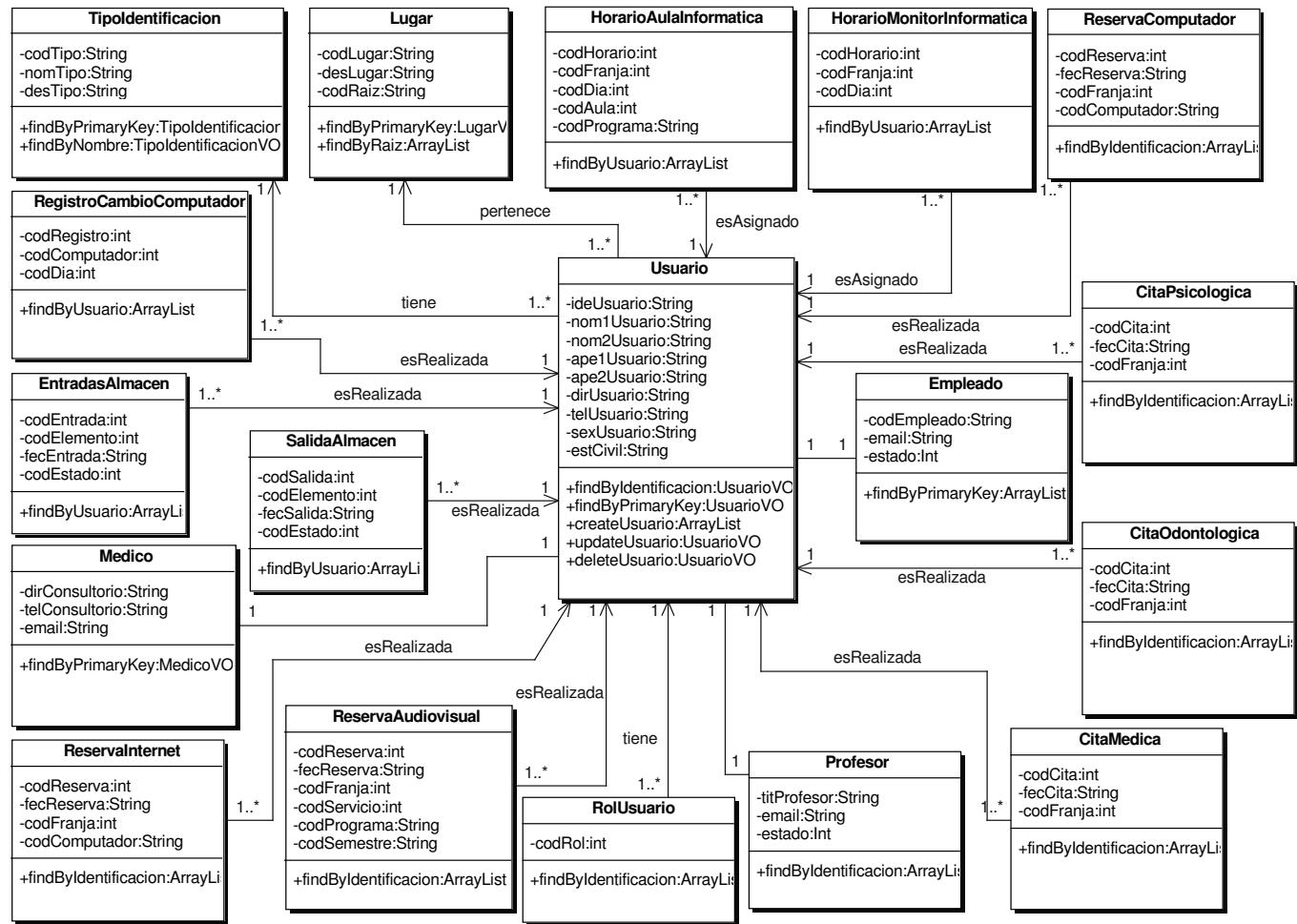
### 10.2.11 Administración de Opciones de Urna Virtual

<b>OpcionUrna</b>
-codOpcion:int -desopcion:String
+findByPrimaryKey:OpcionUrnaVO +findByDescripcion:OpcionUrnaVO +findByAll:ArrayList +createOpcion:String +updateOpcion:String +deleteOpcion:String

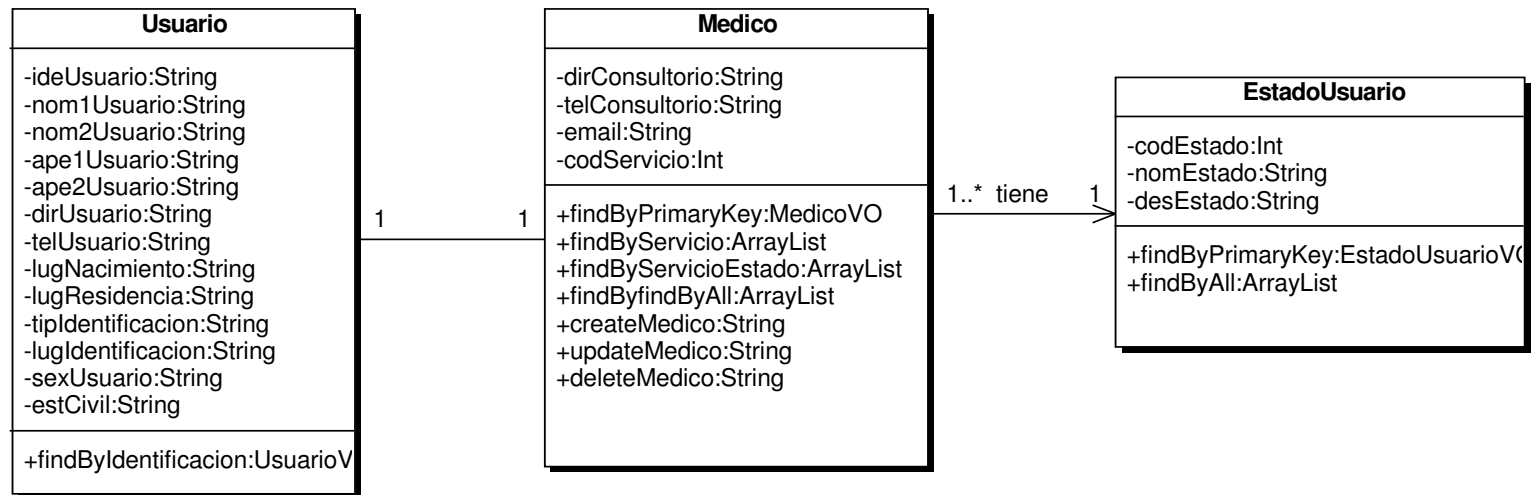
### 10.2.12 Administración de Servicios Recreativos

<b>ServicioRecreativo</b>
-codServicio:int -titServicio:String -desServicio:String -imgServicio:String
+findByPrimaryKey:ServicioRecreativoVO +findByAll:ArrayList +createServicioRecreativo:String +updateServicioRecreativo:String +deleteServicioRecreativo:String

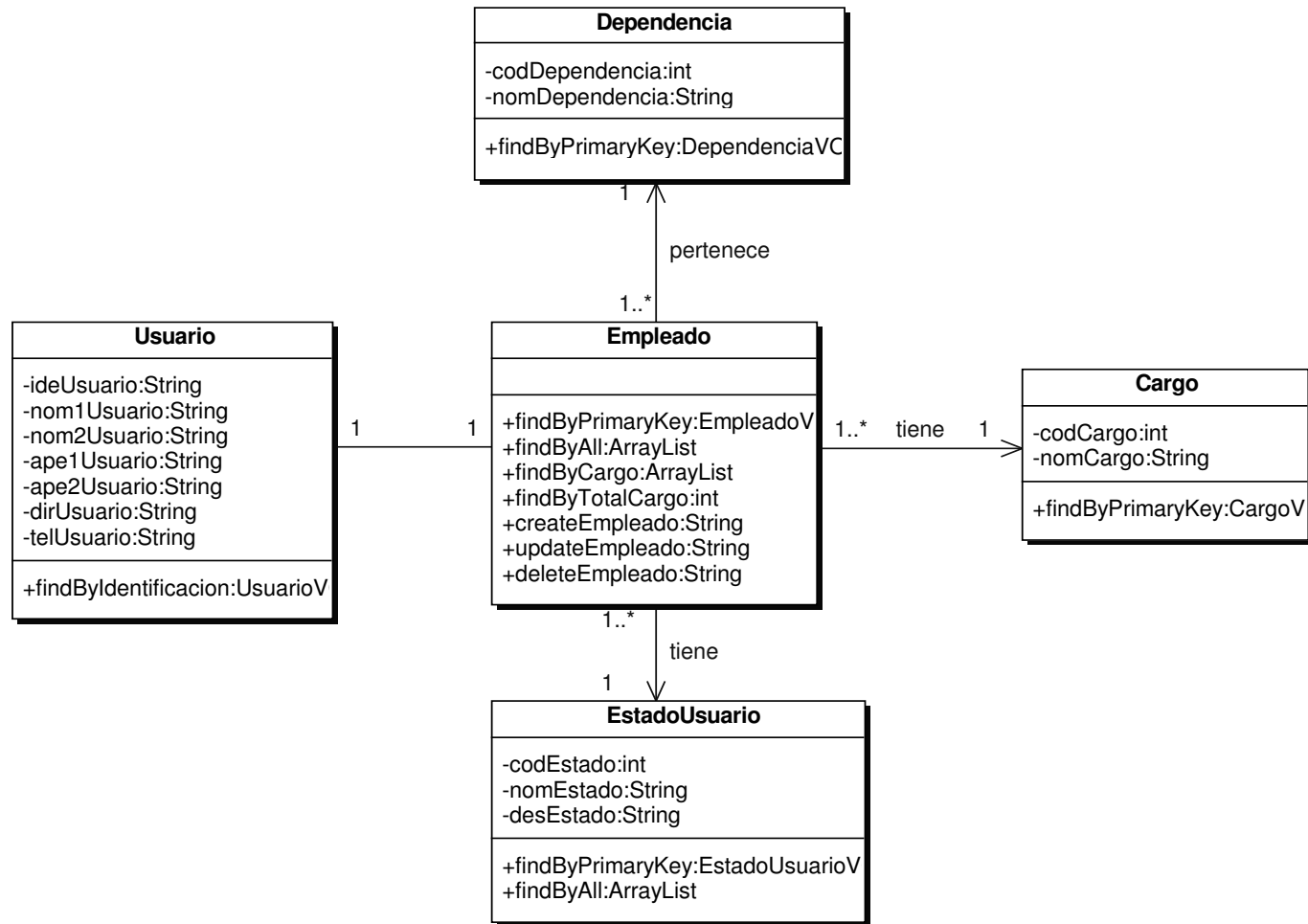
### 10.2.13 Administración de Usuarios



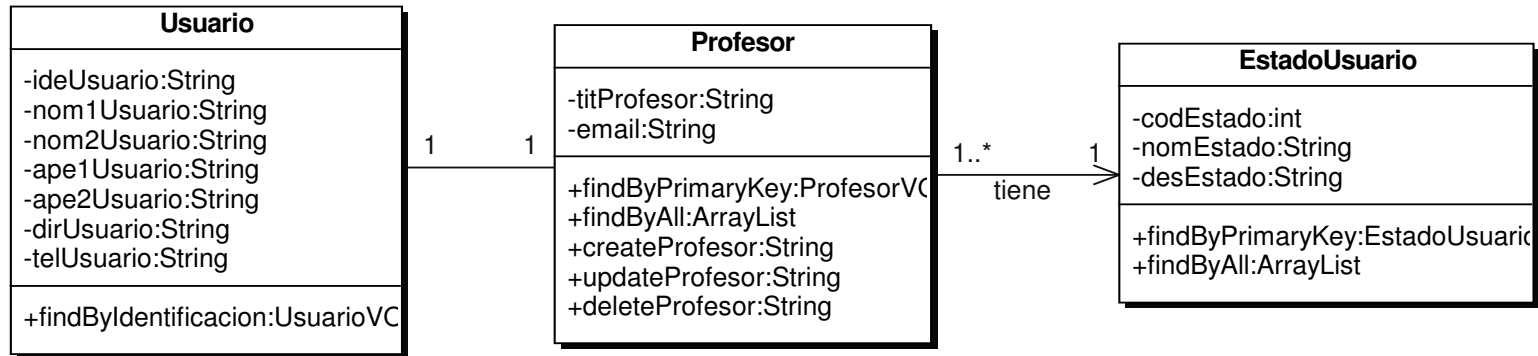
### 10.2.14 Administración de Personal de Salud (Médicos, Odontólogos, Psicólogos)



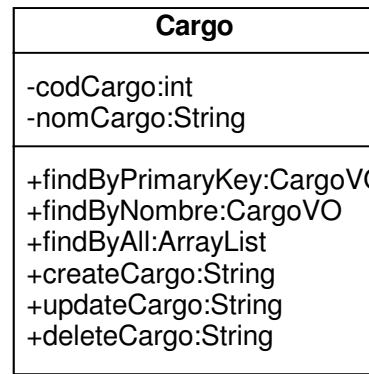
### 10.2.15 Administración de Personal Administrativo y Operativo



### 10.2.16 Administración de Profesores



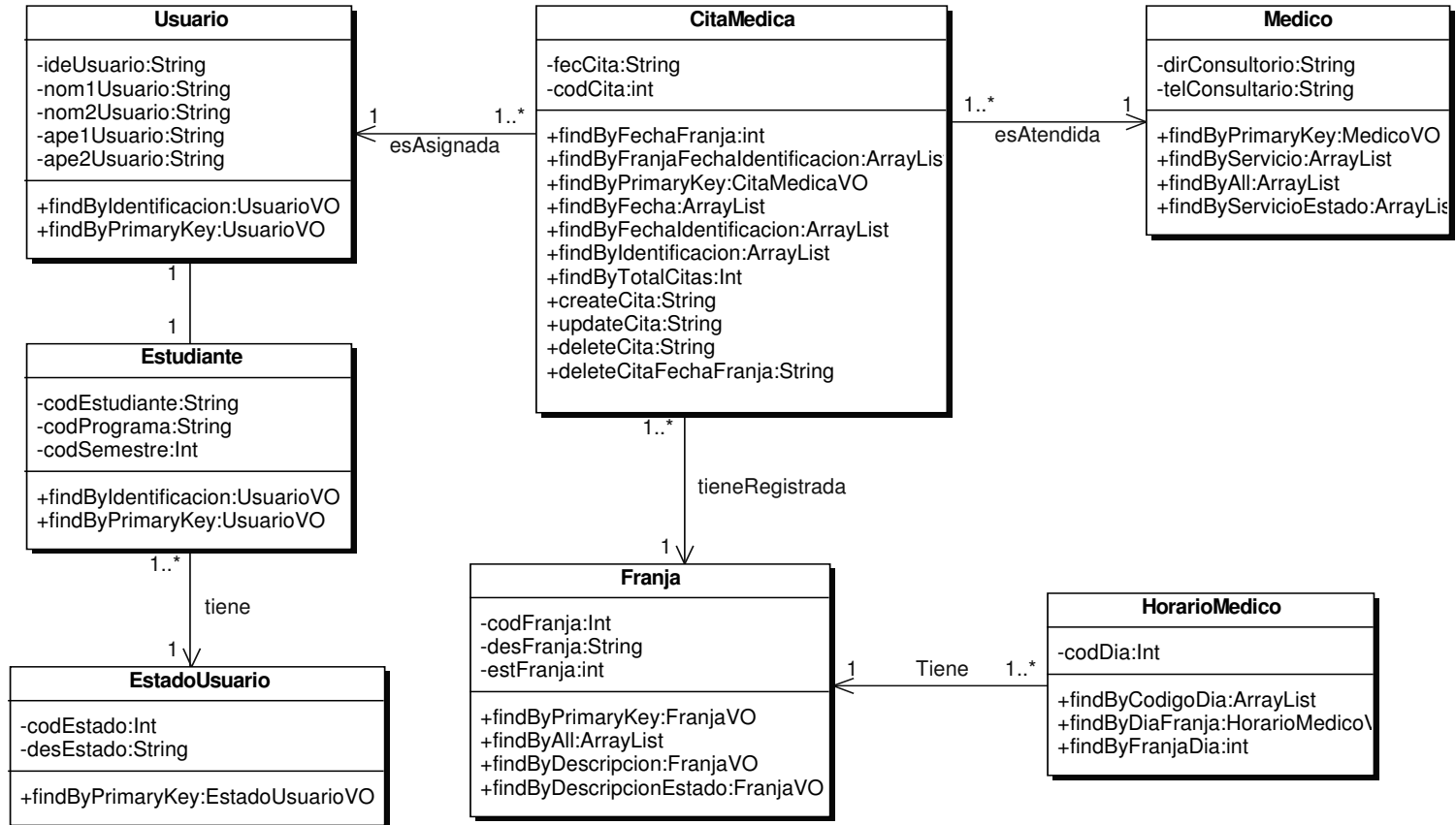
### 10.2.17 Administración de Cargos



## 10.2.18 Administración de Lugares

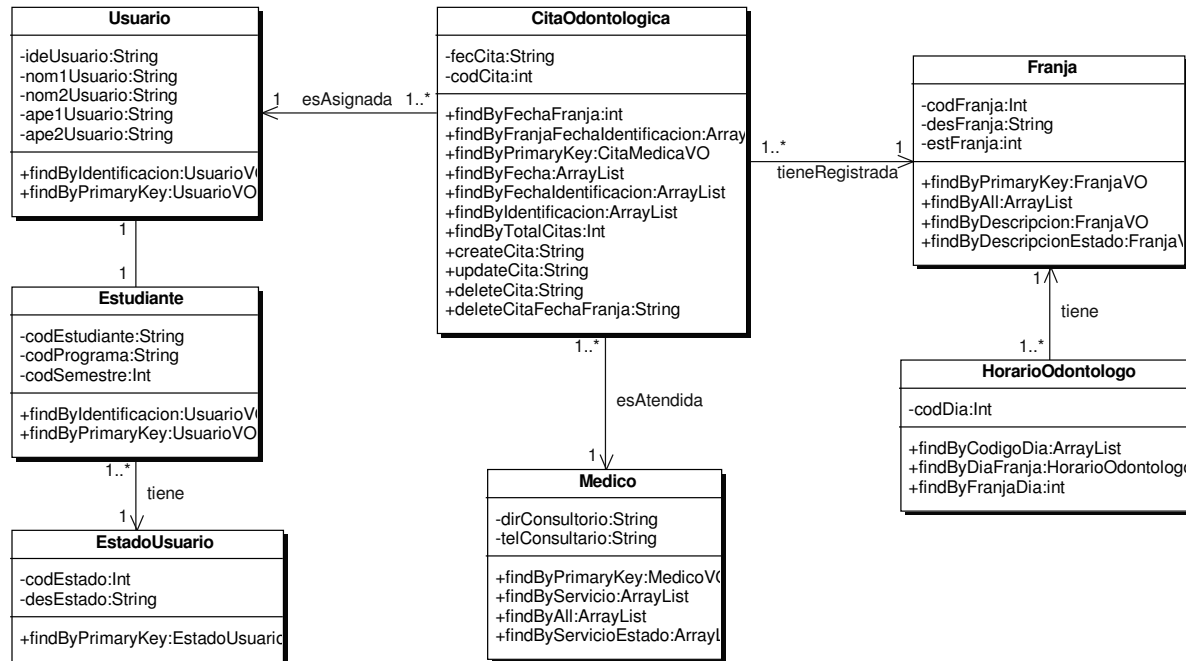
Lugar
-codLugar:String -desLugar:String -codRaiz:String
+findByPrimaryKey:LugarVO +findByRaiz:ArrayList +findByDescripcion:LugarVO +findByDescripcionGeneral:LugarVO +findByDescripcionPais:LugarVO +findByDescripcionRaiz:LugarVO +findByCodigoPais:ArrayList +createLugar:String +updateLugar:String +deleteLugar:String

## 10.2.19 Administración de Citas Médicas

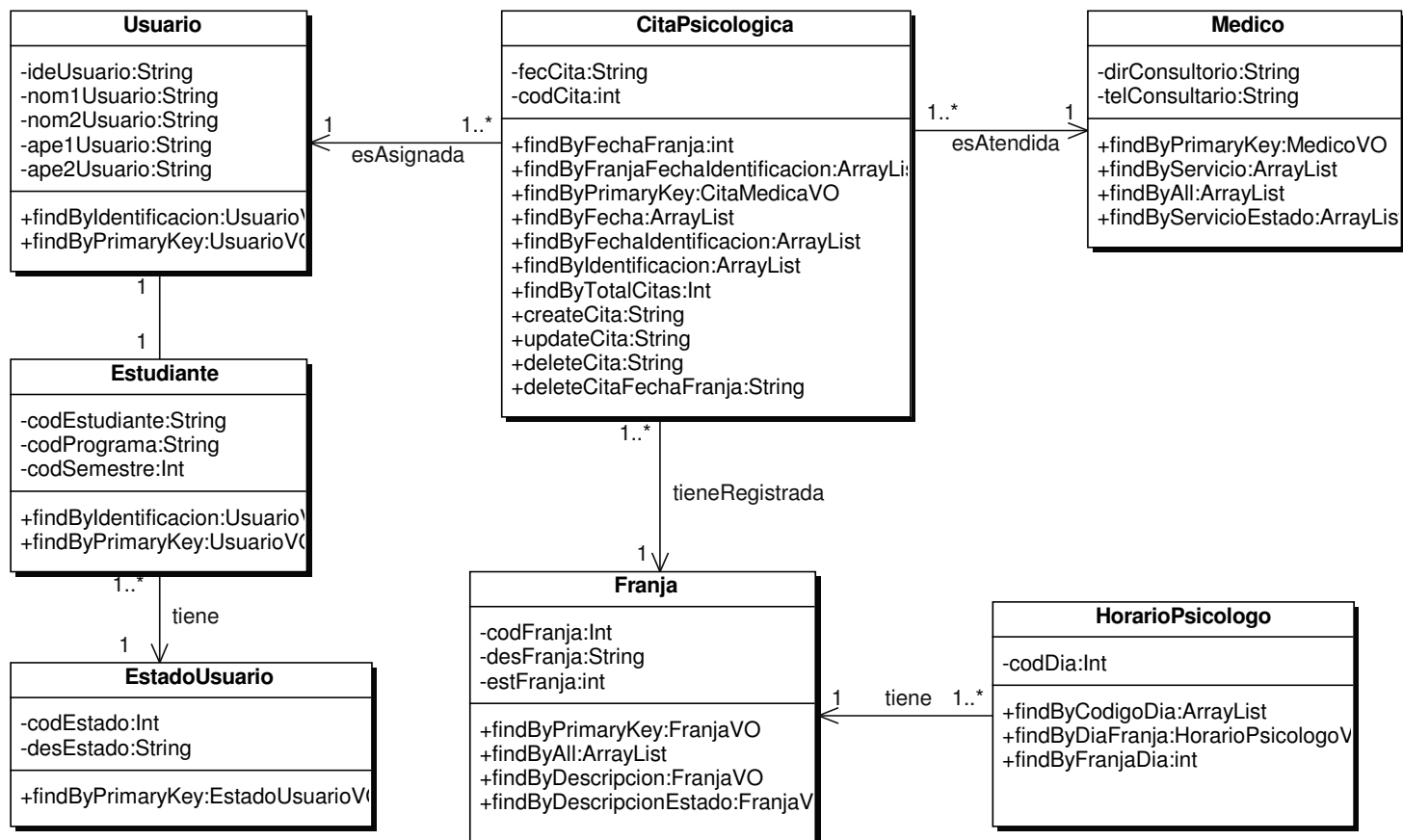




## 10.2.20 Administración de Citas Odontológicas



## 10.2.21 Administración de Citas Psicológicas



### 10.2.22 Administración de Horario Médico

HorarioMedico
-codDia:Int -codFranja:int
+findByCodigoDia:ArrayList +findByDiaFranja:HorarioMedicoVC +findByFranjaDia:int +createHorarioMedico:String +deleteHorarioMedico:String

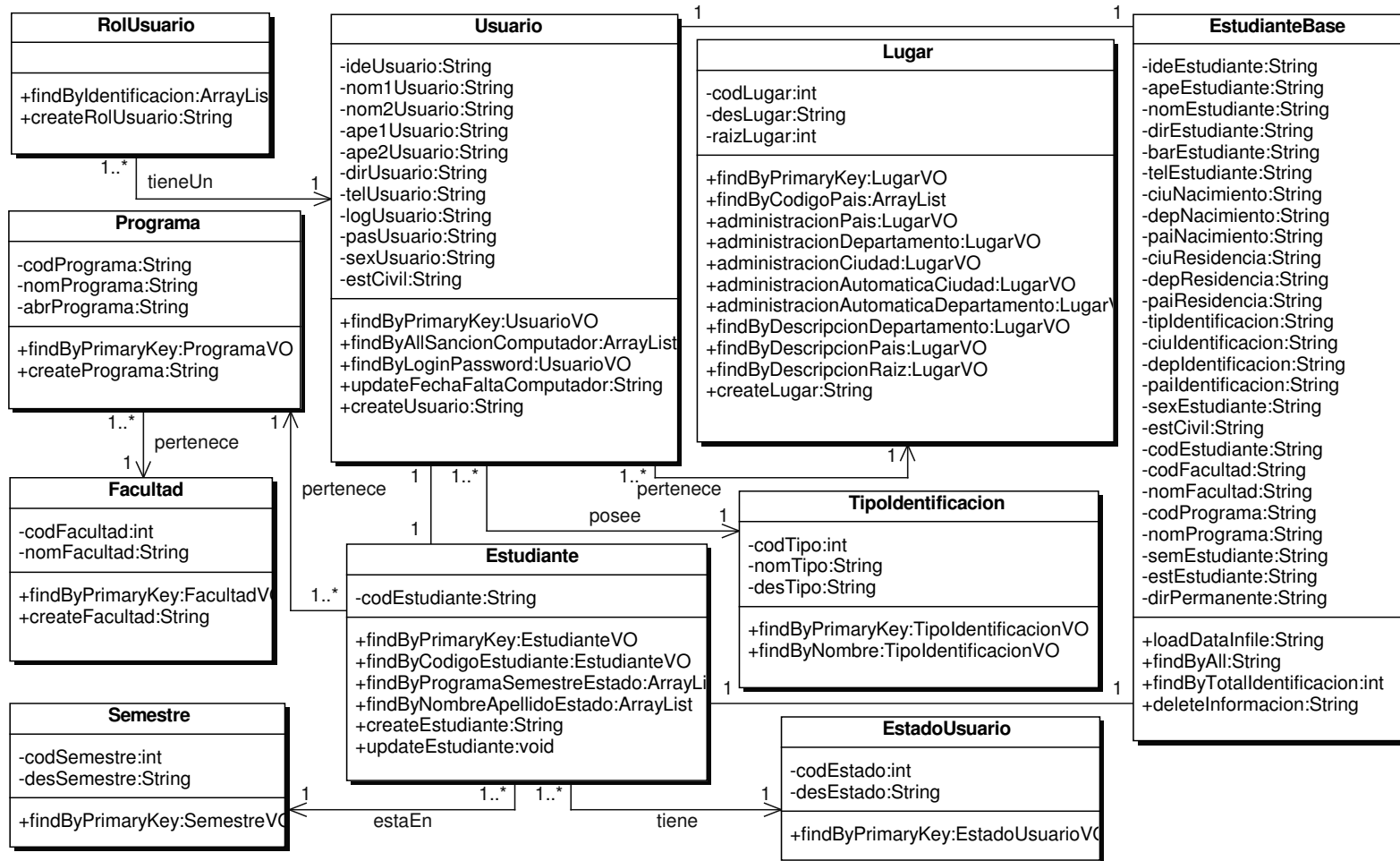
### 10.2.23 Administración de Horario Odontológico

HorarioOdontologo
-codDia:Int -codFranja:int
+findByCodigoDia:ArrayList +findByDiaFranja:HorarioOdontologoVC +findByFranjaDia:int +createHorarioOdontologo:String +deleteHorarioOdontologo:String

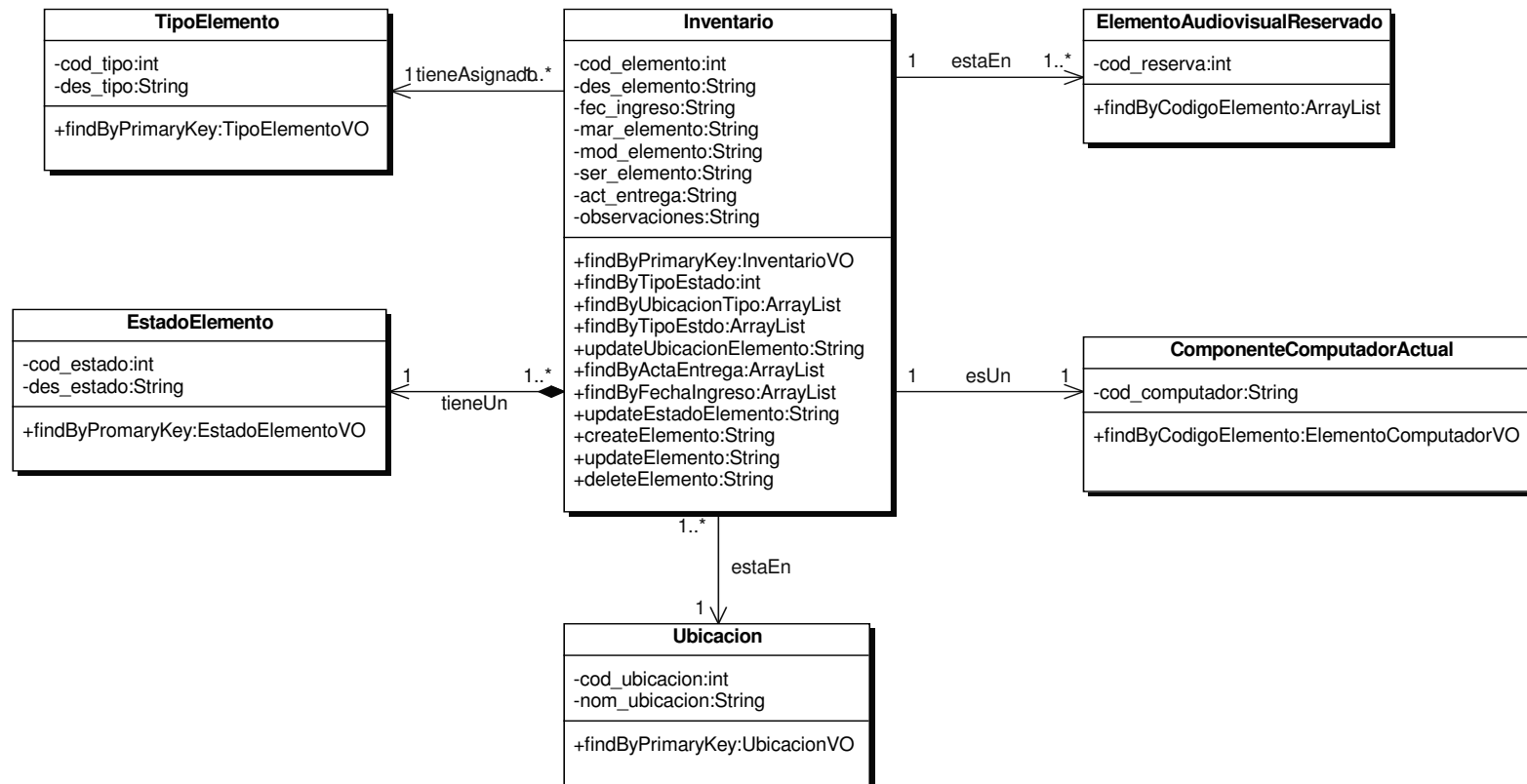
### 10.2.24 Administración de Horario Psicológico

HorarioPsicologo
-codDia:Int -codFranja:int
+findByCodigoDia:ArrayList +findByDiaFranja:HorarioPsicologoV +findByFranjaDia:int +createHorarioPsicologo:String +deleteHorarioPsicologo:String

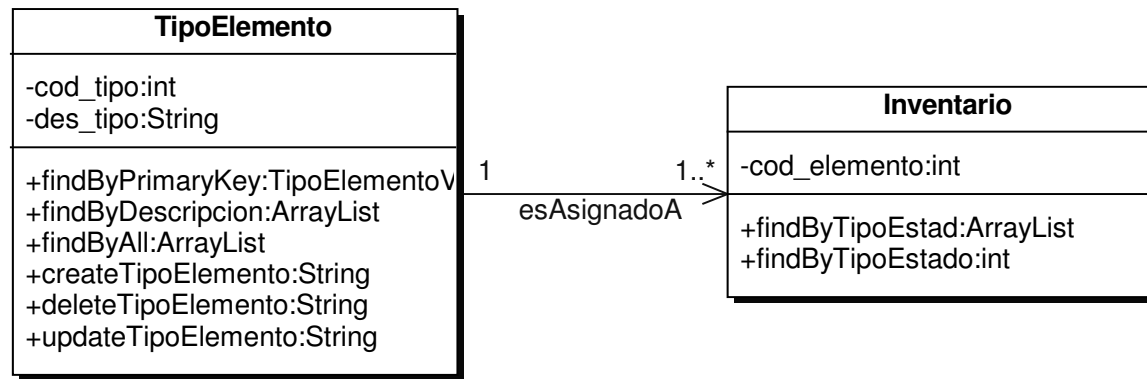
## 10.2.25 Administración de Estudiantes



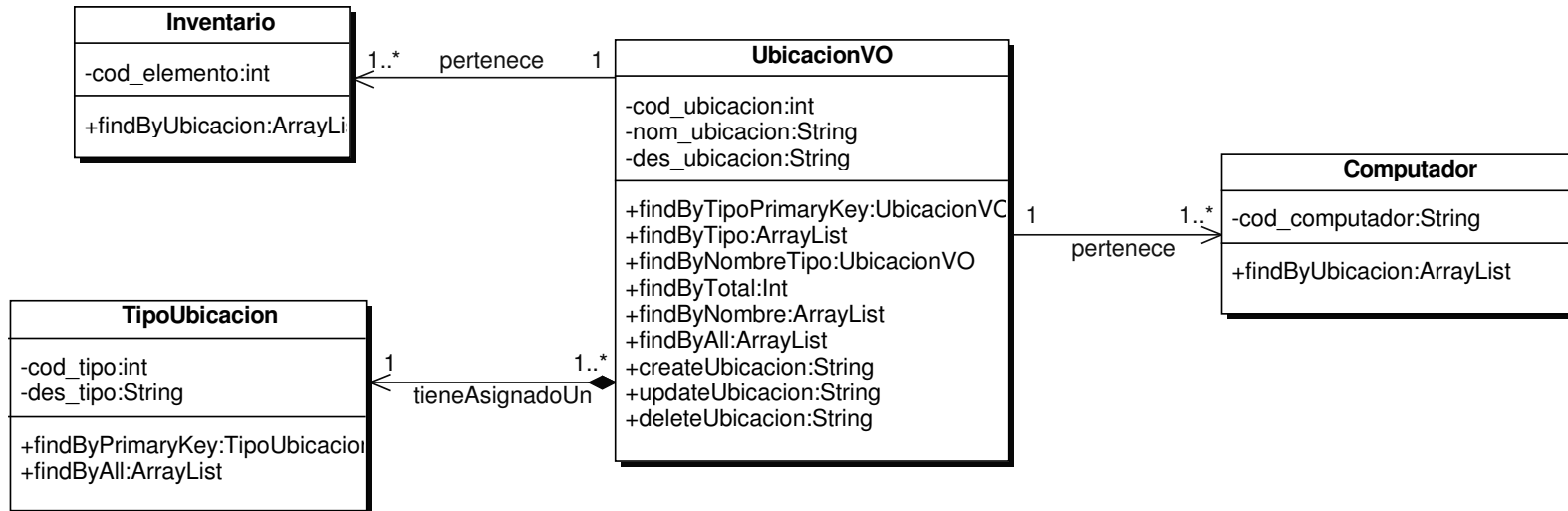
## 10.2.26 Administración de Elementos de Inventario



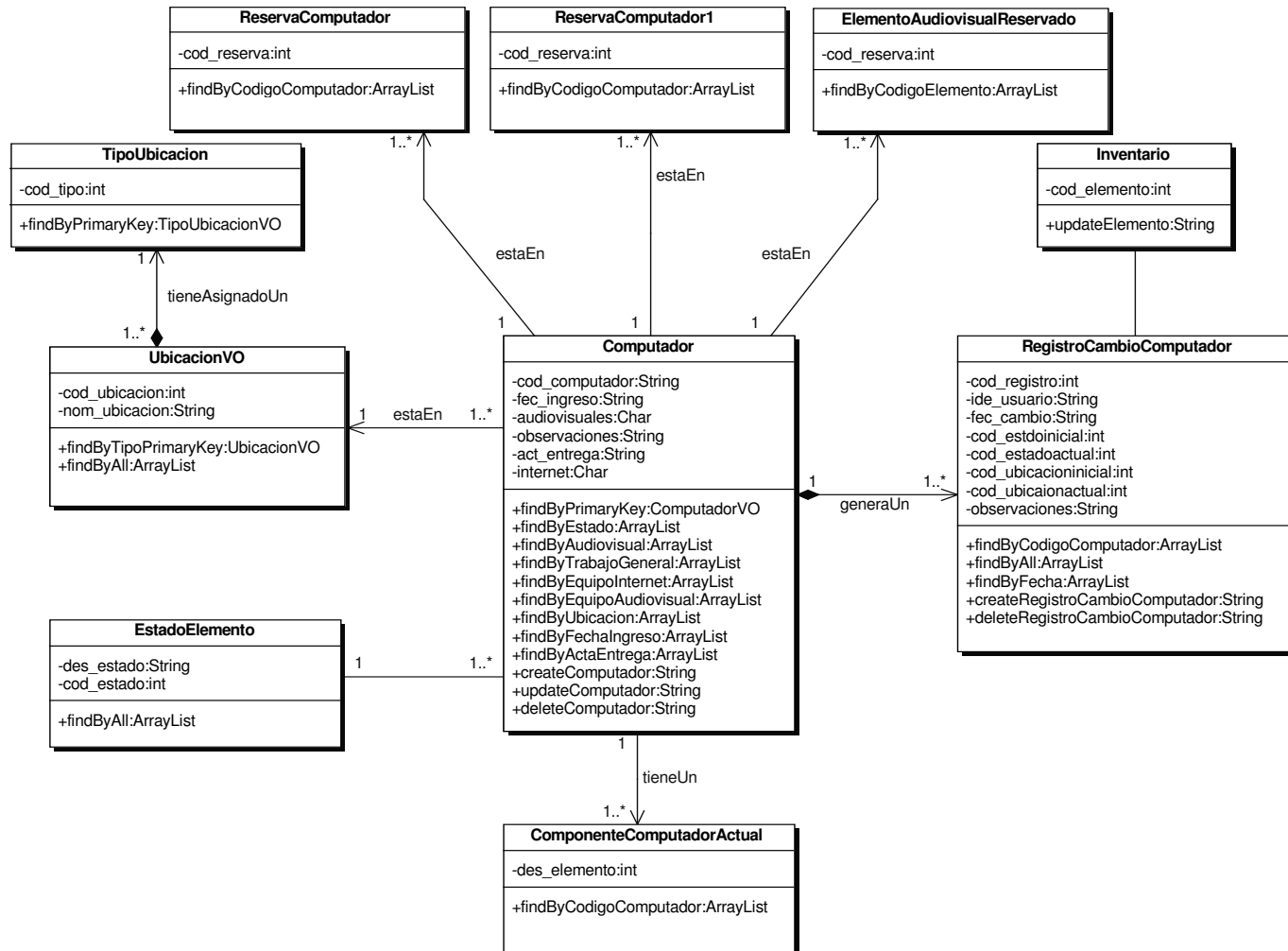
### 10.2.27 Administración Tipos Elementos



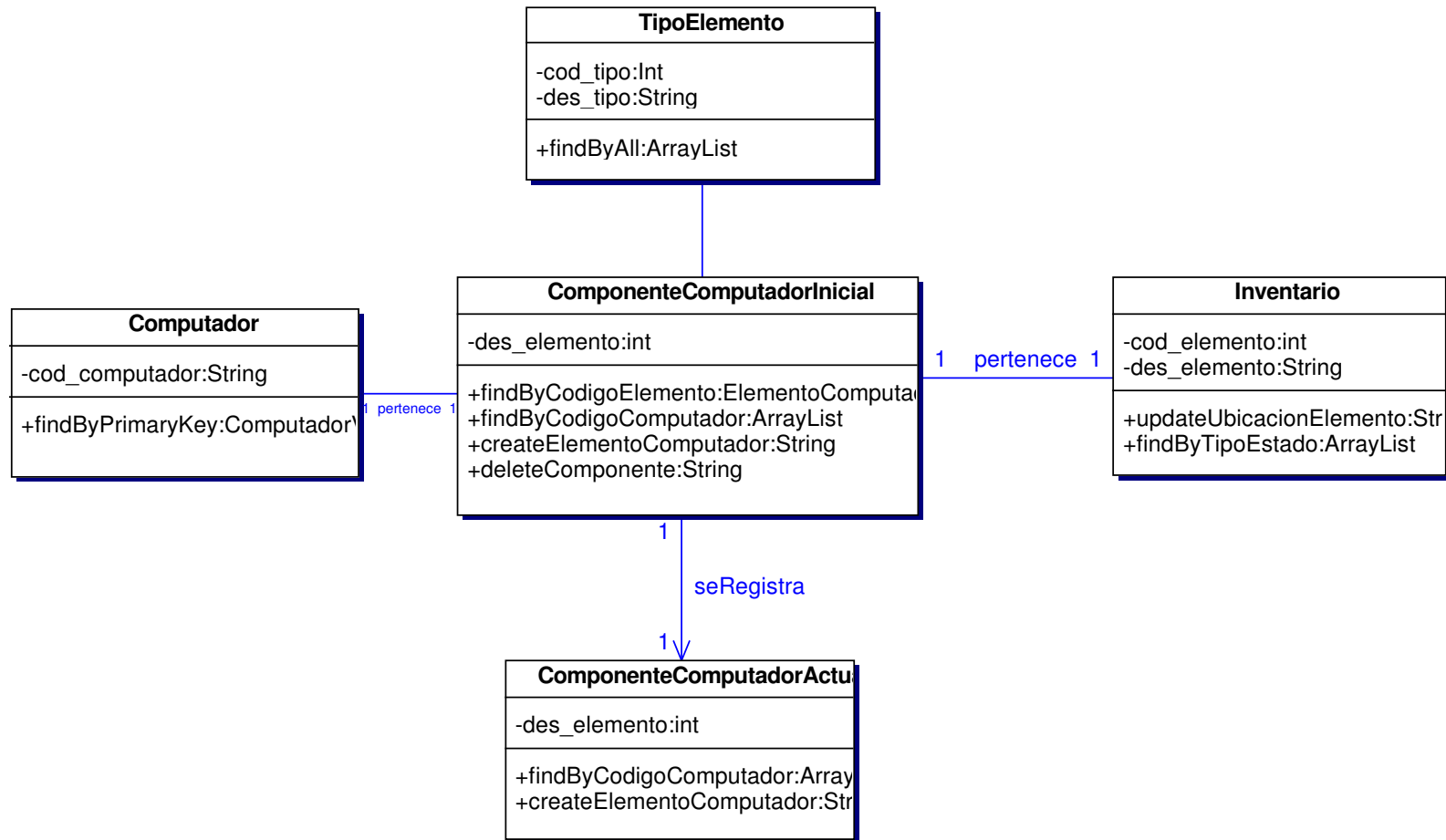
### 10.2.28 Administración Ubicaciones



## 10.2.29 Administración Computadores

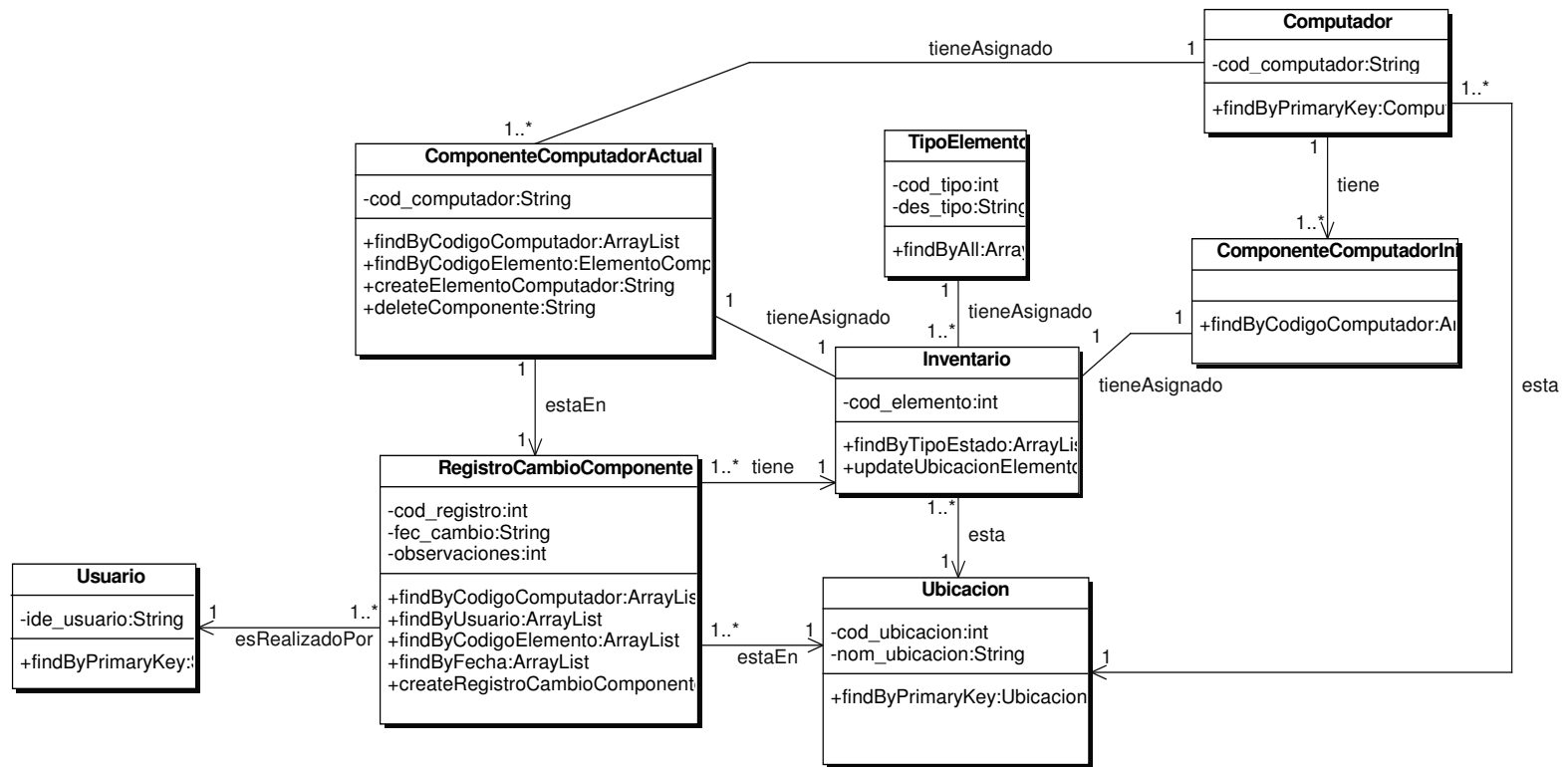


### 10.2.30 Administración de Componentes Computador Inicial

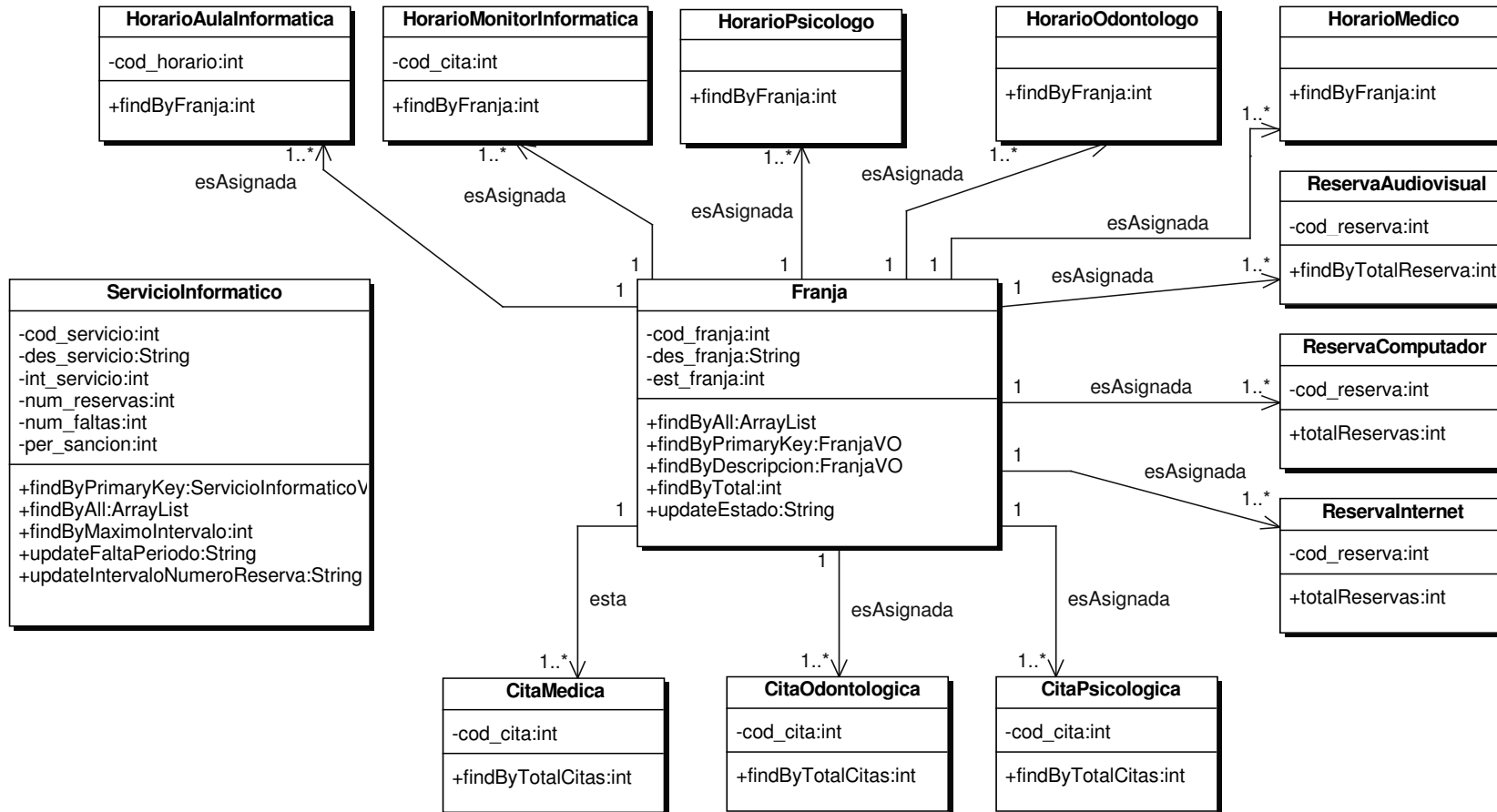




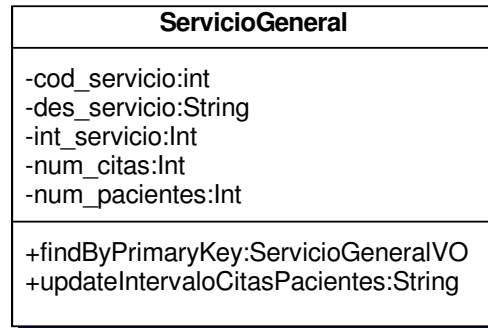
### 10.2.31 Administración De Componentes Computador Actual



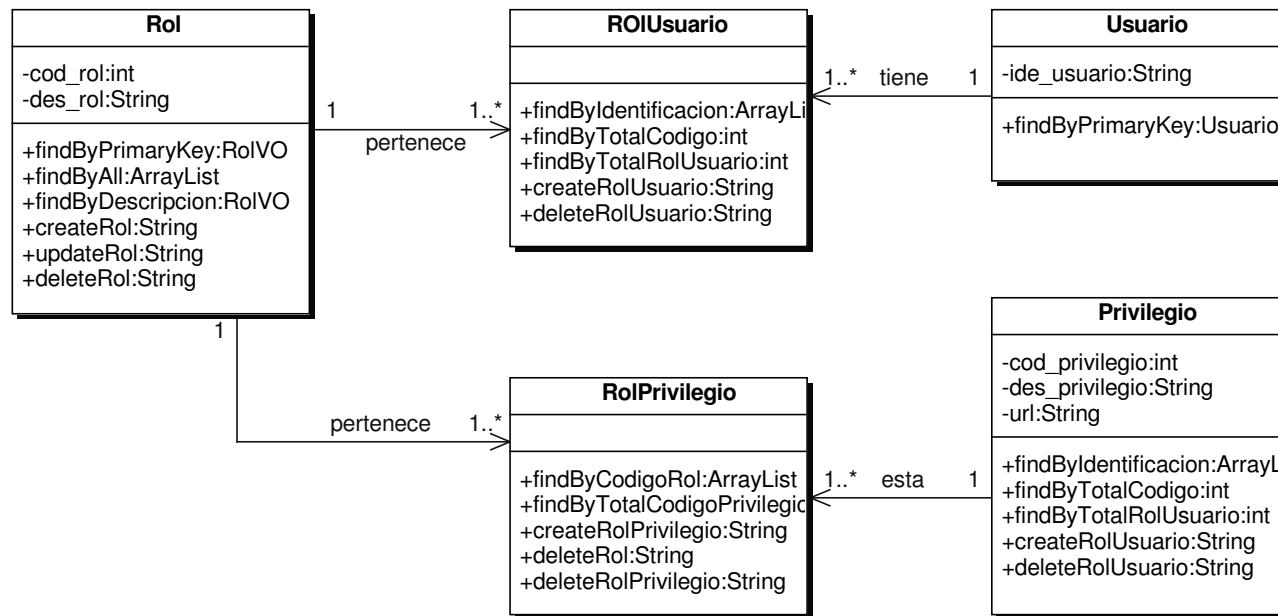
## 10.2.32 Administración de Servicios Informáticos



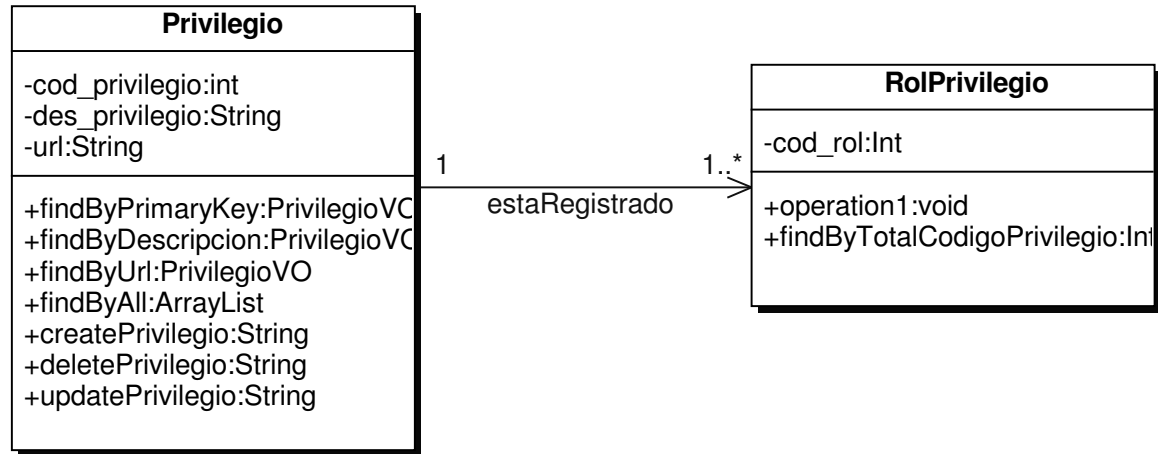
### 10.2.33 Manejo de Restricciones Médicas



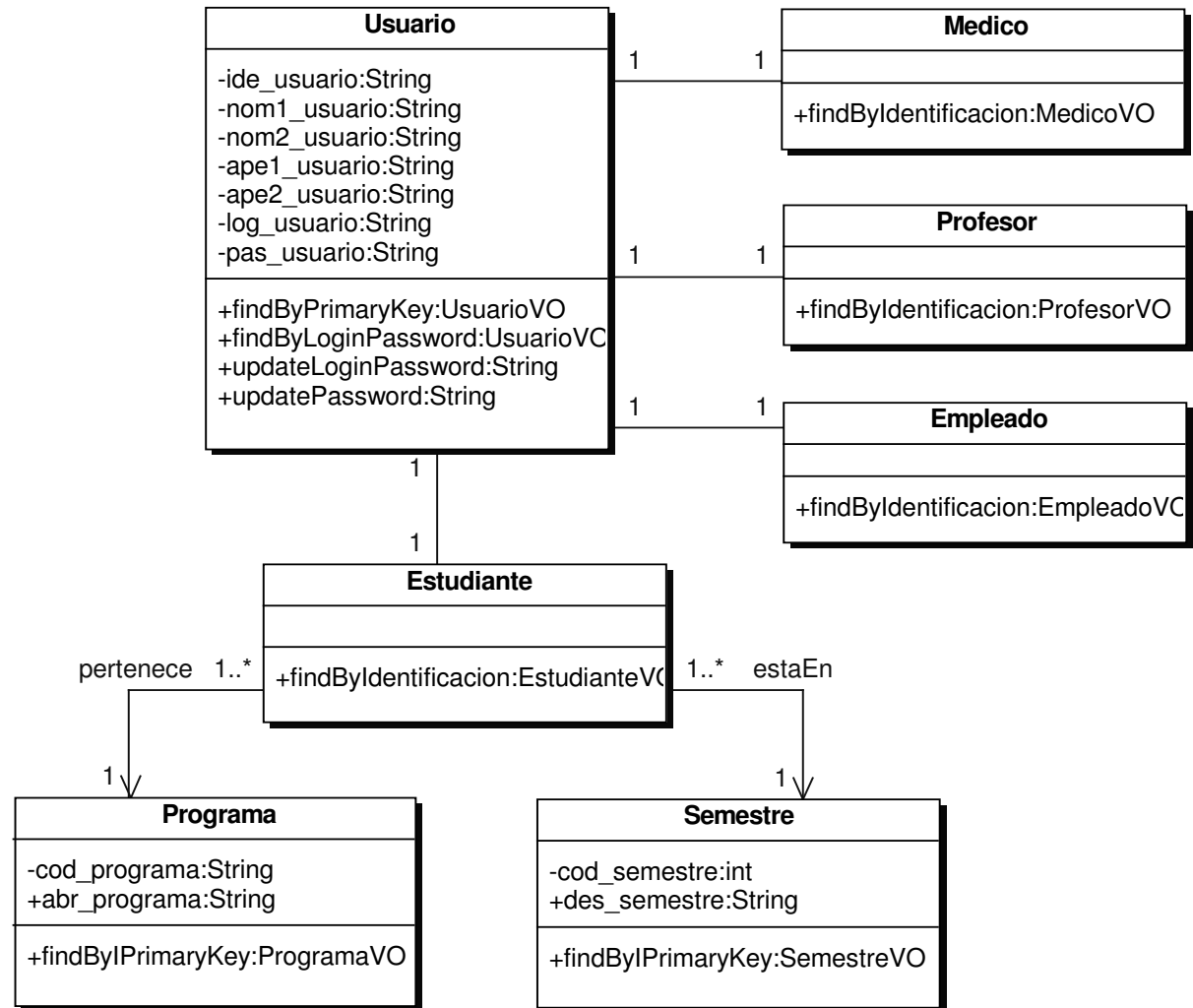
### 10.2.34 Administración de Roles



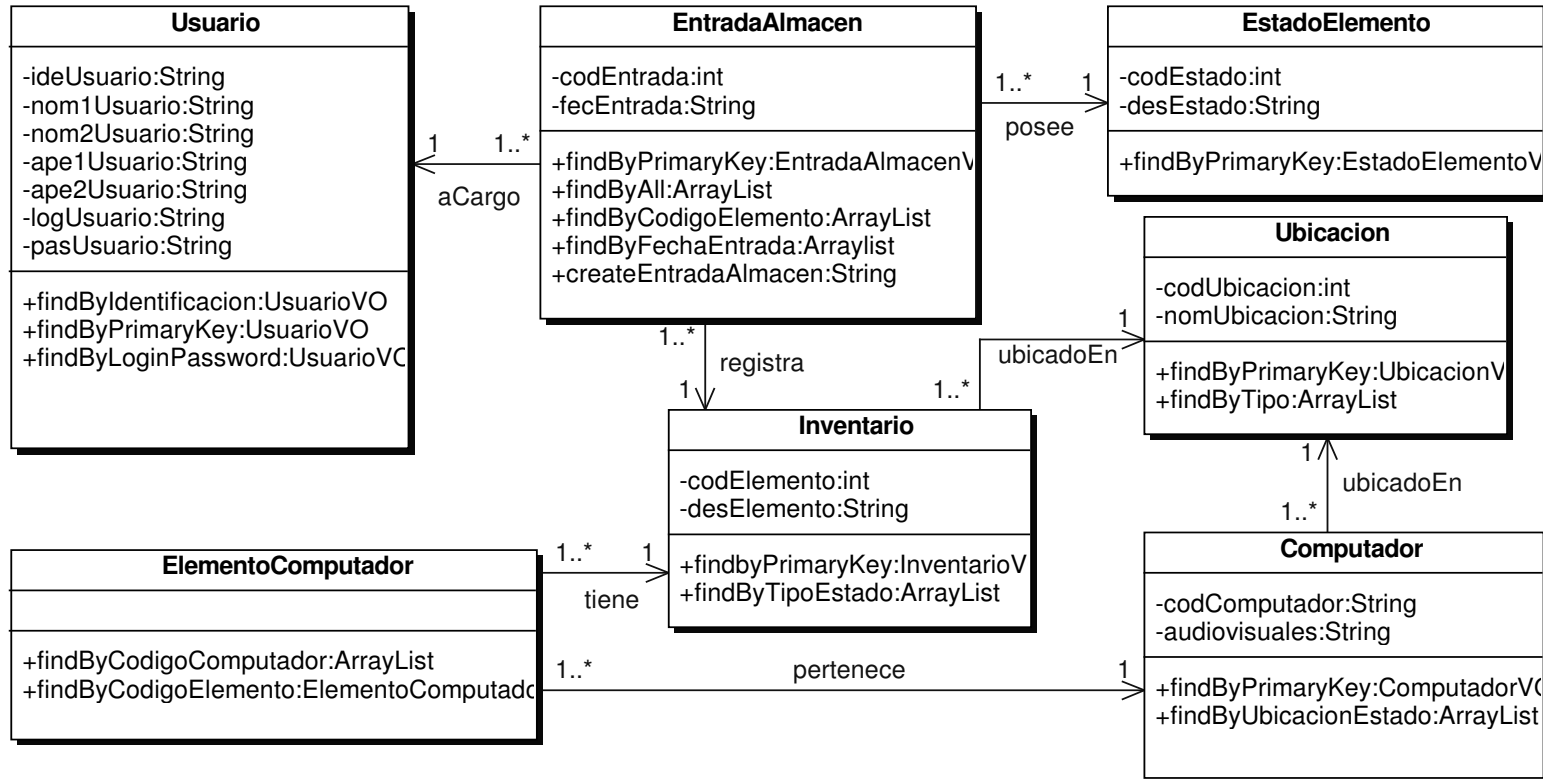
### 10.2.35 Administración de Privilegios



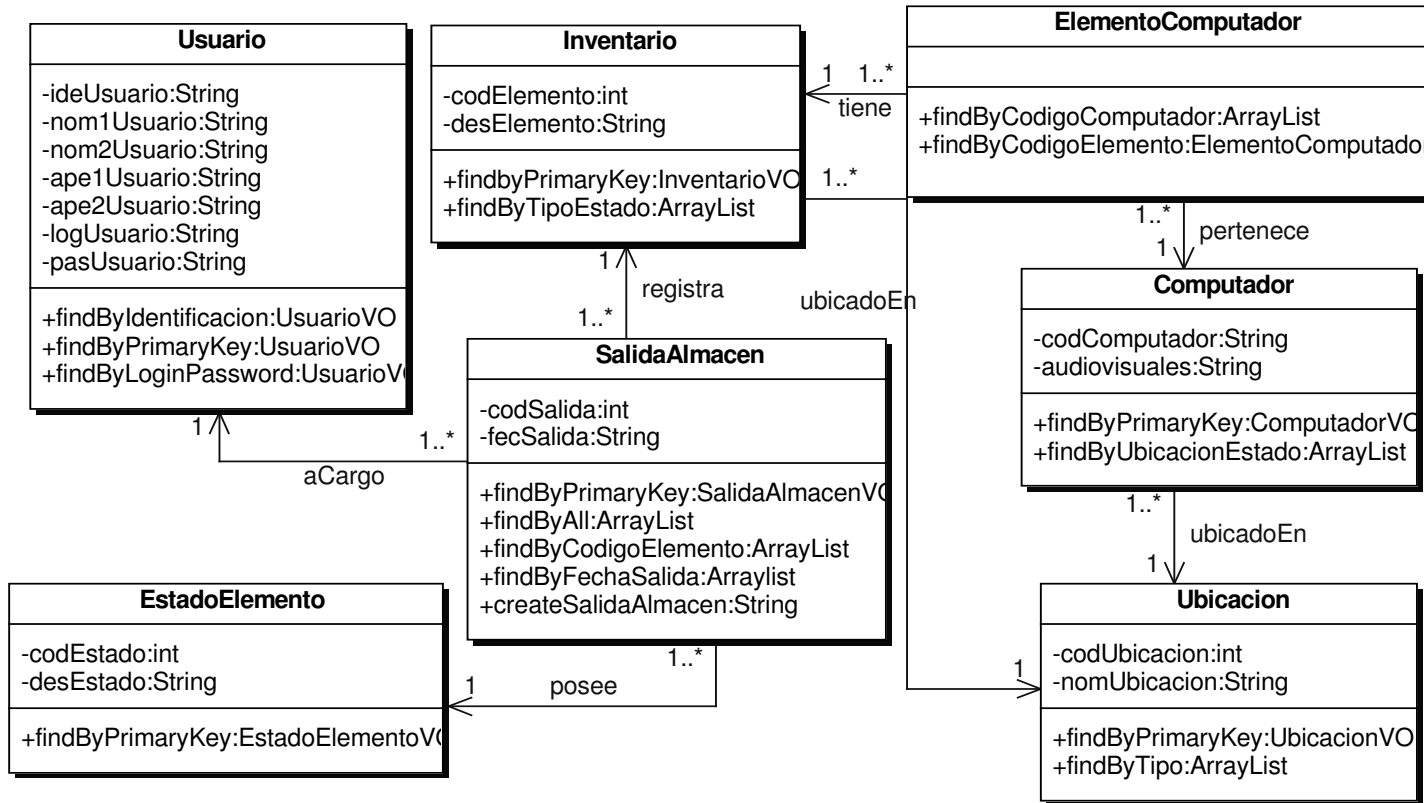
### 10.2.36 Administración de Login Y Password



### 10.2.37 Administración de Entradas de Almacén



### 10.2.38 Administración de Salidas de Almacén



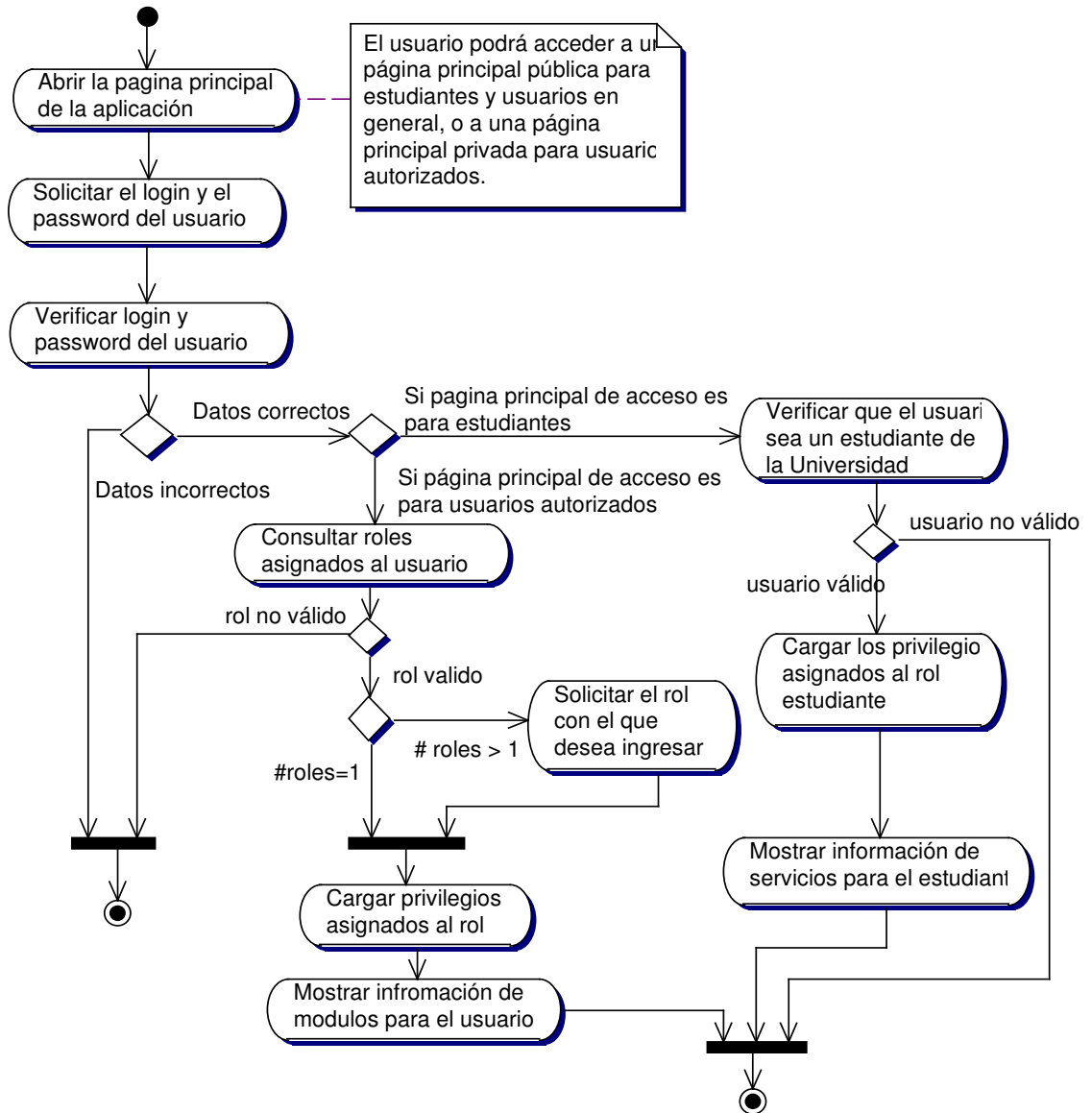
### 10.2.39 Administración de Rubros

<b>Rubro</b>
-codCargo:String -desCargo:String
+findByPrimaryKey:RubroVO +findByAll:ArrayList +findByDescripcion:RubroVO +createRubro:String +updateRubro:String +deleteRubro:String



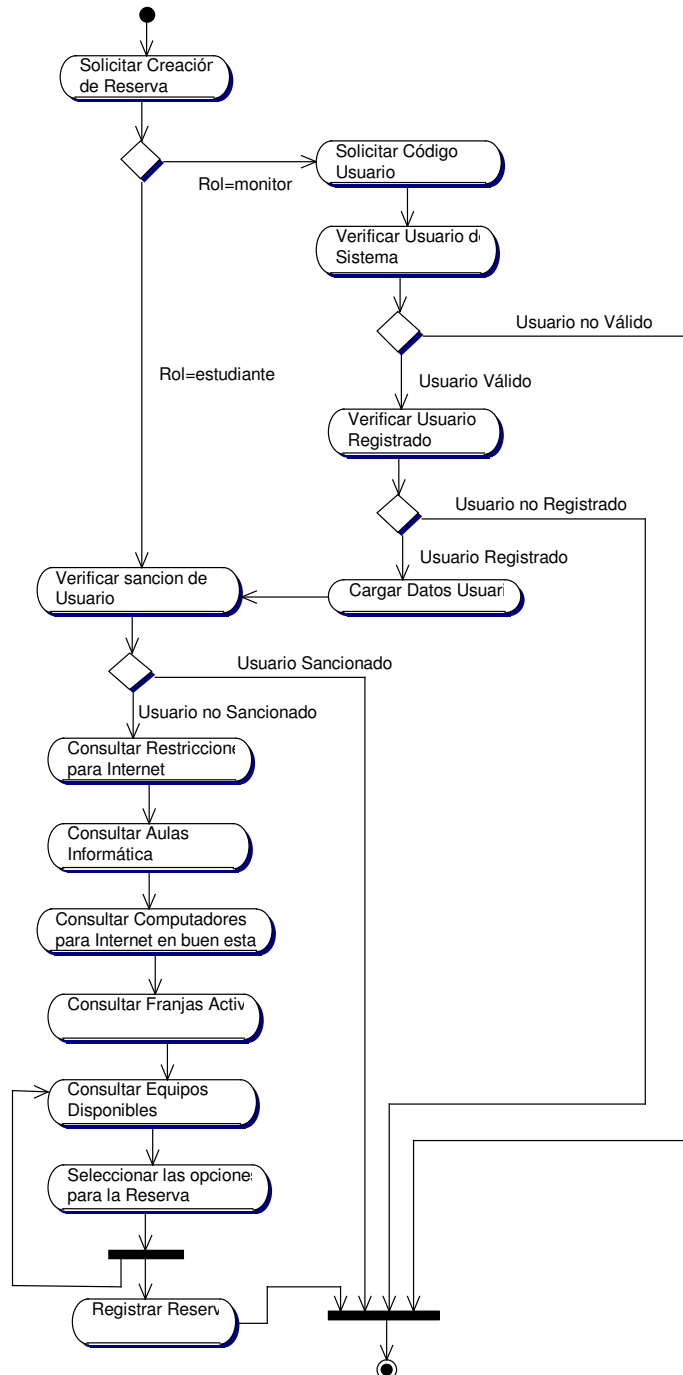
## 10.3 DIAGRAMA DE ACTIVIDADES

### 10.3.1 Diagrama Control de Acceso al Sistema

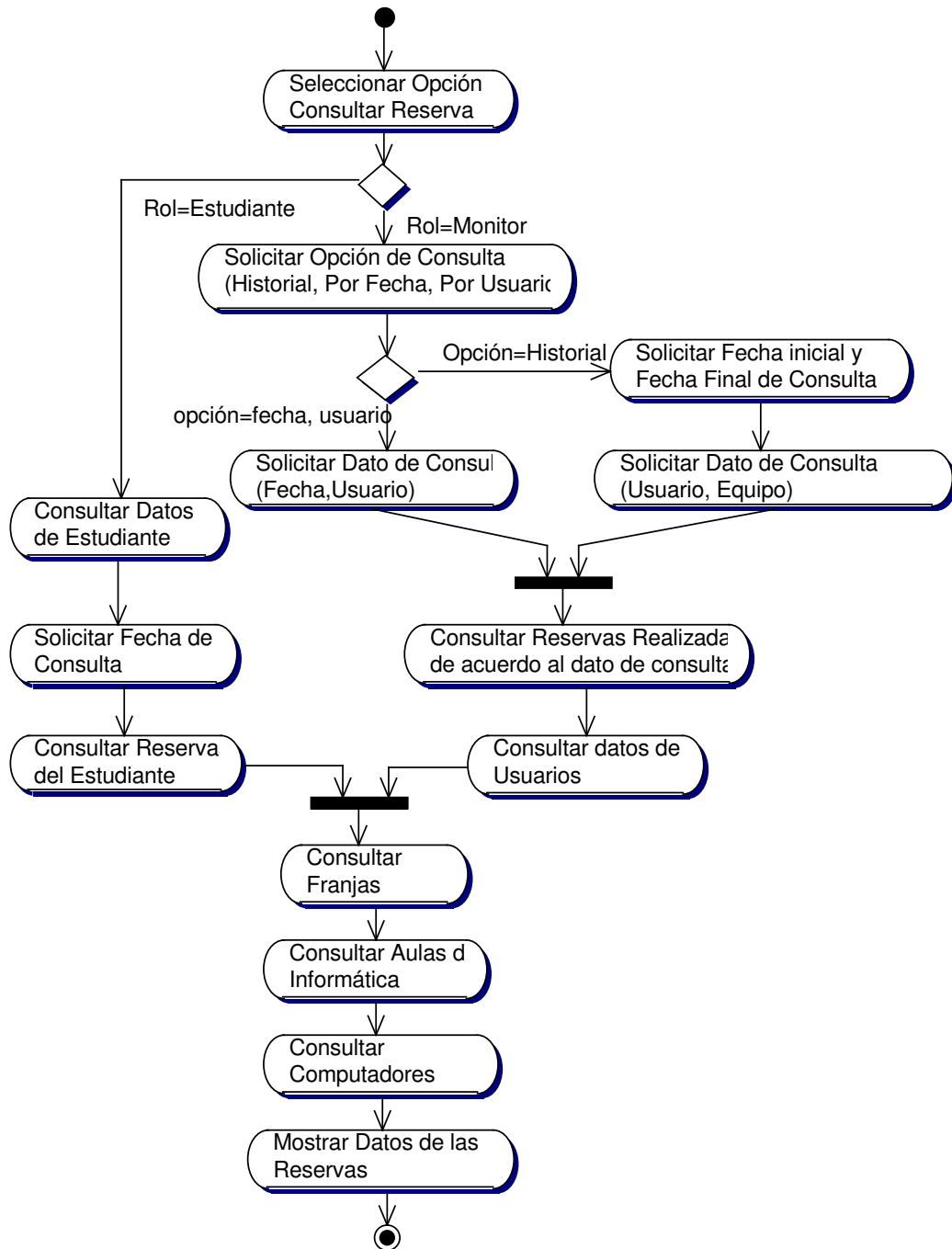


## 10.3.2 Administración de Reservas de Internet

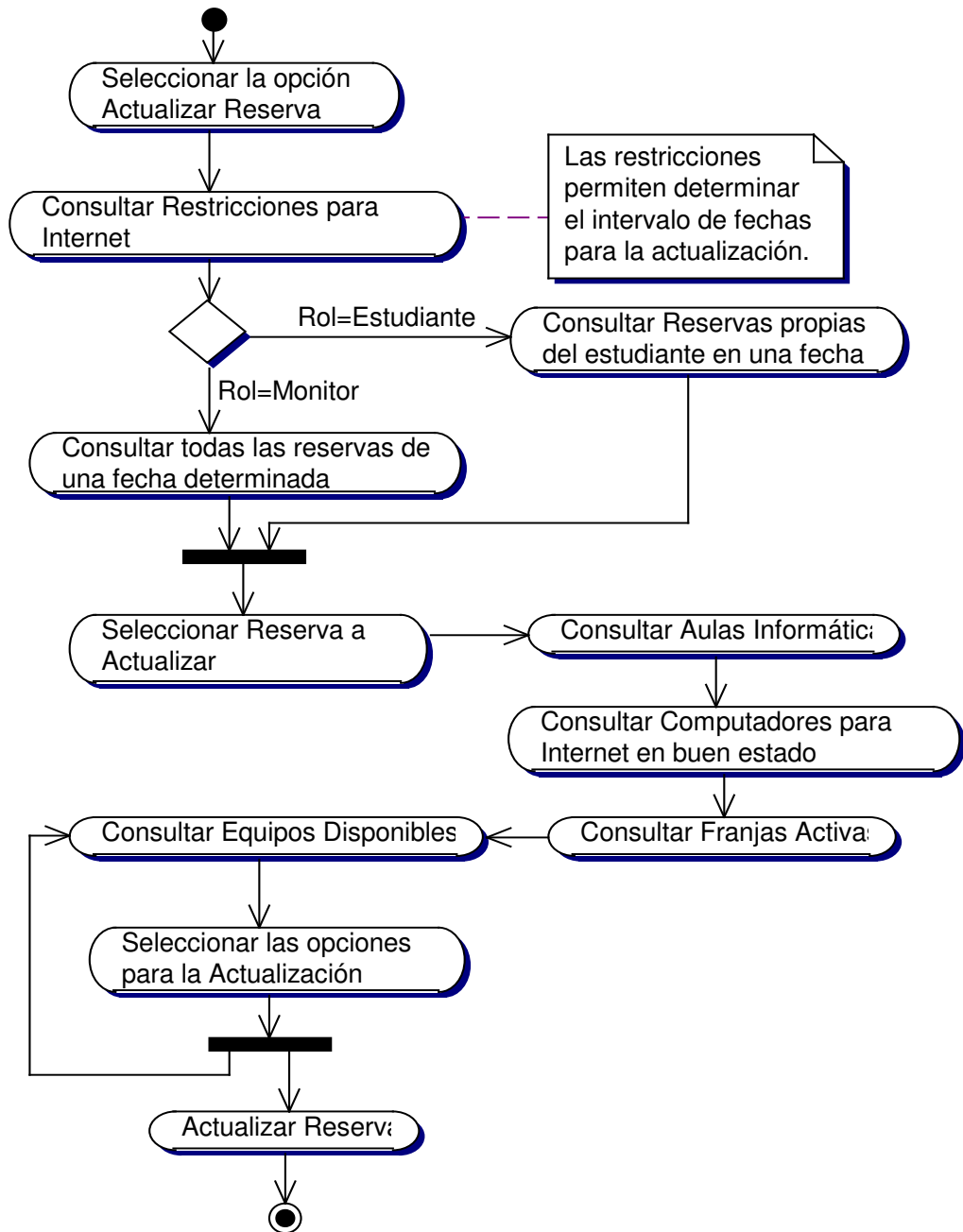
### 10.3.2.1 Diagrama Crear Reservas de Internet



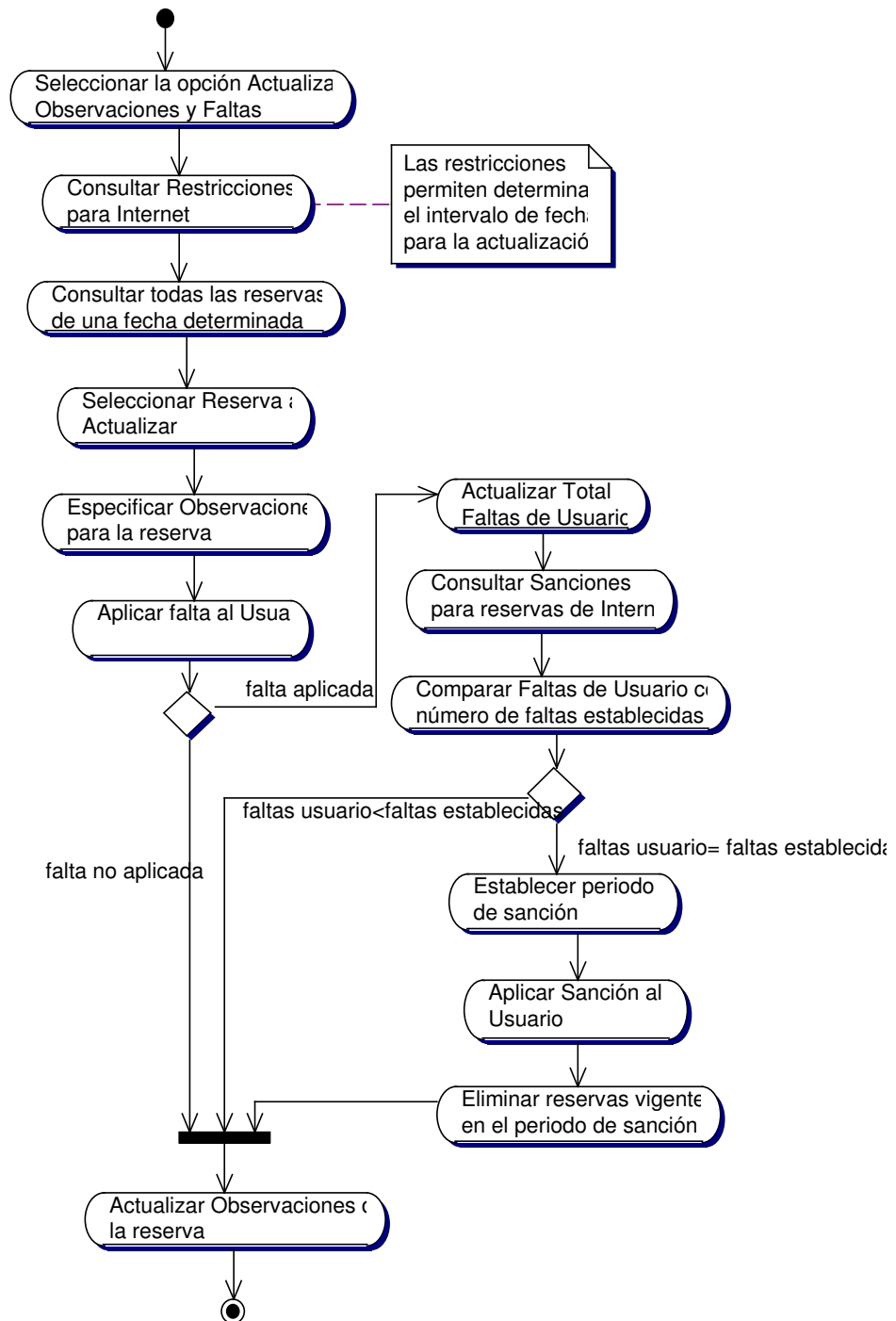
### 10.3.2.2 Diagrama Consultar Reservas de Internet



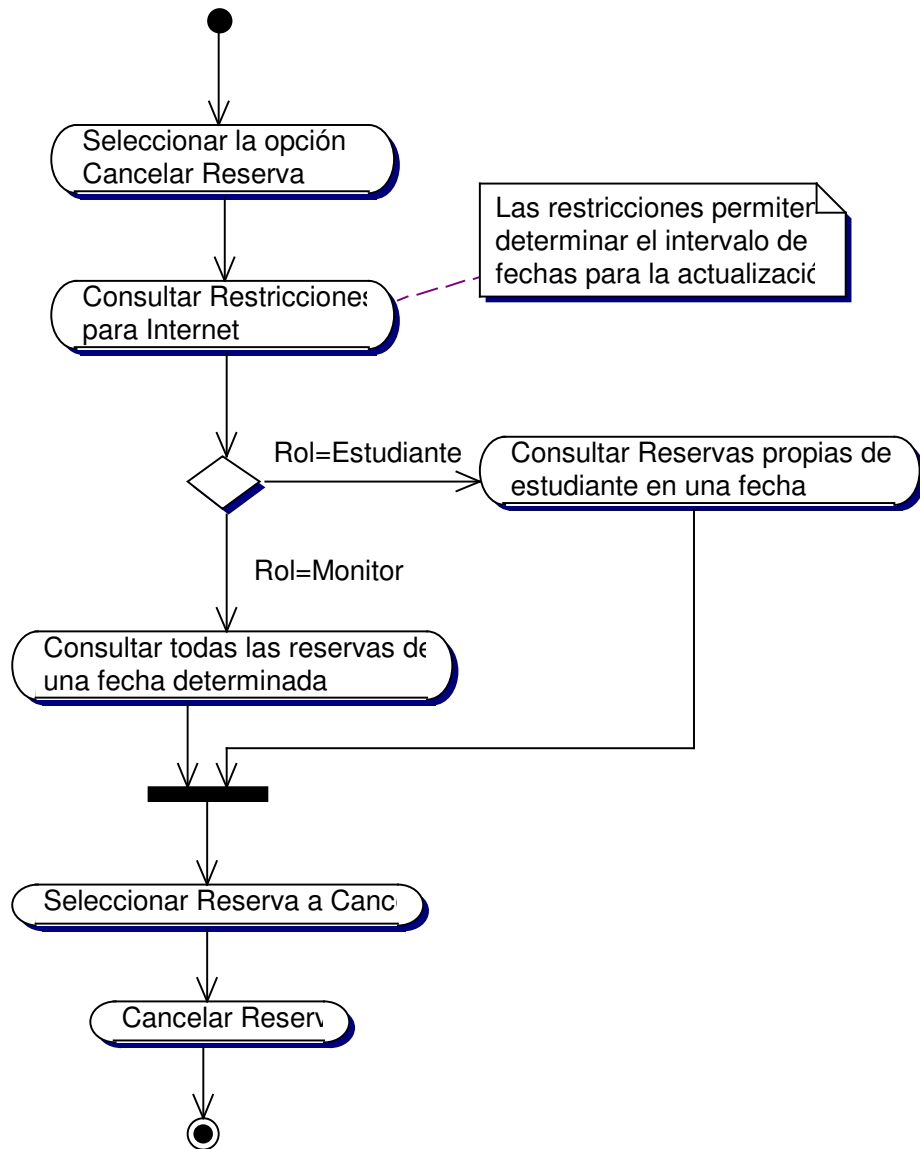
### 10.3.2.3 Diagrama Actualizar Reservas de Internet



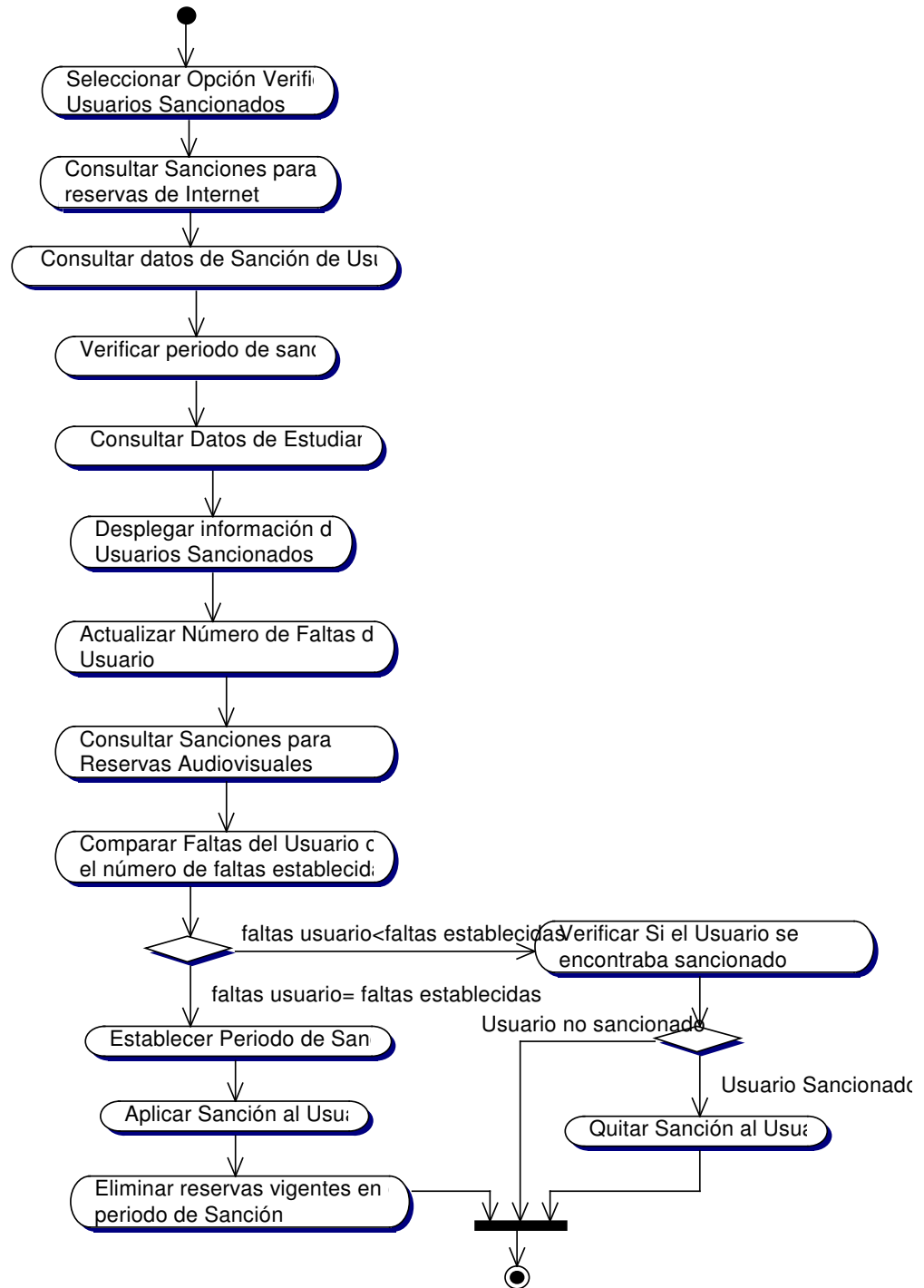
### 10.3.2.4 Diagrama Actualizar Observaciones y Faltas de Reservas de Internet



### 10.3.2.5 Diagrama Cancelar Reservas de Internet

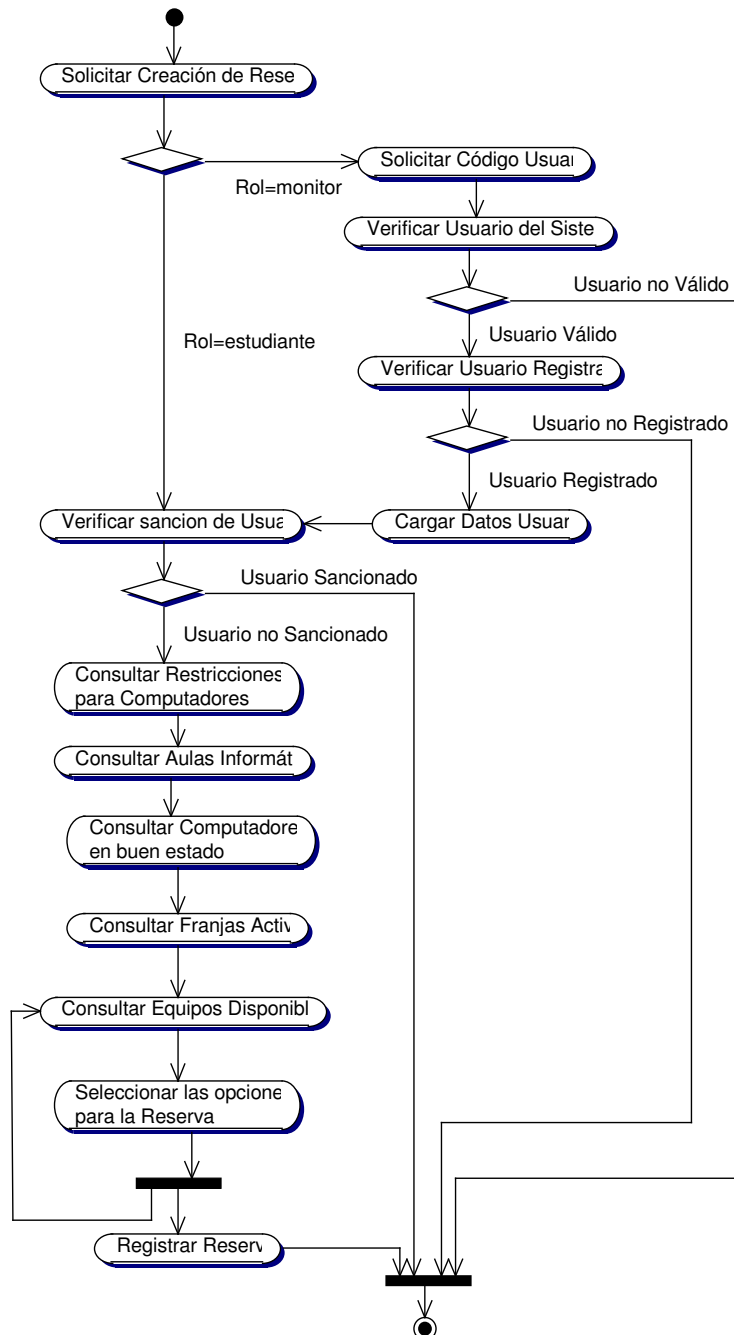


### 10.3.2.6 Diagrama Verificar Usuarios Sancionados para Reservas de Internet



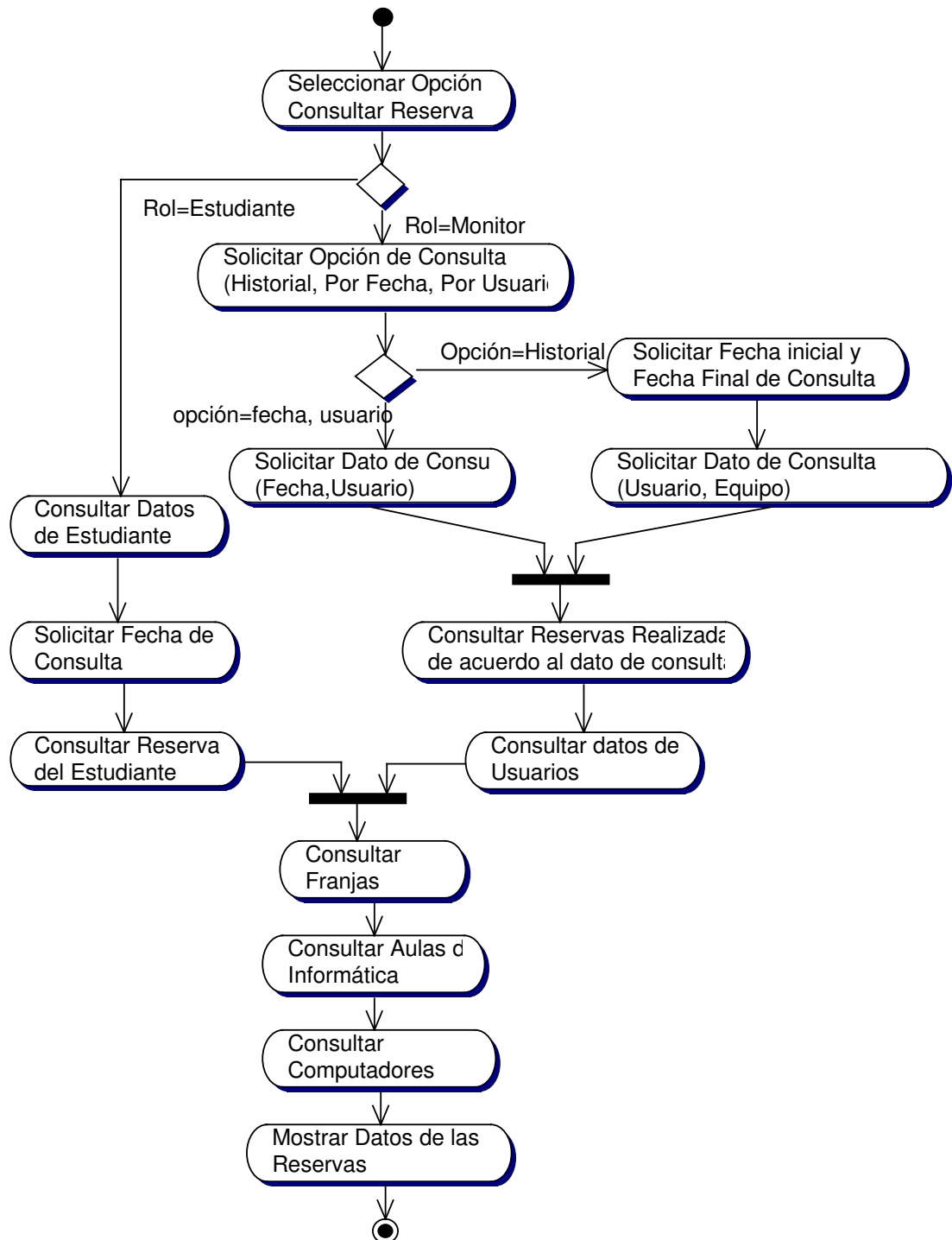
### 10.3.3 Administración de Reservas de Computadores

#### 10.3.3.1 Crear Reservas de Computadores

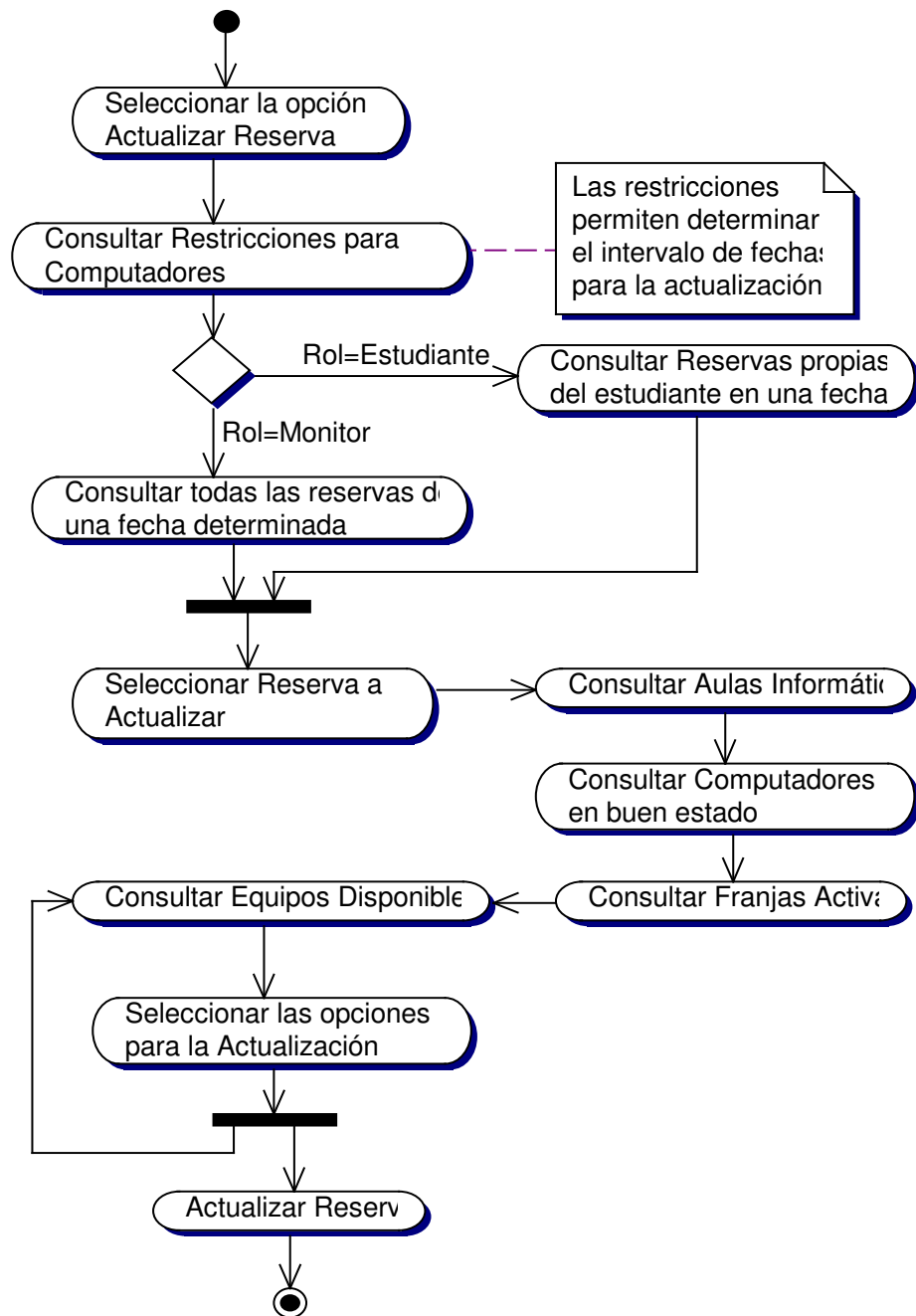




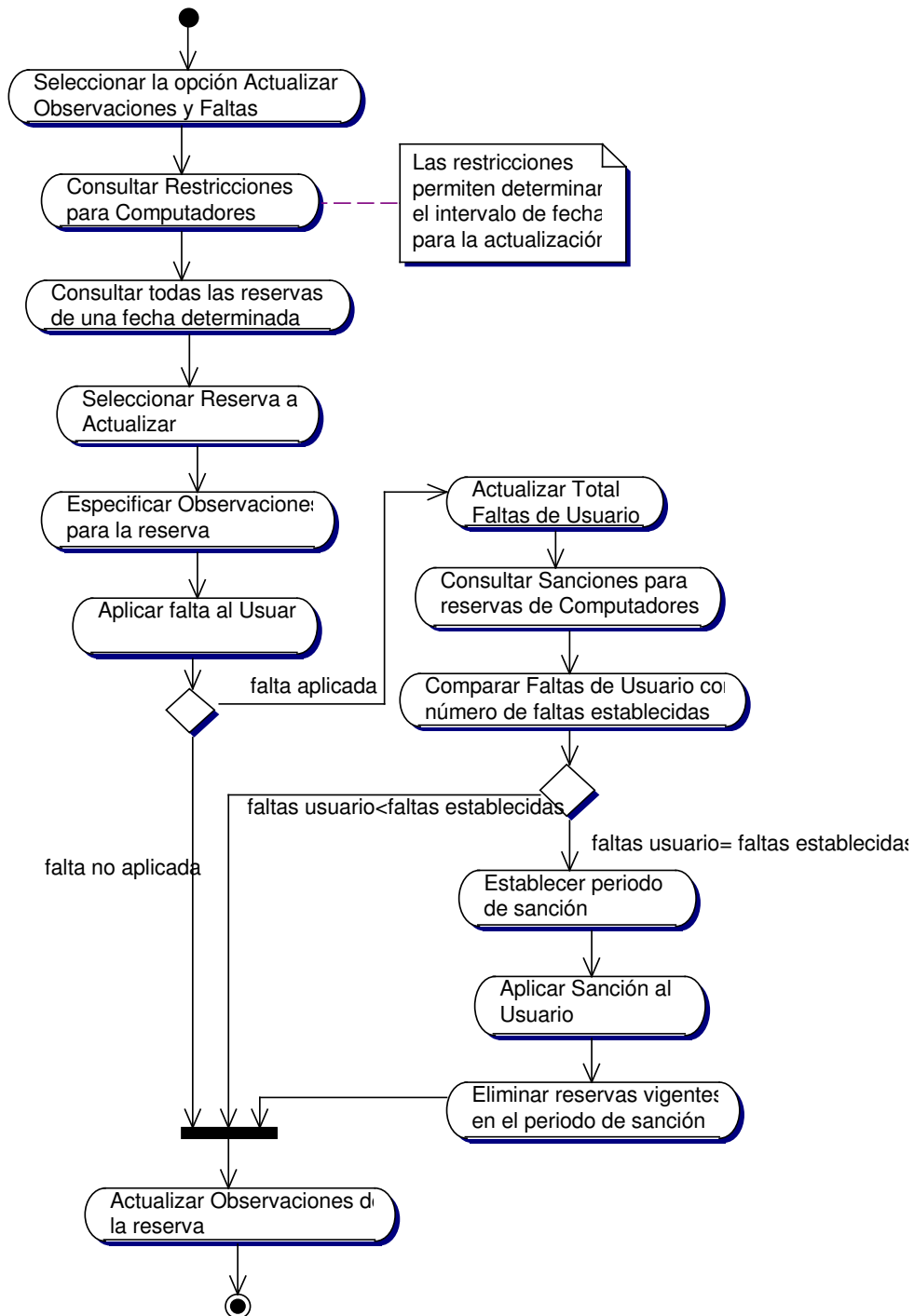
### 10.3.3.2 Consultar Reservas de Computadores



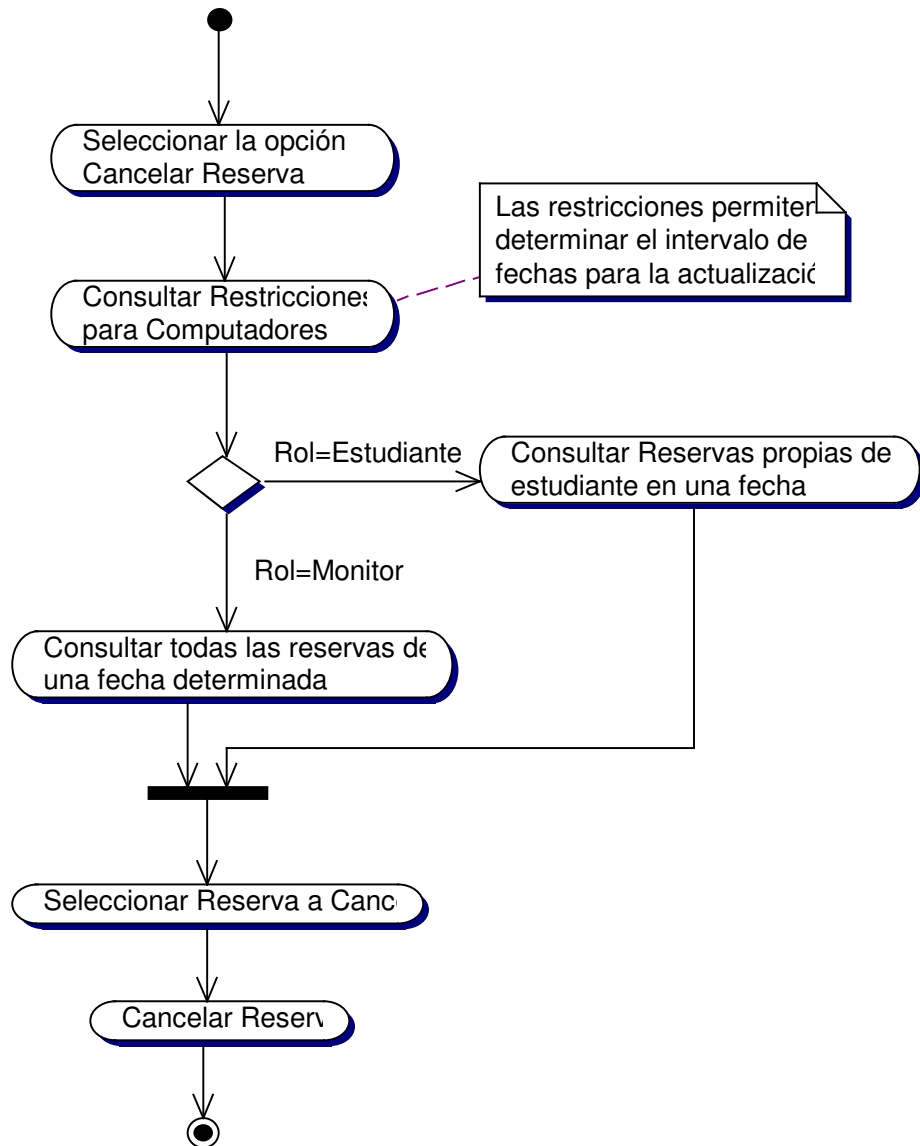
### 10.3.3.3 Actualizar Reservas de Computadores



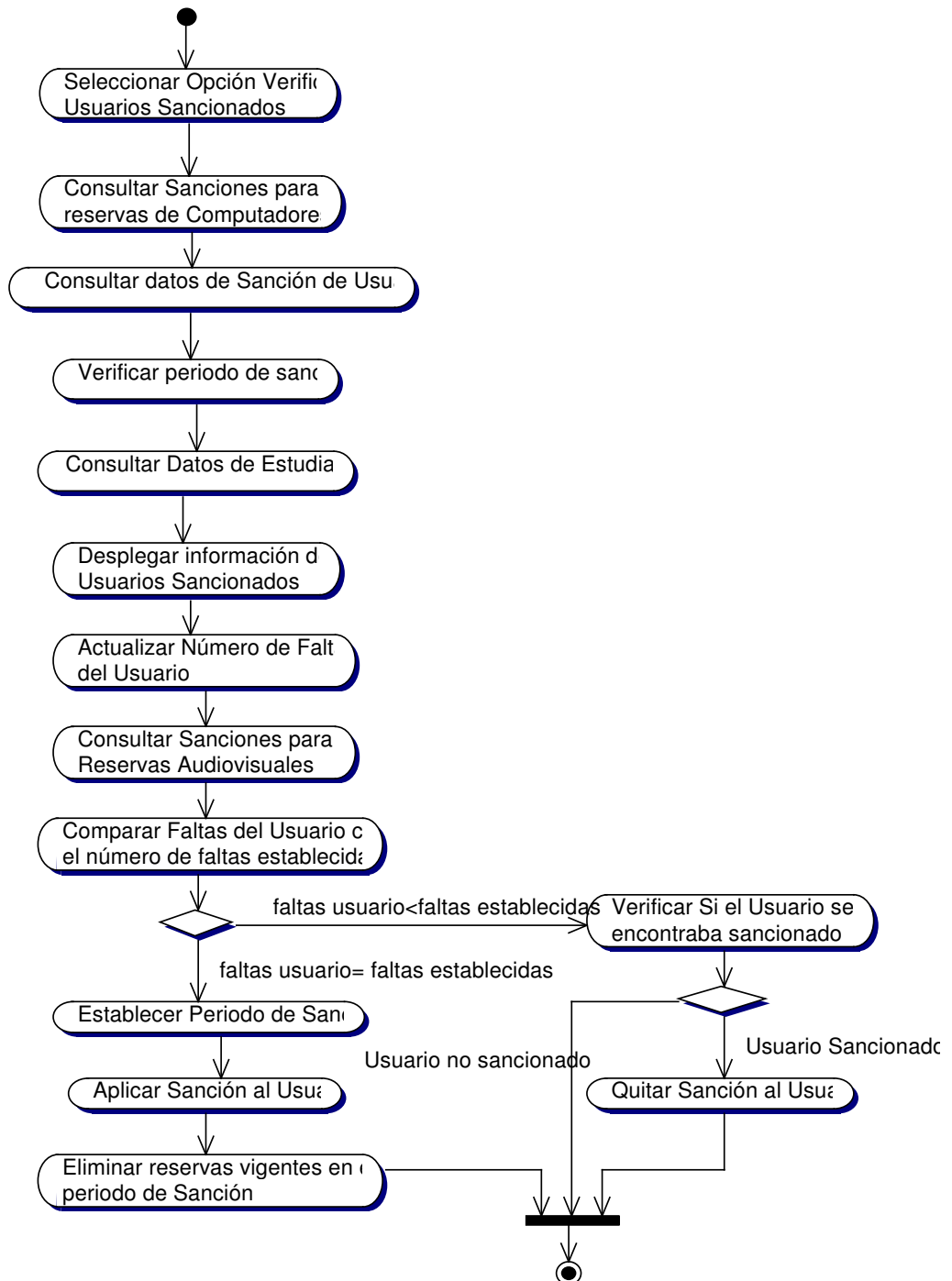
### 10.3.3.4 Diagrama Actualizar Observaciones y Faltas de Reservas de Computadores



### 10.3.3.5 Diagrama Cancelar Reservas de Computadores

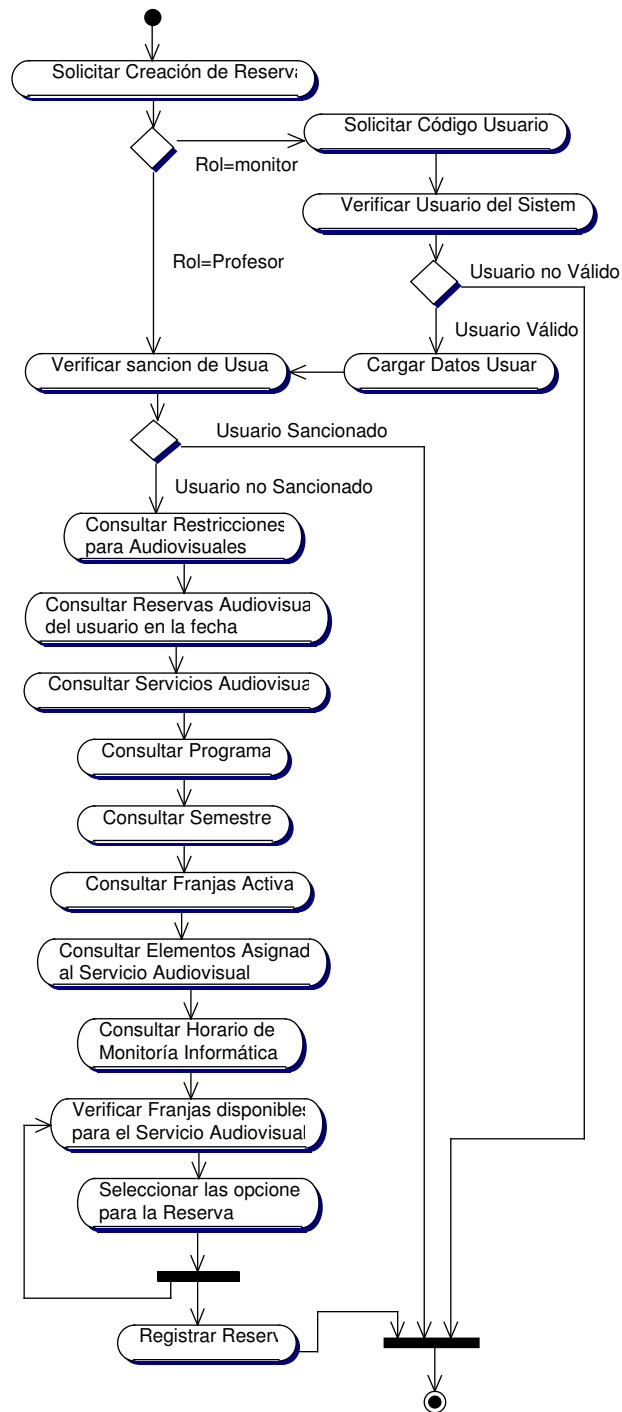


### 10.3.3.6 Diagrama Verificar Usuarios Sancionados para Reservas de Computadores

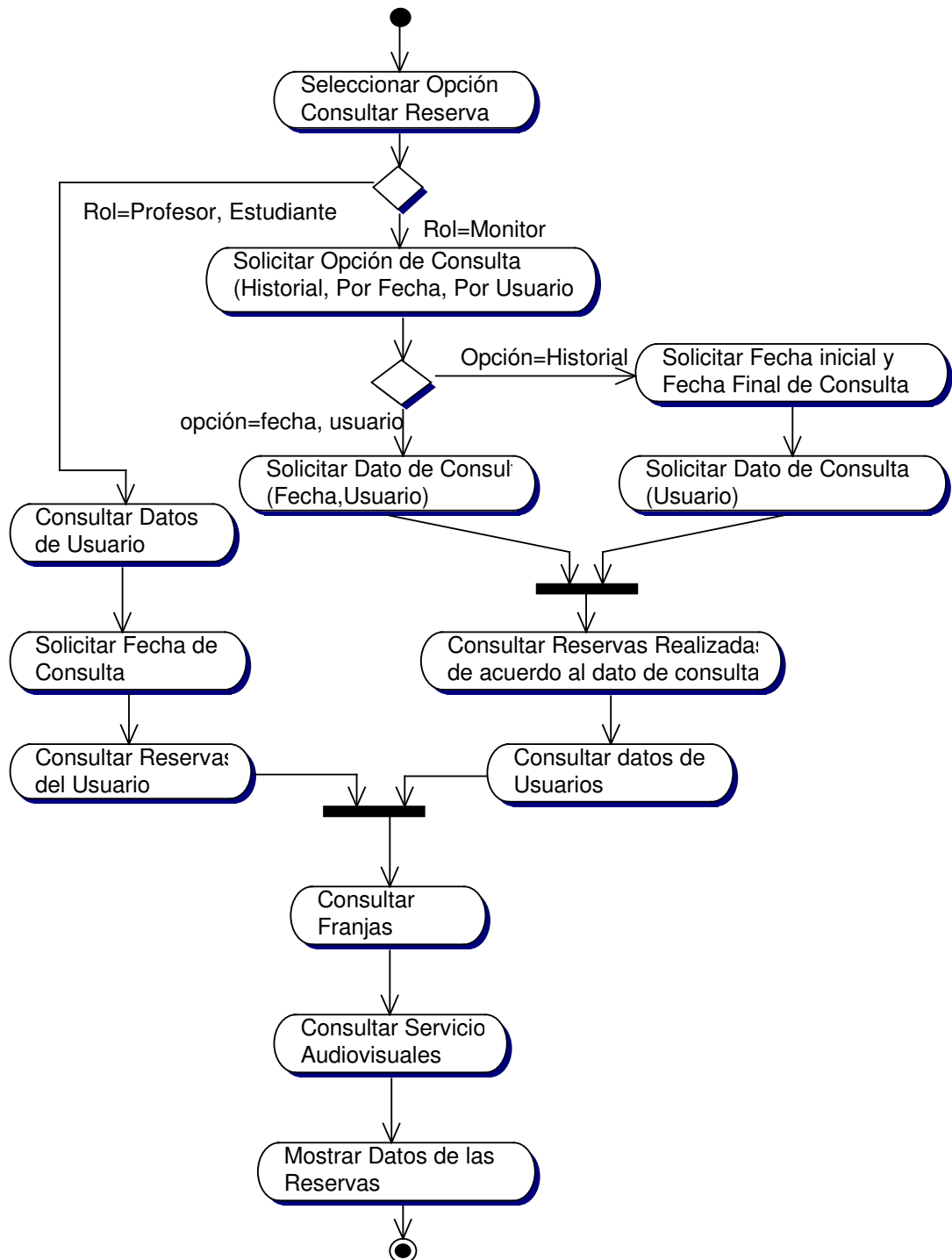


## 10.3.4 Administración de Reservas Audiovisuales

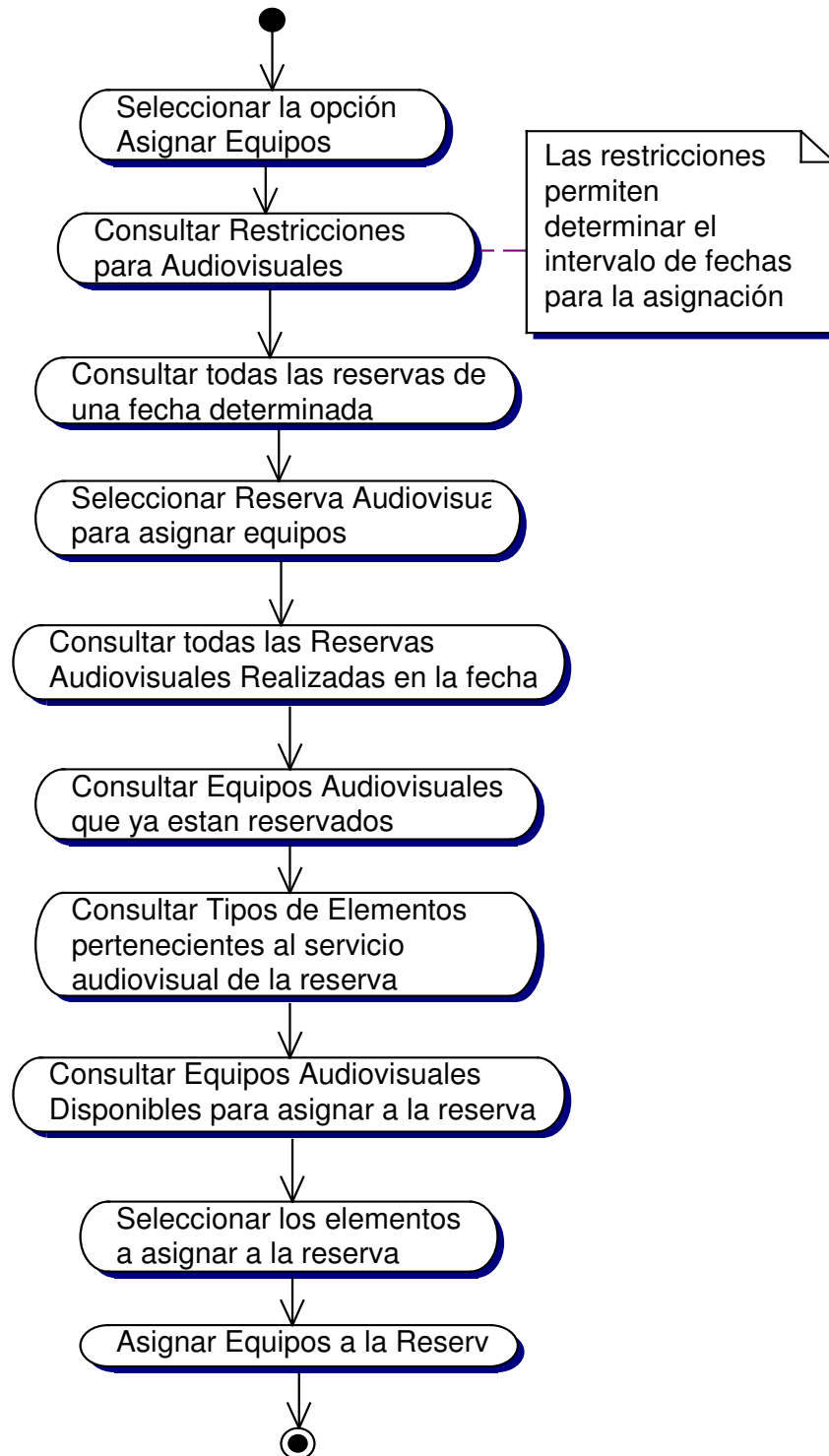
### 10.3.4.1 Diagrama Crear Reservas Audiovisuales



### 10.3.4.2 Diagrama Consultar Reservas Audiovisuales

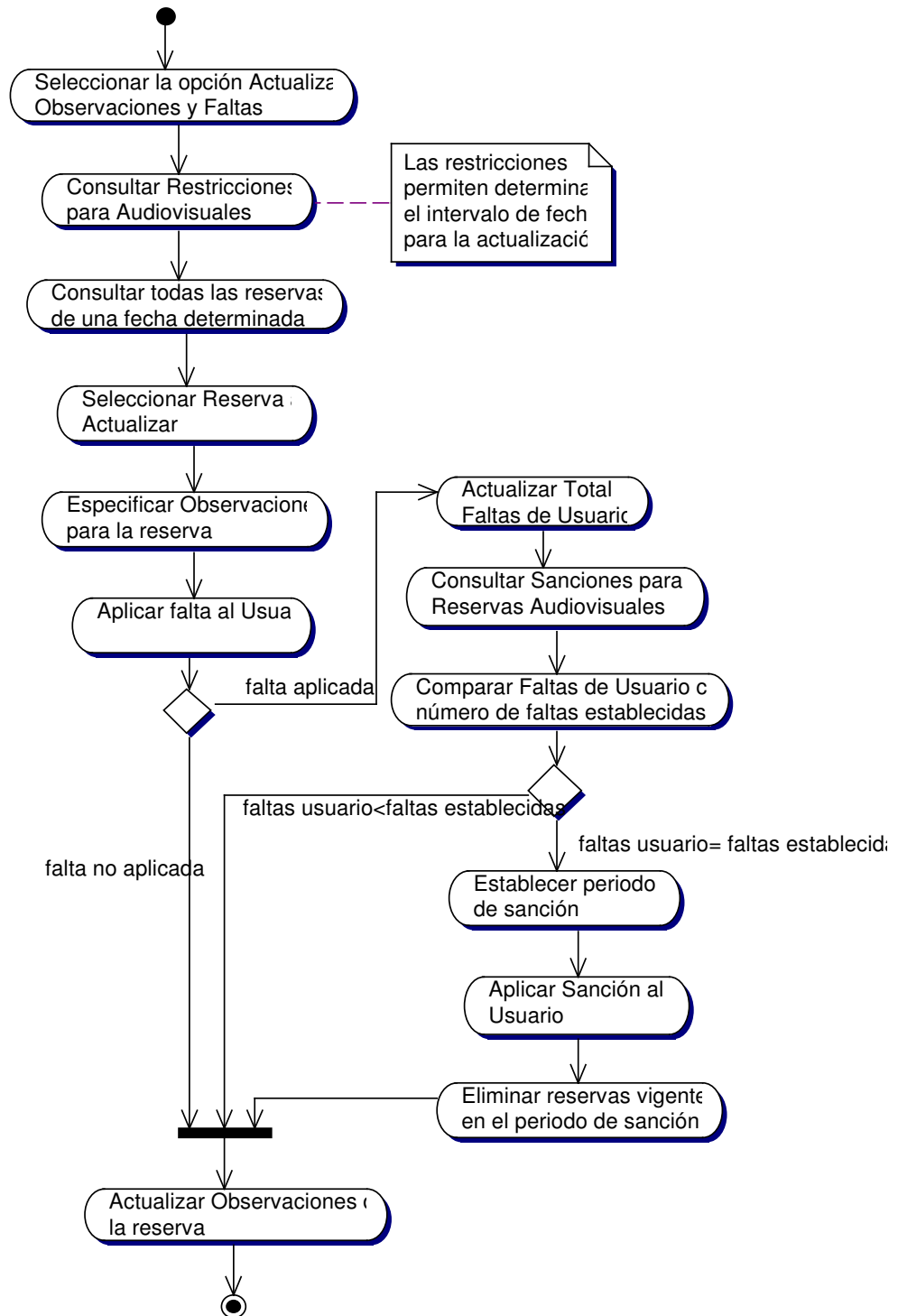


### 10.3.4.3 Diagrama Actualizar Equipos Asignados a la Reserva Audiovisual

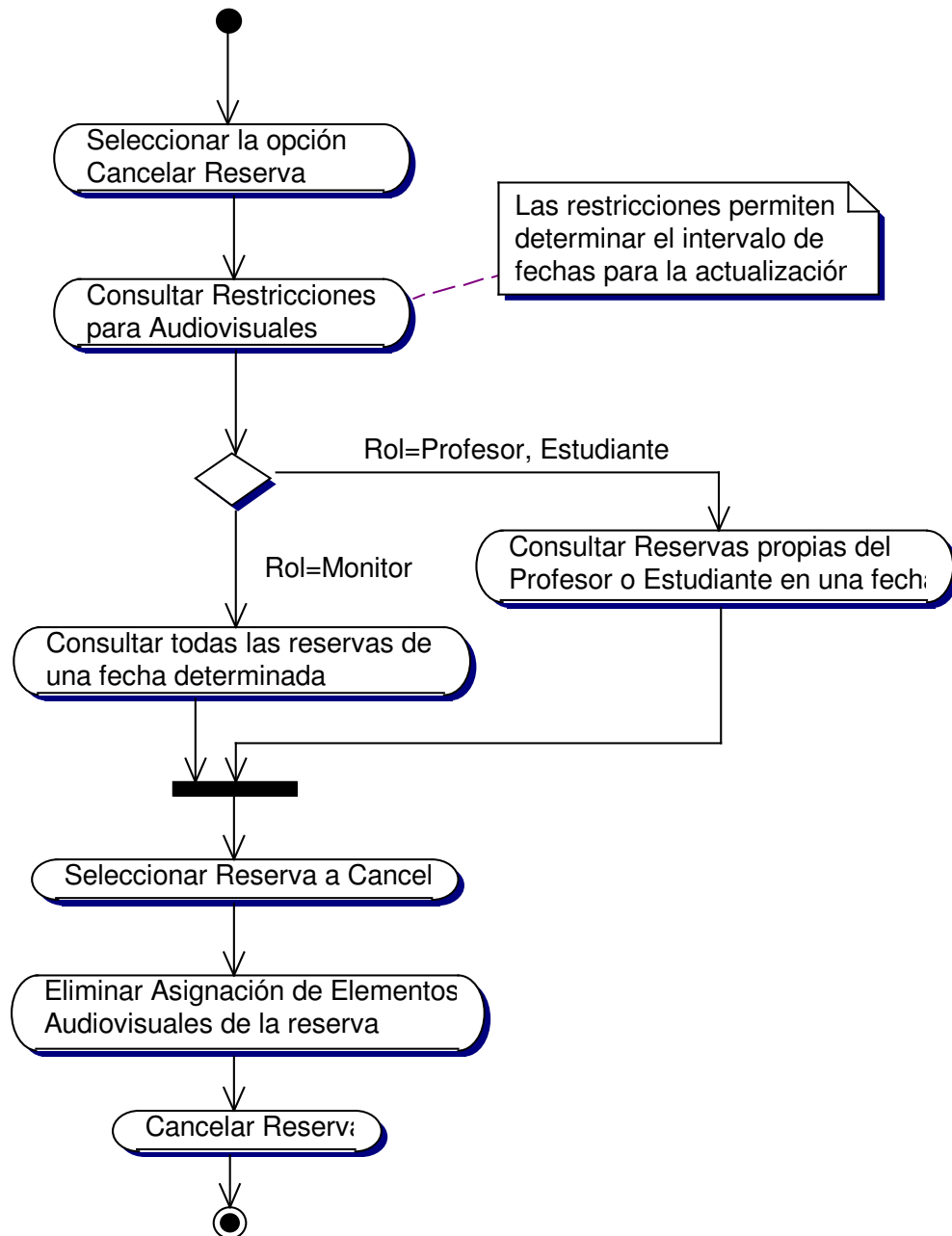




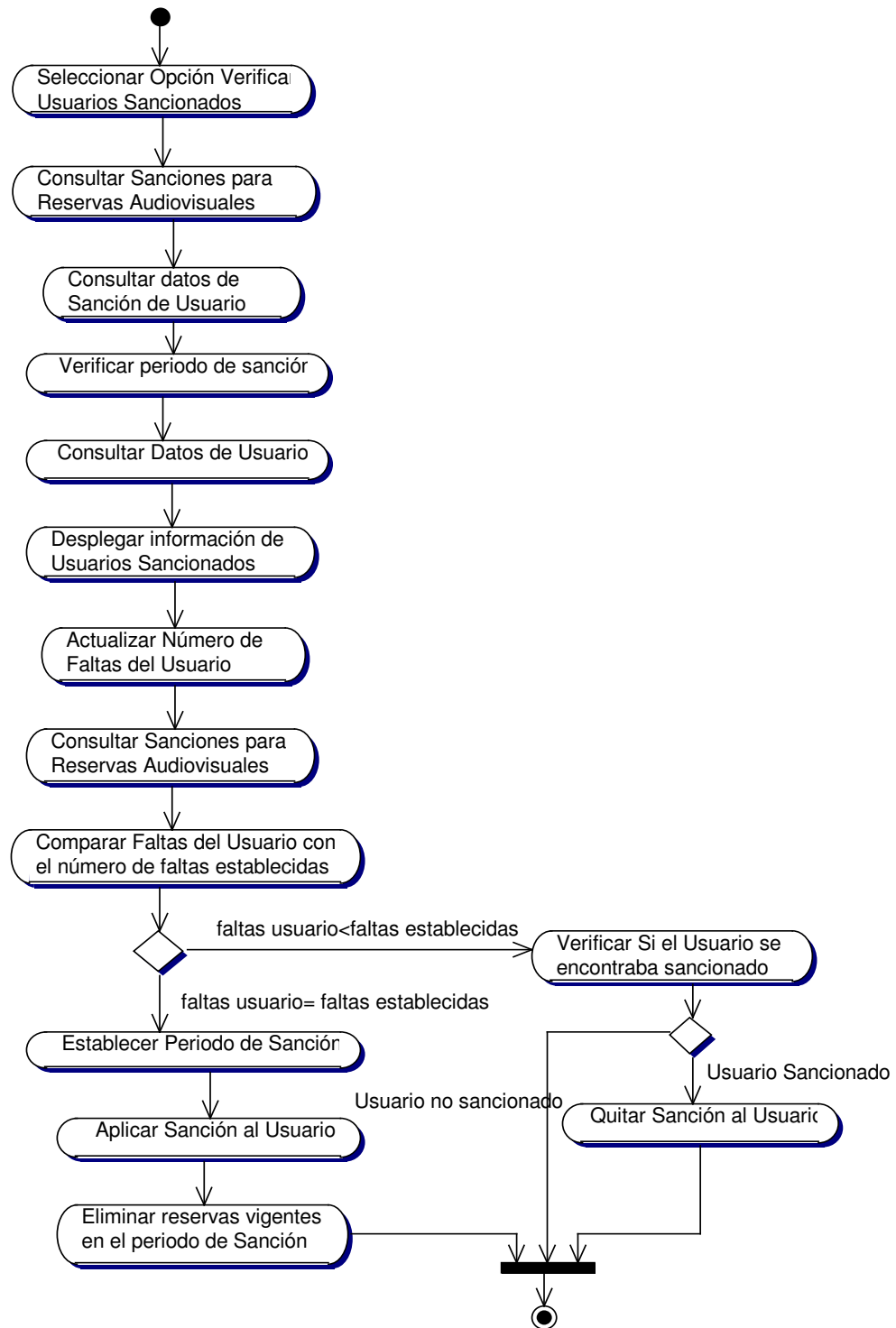
### 10.3.4.4 Diagrama Actualizar Observaciones y Faltas de Reservas Audiovisuales



### 10.3.4.5 Diagrama Cancelar Reservas Audiovisuales

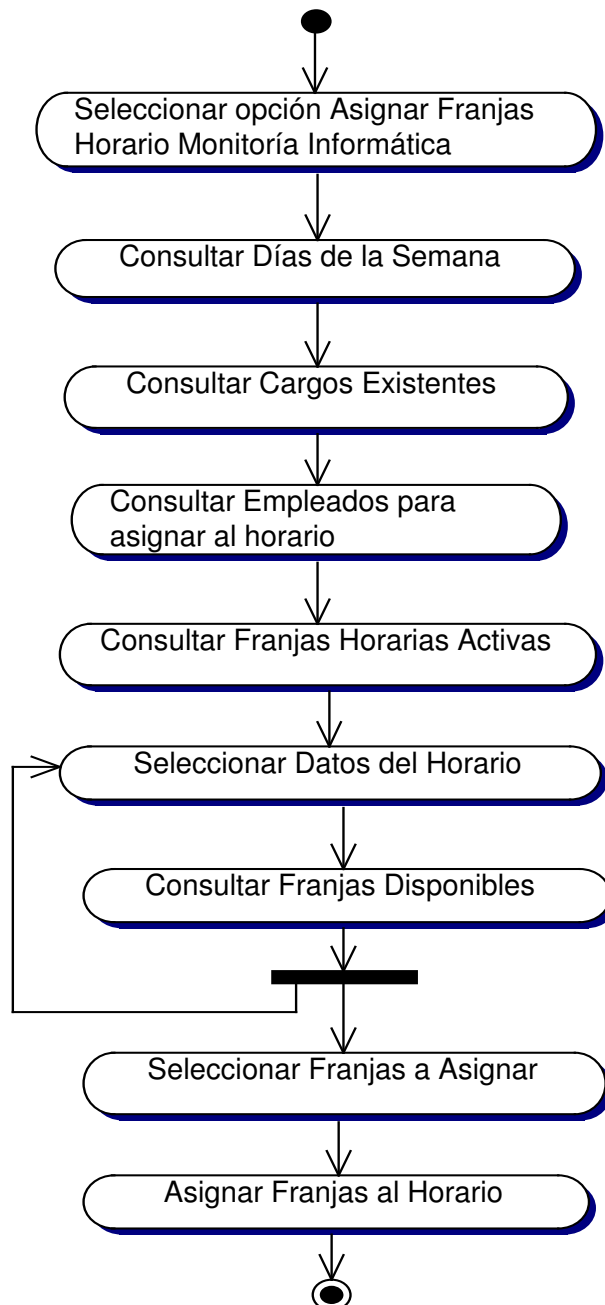


### 10.3.4.6 Diagrama Verificar Usuarios Sancionados para Reservas Audiovisuales

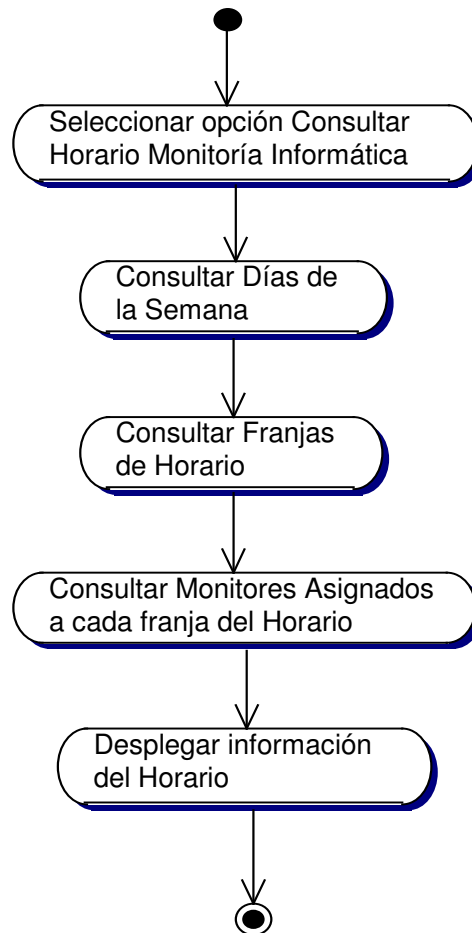


### 10.3.5 Administración de Horario de Monitoría Informática

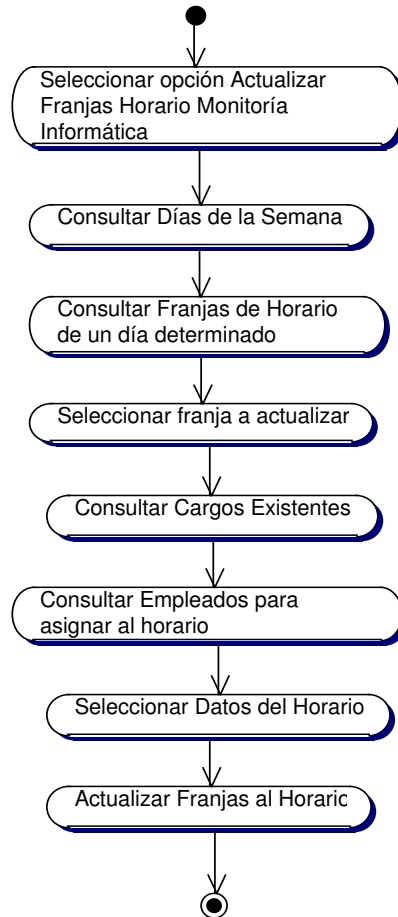
#### 10.3.5.1 Diagrama Asignar Franjas Horario Monitoría Informática



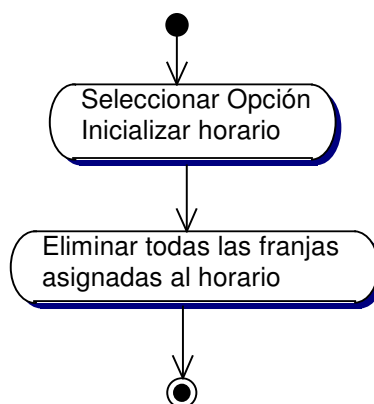
### 10.3.5.2 Diagrama Consultar Horario Monitoría Informática



### 10.3.5.3 Diagrama Actualizar Franja Horario Monitoría Informática

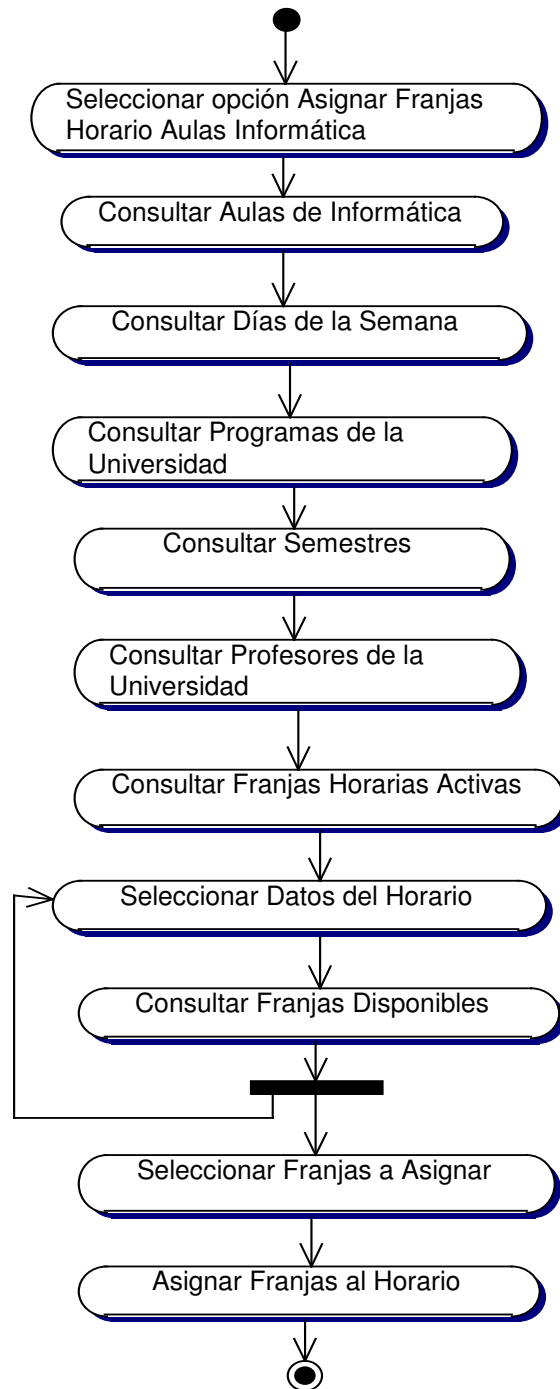


### 10.3.5.4 Diagrama Inicializar Horario Monitoría Informática

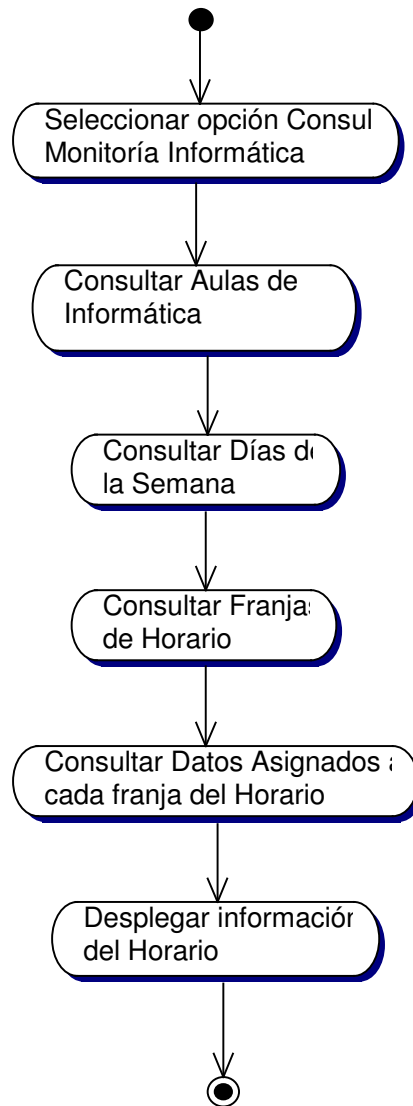


### 10.3.6 Administración de Horarios de Aulas de Informática

#### 10.3.6.1 Diagrama Asignar Franjas Horarios Aulas de Informática

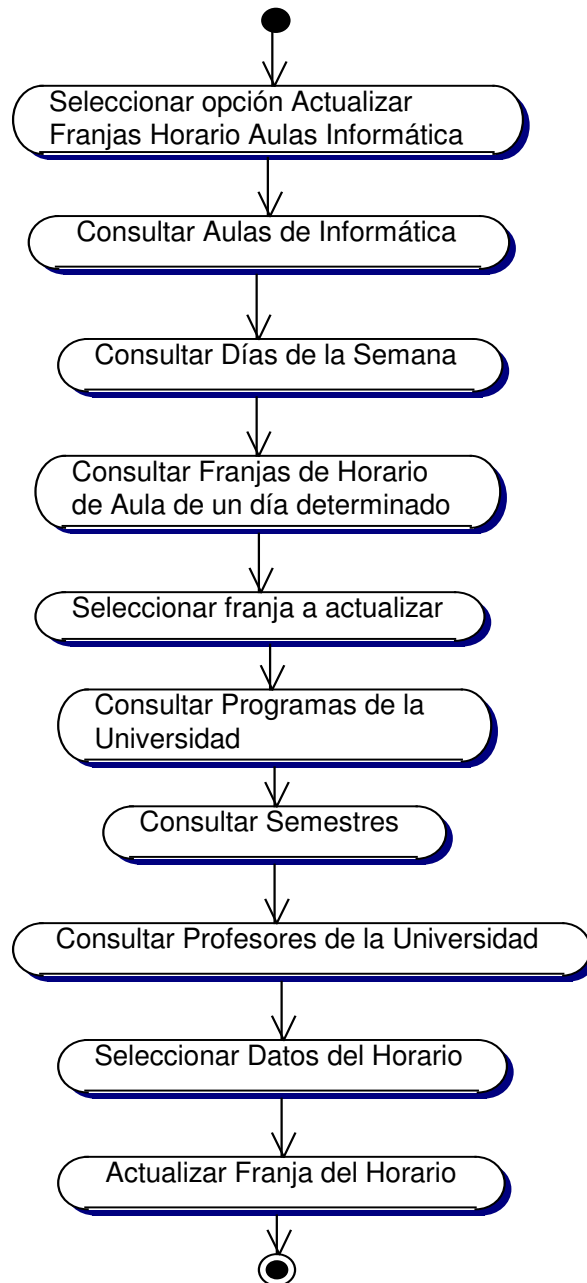


### 10.3.6.2 Diagrama Consultar Horario Aulas de Informática

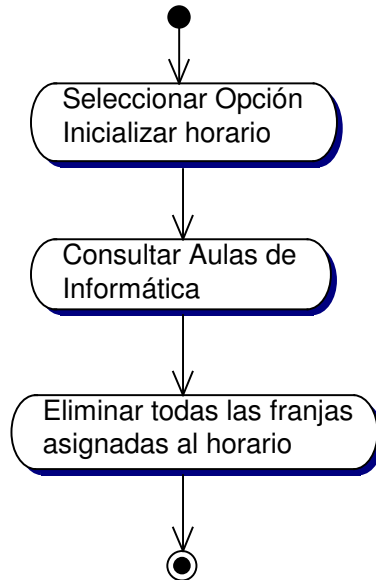




### 10.3.6.3 Diagrama Actualizar Franja Horarios Aulas de Informática

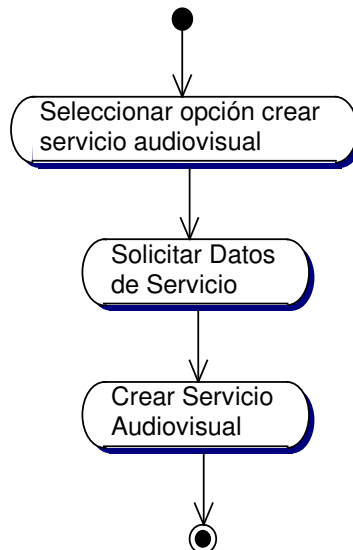


#### 10.3.6.4 Diagrama Inicializar Horarios Aulas de Informática

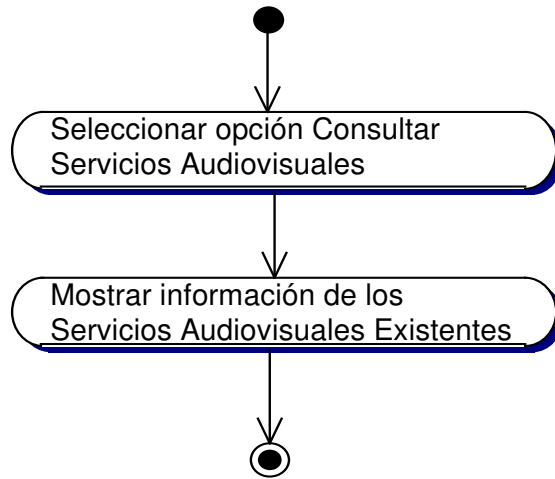


#### 10.3.7 Administración de Servicios Audiovisuales

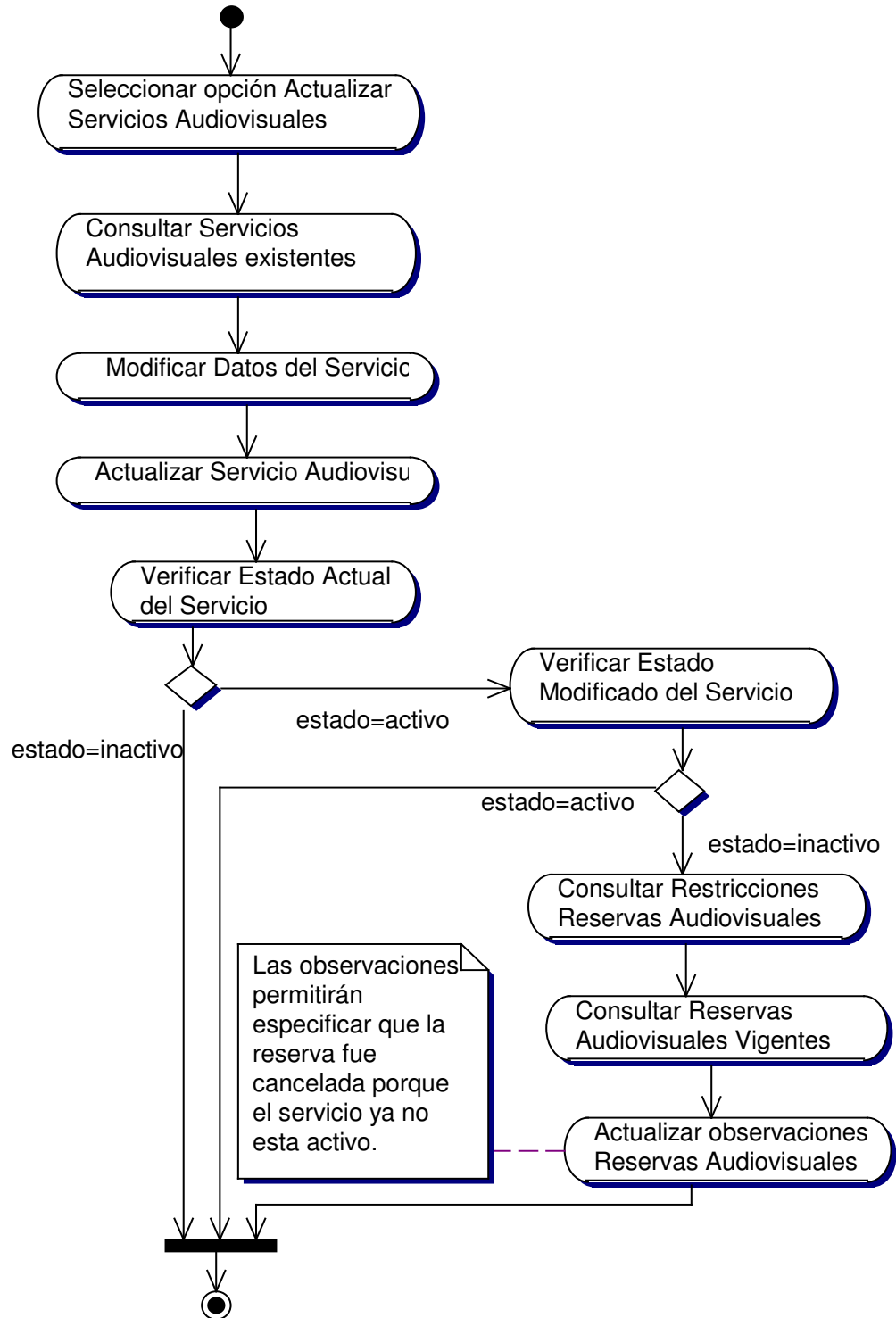
##### 10.3.7.1 Diagrama Crear Servicio Audiovisual



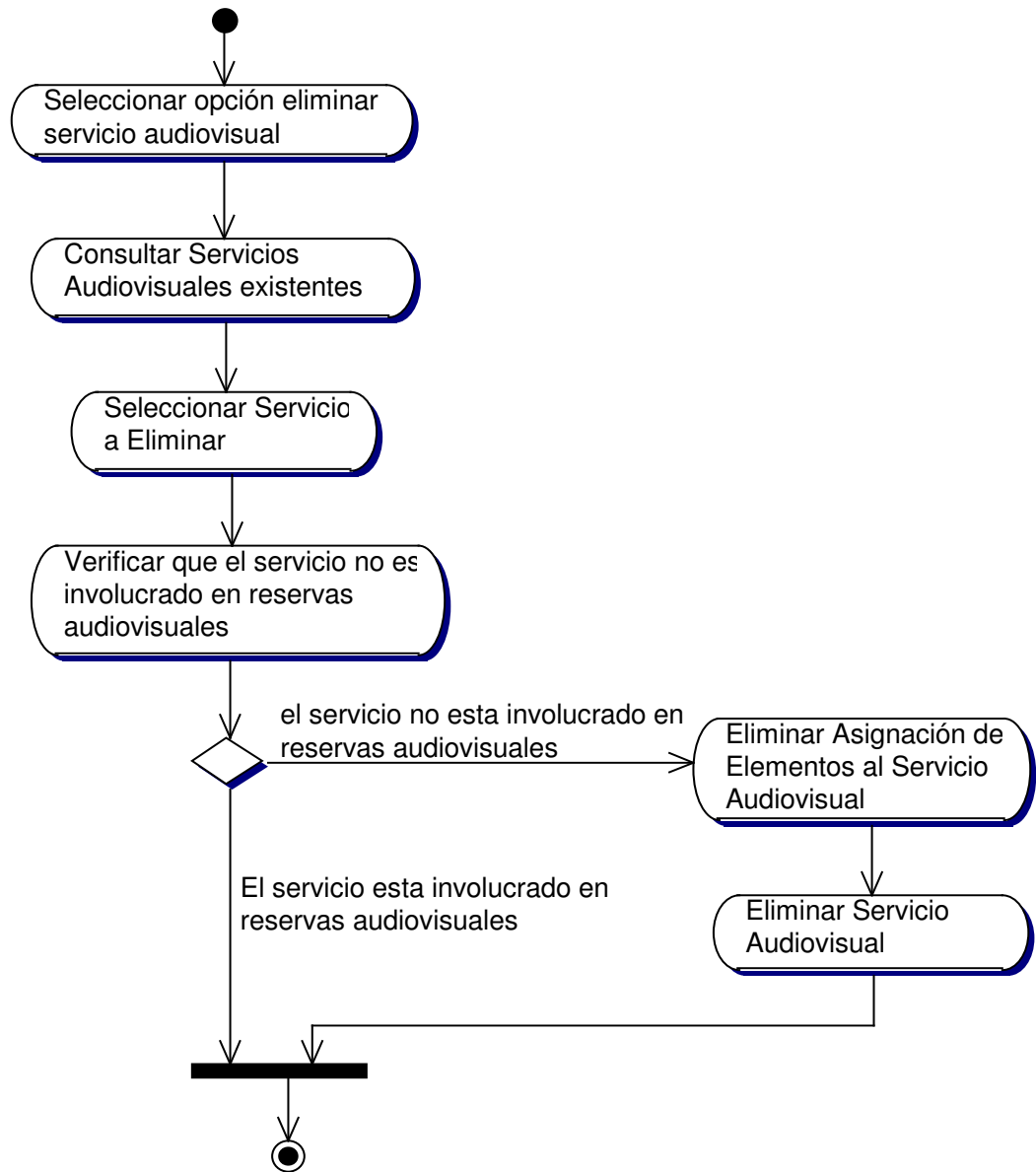
### 10.3.7.2 Diagrama Consultar Servicio Audiovisual



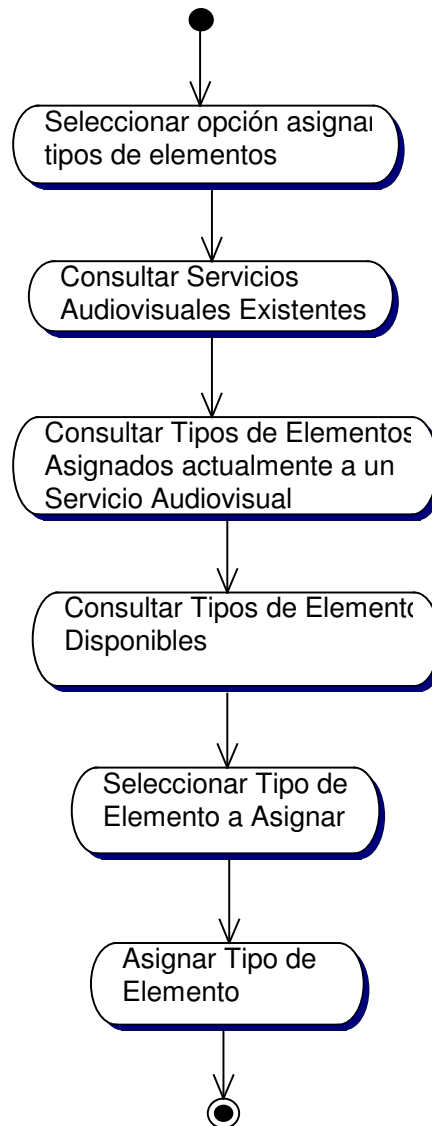
### 10.3.7.3 Diagrama Actualizar Servicio Audiovisual



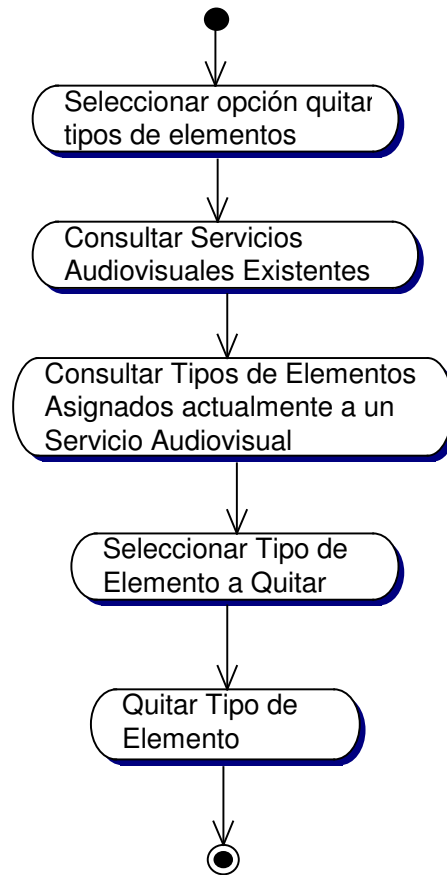
### 10.3.7.4 Diagrama Eliminar Servicio Audiovisual



### 10.3.7.5 Diagrama Asignar Tipos de Elementos

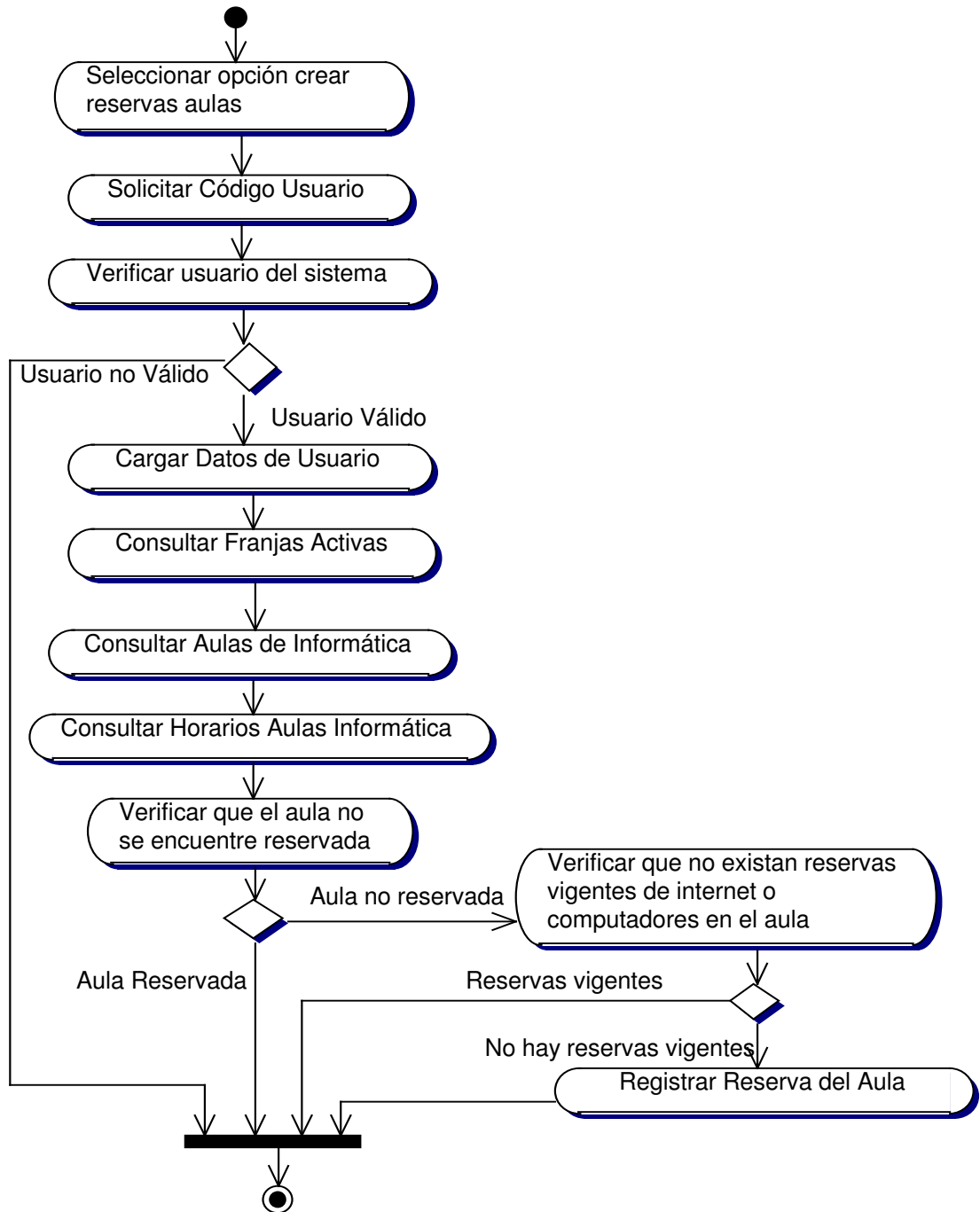


### 10.3.7.6 Diagrama Quitar Tipos de Elementos



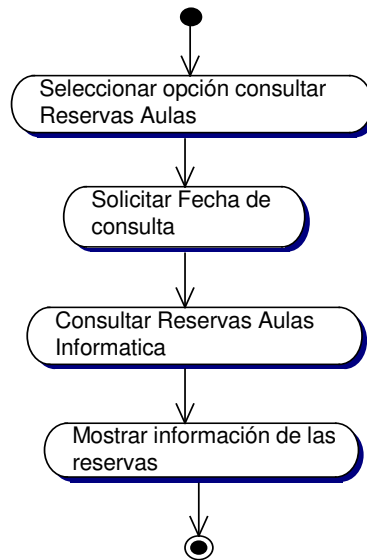
### 10.3.8 Administración de Reservas de Aulas de Informática

#### 10.3.8.1 Diagrama Crear Reserva de Aulas de informática

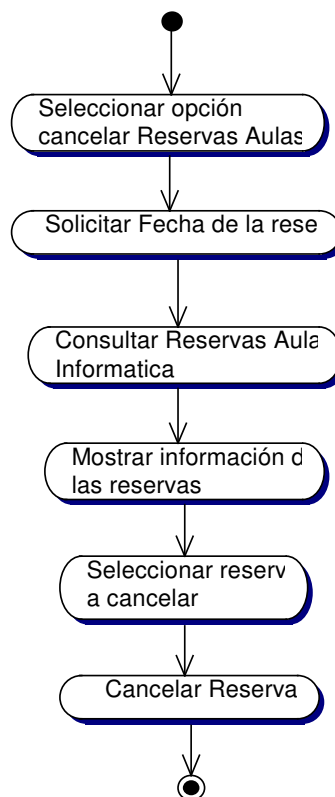




### 10.3.8.2 Diagrama Consultar Reservas de Aulas de Informática

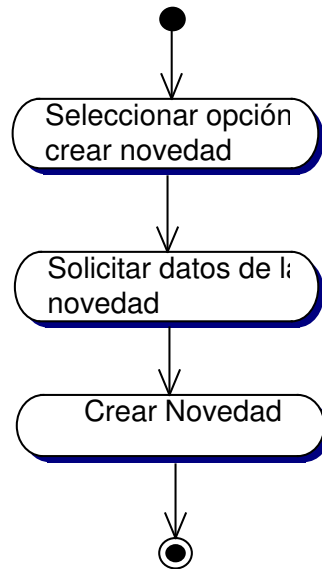


### 10.3.8.3 Diagrama Cancelar Reservas de Aulas de Informática

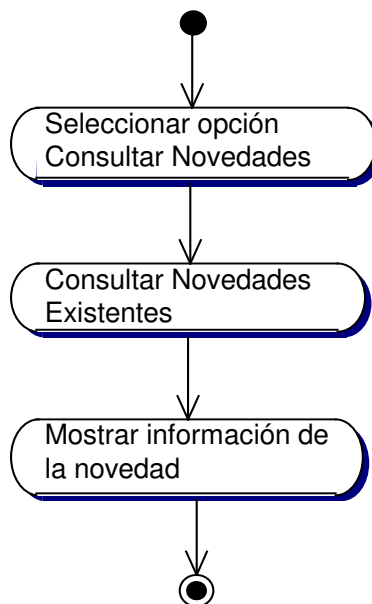


### 10.3.9 Administración de Novedades

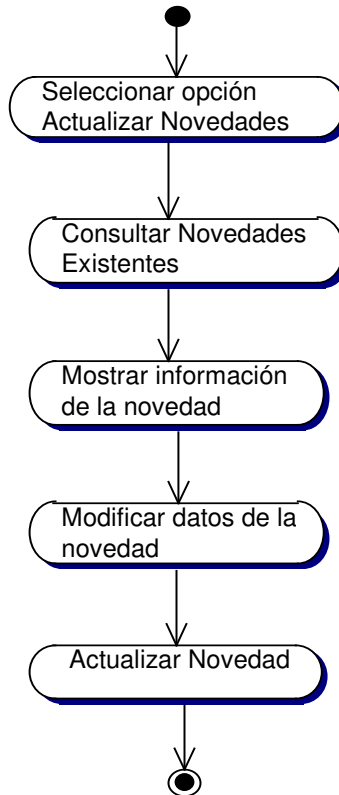
#### 10.3.9.1 Diagrama Crear Novedades



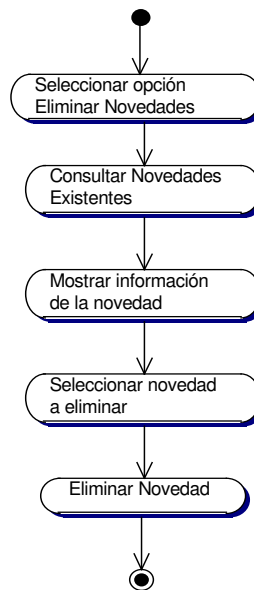
#### 10.3.9.2 Consultar Novedades



### 10.3.9.3 Actualizar Novedades

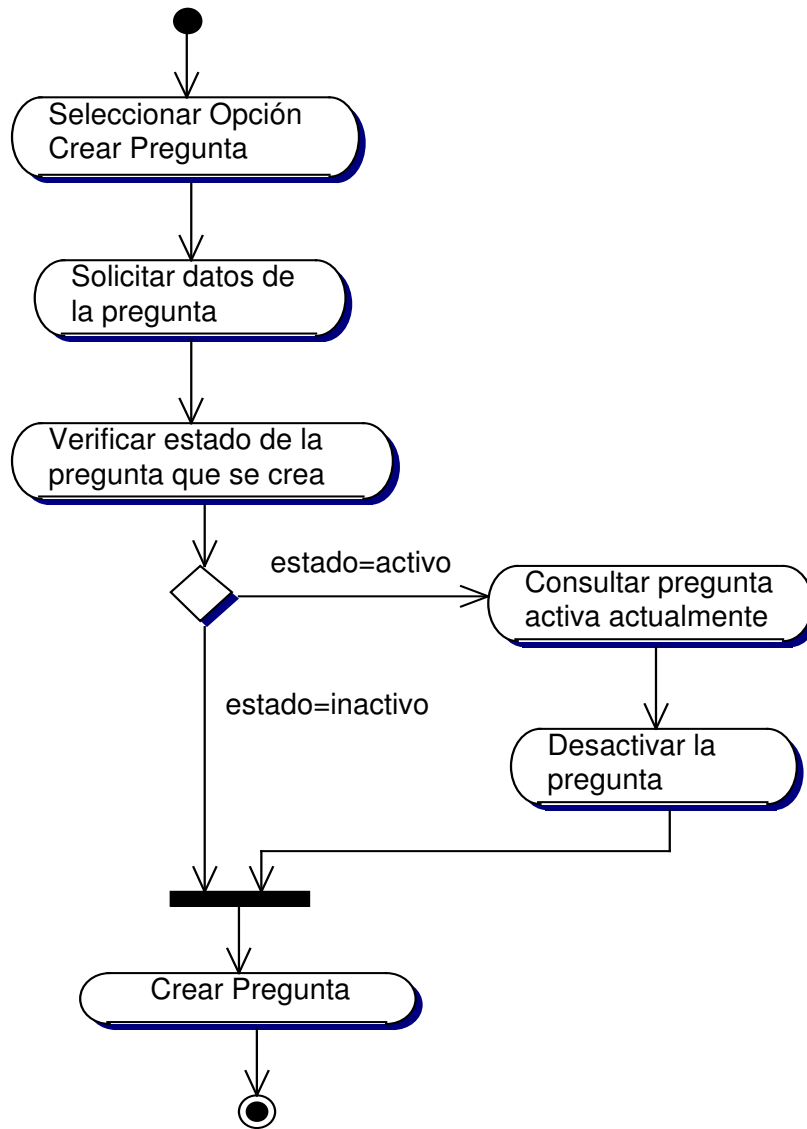


### 10.3.9.4 Diagrama Eliminar Novedades

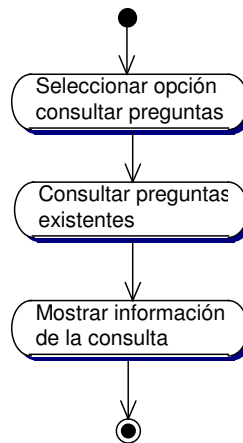


### 10.3.10 Administración de Preguntas de Urna Virtual

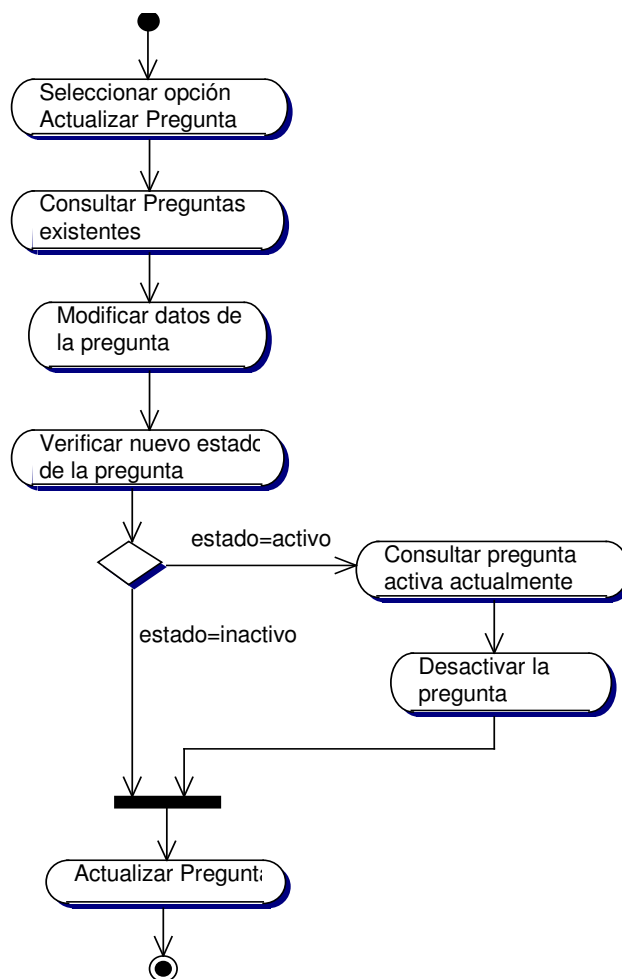
#### 10.3.10.1 Crear Pregunta Urna



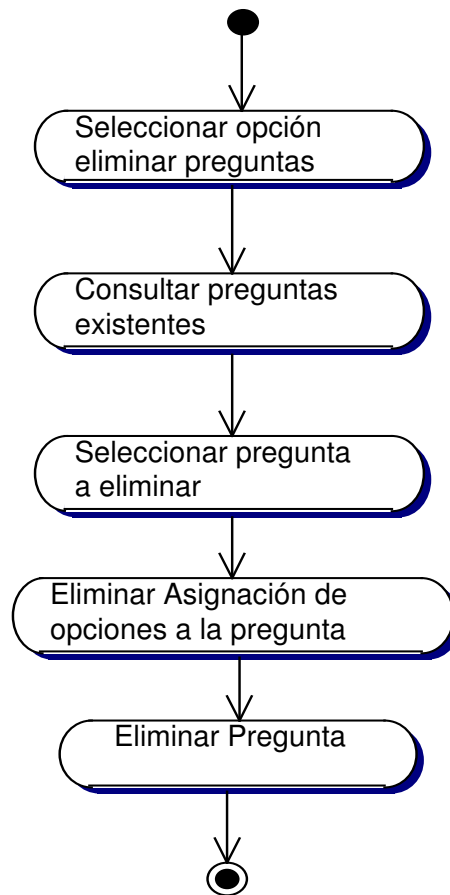
### 10.3.10.2 Diagrama Consultar Preguntas



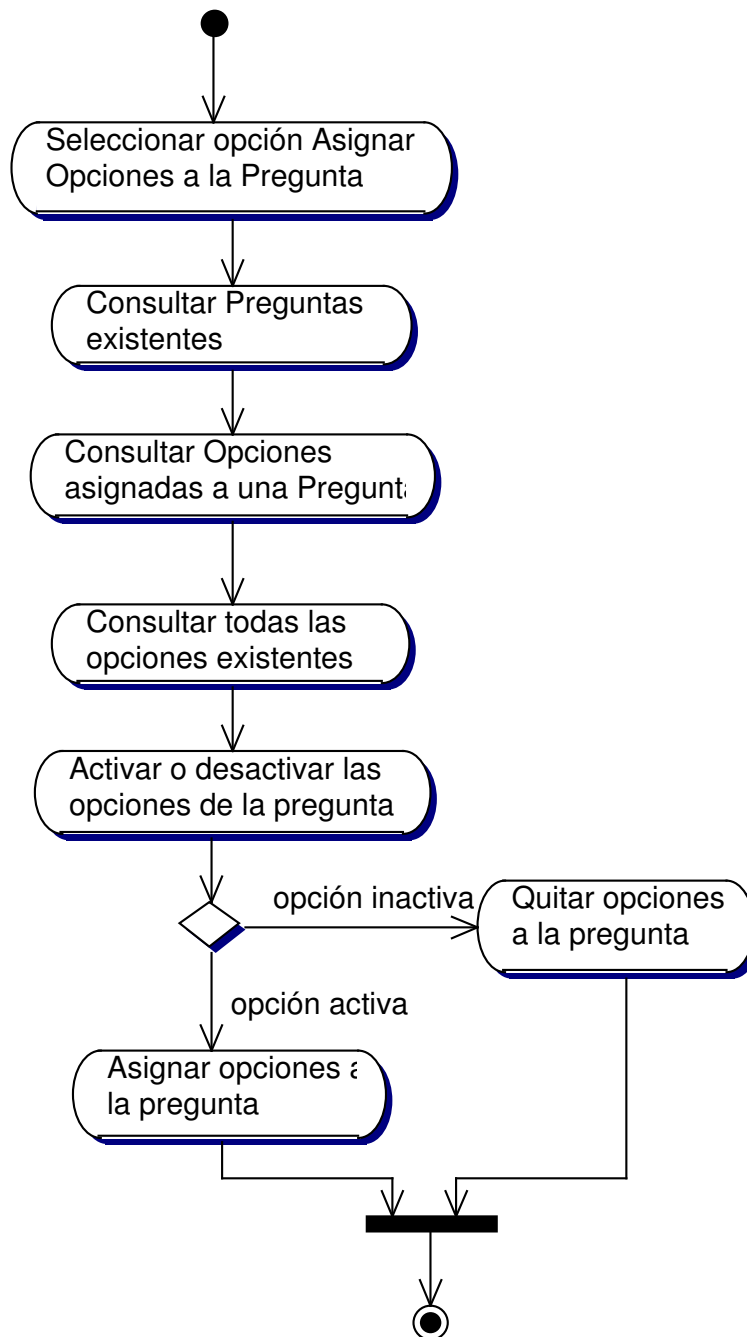
### 10.3.10.3 Diagrama Actualizar Pregunta Urna



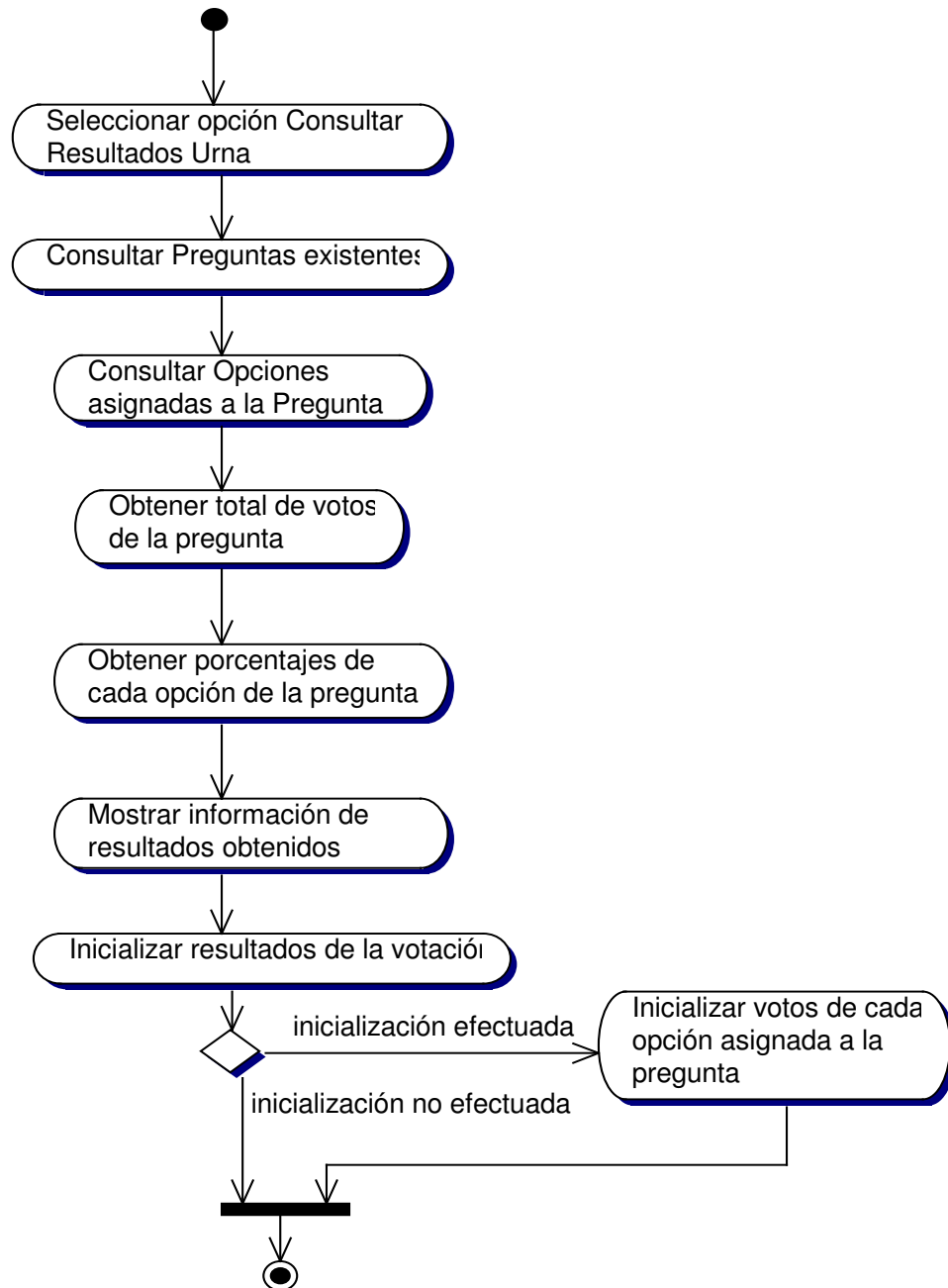
#### 10.3.10.4 Diagrama Eliminar Pregunta Urna



### 10.3.10.5 Diagrama Asignar Opciones a Pregunta Urna

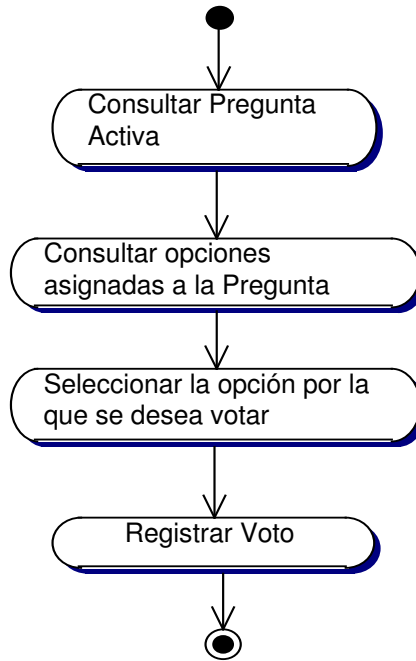


### 10.3.10.6 Diagrama Consultar Resultados Urna



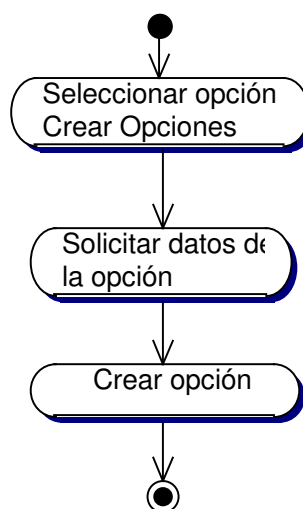


### 10.3.10.7 Diagrama Registrar Voto Urna

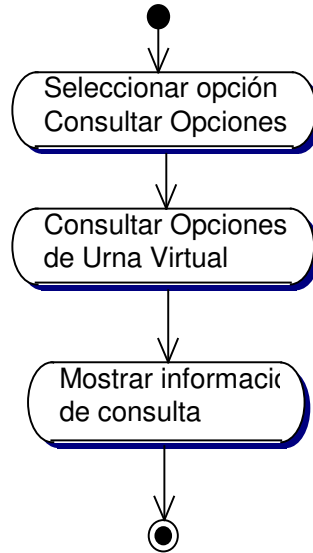


### 10.3.11 Administración de Opciones de Urna Virtual

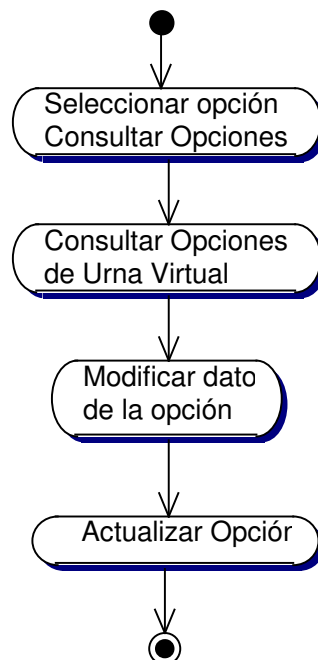
#### 10.3.11.1 Diagrama Crear Opciones Urna



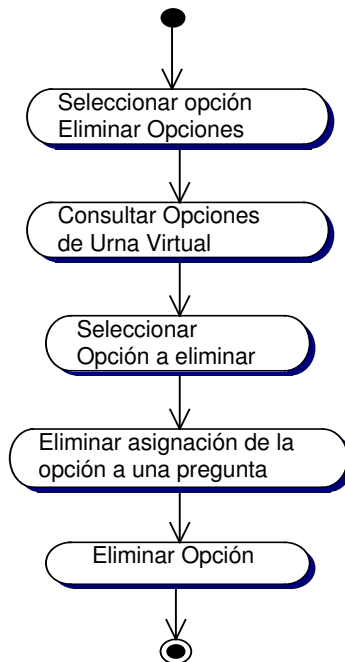
### 10.3.11.2 Diagrama Consultar Opciones Urna



### 10.3.11.3 Diagrama Actualizar Opciones Urna

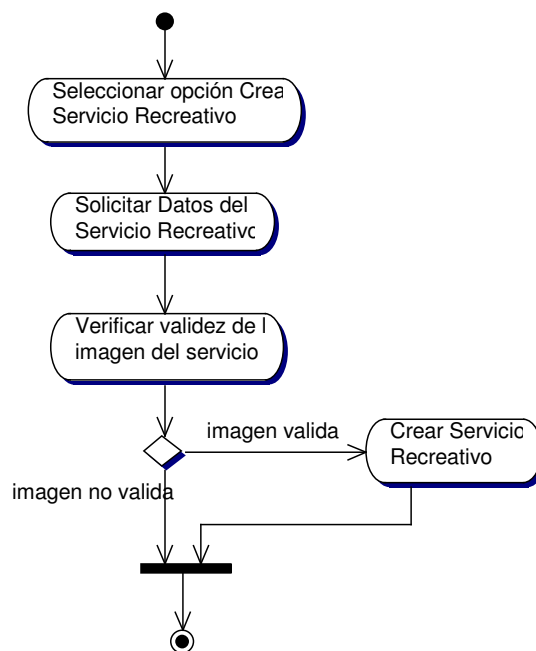


#### 10.3.11.4 Diagrama Eliminar Opciones Urna

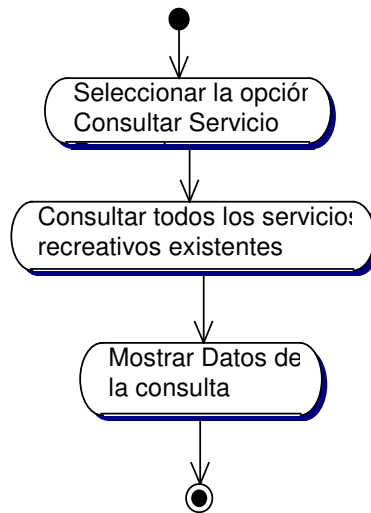


#### 10.3.12 Administración de Servicios Recreativos

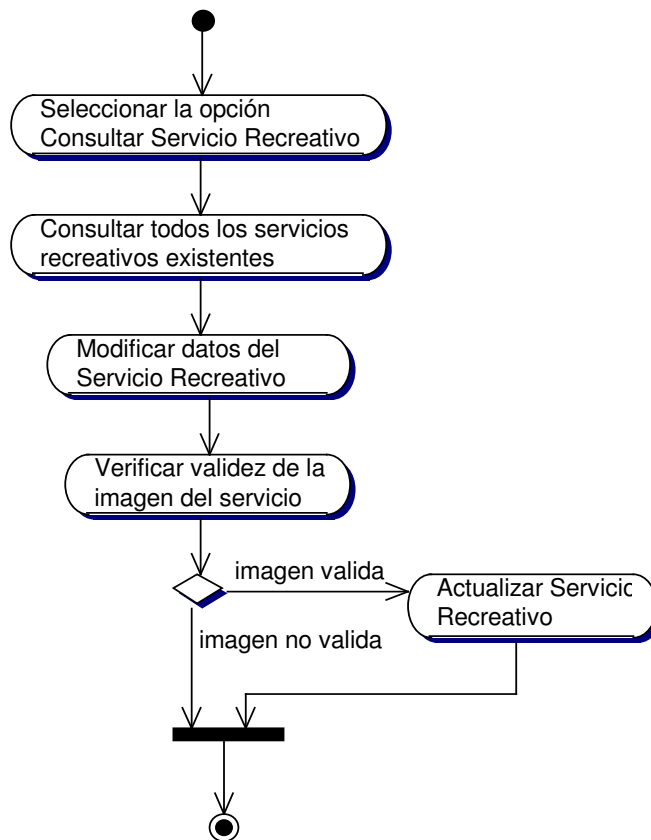
##### 10.3.12.1 Diagrama Crear Servicio Recreativo



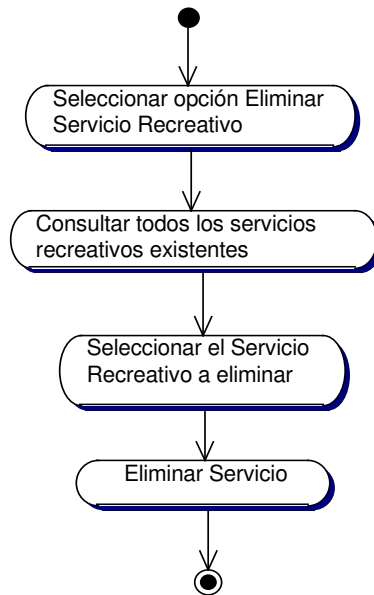
### 10.3.12.2 Diagrama Consultar Servicio Recreativo



### 10.3.12.3 Diagrama Actualizar Servicio Recreativo

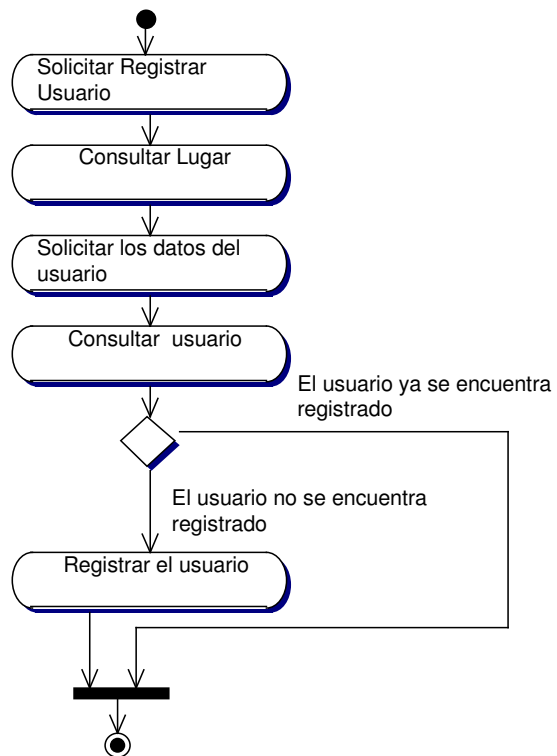


### 10.3.12.4 Diagrama Eliminar Servicio Recreativo

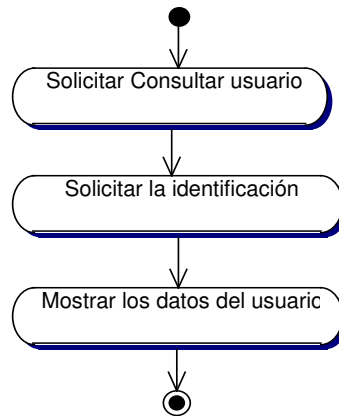


### 10.3.13 Administración de Usuarios

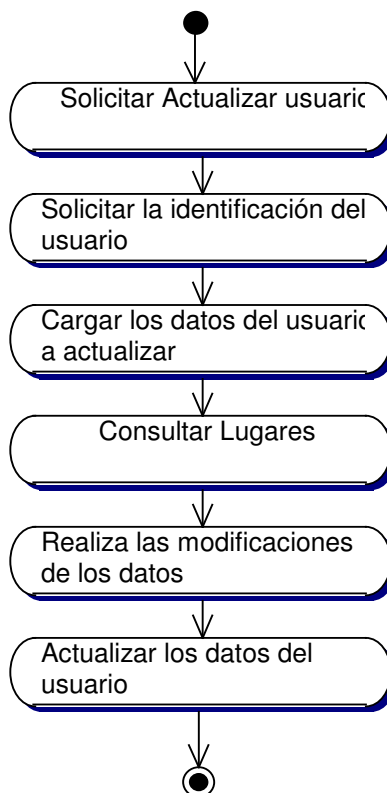
#### 10.3.13.1 Registrar Usuario



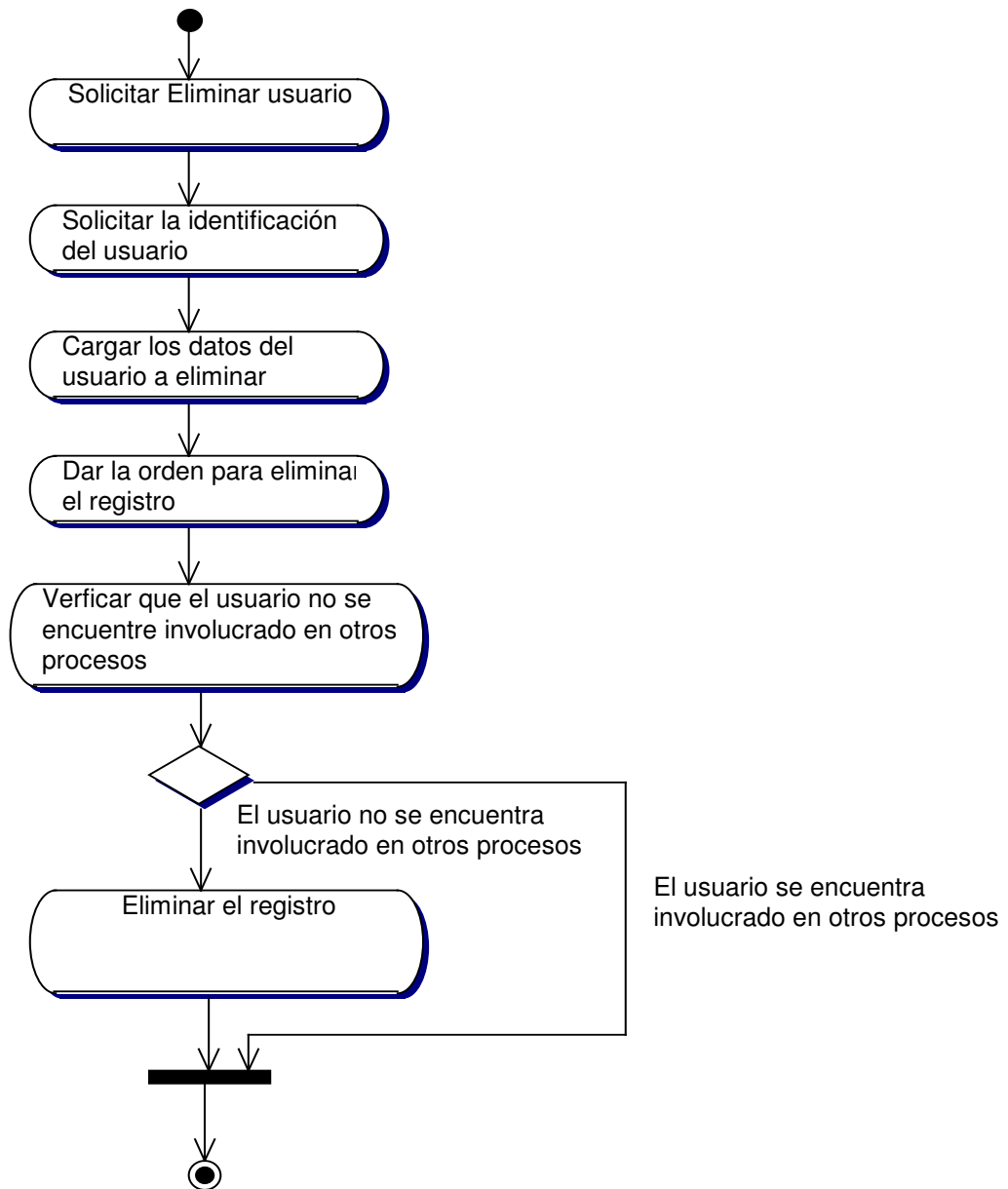
### 10.3.13.2 Consultar Usuario



### 10.3.13.3 Actualizar Usuario

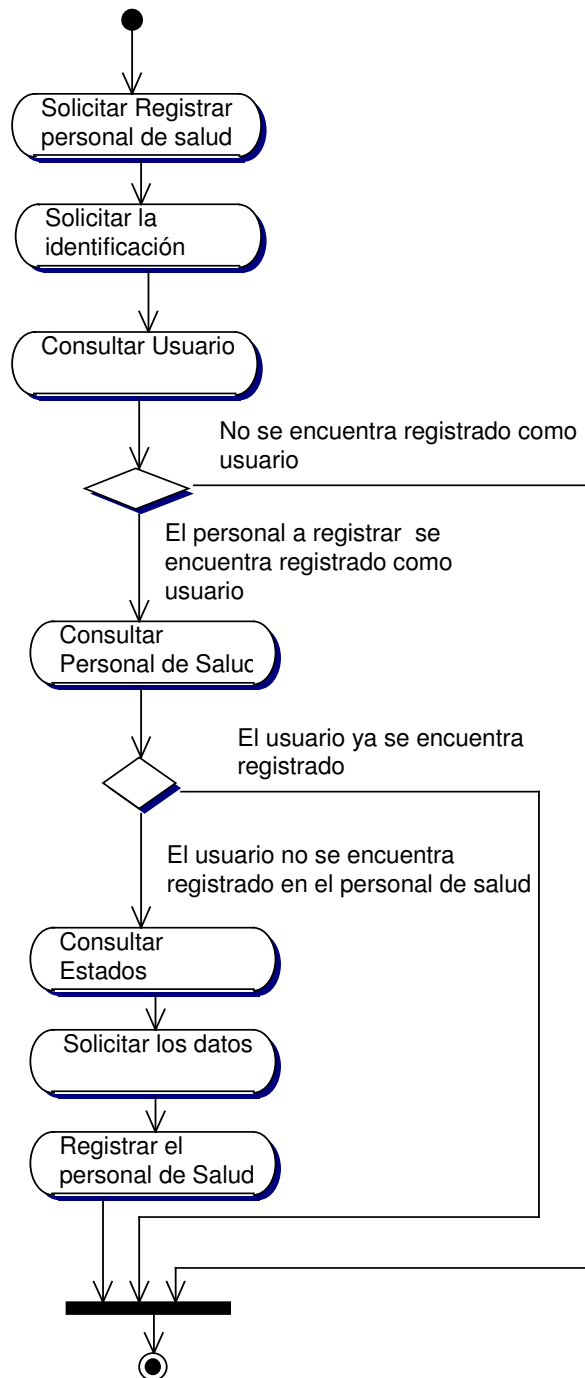


### 10.3.13.4 Eliminar Usuario



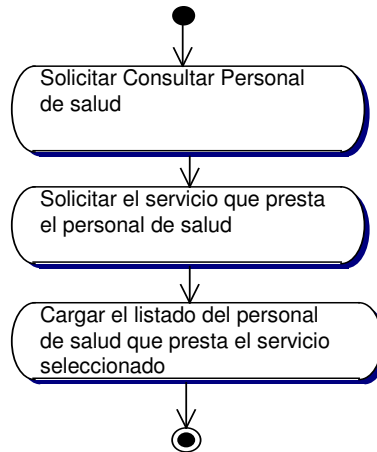
### 10.3.14 Administracion del Personal de Salud

#### 10.3.14.1 Registrar Personal de Salud (Médicos, Odontólogos y Psicólogos)

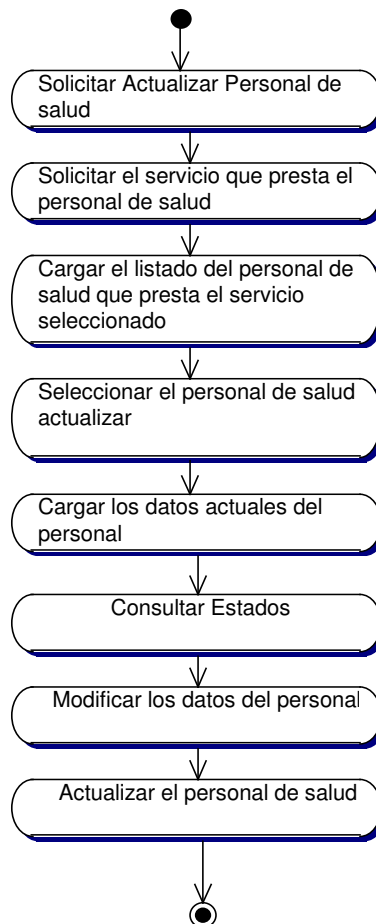




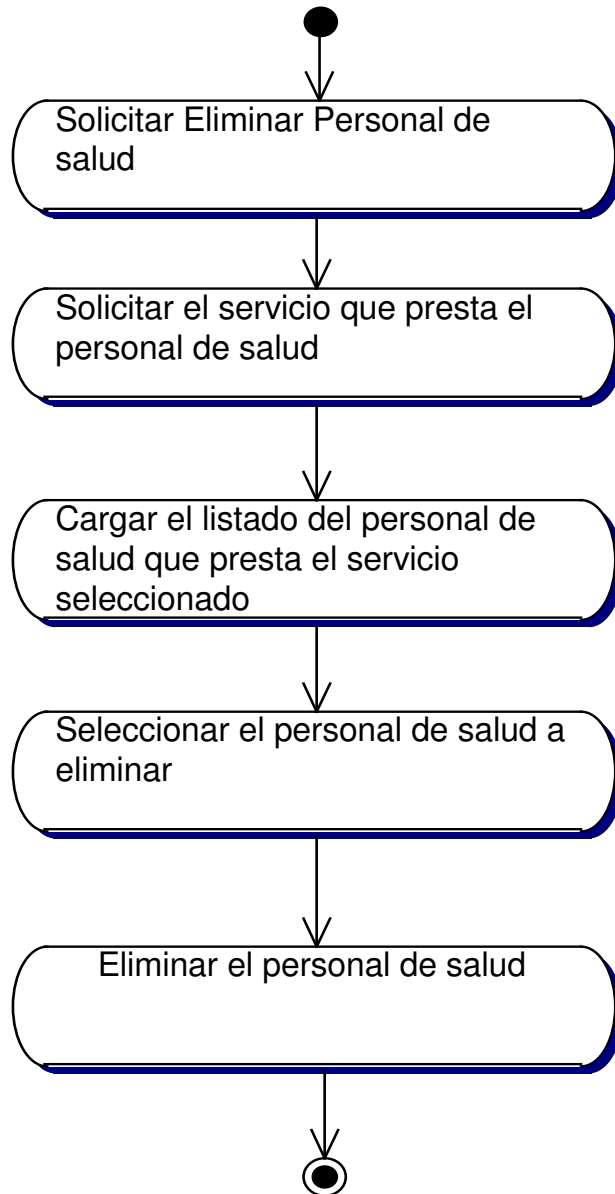
### 10.3.14.2 Consultar Personal de Salud (Médicos, Odontólogos y Psicólogos)



### 10.3.14.3 Actualizar Personal de Salud (Médicos, Odontólogos y Psicólogos)

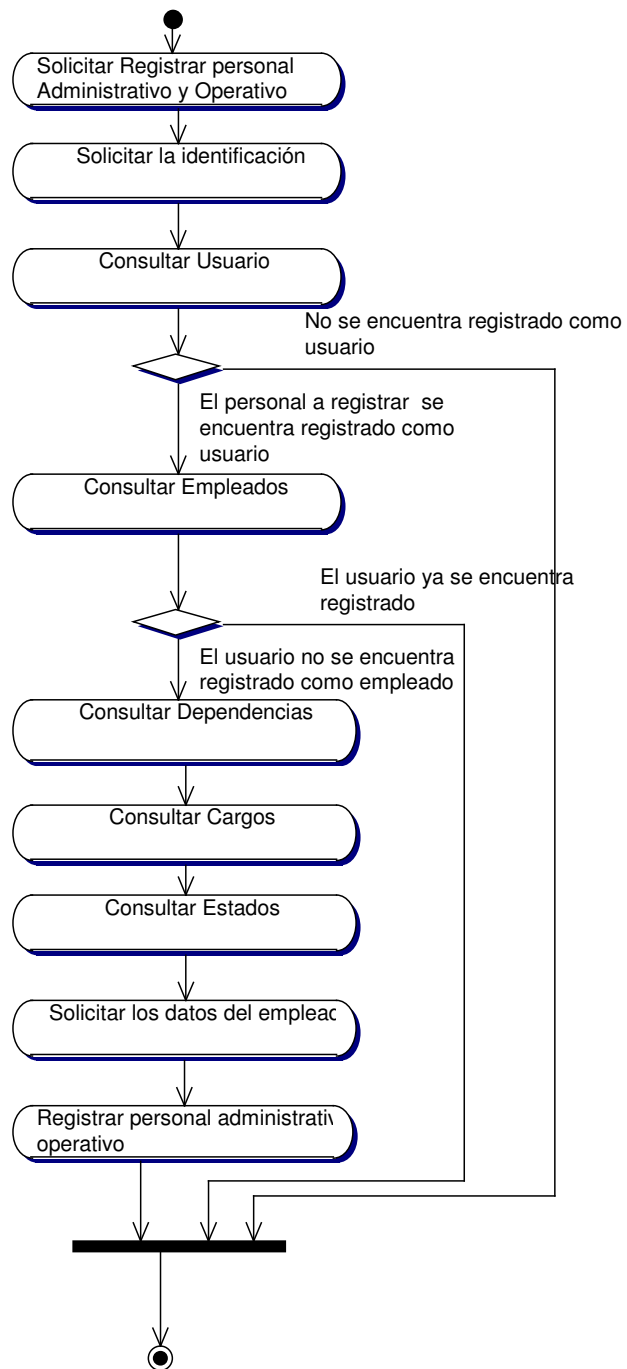


#### 10.3.14.4 Eliminar Personal de Salud (Médicos, Odontólogos y Psicólogos)

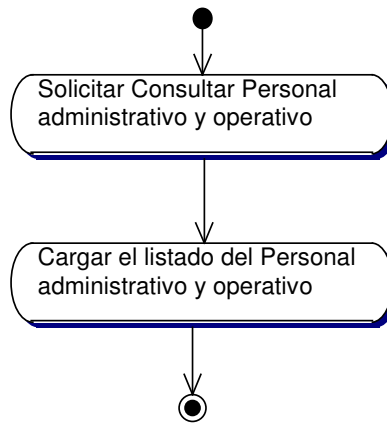


### 10.3.15 Administracion de Personal Administrativo y Operativo

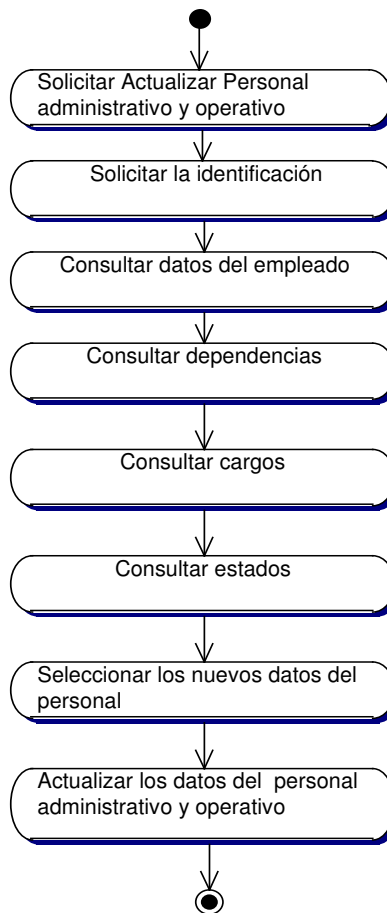
#### 10.3.15.1 Registrar Personal Administrativo y Operativo



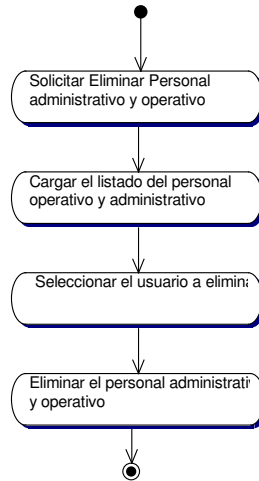
### 10.3.15.2 Consultar Personal Administrativo y Operativo



### 10.3.15.3 Actualizar Personal Administrativo y Operativo

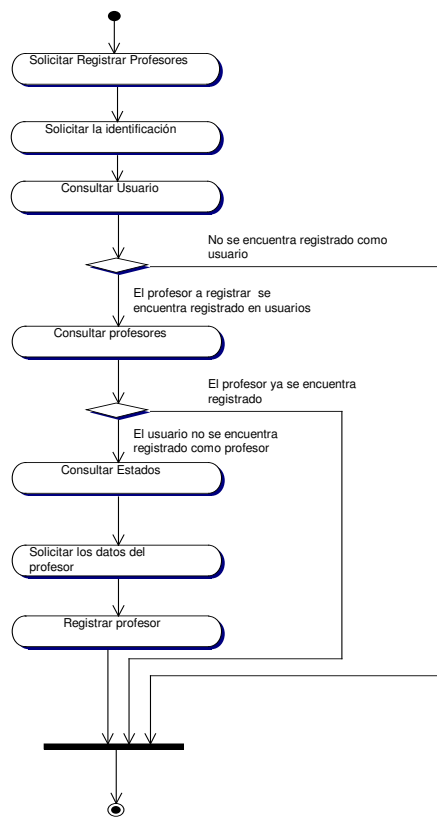


### 10.3.15.4 Eliminar Personal Administrativo y Operativo

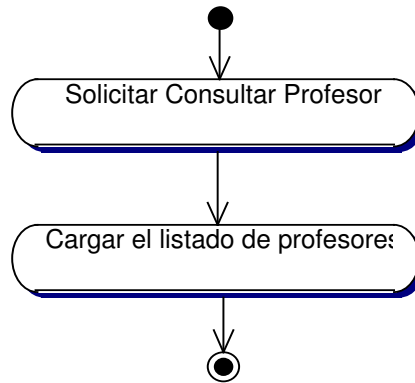


### 10.3.16 Administracion De Profesores

#### 10.3.16.1 Registrar Profesores



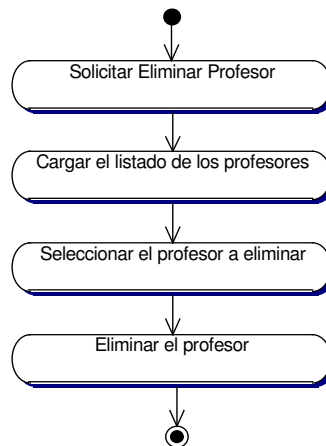
### 10.3.16.2 Consultar Profesores



### 10.3.16.3 Actualizar Profesores

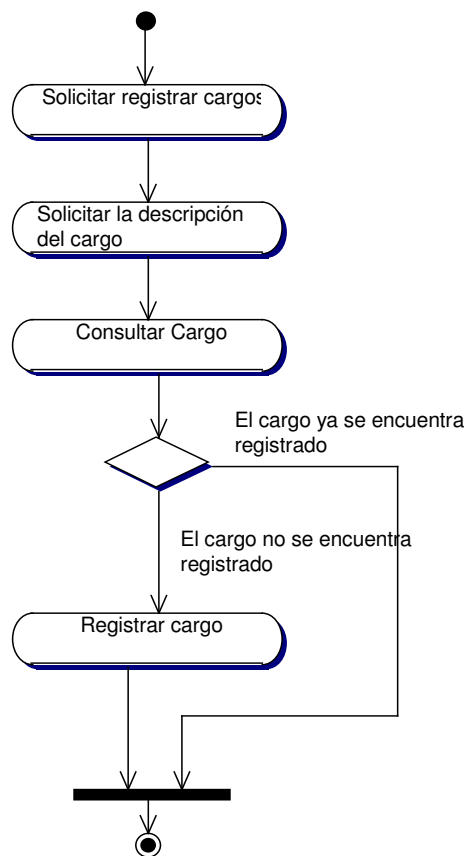


### 10.3.16.4 Eliminar Profesores

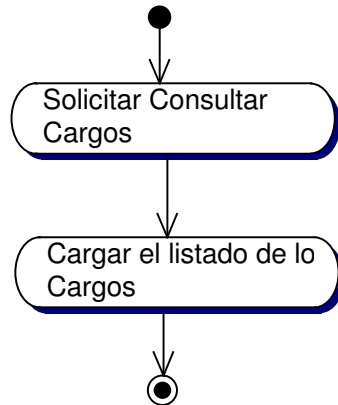


### 10.3.17 Administracion de Cargos

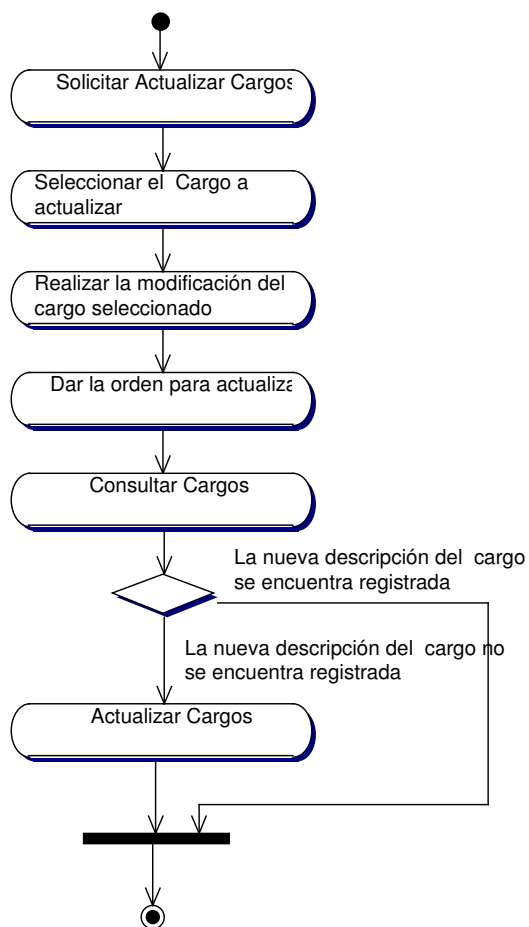
#### 10.3.17.1 Registrar Cargos



### 10.3.17.2 Consultar Cargos

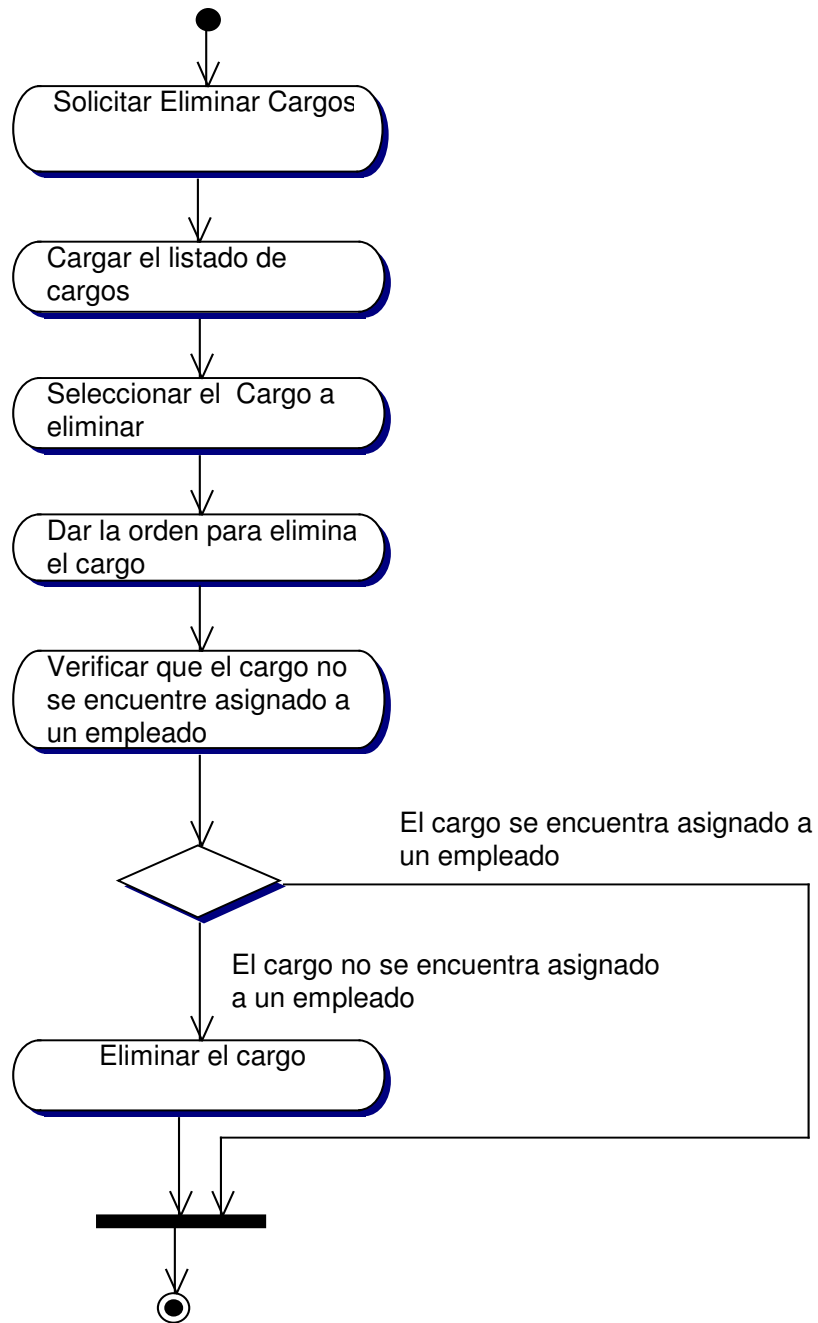


### 10.3.17.3 Actualizar Cargos



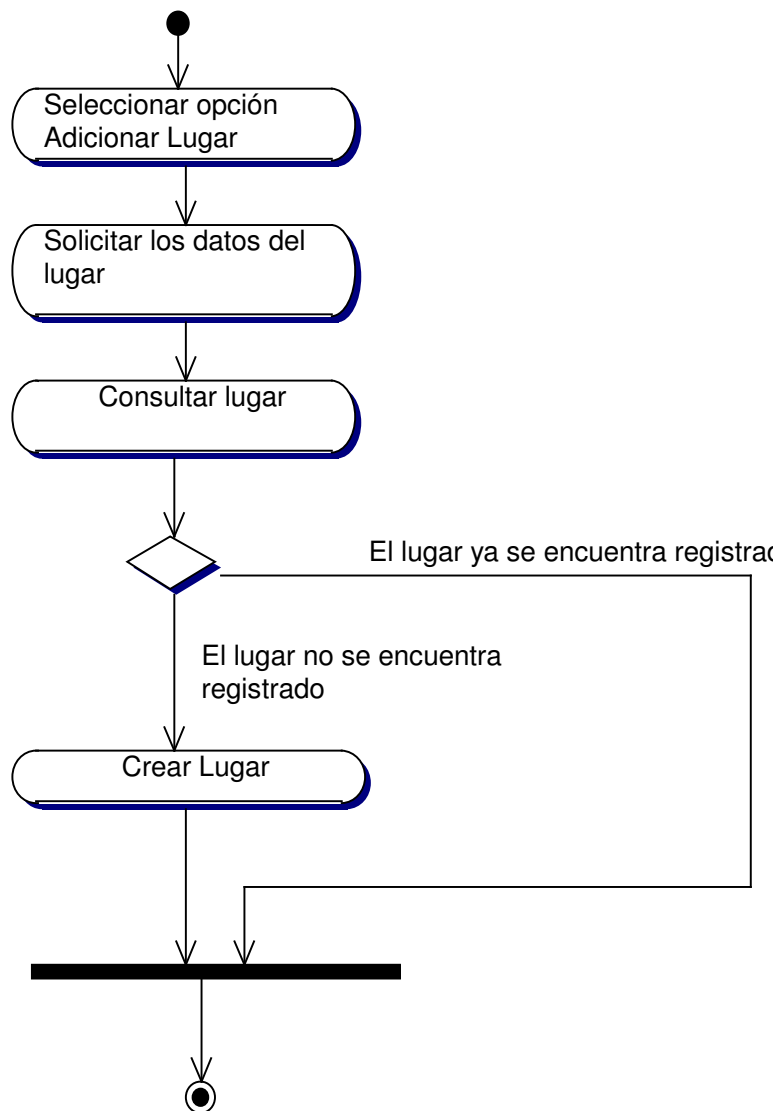


### 10.3.17.4 Eliminar Cargos

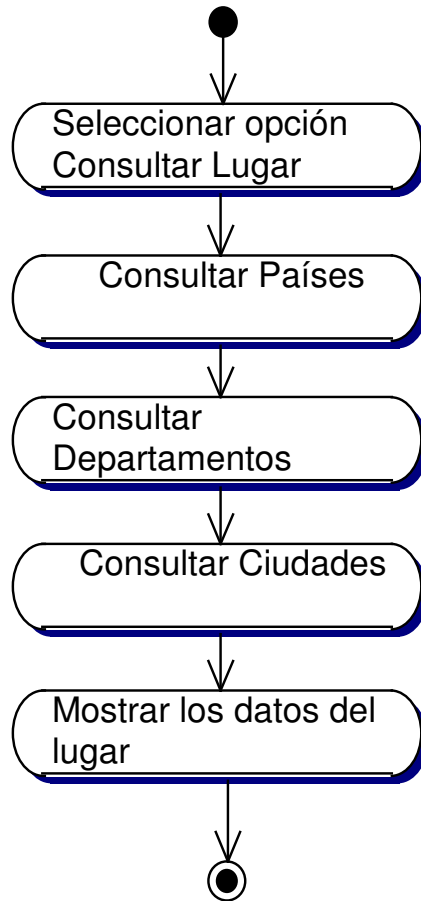


### 10.3.18 Administracion de Lugares

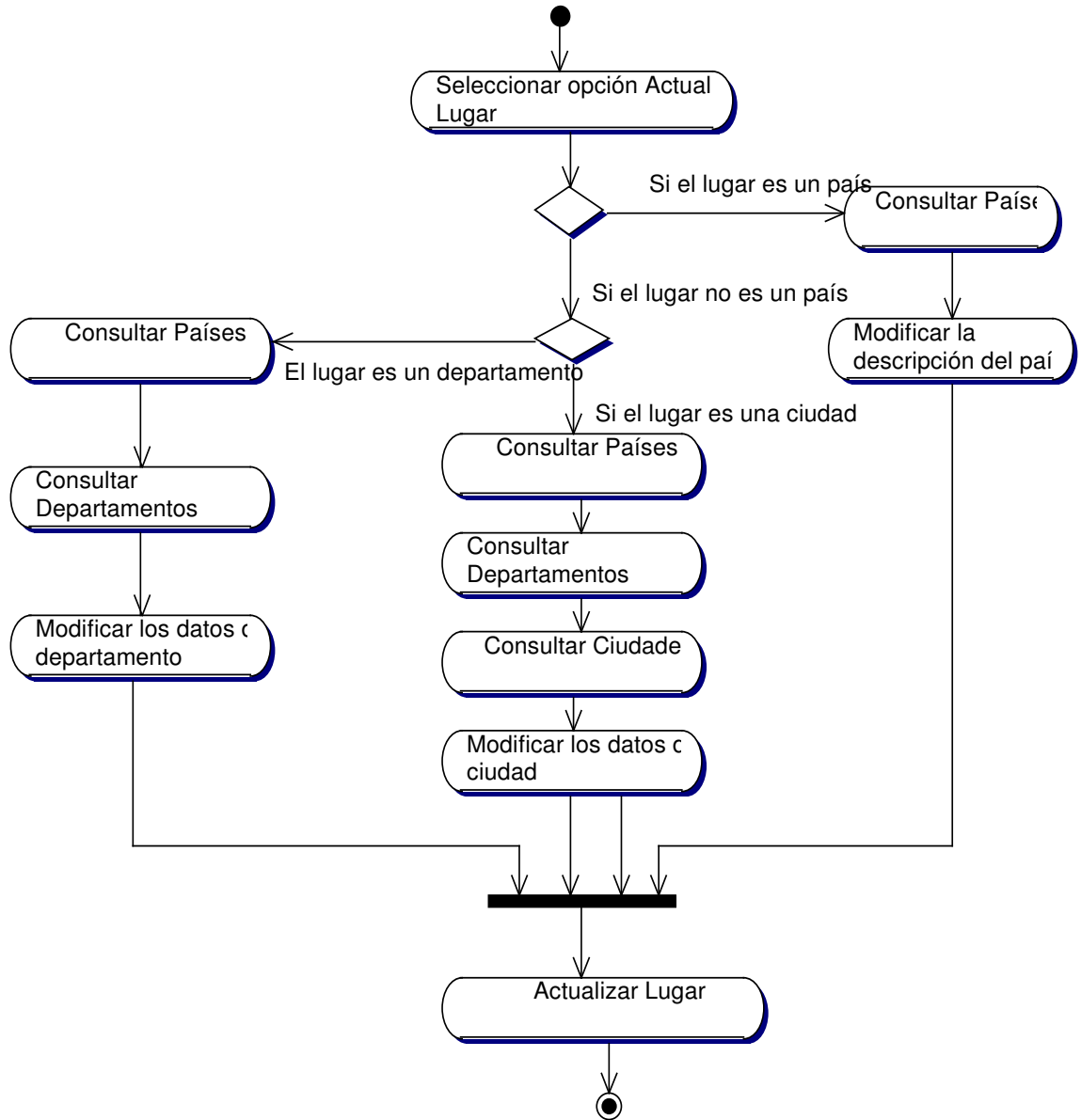
#### 10.3.18.1 Registrar Lugares



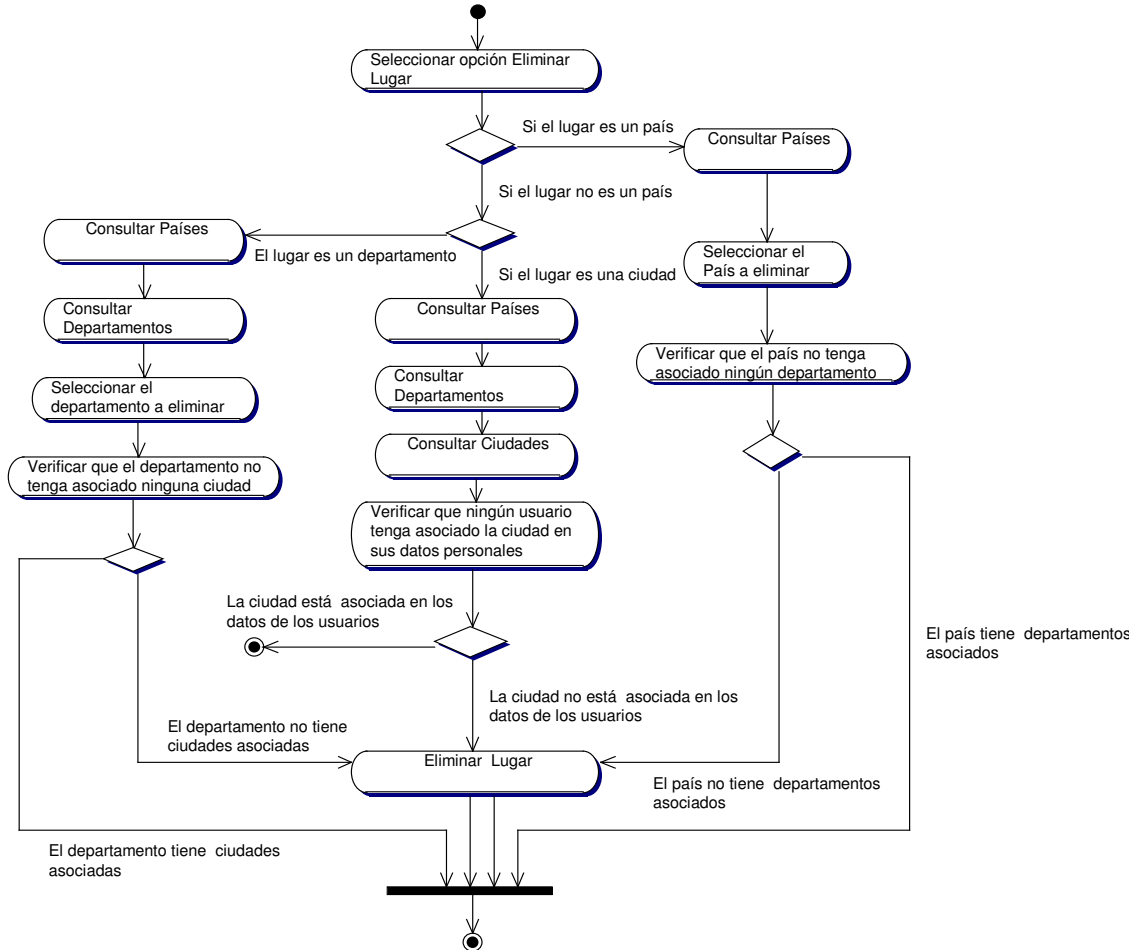
### 10.3.18.2 Consultar Lugares



### 10.3.18.3 Actualizar Lugares

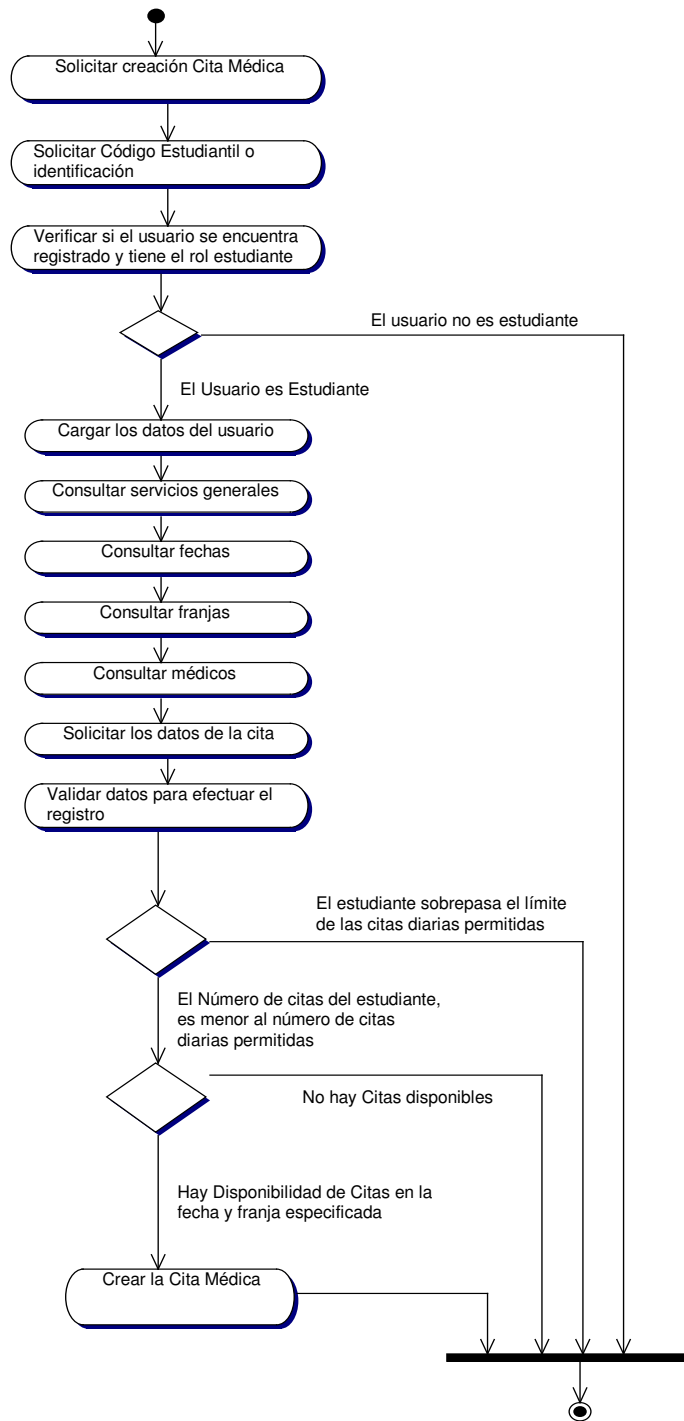


### 10.3.18.4 Eliminar Lugares

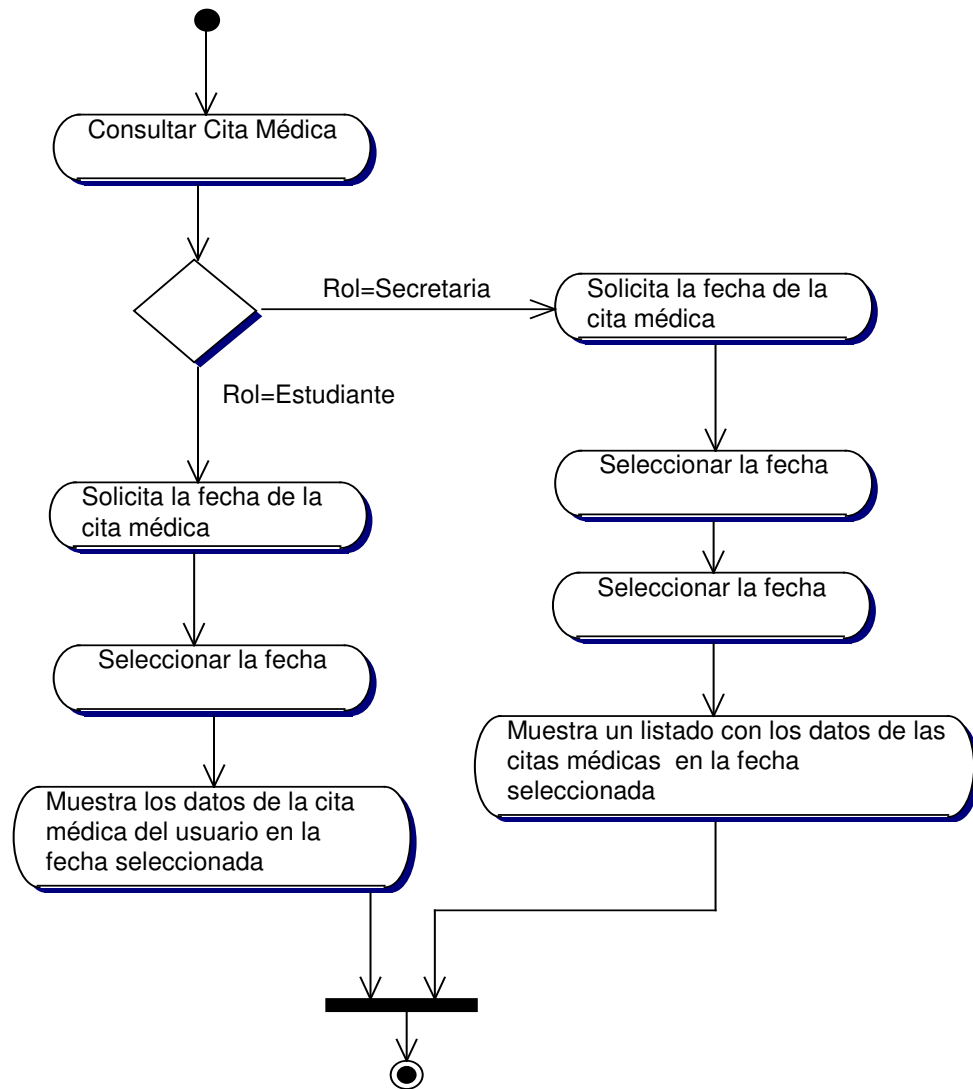


## 10.3.19 Administracion de Citas (Medicas, Odontologicas, Psicologicas)

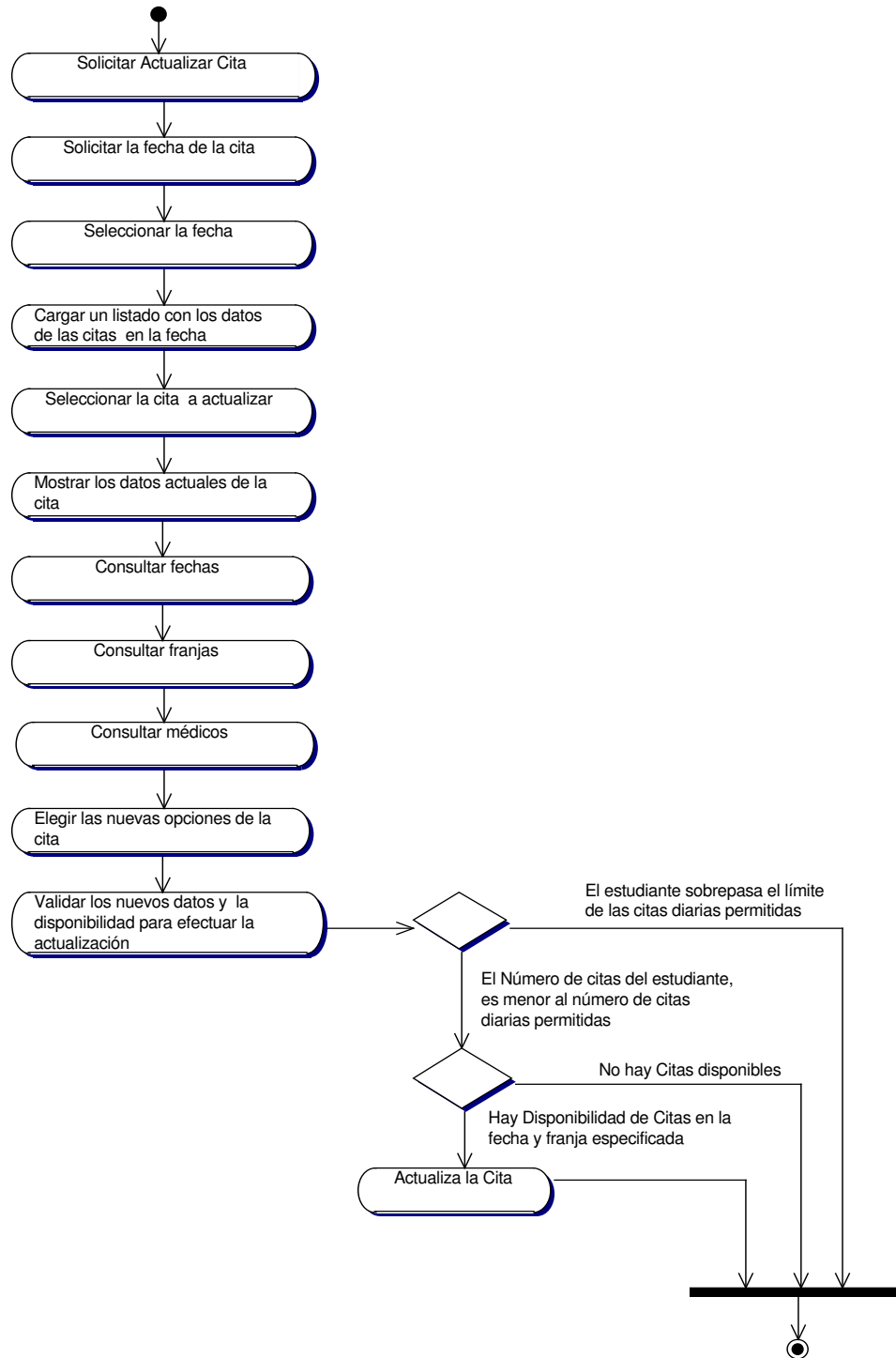
### 10.3.19.1 Crear Cita



### 10.3.19.2 Consultar Cita

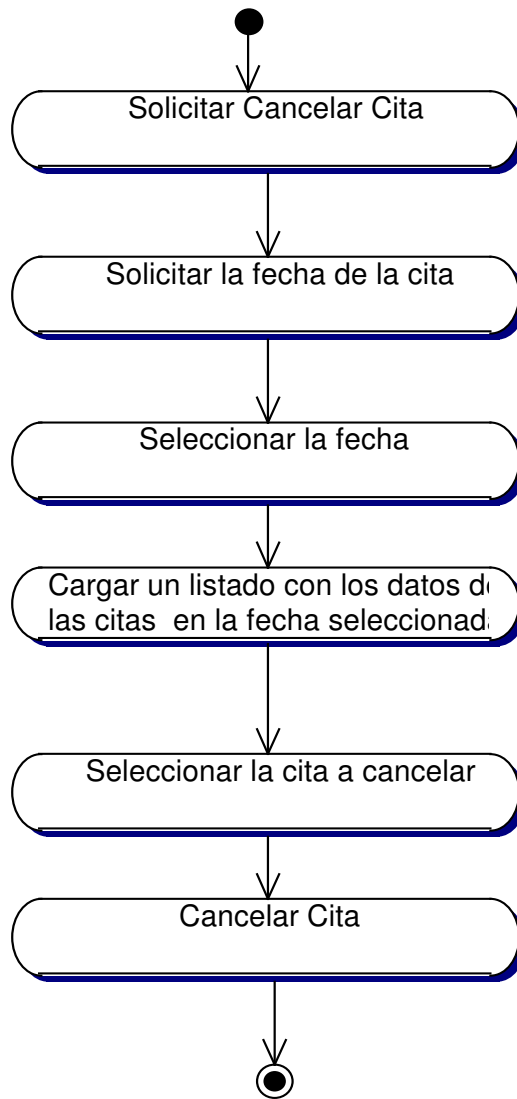


### 10.3.19.3 Actualizar Cita



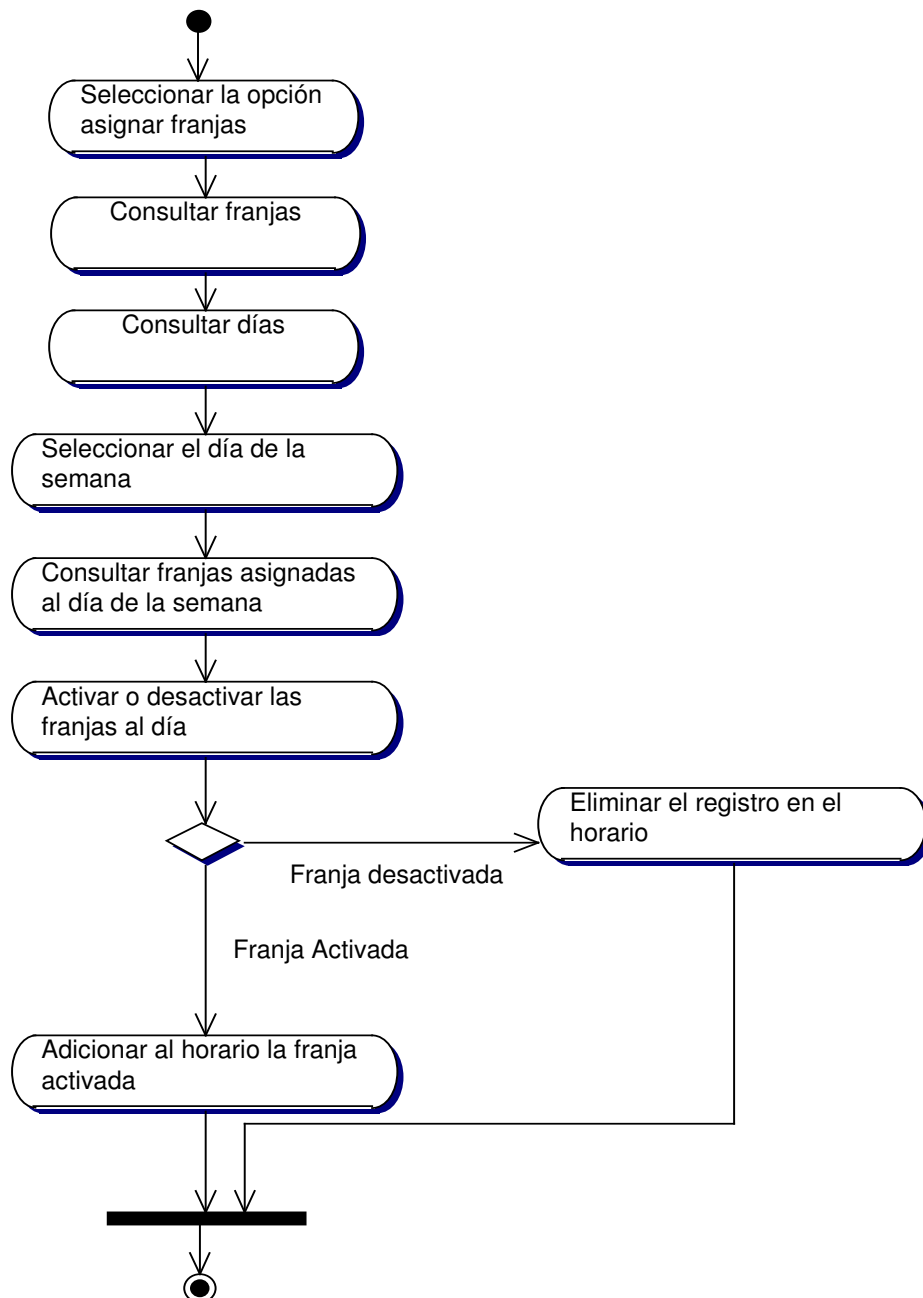


#### 10.3.19.4 Cancelar Cita

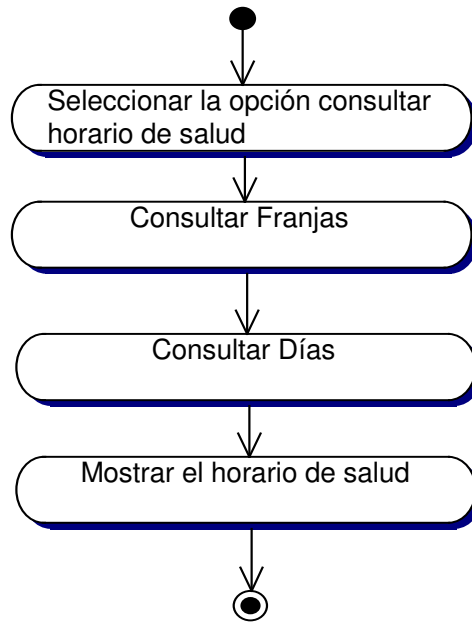


### 10.3.20 Administracion de Horarios de Salud

#### 10.3.20.1 Registrar Horario de Salud

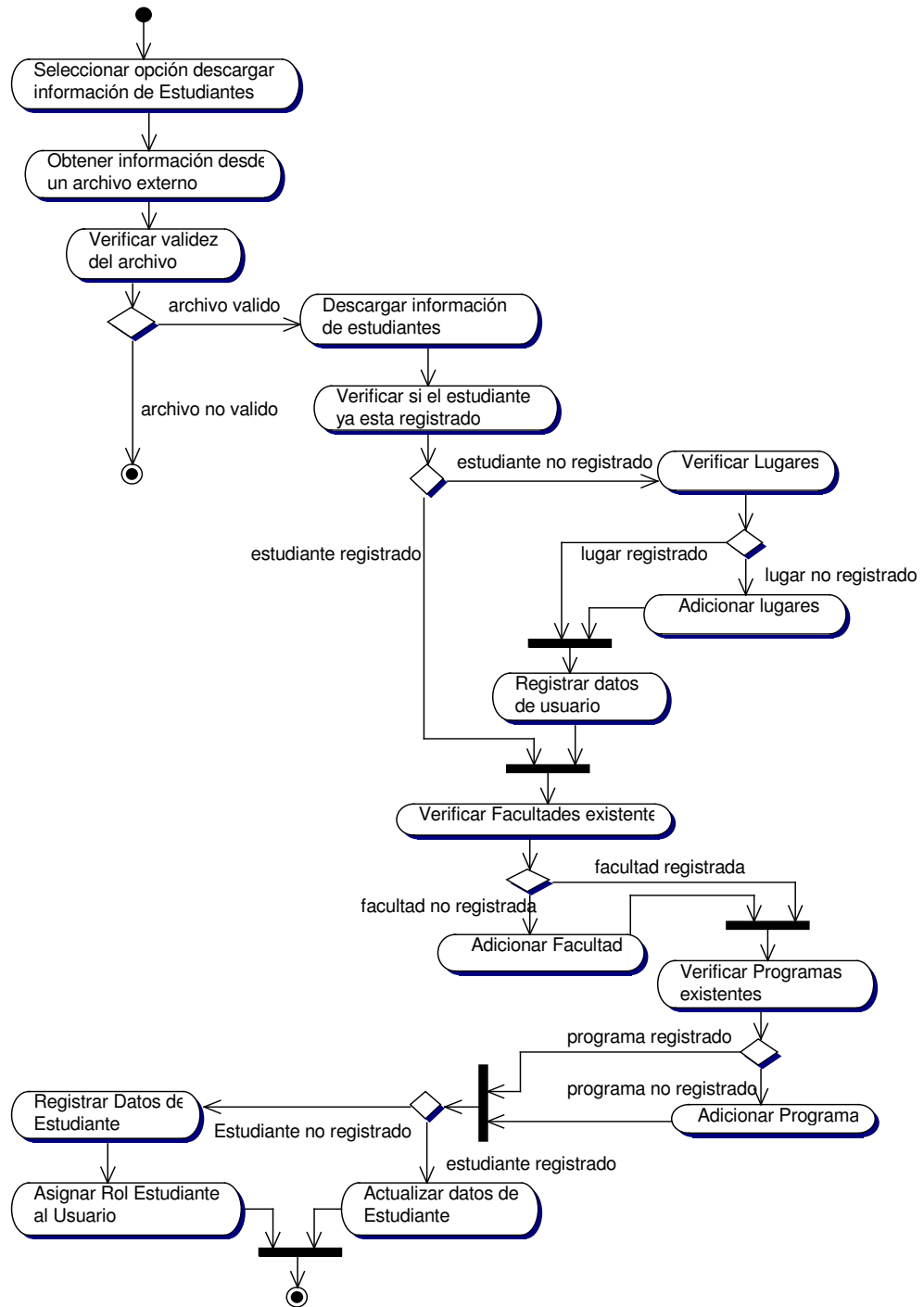


### 10.3.20.2 Consultar Horario de Salud

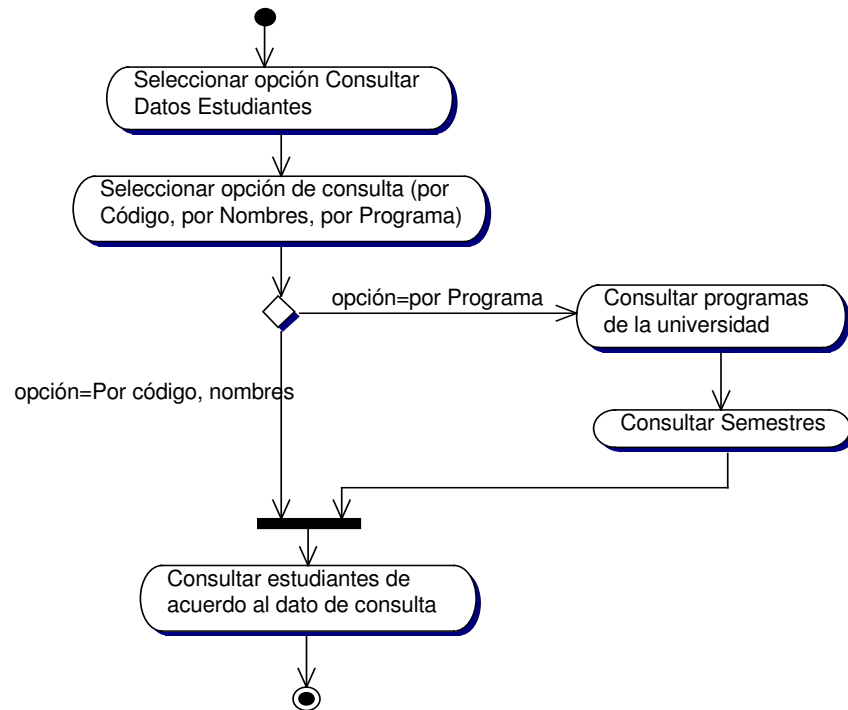


### 10.3.21 Administración de Estudiantes

#### 10.3.21.1 Diagrama Descargar Información de Estudiantes de la Extensión

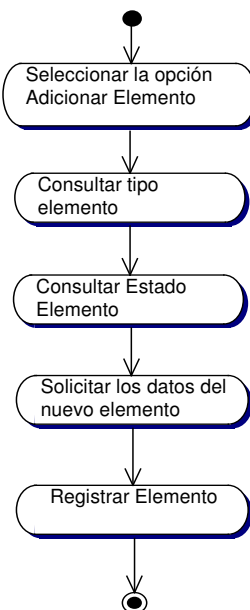


### 10.3.21.2 Diagrama Consultar Datos de Estudiantes

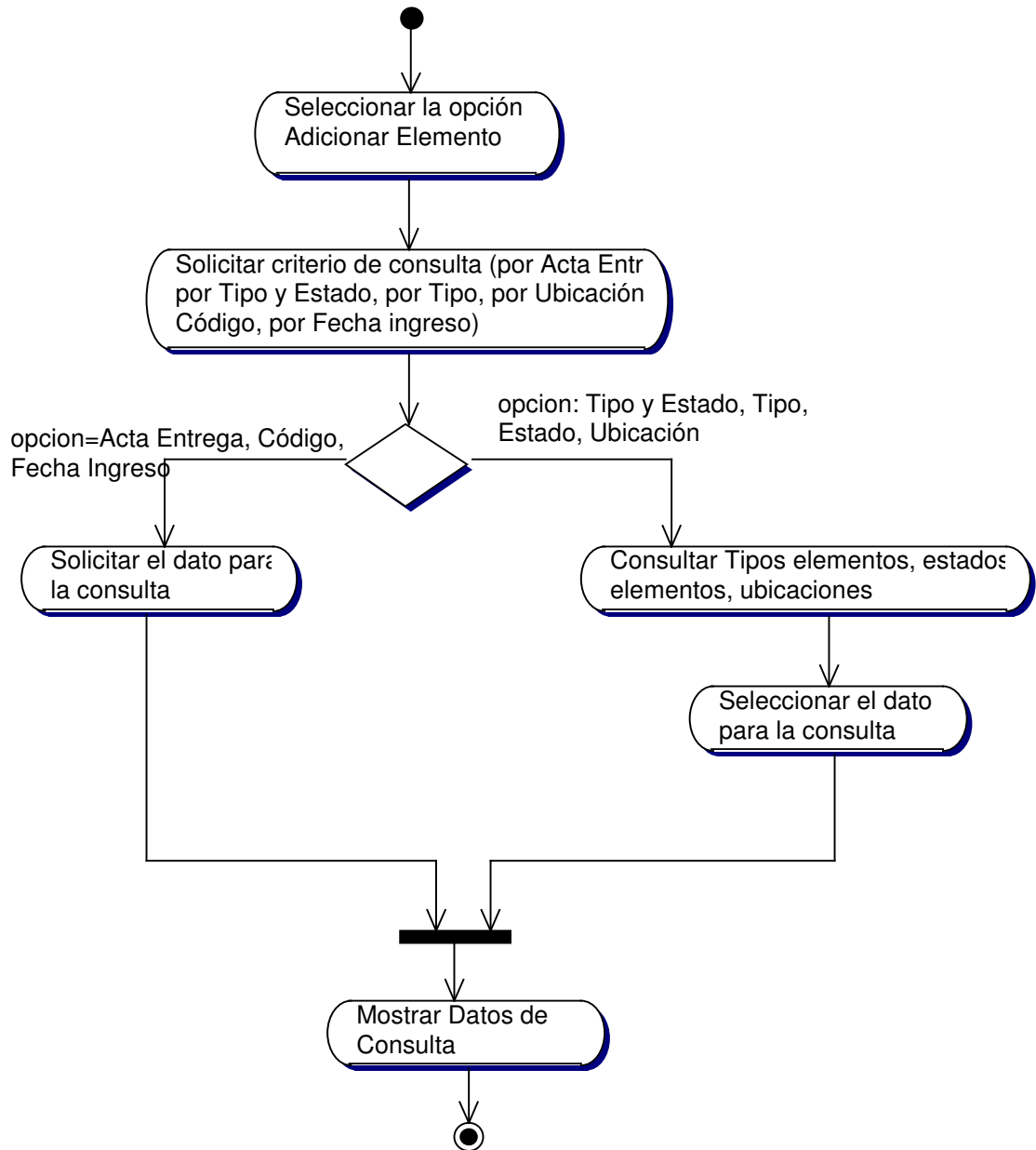


### 10.3.22 Administración de Elementos de Inventario

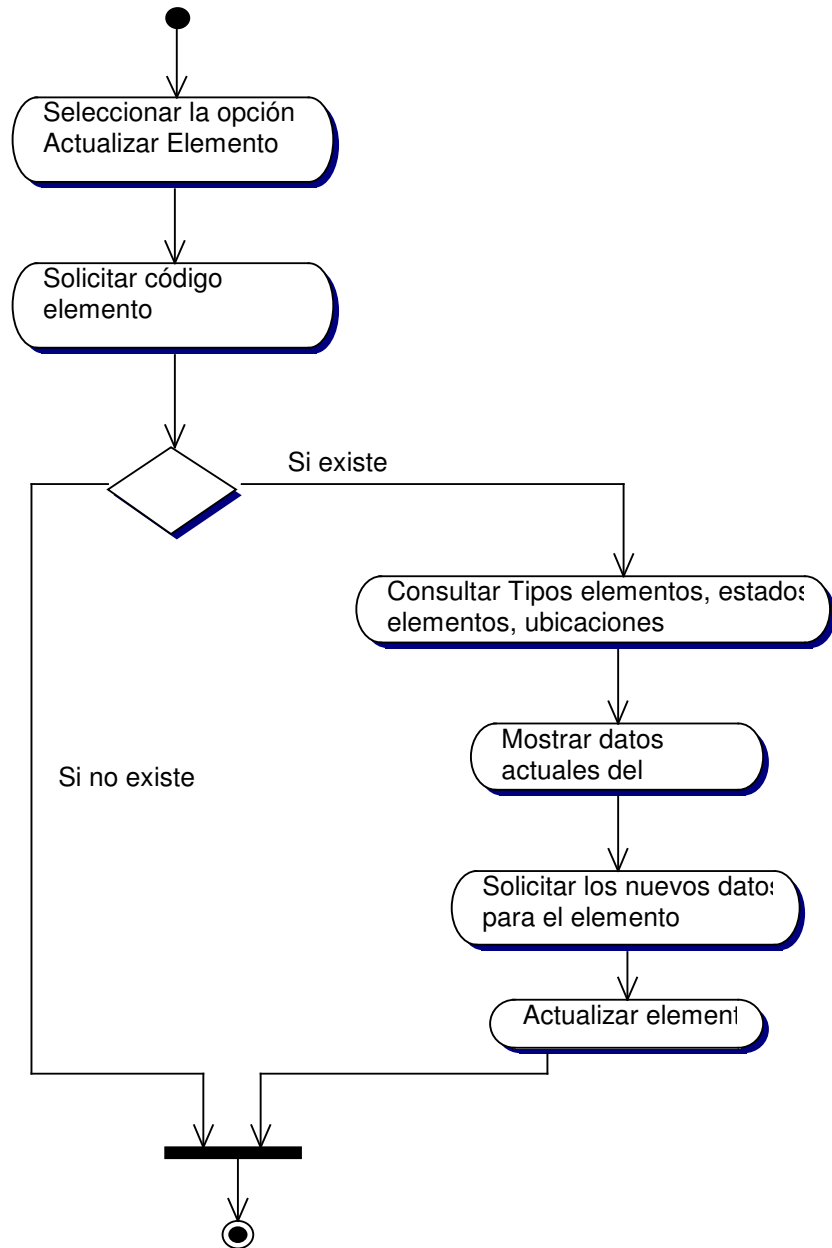
#### 10.3.22.1 Adicionar Elemento



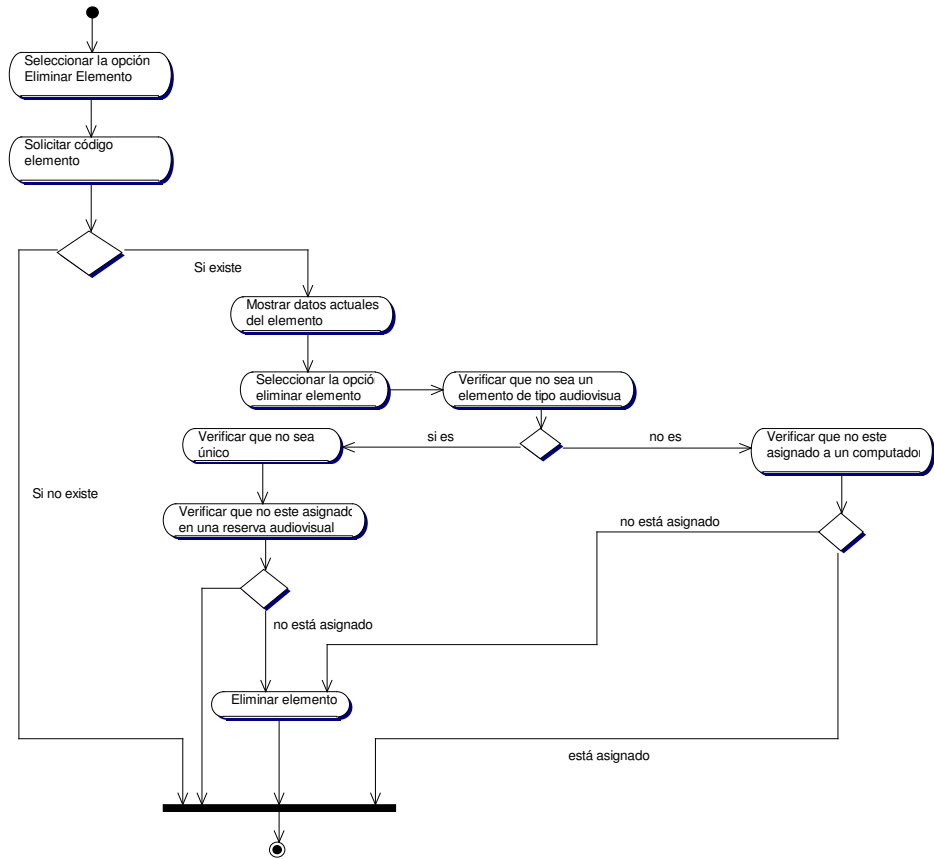
### 10.3.22.2 Consultar Elemento



### 10.3.22.3 Actualizar Elemento

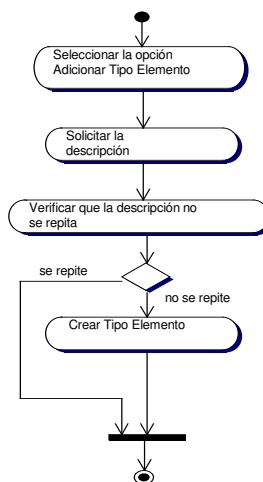


### 10.3.22.3 Eliminar Elemento



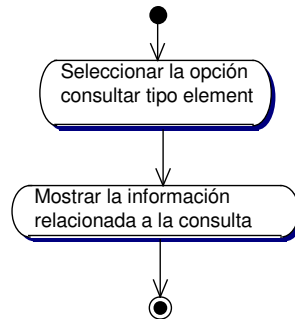
### 10.3.23 Administración Tipos de Elementos

#### 10.3.23.1 Crear Tipo Elemento

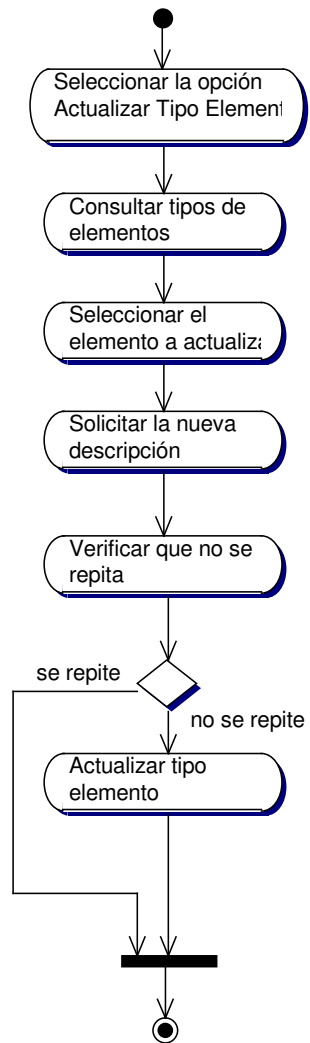




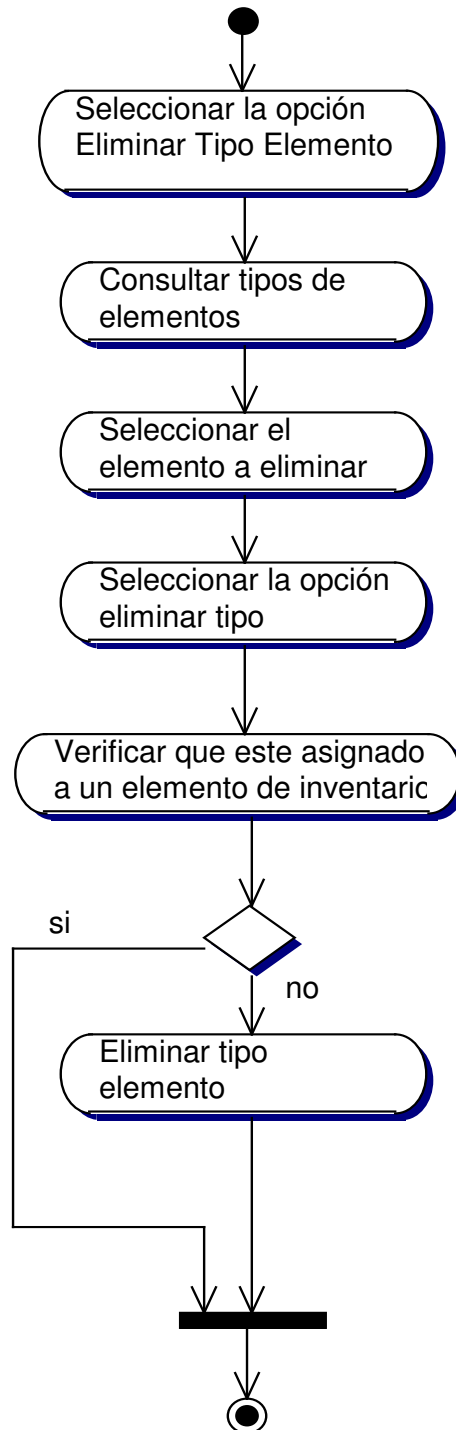
### 10.3.23.2 Consultar Tipo Elemento



### 10.3.23.3 Actualizar Tipo Elemento

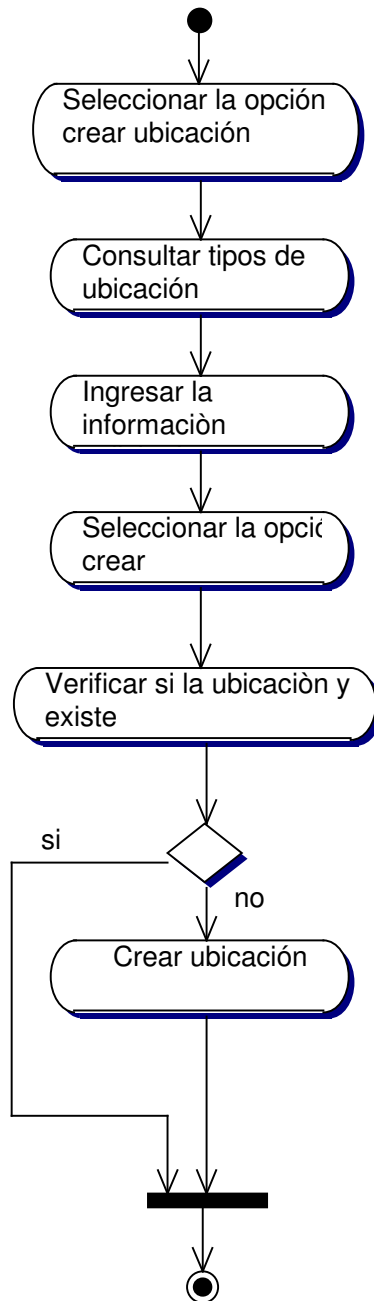


#### 10.3.23.4 Eliminar Tipo Elemento

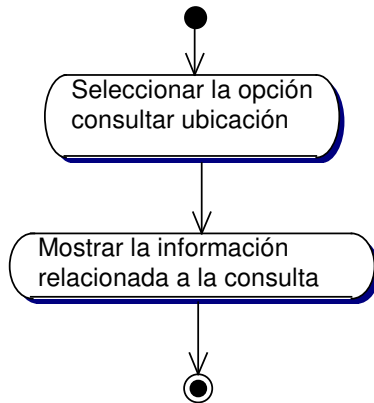


## 10.3.24 Administración Ubicaciones

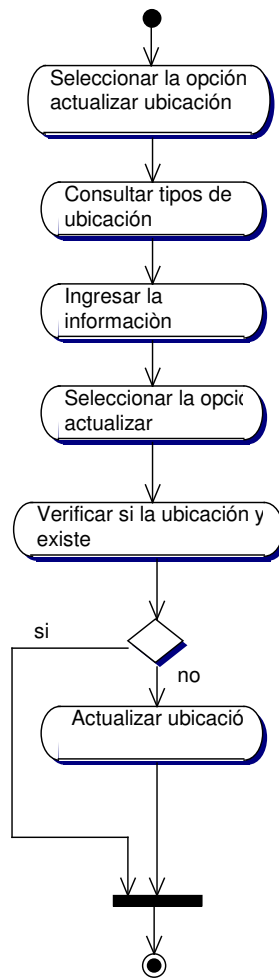
### 10.3.24.1 Crear Ubicación



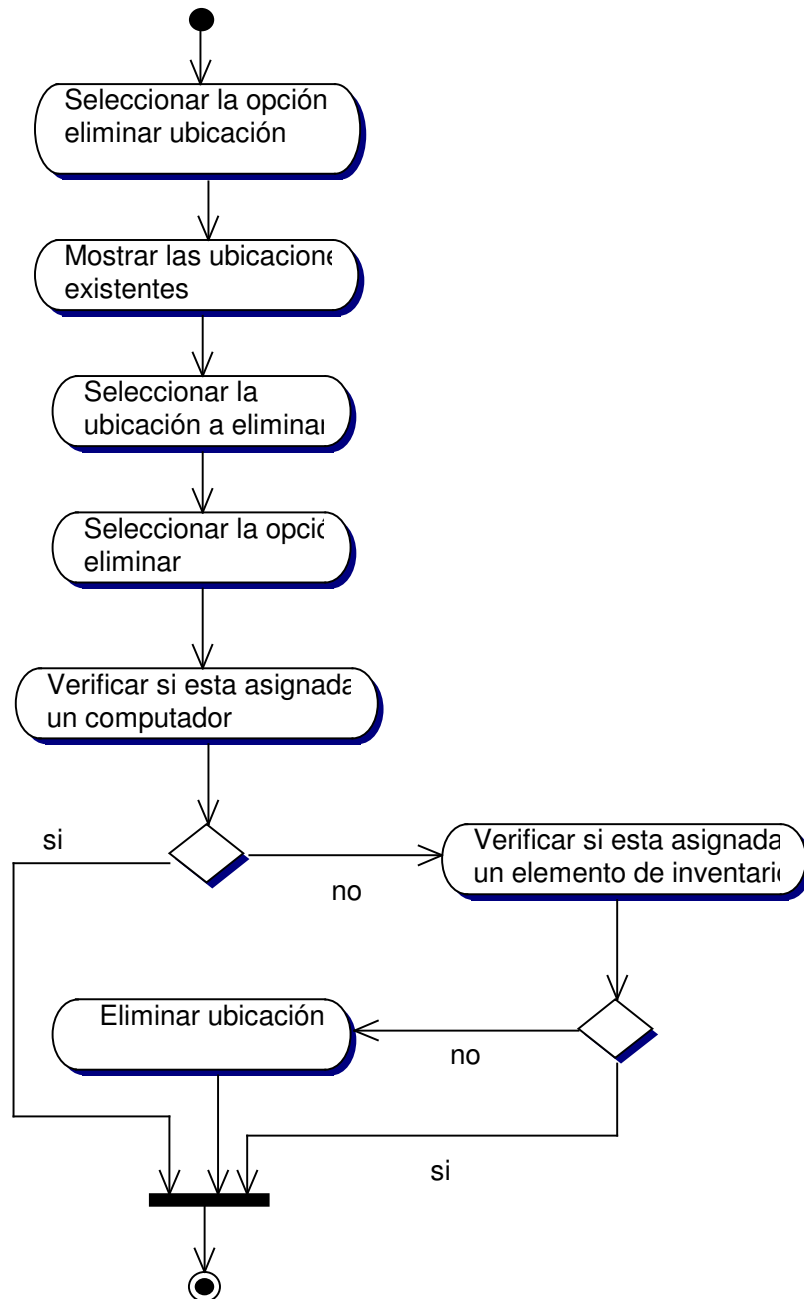
### 10.3.24.2 Consultar Ubicación



### 10.3.24.3 Actualizar Ubicación

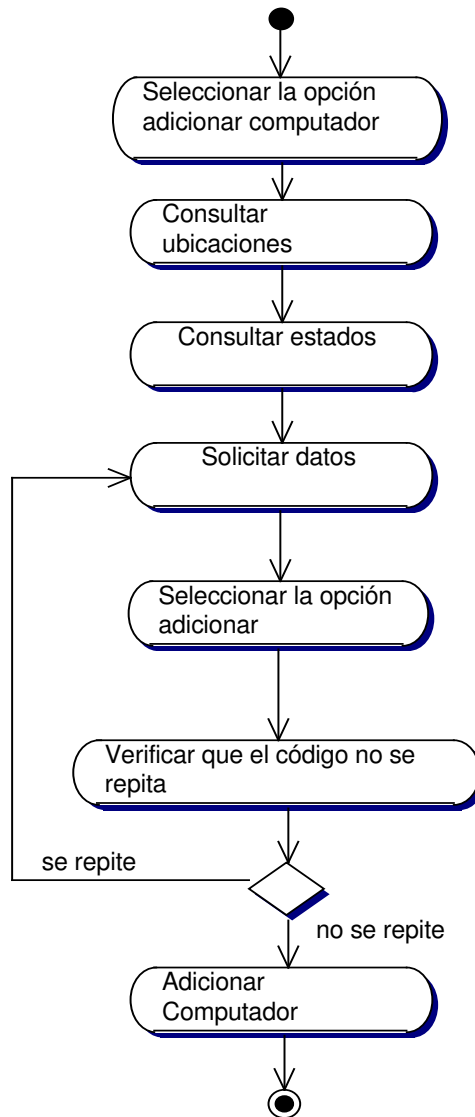


#### 10.3.24.4 Eliminar Ubicación

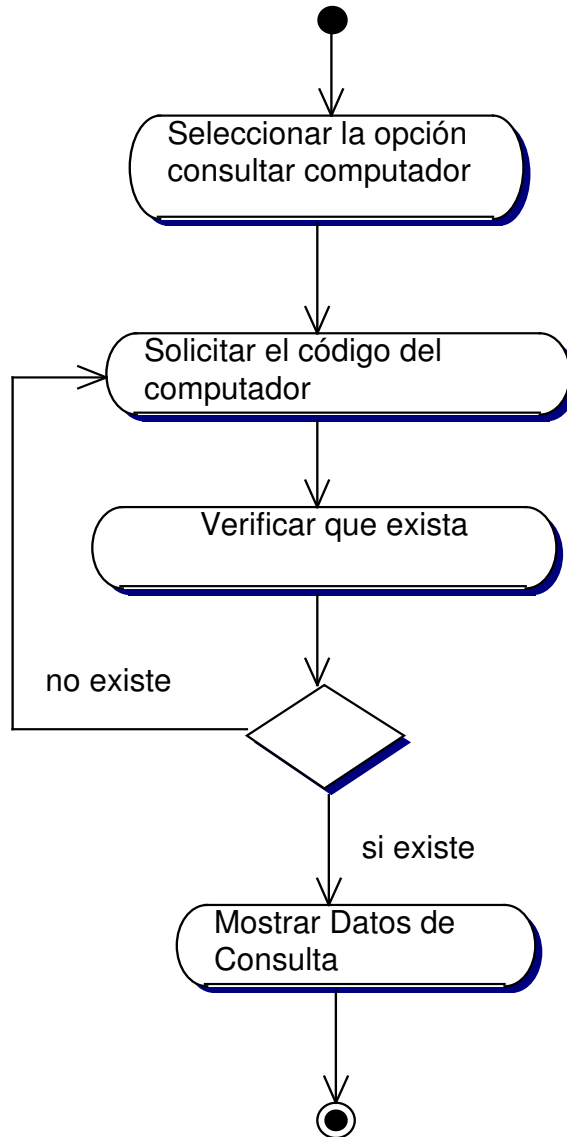


## 10.3.25 Administración De Reservas De Computadores

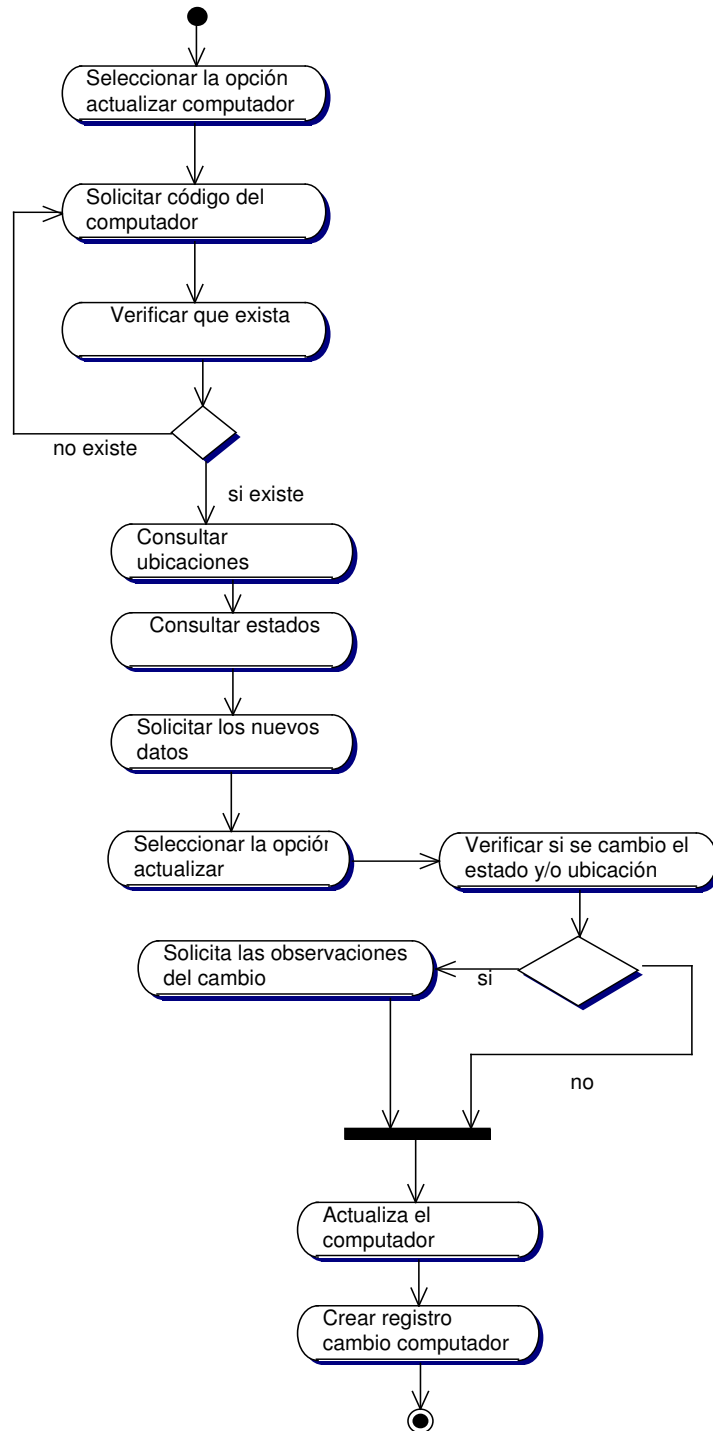
### 10.3.25.1 Adicionar Computador



### 10.3.25.2 Consultar Computador

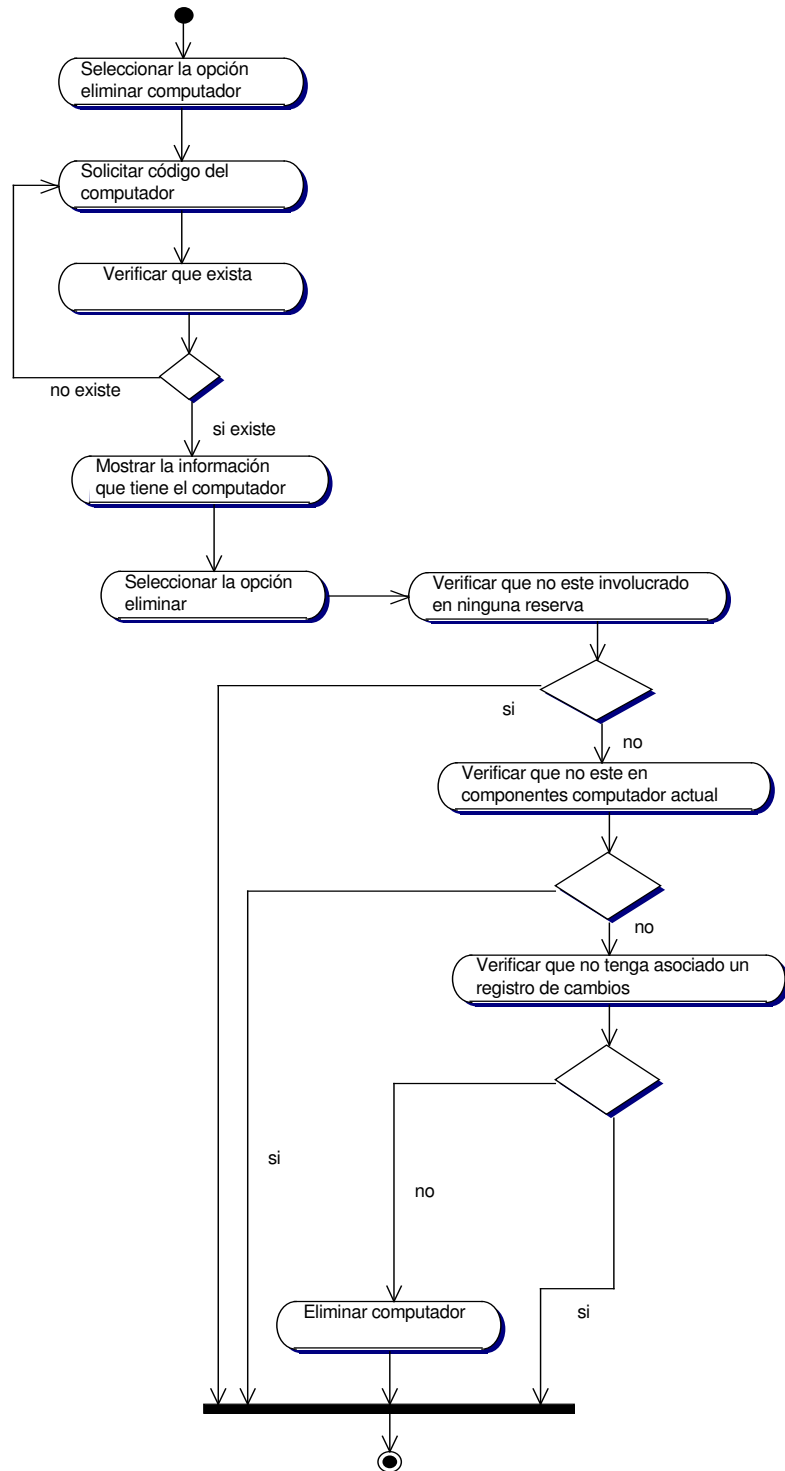


### 10.3.25.3 Actualizar Computador

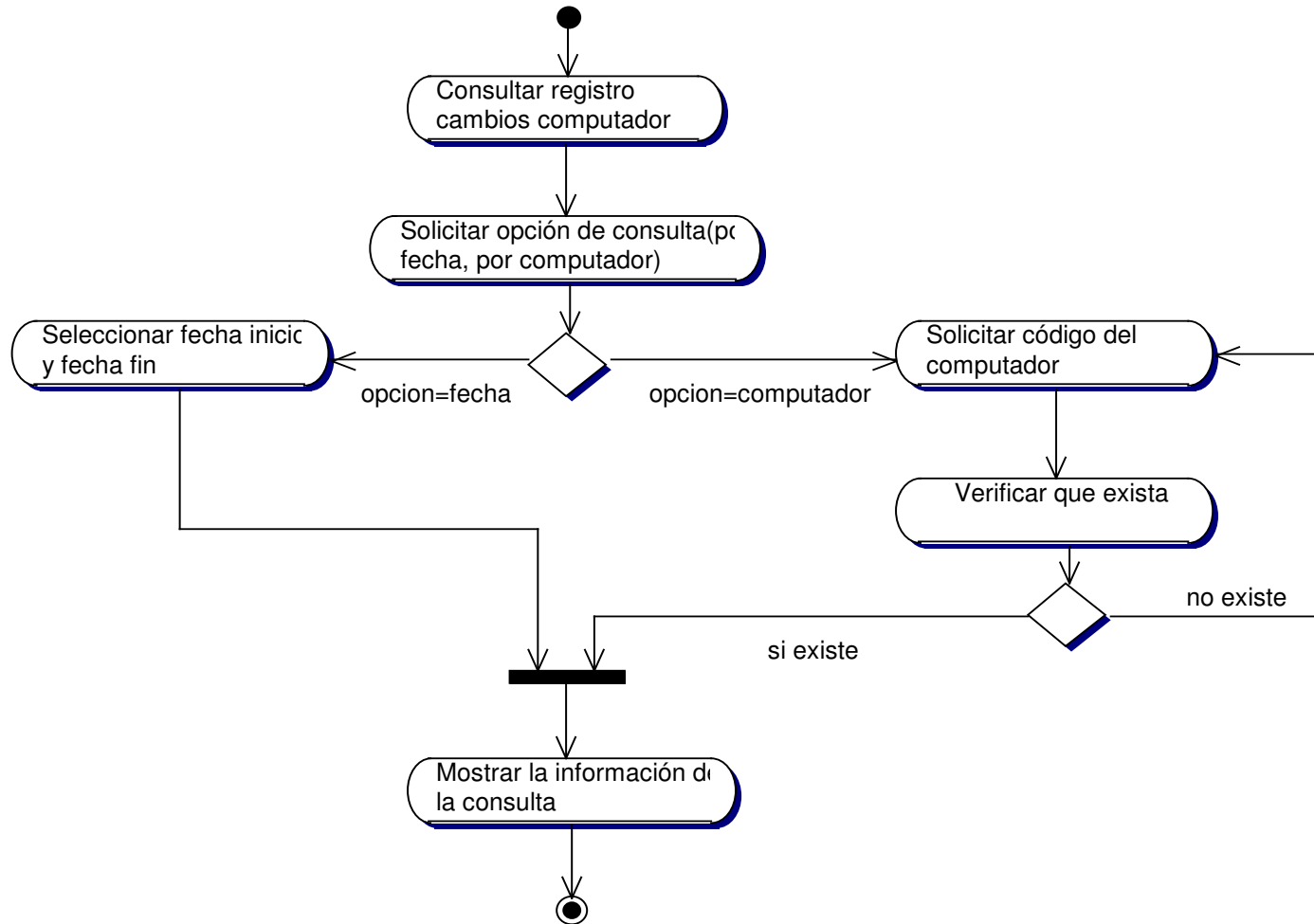




### 10.3.25.4 Eliminar Computador

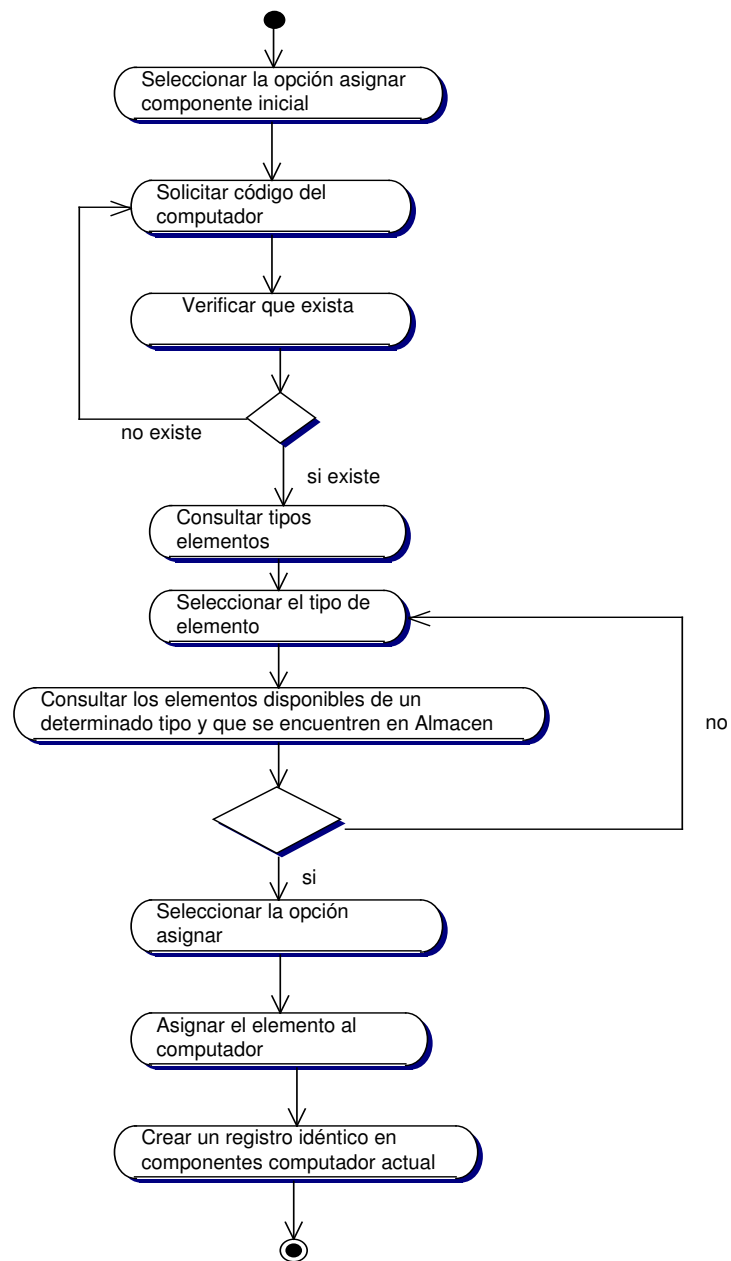


### 10.3.25.5 Consultar Registro Cambios Computador

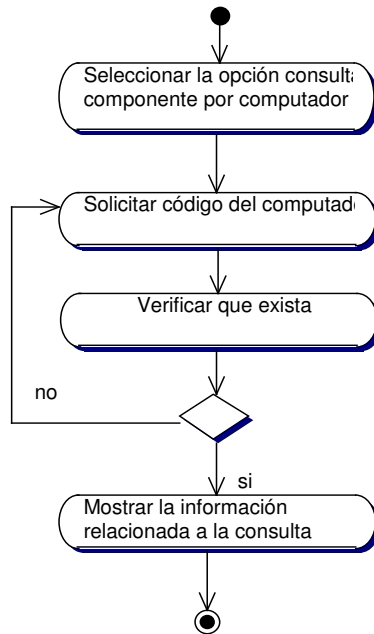


## 10.3.26 Administración de Componentes Computador Inicial

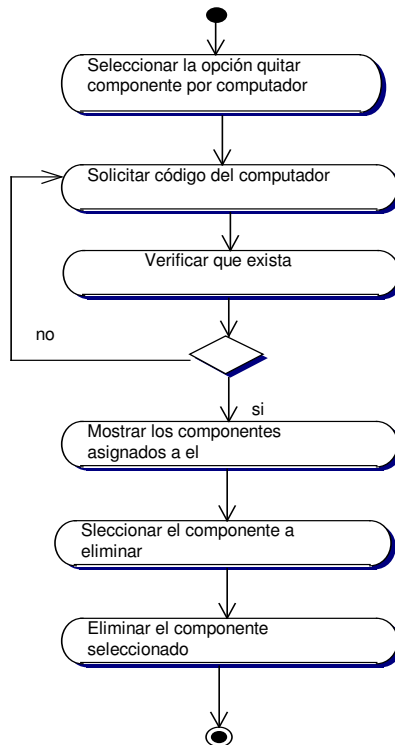
### 10.3.26.1 Asignar Componentes Iniciales



### 10.3.26.2 Consultar Componente por Computador

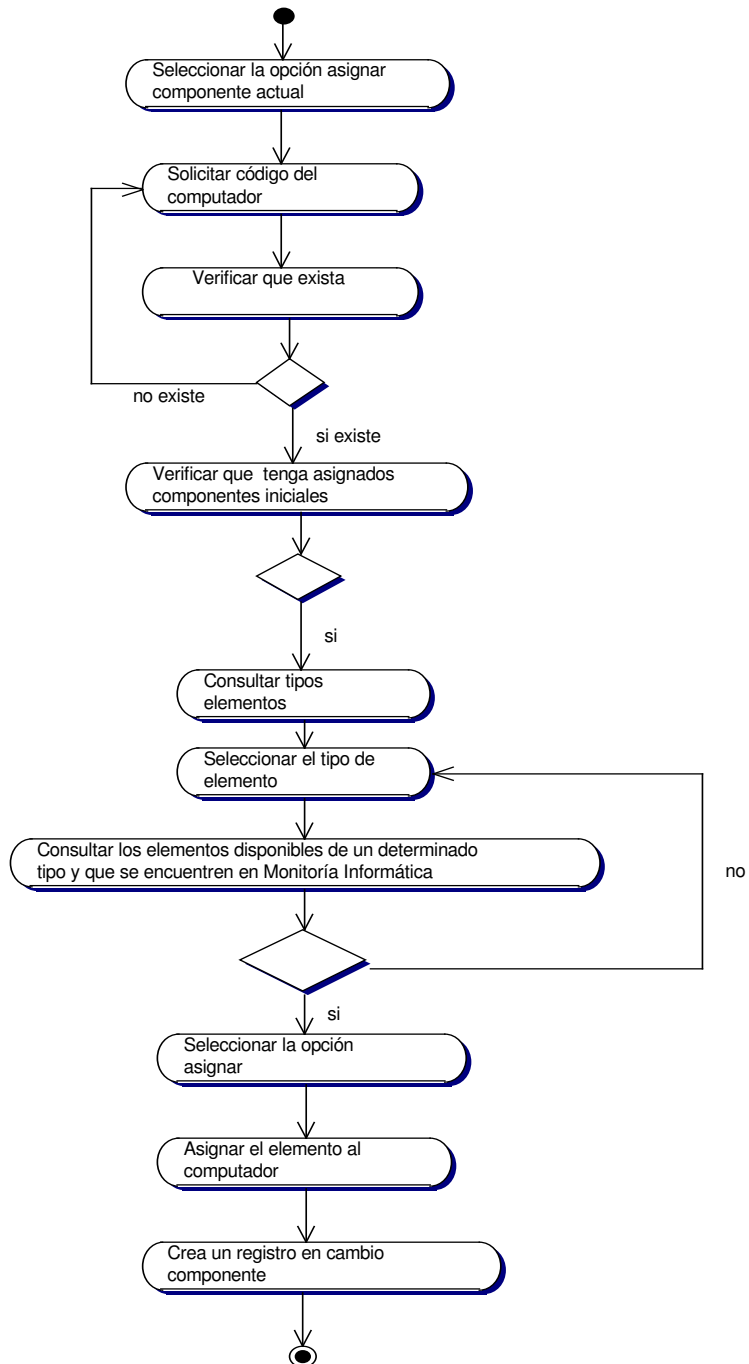


### 10.3.26.3 Quitar Componente Inicial

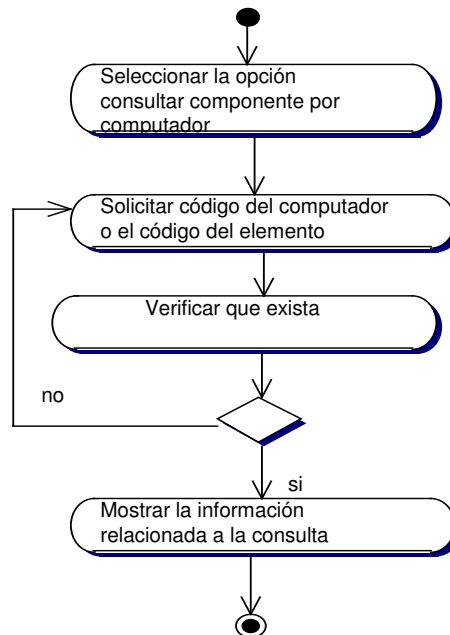


## 10.3.27 Administración de Componentes Computador Actual

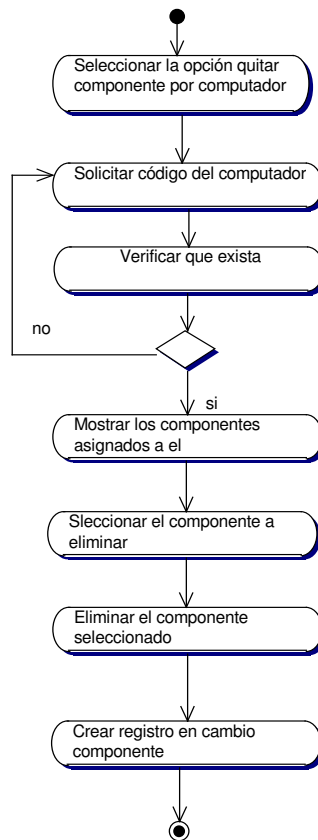
### 10.3.27.1 Asignar Componentes Actuales



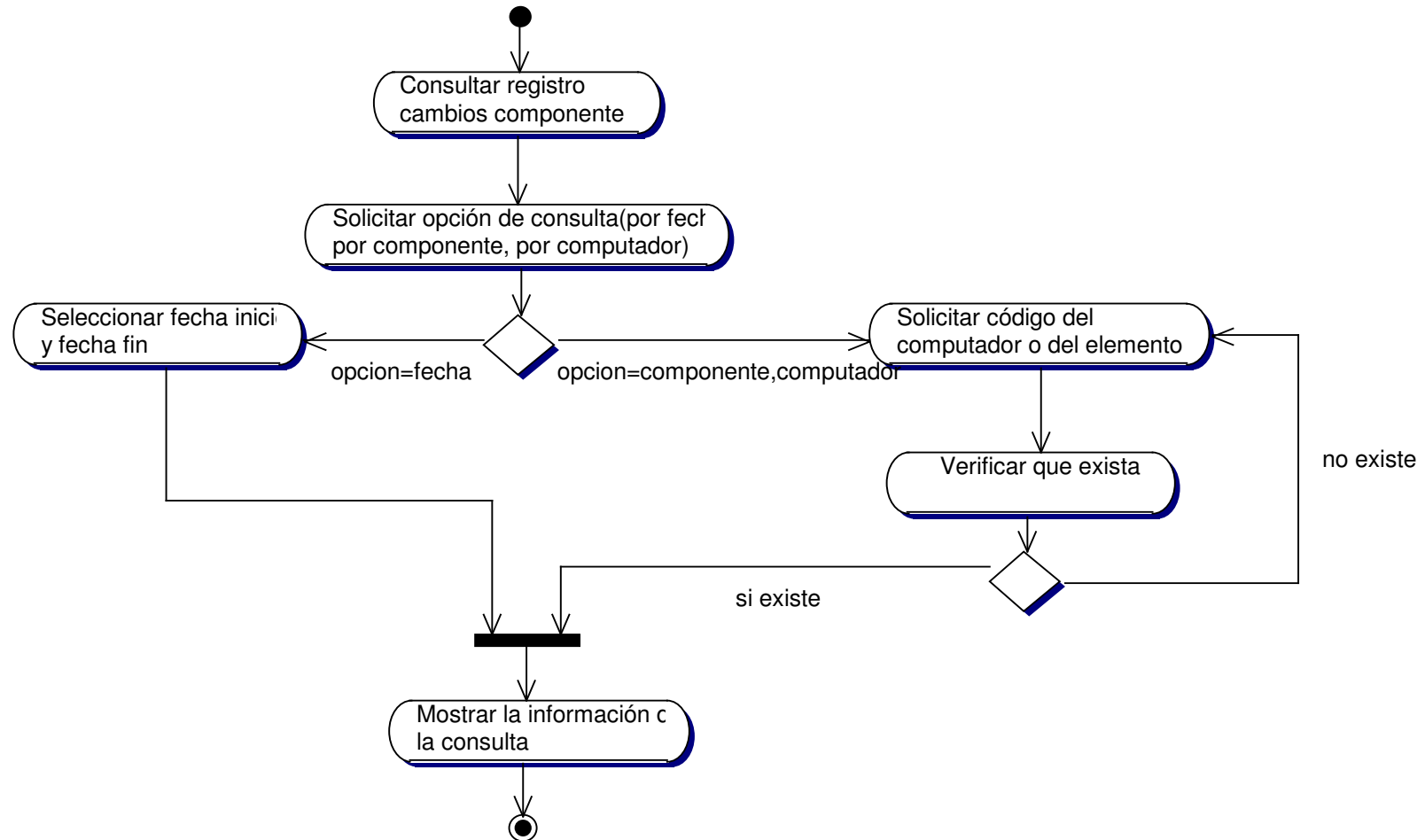
### 10.3.27.2 Consultar Componente por Computador



### 10.3.27.3 Quitar Componente Actual

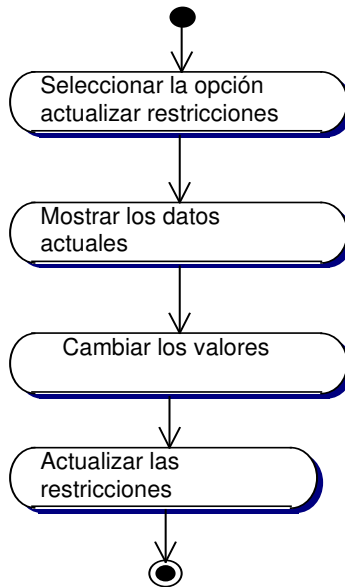


#### 10.3.27.4 Consultar Registro Cambio Componente

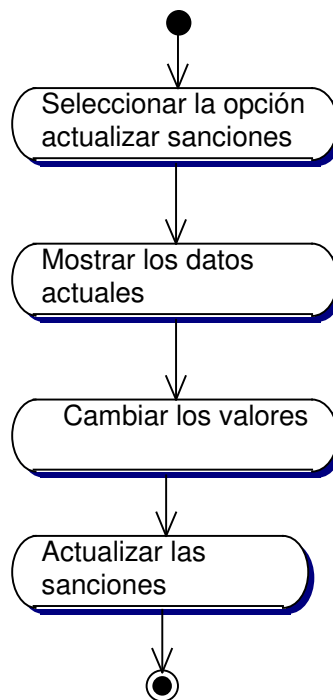


## 10.3.28 Administración Servicios Informáticos

### 10.3.28.1 Actualizar Restricciones

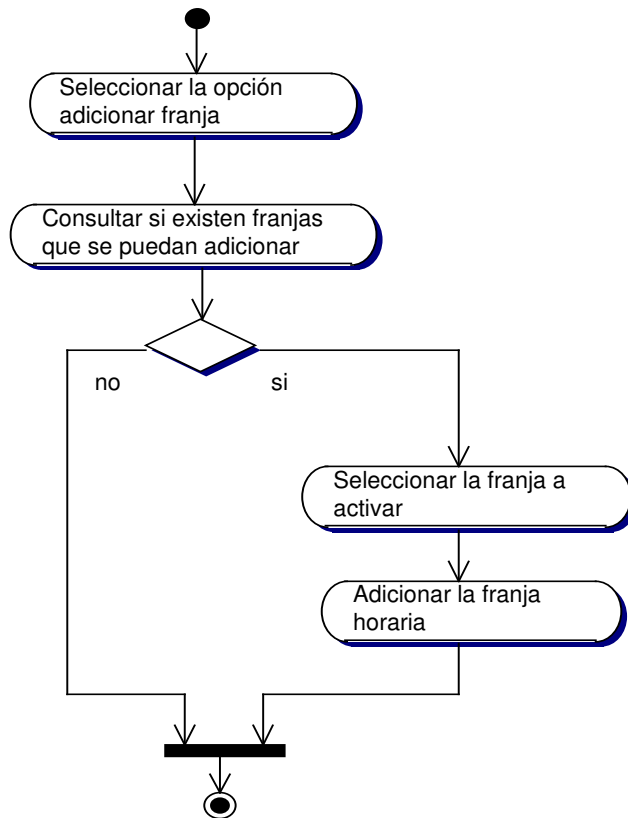


### 10.3.28.2 Actualizar Sanciones

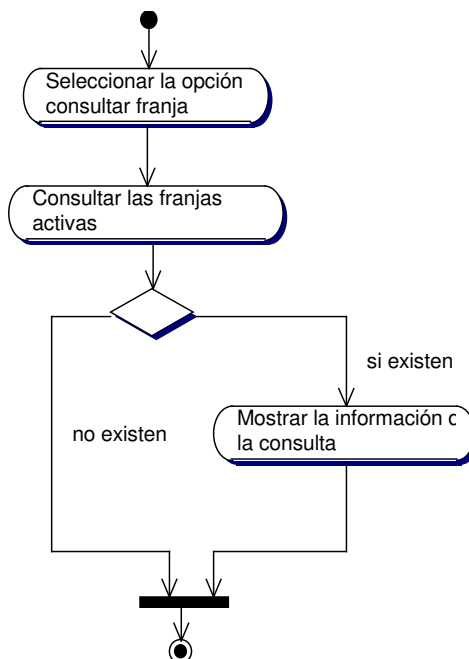




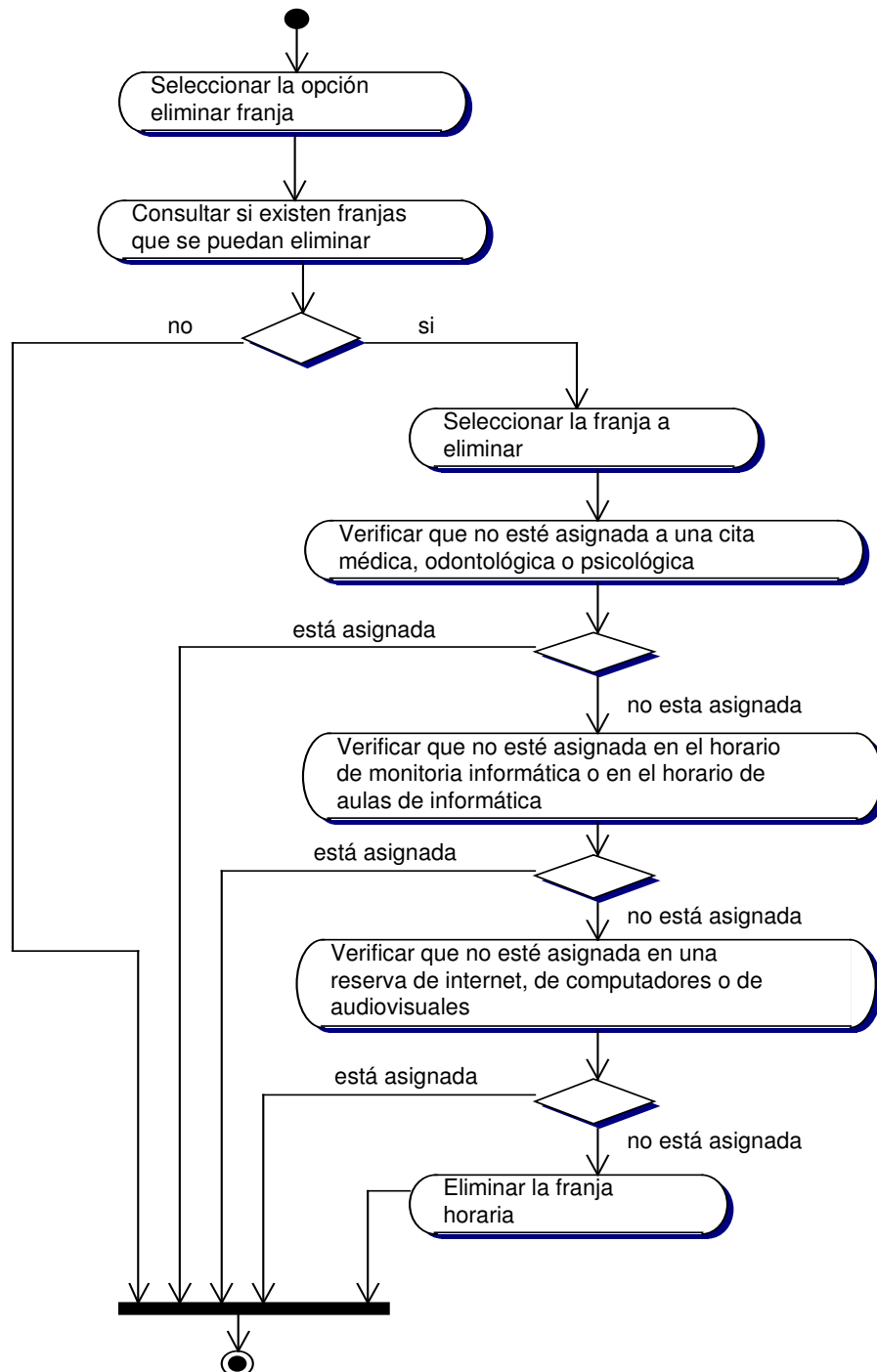
### 10.3.28.3 Adicionar Franja Horaria



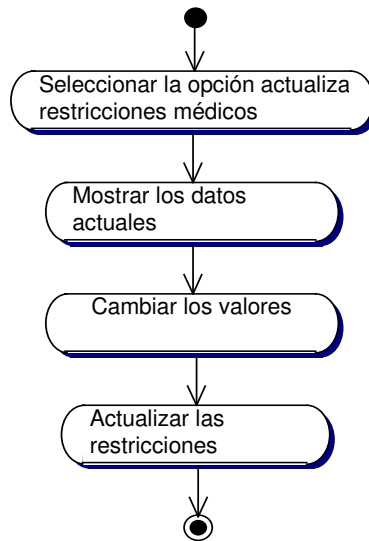
### 10.3.28.4 Consultar Franja Horaria



### 10.3.28.5 Eliminar Franja Horaria

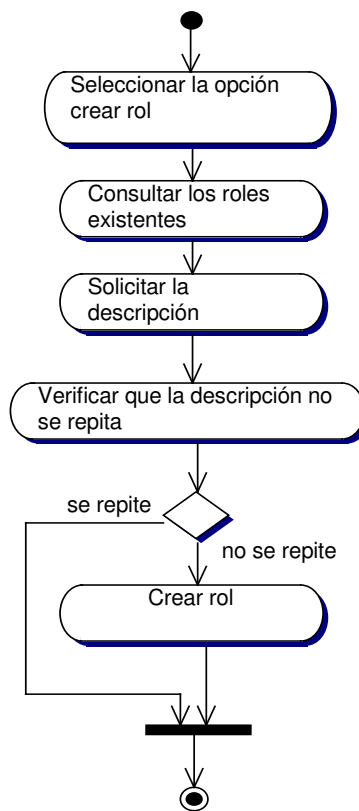


### 10.3.29 Actualizar Restricciones Médicas

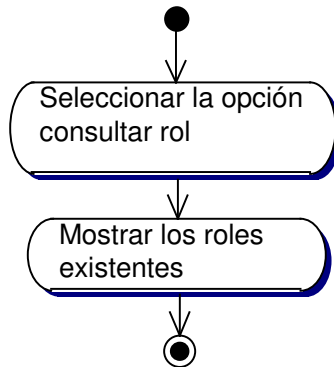


### 10.3.30 Administración de Roles

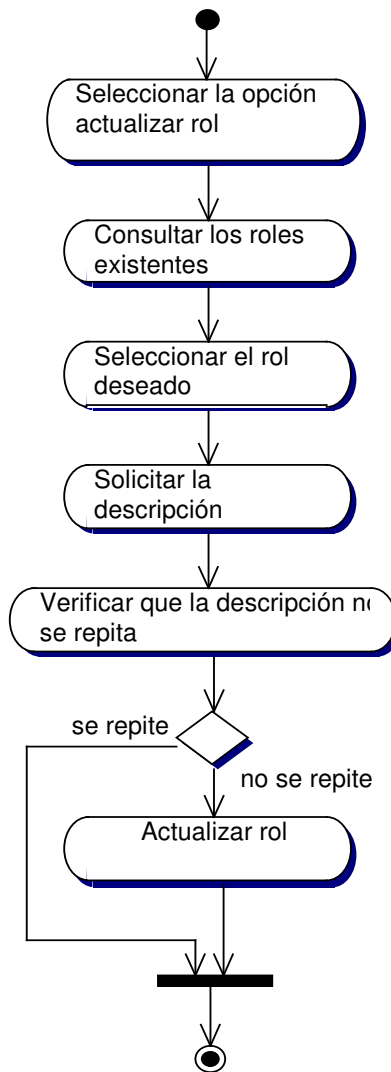
#### 10.3.30.1 Crear Rol



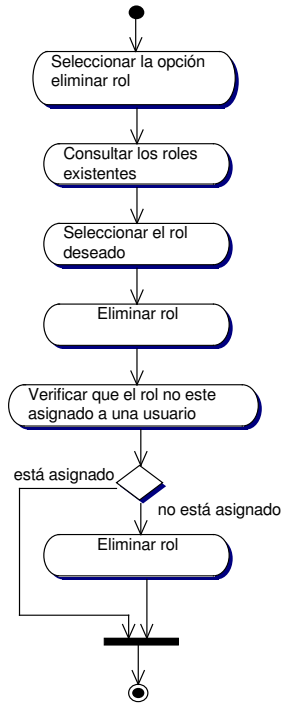
### 10.3.30.2 Consultar Rol



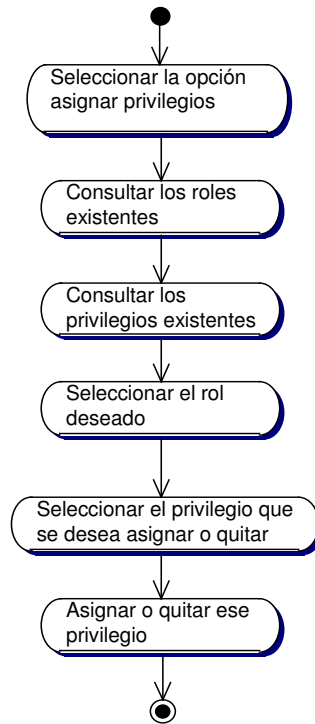
### 10.3.30.3 Actualizar Rol



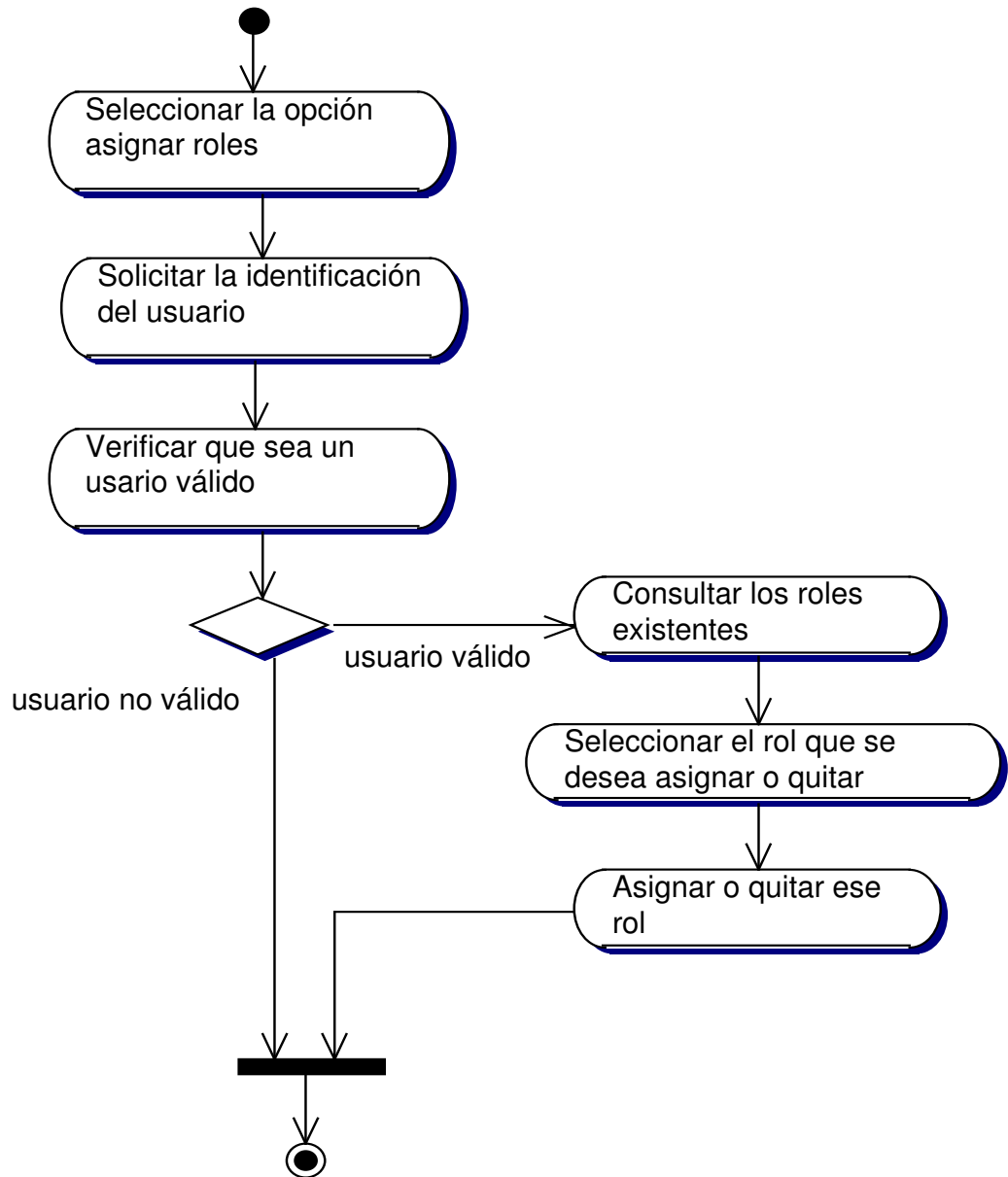
### 10.3.30.4 Eliminar Rol



### 10.3.30.5 Asignar Privilegios Rol

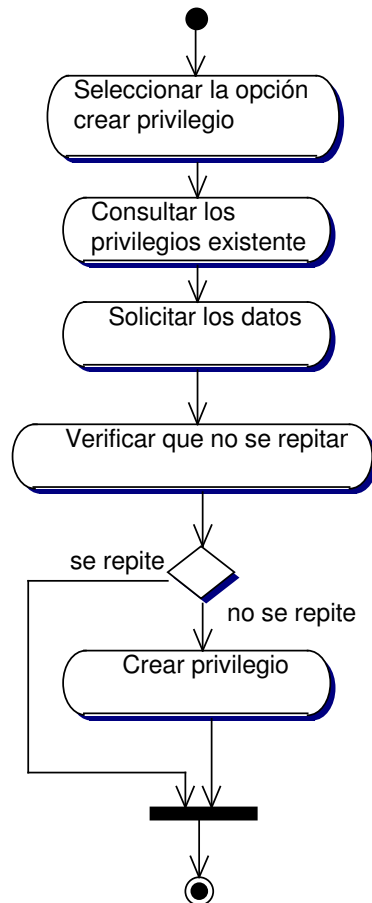


### 10.3.30.6 Asignar Roles Usuario

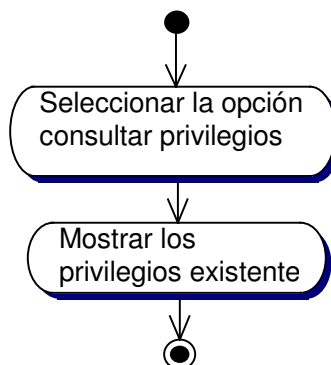


### 10.3.31 Administración de Privilegios

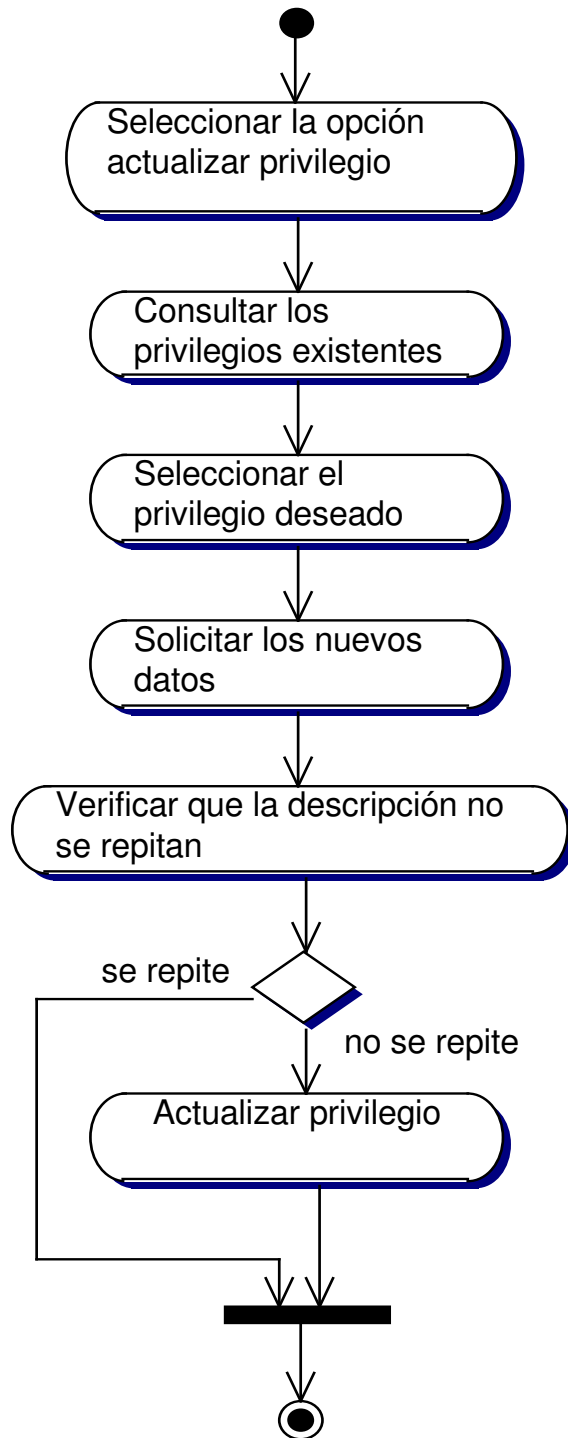
#### 10.3.31.1 Crear Privilegio



#### 10.3.31.2 Consultar Privilegios

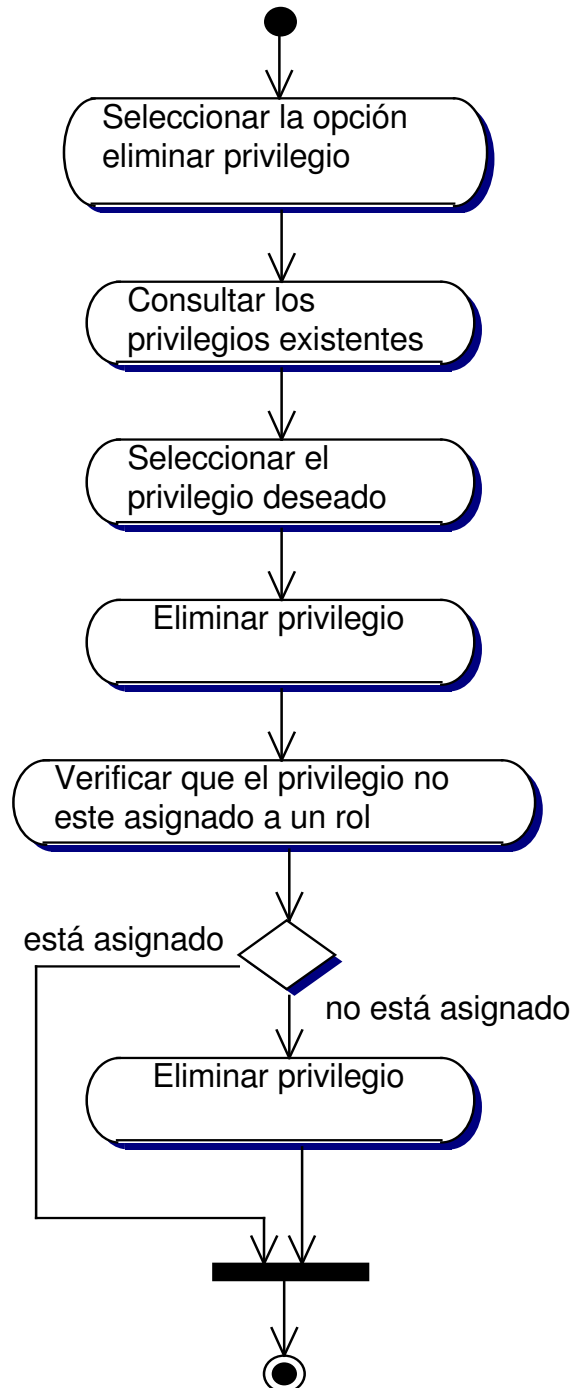


### 10.3.31.3 Actualizar Privilegio



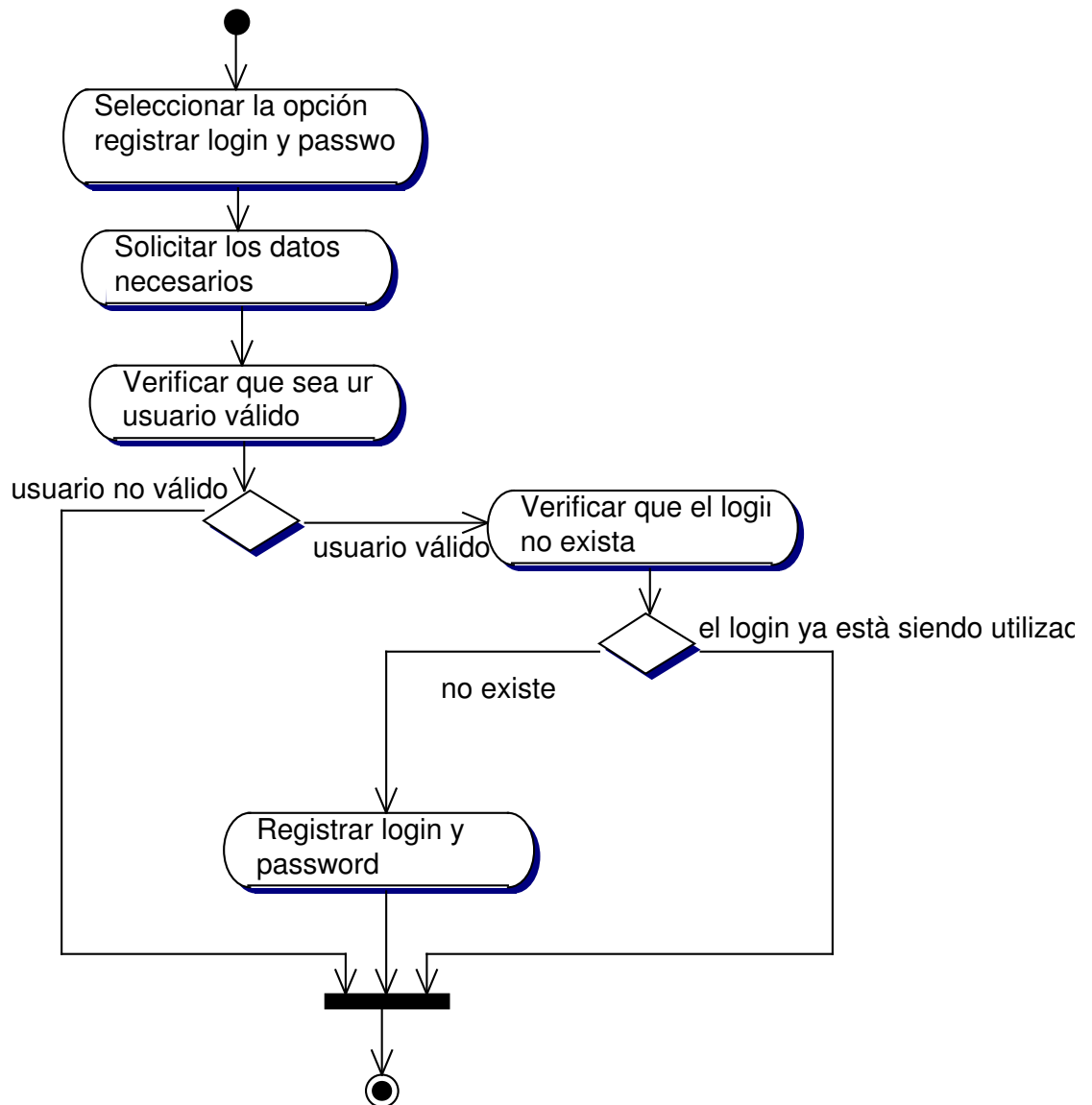


#### 10.3.31.4 Eliminar Privilegio

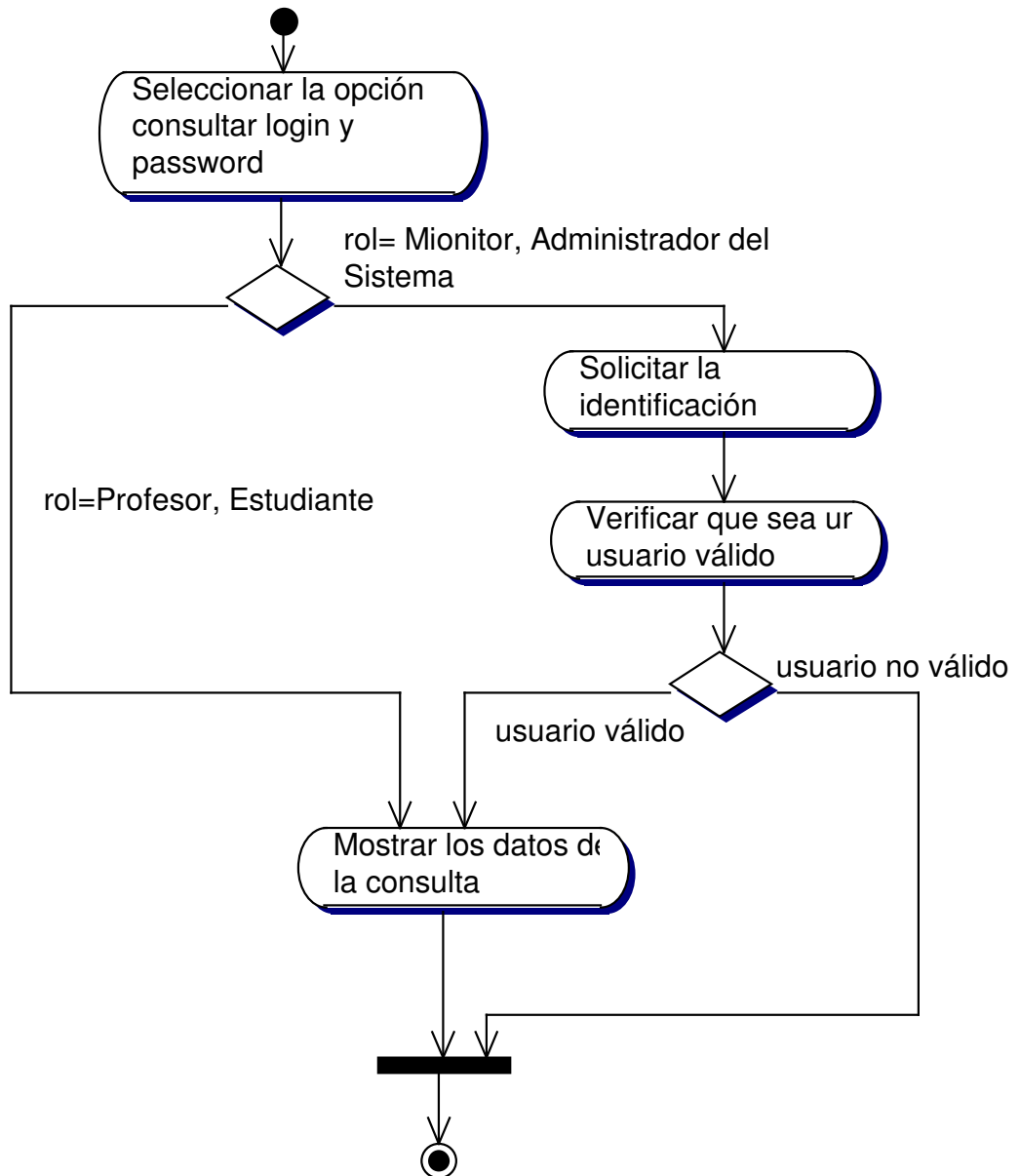


### 10.3.32 Administración de Login y Password

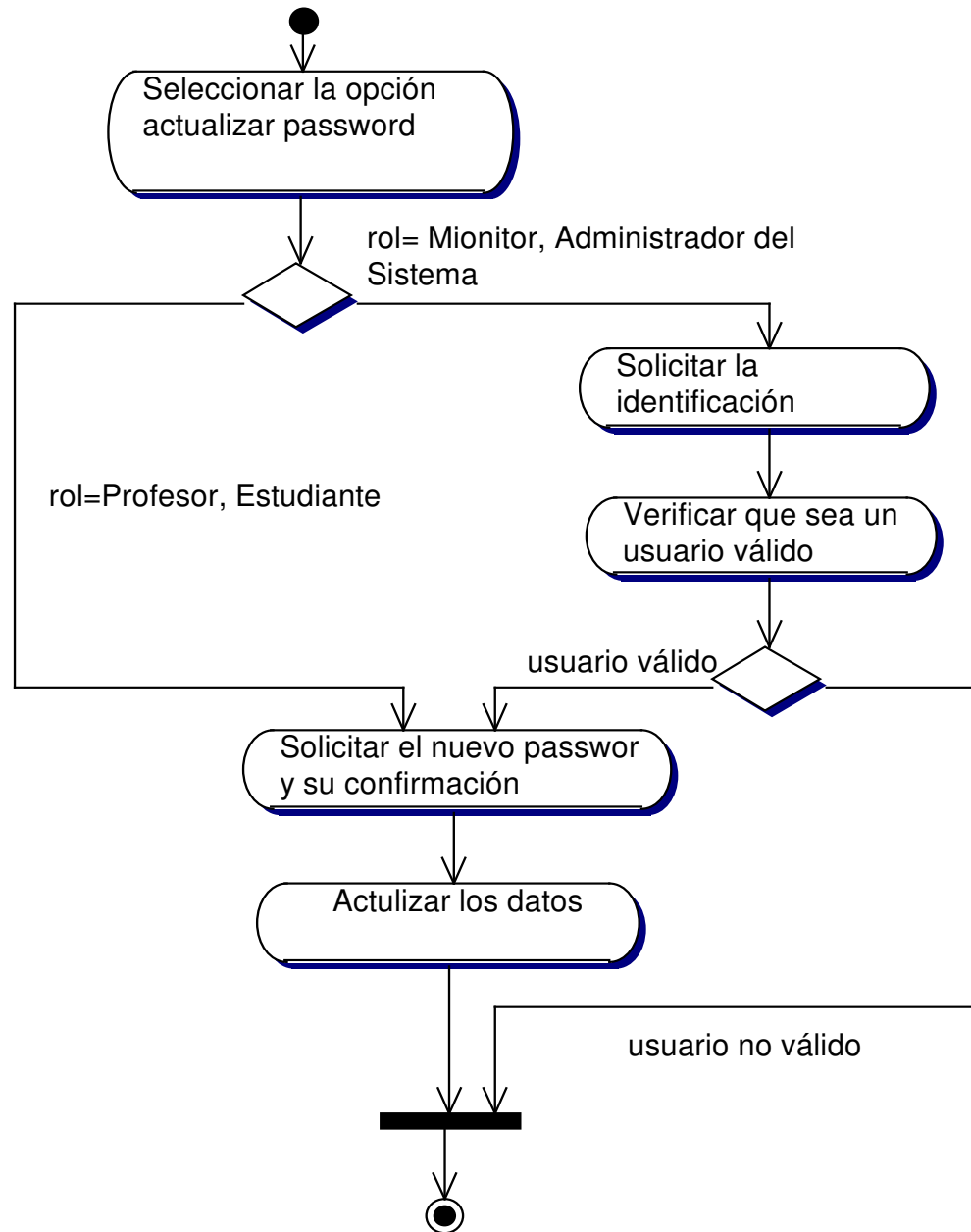
#### 10.3.32.1 Registrar Login y Password



### 10.3.32.2 Consultar Login y Password

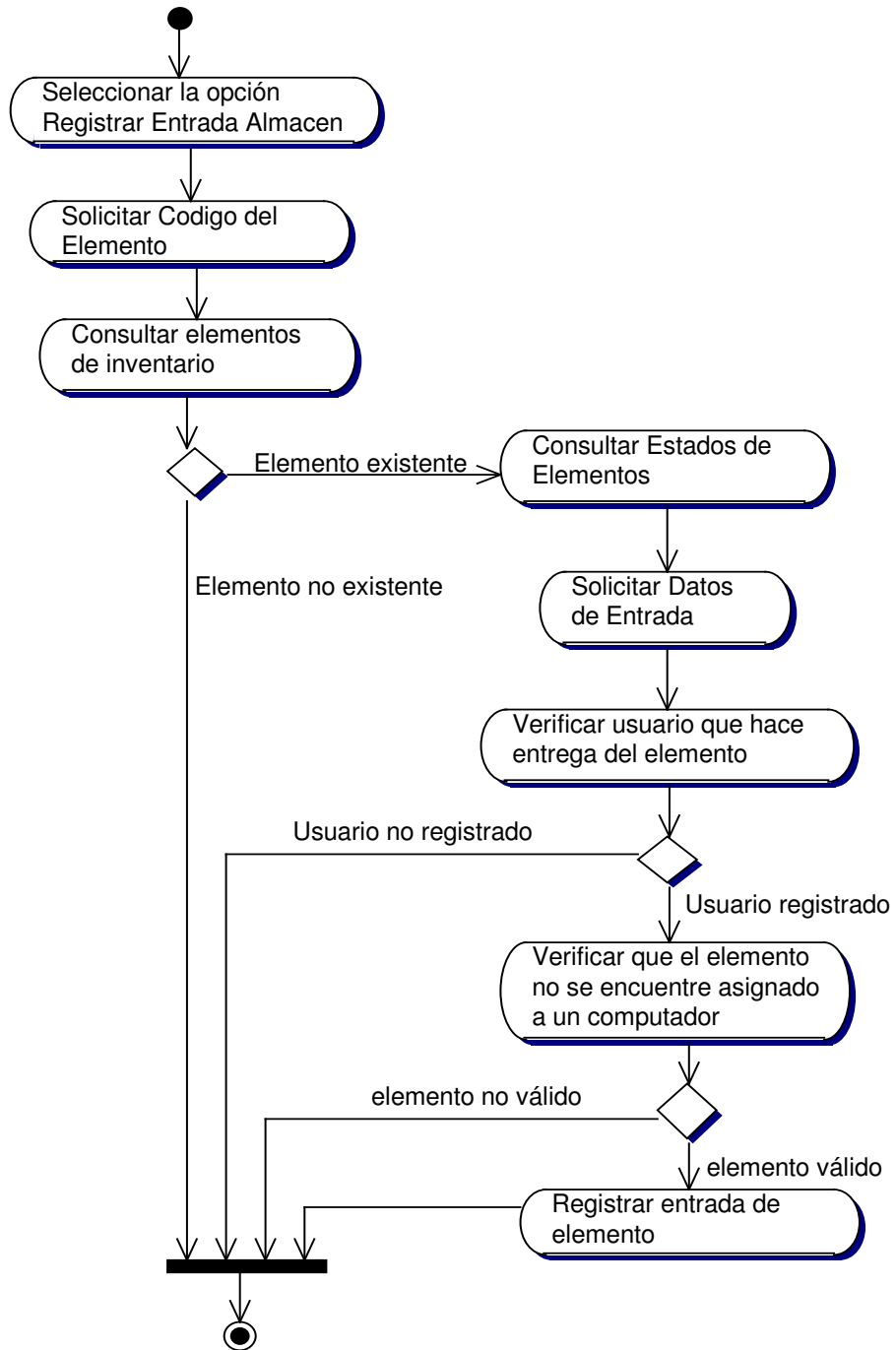


### 10.3.32.3 Actualizar Password

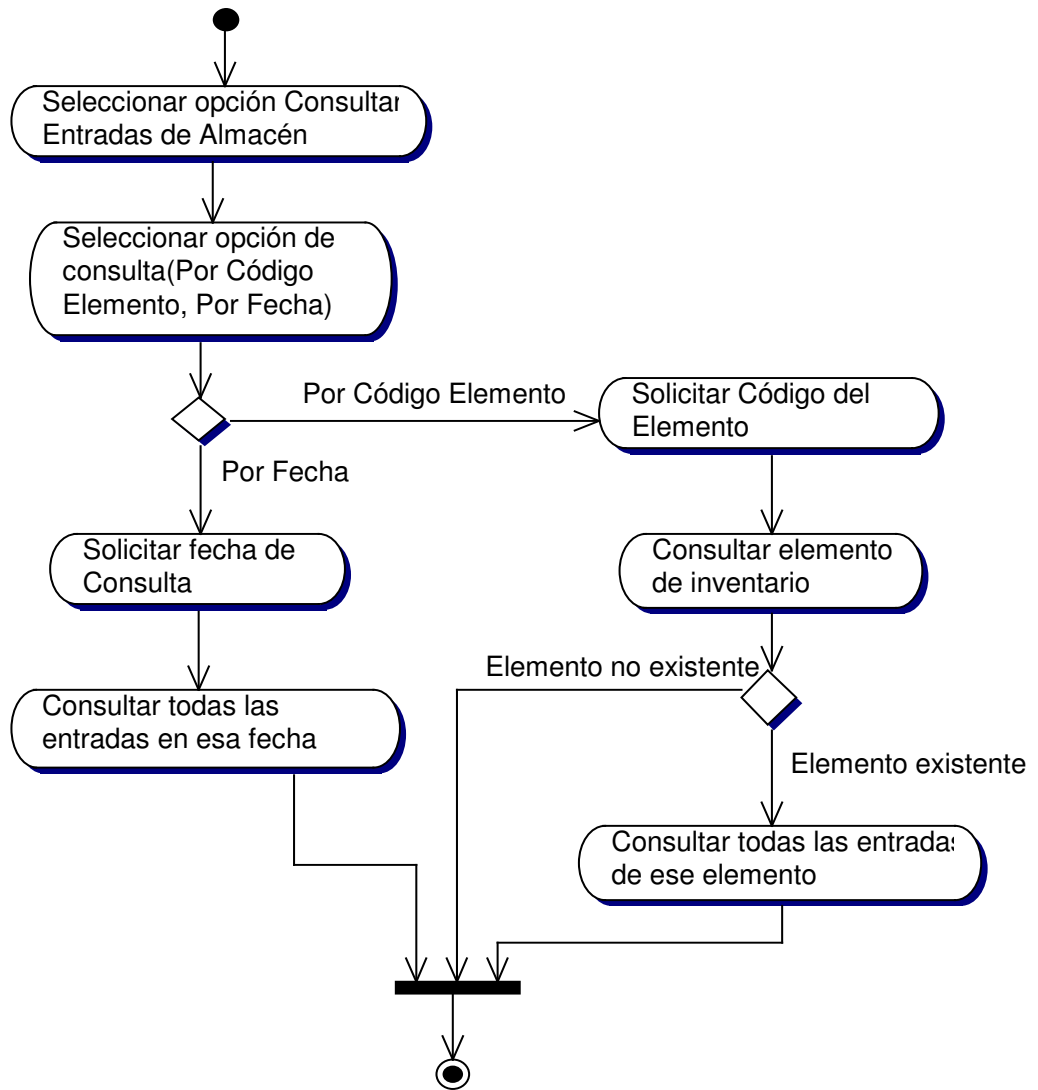


### 10.3.33 Administración de Entradas de Almacén

#### 10.3.33.1 Diagrama Registrar Entrada Almacén

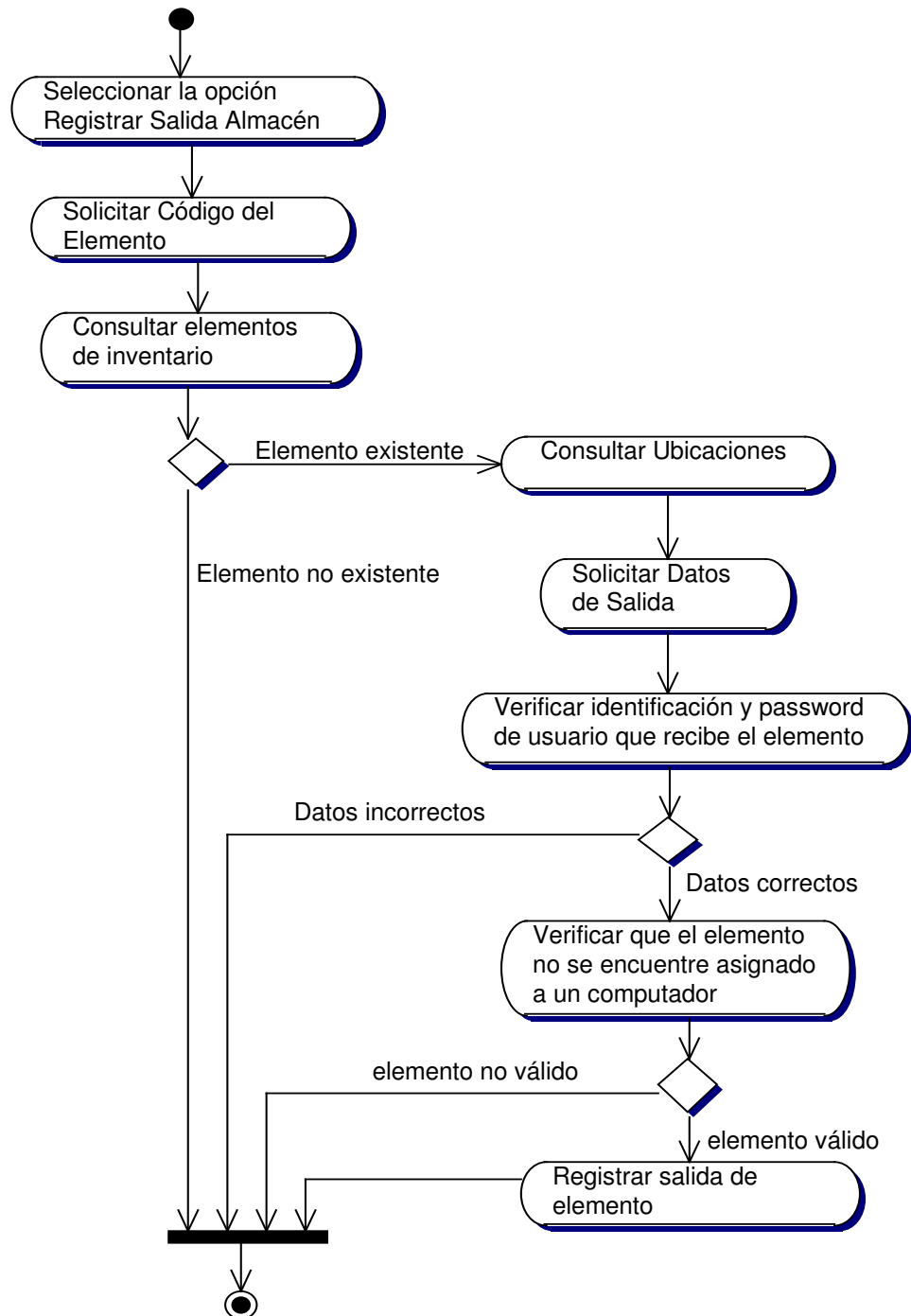


### 10.3.33.2 Diagrama Consultar Entrada Almacén

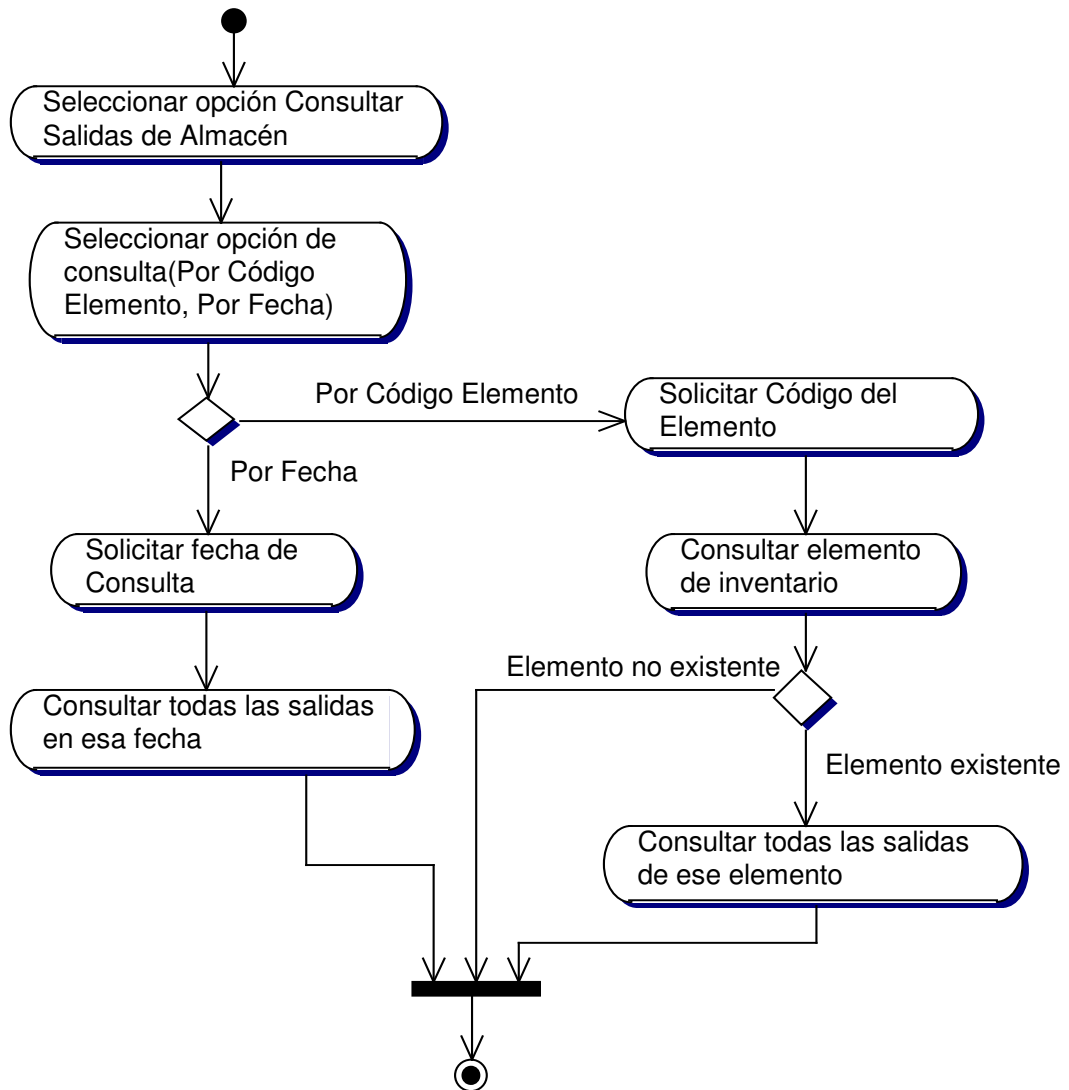


### 10.3.34 Administración de Salidas de Almacén

#### 10.3.34.1 Diagrama Registrar Salida Almacén



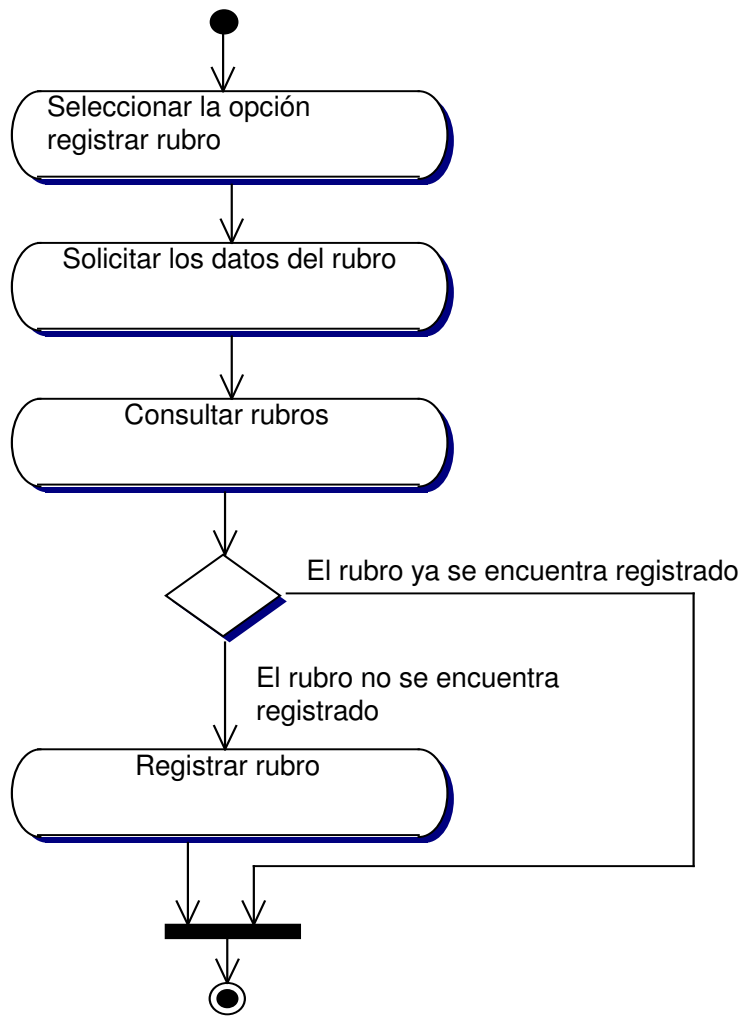
### 10.3.34.2 Diagrama Consultar Salida Almacén



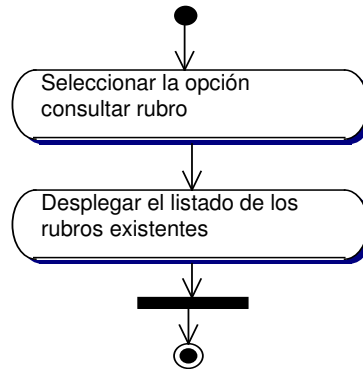


### 10.3.35 Administración Rubros

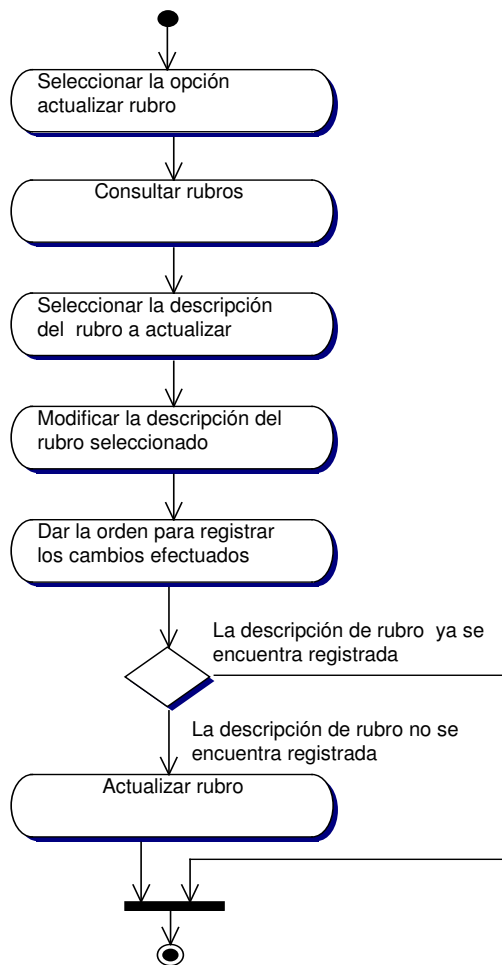
#### 10.3.35.1 Diagrama Registrar Rubro



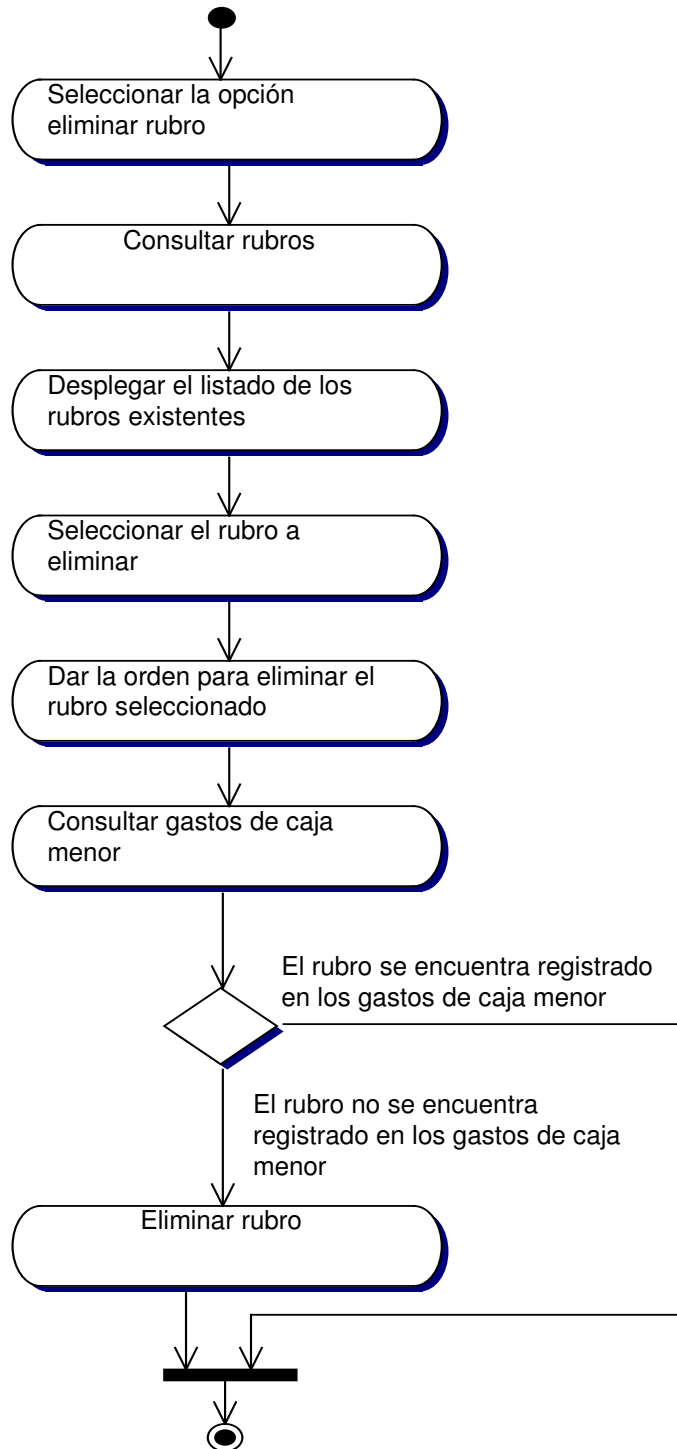
### 10.3.35.2 Diagrama Consultar Rubro



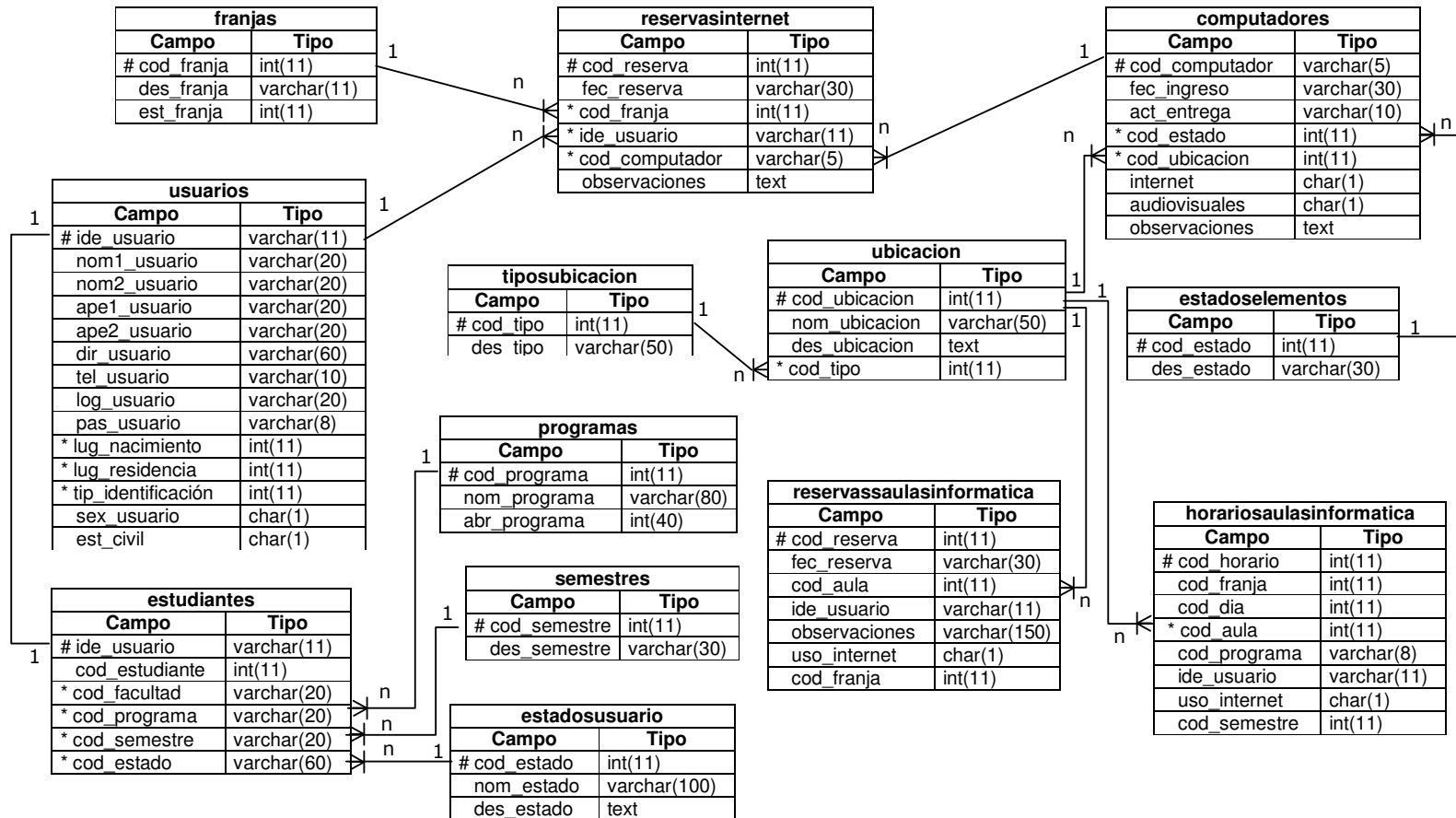
### 10.3.35.3 Diagrama Actualizar Rubro



### 10.3.35.4 Diagrama Eliminar Rubro



## 10.4 DIAGRAMA FISICO DE LA BASE DE DATOS (Administración Reservas Internet)



## 11. IMPLEMENTACION

### 11.1 DOCUMENTACION DE CLASES

#### 11.1.1 Monitoría Informática

##### 11.1.1.1 Reserva Internet Value Object

<b>Nombre de la Clase:</b> ReservaInternetVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las reservas de internet realizadas por un usuario		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codReserva	Identifica de manera única una Reserva de Internet	
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva	
String str_observacion	Cadena que permite almacenar anotaciones de alguna eventualidad que se presente	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	registradaEn	Asociación
UsuarioVO	pertenece	Asociación
ComputadorVO	tieneAsignado	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ReservaInternetVO	Método constructor sin parámetros	
getCodigoReserva	Retorna el código de la reserva	
getFechaReserva	Retorna la fecha en que se hizo la reserva	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getComputadorVO	Retorna un objeto de tipo ComputadorVO	
getObservacion	Retorna las observaciones hechas para la reserva	
setCodigoReserva	Permite asignar el código de la reserva	
setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setComputadorVO	Permite asignar o cambiar un objeto de tipo ComputadorVO	

setObservacion	Permite asignar o cambiar las observaciones de la reserva
<b>Observaciones:</b>	

### 11.1.1.2 Reserva Internet Visitor

<b>Nombre de la Clase:</b> ReservaInternetVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de las reservas de internet realizadas por un usuario, hace referencia directa con cada uno de los campos de la tabla reservasinternet	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codReserva	Identifica de manera única una Reserva de Internet
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva
int int_codFranja	Identifica el código de la franja en la cual se hizo la reserva
String str_ideUsuario	Determina la identificación del usuario que hizo la reserva
String str_codComputador	Identifica el código del computador asignado a una reserva.
String str_observacion	permite almacenar anotaciones de alguna eventualidad que se presente
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ReservaInternetVisitor	Método constructor sin parámetros
getCodigoReserva	Retorna el código de la reserva
getFechaReserva	Retorna la fecha en que se hizo la reserva
getCodigoFranja	Retorna el código de la franja en que se hizo la reserva
getIdentificacionUsuario	Retorna la identificación del Usuario dueño de la reserva
getCodigoComputador	Retorna el código del computador asignado a la reserva
getObservacion	Retorna las observaciones hechas para la reserva
setCodigoReserva	Permite asignar el código de la reserva
setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva
setCodigoFranja	Permite asignar o cambiar el código de la franja
setIdentificacionUsuario	Permite asignar o cambiar la identificación del usuario
setCodigoComputador	Permite asignar o cambiar el código del computador

setObservacion	Permite asignar o cambiar las observaciones de la reserva
<b>Observaciones:</b>	

### 11.1.1.3 Reserva Internet Manager

<b>Nombre de la Clase:</b> ReservaInternetMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las reservas de internet realizadas. Tiene toda la lógica de la clase ReservainternetVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_reservaInternet	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaInternetVO	retorna	Asociación
FranjaVO	utiliza	Asociación
UsuarioVO	utiliza	Asociación
ComputadorVO	utiliza	Asociación
ReservaInternetVisitor	utiliza	Asociación
FranjaMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
ComputadorMgr	utiliza	Asociación
ReservaInternetDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ReservaInternetVisitor para la consulta de una Reserva de acuerdo a un código. Retorna un objeto de tipo ReservaInternetVO	
findByCodigoComputador	Método que permite acceder a la consulta de todas las reservas de internet que involucra un determinado computador	
totalReservas	Método que permite acceder a la consulta del número de reservas de un usuario en una fecha y franja determinada	
totalReservas	Método que permite acceder a la consulta del número de reservas de un usuario en una fecha determinada	
totalReservas	Método que permite acceder a la consulta del número de reservas en una fecha y franja determinada	

findByFecha	Método que permite acceder a la consulta de todas las reservas de internet en una fecha determinada
findByFechaFranjaComputador	Método que permite establecer los atributos de un objeto de tipo ReservaInternetVisitor para la consulta de una reserva de acuerdo a una fecha, franja y computador determinado
findByFechaIdentificacion	Método que permite acceder a la consulta de todas las reservas de internet de un usuario en una fecha determinada
findByFechaComputador	Método que permite acceder a la consulta de todas las reservas de internet que involucran un computador en una fecha determinada
createReserva	Método que permite establecer los atributos de un objeto de tipo ReservaInternetVisitor para la creación de una reserva de internet
updateReserva	Método que permite establecer los atributos de un objeto de tipo ReservaInternetVisitor para la actualización de una reserva de internet
updateObservacion	Método que permite establecer los atributos de un objeto de tipo ReservaInternetVisitor para la actualización de las observaciones de una reserva de internet
deleteReserva	Método que permite establecer los atributos de un objeto de tipo ReservaInternetVisitor para la eliminación de una reserva de internet
deleteReservaSancionInternet	Método que permite establecer los atributos de un objeto de tipo ReservaInternetVisitor para la eliminación de una reserva de internet de un usuario en una fecha determinada
<b>Observaciones:</b>	

#### 11.1.1.4 Reserva Internet Data Access Object

<b>Nombre de la Clase:</b> ReservaInternetDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las reservas de internet		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>



ReservaInternetVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Retorna un objeto de tipo ReservaInternetVisitor de acuerdo al código de la reserva	
findByCodigoComputador	Método que retorna un ArrayList(colección de objetos) de tipo ReservaInternetVisitor con todas las reservas de internet que involucren un determinado computador	
totalReservasFranjaUsuario	Retorna el número de reservas que un usuario tiene en una determinada fecha y franja	
totalReservasFechaUsuario	Retorna el número de reservas que un usuario tiene en una determinada fecha	
totalReservasFechaFranja	Retorna el número de reservas existentes en una determinada fecha y franja	
findByFecha	Retorna un ArrayList(colección de objetos) de tipo ReservaInternetVisitor con los datos de todas las reservas realizadas en una determinada fecha	
findByFechaFranjaComputador	Retorna un objeto de tipo ReservaInternetVisitor de acuerdo a una fecha, franja y computador determinado	
findByFechaIdentificacion	Retorna un ArrayList de objetos de tipo ReservaInternetVisitor con los datos de todas las reservas realizadas por un usuario en una determinada fecha	
findByFechaComputador	Retorna un ArrayList de objetos de tipo ReservaInternetVisitor con los datos de todas las reservas que involucran un computador en una determinada fecha	
createReserva	Método que permite crear una reserva de internet	
updateReserva	Método que permite actualizar los datos de una reserva de internet	
updateObservacion	Método que permite actualizar las observaciones realizadas a una determinada reserva	
deleteReserva	Método que elimina una reserva de internet de acuerdo a un código determinado	
deleteReservaSancionInternet	Método que elimina todas las reservas de internet de un usuario en una fecha determinada	
<b>Observaciones:</b>		

#### 11.1.1.5 Reserva Computador Value Object

<b>Nombre de la Clase:</b> ReservaComputadorVO
---

<b>Descripción de la clase</b>		
Clase que representa de forma lógica las reservas de computadores realizadas por un usuario		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codReserva	Identifica de manera única una Reserva de Computadores	
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva	
String str_observacion	Cadena que permite almacenar anotaciones de alguna eventualidad que se presente	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	registradaEn	Asociación
UsuarioVO	pertenece	Asociación
ComputadorVO	tieneAsignado	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ReservaComputadorVO	Método constructor sin parámetros	
getCodigoReserva	Retorna el código de la reserva	
getFechaReserva	Retorna la fecha en que se hizo la reserva	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getComputadorVO	Retorna un objeto de tipo ComputadorVO	
getObservacion	Retorna las observaciones hechas para la reserva	
setCodigoReserva	Permite asignar el código de la reserva	
setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setComputadorVO	Permite asignar o cambiar un objeto de tipo ComputadorVO	
setObservacion	Permite asignar o cambiar las observaciones de la reserva	
<b>Observaciones:</b>		

### 11.1.1.6 Reserva Computador Visitor

<b>Nombre de la Clase:</b> ReservaComputadorVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de las reservas de computadores realizadas por un usuario, hace referencia directa con cada uno de los campos de la tabla reservascomputadores	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

int int_codReserva	Identifica de manera única una Reserva de Computadores
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva
int int_codFranja	Identifica el código de la franja en la cual se hizo la reserva
String str_ideUsuario	Determina la identificación del usuario que hizo la reserva
String str_codComputador	Identifica el código del computador asignado a una reserva.
String str_observacion	permite almacenar anotaciones de alguna eventualidad que se presente
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ReservaComputadorVisitor	Método constructor sin parámetros
getCodigoReserva	Retorna el código de la reserva
getFechaReserva	Retorna la fecha en que se hizo la reserva
getCodigoFranja	Retorna el código de la franja en que se hizo la reserva
getIdentificacionUsuario	Retorna la identificación del Usuario dueño de la reserva
getCodigoComputador	Retorna el código del computador asignado a la reserva
getObservacion	Retorna las observaciones hechas para la reserva
setCodigoReserva	Permite asignar el código de la reserva
setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva
setCodigoFranja	Permite asignar o cambiar el código de la franja
setIdentificacionUsuario	Permite asignar o cambiar la identificación del usuario
setCodigoComputador	Permite asignar o cambiar el código del computador
setObservacion	Permite asignar o cambiar las observaciones de la reserva
<b>Observaciones:</b>	

### 11.1.1.7 Reserva Computador Manager

<b>Nombre de la Clase:</b> ReservaComputadorMgr	
<b>Descripción de la clase</b> Clase que permite la administración de las reservas de computadores realizadas. Tiene toda la lógica de la clase ReservaComputadorVO a nivel de modelo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

ArrayList al_reservaComputador	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaComputadorVO	retorna	Asociación
FranjaVO	utiliza	Asociación
UsuarioVO	utiliza	Asociación
ComputadorVO	utiliza	Asociación
ReservaComputadorVisitor	utiliza	Asociación
FranjaMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
ComputadorMgr	utiliza	Asociación
ReservaComputadorDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ReservaComputadorVisitor para la consulta de una Reserva de acuerdo a un código. Retorna un objeto de tipo ReservaComputadorVO	
findByCodigoComputador	Método que permite acceder a la consulta de todas las reservas de computadores que involucra un determinado computador	
totalReservas	Método que permite acceder a la consulta del número de reservas de un usuario en una fecha y franja determinada	
totalReservas	Método que permite acceder a la consulta del número de reservas de un usuario en una fecha determinada	
totalReservas	Método que permite acceder a la consulta del número de reservas en una fecha y franja determinada	
findByFecha	Método que permite acceder a la consulta de todas las reservas de computadores en una fecha determinada	
findByFechaFranjaComputador	Método que permite establecer los atributos de un objeto de tipo ReservaComputadorVisitor para la consulta de una reserva de acuerdo a una fecha, franja y computador determinado	
findByFechaIdentificacion	Método que permite acceder a la consulta de todas las reservas de computadores de un usuario en una fecha determinada	
findByFechaComputador	Método que permite acceder a la consulta de todas las reservas de computadores que involucran un computador en una fecha determinada	
createReserva	Método que permite establecer los atributos de un objeto de tipo ReservaComputadorVisitor para la creación de una reserva de computadores	

updateReserva	Método que permite establecer los atributos de un objeto de tipo ReservaComputadorVisitor para la actualización de una reserva de computadores
updateObservacion	Método que permite establecer los atributos de un objeto de tipo ReservaComputadorVisitor para la actualización de las observaciones de una reserva de computadores
deleteReserva	Método que permite establecer los atributos de un objeto de tipo ReservaComputadorVisitor para la eliminación de una reserva de computadores
deleteReservaSancionComputador	Método que permite establecer los atributos de un objeto de tipo ReservaComputadorVisitor para la eliminación de una reserva de computadores de un usuario en una fecha determinada
<b>Observaciones:</b>	

#### 11.1.1.8 Reserva Computador Data Access Object

<b>Nombre de la Clase:</b> ReservaComputadorDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las reservas de computadores		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaComputadorVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Retorna un objeto de tipo ReservaComputadorVisitor de acuerdo al código de la reserva	
findByCodigoComputador	Método que retorna un ArrayList(colección de objetos) de tipo ReservaComputadorVisitor con todas las reservas de computadores que involucren un determinado computador	
totalReservasFranjaUsuario	Retorna el número de reservas que un usuario tiene en una determinada fecha y franja	
totalReservasFechaUsuario	Retorna el número de reservas que un usuario tiene en una determinada fecha	

totalReservasFechaFranja	Retorna el número de reservas existentes en una determinada fecha y franja
findByFecha	Retorna un ArrayList(colección de objetos) de tipo ReservaComputadorVisitor con los datos de todas las reservas realizadas en una determinada fecha
findByFechaFranjaComputador	Retorna un objeto de tipo ReservaComputadorVisitor de acuerdo a una fecha, franja y computador determinado
findByFechaIdentificacion	Retorna un ArrayList de objetos de tipo ReservaComputadorVisitor con los datos de todas las reservas realizadas por un usuario en una determinada fecha
findByFechaComputador	Retorna un ArrayList de objetos de tipo ReservaComputadorVisitor con los datos de todas las reservas que involucran un computador en una determinada fecha
createReserva	Método que permite crear una reserva de computadores
updateReserva	Método que permite actualizar los datos de una reserva de computadores
updateObservacion	Método que permite actualizar las observaciones realizadas a una determinada reserva
deleteReserva	Método que elimina una reserva de computadores de acuerdo a un código determinado
deleteReservaSancionComputador	Método que elimina todas las reservas de computadores de un usuario en una fecha determinada
<b>Observaciones:</b>	

### 11.1.1.9 Reserva Audiovisual Value Object

<b>Nombre de la Clase:</b> ReservaAudiovisualVO	
<b>Descripción de la clase</b> Clase que representa de forma lógica las reservas de audiovisuales realizadas por un usuario	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codReserva	Identifica de manera única una Reserva de Audiiovisuales
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva
String str_observacion	Cadena que permite almacenar anotaciones de alguna eventualidad que se presente

<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	registradaEn	Asociación
ServicioAudiovisual	tieneReservado	Asociación
UsuarioVO	pertenece	Asociación
ProgramaVO	pertenece	Asociación
SemestreVO	pertenece	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ReservaAudiovisualVO	Método constructor sin parámetros	
getCodigoReserva	Retorna el código de la reserva	
getFechaReserva	Retorna la fecha en que se hizo la reserva	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getServicioAudiovisualVO	Retorna un objeto de tipo ServicioAudiovisualVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getProgramaVO	Retorna un objeto de tipo ProgramaVO	
getSemestreVO	Retorna un objeto de tipo SemestreVO	
getObservacion	Retorna las observaciones hechas para la reserva	
setCodigoReserva	Permite asignar el código de la reserva	
setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
setServicioAudiovisualVO	Permite asignar o cambiar un objeto de tipo ServicioAudiovisualVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setProgramaVO	Permite asignar o cambiar un objeto de tipo ProgramaVO	
setSemestreVO	Permite asignar o cambiar un objeto de tipo SemestreVO	
setObservacion	Permite asignar o cambiar las observaciones de la reserva	
<b>Observaciones:</b>		

#### 11.1.1.10 Reserva Audiovisual Visitor

<b>Nombre de la Clase:</b> ReservaAudiovisualVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de las reservas de audiovisuales realizadas por un usuario, hace referencia directa con cada uno de los campos de la tabla reservasaudiovisuales	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codReserva	Identifica de manera única una Reserva de Audiovisuales

String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva
int int_codFranja	Identifica el código de la franja en la cual se hizo la reserva
int int_codServicio	Identifica el código del servicio audiovisual reservado
String str_ideUsuario	Determina la identificación del usuario que hizo la reserva
String str_codPrograma	Identifica el código del programa que hará uso de la reserva
String str_codSemestre	Identifica el código del semestre que hará uso de la reserva
String str_observacion	permite almacenar anotaciones de alguna eventualidad que se presente
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ReservaAudiovisualVisitor	Método constructor sin parámetros
getCodigoReserva	Retorna el código de la reserva
getFechaReserva	Retorna la fecha en que se hizo la reserva
getCodigoFranja	Retorna el código de la franja en que se hizo la reserva
getCodigoServicio	Retorna el código del servicio audiovisual reservado
getIdentificacionUsuario	Retorna la identificación del Usuario dueño de la reserva
getCodigoPrograma	Retorna el código del programa que hará uso la reserva
getCodigoSemestre	Retorna el código del semestre que hará uso la reserva
getObservacion	Retorna las observaciones hechas para la reserva
setCodigoReserva	Permite asignar el código de la reserva
setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva
setCodigoFranja	Permite asignar o cambiar el código de la franja
setCodigoServicio	Permite asignar o cambiar el código del servicio
setIdentificacionUsuario	Permite asignar o cambiar la identificación del usuario
setCodigoPrograma	Permite asignar o cambiar el código del programa que hará uso de la reserva
setCodigoSemestre	Permite asignar o cambiar el código del semestre que hará uso de la reserva
setObservacion	Permite asignar o cambiar las observaciones de la reserva
<b>Observaciones:</b>	

#### 11.1.1.11 Reserva Audiovisual Manager



<b>Nombre de la Clase:</b> ReservaAudiovisualMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las reservas de audiovisuales realizadas. Tiene toda la lógica de la clase ReservaAudiovisualVO a nivel de modelo		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_reservaAudiovisual		ArrayList (colección de objetos) que almacena los resultados de las consultas
String v_str_mensaje		Cadena que almacena el mensaje que retorna una determinada operación
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaAudiovisualVO	retorna	Asociación
FranjaVO	utiliza	Asociación
UsuarioVO	utiliza	Asociación
ProgramaVO	utiliza	Asociación
SemestreVO	utiliza	Asociación
ServicioAudiovisualVO	utiliza	Asociación
ReservaAudiovisualVisitor	utiliza	Asociación
FranjaMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
ProgramaMgr	utiliza	Asociación
SemestreMgr	utiliza	Asociación
ServicioAudiovisualMgr	utiliza	Asociación
ReservaAudiovisualDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>		<b>Descripción</b>
findByPrimaryKey		Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la consulta de una Reserva de acuerdo a un código. Retorna un objeto de tipo ReservaAudiovisualVO
findByFechaFranja		Método que permite acceder a la consulta de las reservas de audiovisuales en una fecha y franja determinada
findByFechaServicio		Método que permite acceder a la consulta de las reservas de audiovisuales en una fecha y de un servicio audiovisual determinado
findByFechaIdentificacion		Método que permite acceder a la consulta de las reservas de audiovisuales realizadas por un usuario en una fecha determinada
findByFecha		Método que permite acceder a la consulta de las reservas de audiovisuales en una fecha determinada
findByTotalReserva		Método que permite acceder a la consulta del número de reservas de un usuario en una fecha y franja determinada

findByTotalReserva	Método que permite acceder a la consulta del número de reservas en una fecha y franja determinada
findByTotalReserva	Método que permite acceder a la consulta del número de reservas de un usuario en una determinada fecha
findByFechaComputador	Método que permite acceder a la consulta de todas las reservas de computadores que involucran un computador en una fecha determinada
findByTotalFranjaUsuario	Método que permite acceder a la consulta del número de reservas de un usuario en una determinada fecha y franja
findByTotalServicio	Método que permite acceder a la consulta del número de reservas en las cuales se encuentra involucrado un determinado Servicio Audiovisual
findByIdentificacion	Método que permite acceder a la consulta de las reservas de audiovisuales realizadas por un usuario determinado
createReserva	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la creación de una reserva de audiovisuales
updateObservacion	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la actualización de una reserva de audiovisuales
deleteReservaSancionAudiovisual	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la eliminación de todas las reservas de audiovisuales de un usuario en una fecha determinada
deleteReserva	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la eliminación de una reserva de audiovisuales
<b>Observaciones:</b>	

#### 11.1.1.12 Reserva Audiovisual Data Access Object

<b>Nombre de la Clase:</b> ReservaAudiovisualDAO	
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las reservas de audiovisuales	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
Connection conn	Recibe la conexión creada
ArrayList al_aux	Arreglo que almacena los resultados de las consultas

<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaComputadorVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ReservaAudiovisualVisitor con los datos de una determinada reserva audiovisual	
findByFechaFranja	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas en una determinada fecha y franja	
findByFechaServicio	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas en una determinada fecha y de un determinado servicio	
findByFechaIdentificacion	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas por un usuario en una determinada fecha	
findByFecha	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas en una determinada fecha	
findByTotalFechaFranjaUsuario	Método que permite retornar el número de reservas de un usuario en una determinada fecha y franja	
findByTotalFechaFranja	Método que permite retornar el número de reservas en una determinada fecha y franja	
findByTotalFechaUsuario	Método que permite retornar el número de reservas de un usuario en una determinada fecha	
findByTotalFranjaUsuario	Método que permite retornar el número de reservas de un usuario en una determinada fecha y franja	
findByTotalServicio	Método que permite retornar el número de reservas en las cuales está involucrado un servicio audiovisual	
findByIdentificacion	Método que retorna un ArrayList con todas las reservas audiovisuales realizadas por un determinado usuario	
createReserva	Método que permite crear una reserva de audiovisuales	
updateObservacion	Método que actualiza las observaciones de una determinada Reserva	
deleteReservaSancionAudiovisual	Método que elimina todas las reservas de Audiovisuales de un usuario en una fecha determinada	
deleteReserva	Método que elimina una Reserva Audiovisual	
<b>Observaciones:</b>		

### 11.1.1.13 Reserva Audiovisual Value Object

<b>Nombre de la Clase:</b> ReservaAudiovisualVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las reservas de audiovisuales realizadas por un usuario		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codReserva	Identifica de manera única una Reserva de Audiovisuales	
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva	
String str_observacion	Cadena que permite almacenar anotaciones de alguna eventualidad que se presente	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	registradaEn	Asociación
ServicioAudiovisual	tieneReservado	Asociación
UsuarioVO	pertenece	Asociación
ProgramaVO	pertenece	Asociación
SemestreVO	pertenece	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ReservaAudiovisualVO	Método constructor sin parámetros	
getCodigoReserva	Retorna el código de la reserva	
getFechaReserva	Retorna la fecha en que se hizo la reserva	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getServicioAudiovisualVO	Retorna un objeto de tipo ServicioAudiovisualVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getProgramaVO	Retorna un objeto de tipo ProgramaVO	
getSemestreVO	Retorna un objeto de tipo SemestreVO	
getObservacion	Retorna las observaciones hechas para la reserva	
setCodigoReserva	Permite asignar el código de la reserva	
setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
setServicioAudiovisualVO	Permite asignar o cambiar un objeto de tipo ServicioAudiovisualVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setProgramaVO	Permite asignar o cambiar un objeto de tipo ProgramaVO	
setSemestreVO	Permite asignar o cambiar un objeto de tipo SemestreVO	
setObservacion	Permite asignar o cambiar las observaciones de la reserva	

**Observaciones:**

#### 11.1.1.14 Reserva Audiovisual Visitor

<b>Nombre de la Clase:</b> ReservaAudiovisualVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de las reservas de audiovisuales realizadas por un usuario, hace referencia directa con cada uno de los campos de la tabla reservasaudiovisuales	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codReserva	Identifica de manera única una Reserva de Audiovisuales
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva
int int_codFranja	Identifica el código de la franja en la cual se hizo la reserva
int int_codServicio	Identifica el código del servicio audiovisual reservado
String str_ideUsuario	Determina la identificación del usuario que hizo la reserva
String str_codPrograma	Identifica el código del programa que hará uso de la reserva
String str_codSemestre	Identifica el código del semestre que hará uso de la reserva
String str_observacion	permite almacenar anotaciones de alguna eventualidad que se presente
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ReservaAudiovisualVisitor	Método constructor sin parámetros
getCodigoReserva	Retorna el código de la reserva
getFechaReserva	Retorna la fecha en que se hizo la reserva
getCodigoFranja	Retorna el código de la franja en que se hizo la reserva
getCodigoServicio	Retorna el código del servicio audiovisual reservado
getIdentificacionUsuario	Retorna la identificación del Usuario dueño de la reserva
getCodigoPrograma	Retorna el código del programa que hará uso la reserva
getCodigoSemestre	Retorna el código del semestre que hará uso la reserva
getObservacion	Retorna las observaciones hechas para la reserva
setCodigoReserva	Permite asignar el código de la reserva

setFechaReserva	Permite asignar o cambiar la fecha en que se hizo la reserva
setCodigoFranja	Permite asignar o cambiar el código de la franja
setCodigoServicio	Permite asignar o cambiar el código del servicio
setIdenticacionUsuario	Permite asignar o cambiar la identificación del usuario
setCodigoPrograma	Permite asignar o cambiar el código del programa que hará uso de la reserva
setCodigoSemestre	Permite asignar o cambiar el código del semestre que hará uso de la reserva
setObservacion	Permite asignar o cambiar las observaciones de la reserva
<b>Observaciones:</b>	

### 11.1.1.15 Reserva Audiovisual Manager

<b>Nombre de la Clase:</b> ReservaAudiovisualMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las reservas de audiovisuales realizadas. Tiene toda la lógica de la clase ReservaAudiovisualVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_reservaAudiovisual	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaAudiovisualVO	retorna	Asociación
FranjaVO	utiliza	Asociación
UsuarioVO	utiliza	Asociación
ProgramaVO	utiliza	Asociación
SemestreVO	utiliza	Asociación
ServicioAudiovisualVO	utiliza	Asociación
ReservaAudiovisualVisitor	utiliza	Asociación
FranjaMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
ProgramaMgr	utiliza	Asociación
SemestreMgr	utiliza	Asociación
ServicioAudiovisualMgr	utiliza	Asociación
ReservaAudiovisualDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	

findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la consulta de una Reserva de acuerdo a un código. Retorna un objeto de tipo ReservaAudiovisualVO
findByFechaFranja	Método que permite acceder a la consulta de las reservas de audiovisuales en una fecha y franja determinada
findByFechaServicio	Método que permite acceder a la consulta de las reservas de audiovisuales en una fecha y de un servicio audiovisual determinado
findByFechaIdentificacion	Método que permite acceder a la consulta de las reservas de audiovisuales realizadas por un usuario en una fecha determinada
findByFecha	Método que permite acceder a la consulta de las reservas de audiovisuales en una fecha determinada
findByTotalReserva	Método que permite acceder a la consulta del número de reservas de un usuario en una fecha y franja determinada
findByTotalReserva	Método que permite acceder a la consulta del número de reservas en una fecha y franja determinada
findByTotalReserva	Método que permite acceder a la consulta del número de reservas de un usuario en una determinada fecha
findByFechaComputador	Método que permite acceder a la consulta de todas las reservas de computadores que involucran un computador en una fecha determinada
findByTotalFranjaUsuario	Método que permite acceder a la consulta del número de reservas de un usuario en una determinada fecha y franja
findByTotalServicio	Método que permite acceder a la consulta del número de reservas en las cuales se encuentra involucrado un determinado Servicio Audiovisual
findByIdentificacion	Método que permite acceder a la consulta de las reservas de audiovisuales realizadas por un usuario determinado
createReserva	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la creación de una reserva de audiovisuales
updateObservacion	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la actualización de una reserva de audiovisuales
deleteReservaSancionAudiovisual	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la eliminación de todas las reservas de audiovisuales de un usuario en una fecha determinada

deleteReserva	Método que permite establecer los atributos de un objeto de tipo ReservaAudiovisualVisitor para la eliminación de una reserva de audiovisuales
<b>Observaciones:</b>	

### 11.1.1.16 Reserva Audiovisual Data Access Object

<b>Nombre de la Clase:</b> ReservaAudiovisualDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las reservas de audiovisuales		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaComputadorVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ReservaAudiovisualVisitor con los datos de una determinada reserva audiovisual	
findByFechaFranja	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas en una determinada fecha y franja	
findByFechaServicio	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas en una determinada fecha y de un determinado servicio	
findByFechaIdentificacion	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas por un usuario en una determinada fecha	
findByFecha	Método que retorna un ArrayList(colección de objetos) de tipo ReservaAudiovisualVisitor con todas las reservas audiovisuales realizadas en una determinada fecha	
findByTotalFechaFranjaUsuario	Método que permite retornar el número de reservas de un usuario en una determinada fecha y franja	
findByTotalFechaFranja	Método que permite retornar el número de reservas en una determinada fecha y franja	



findByTotalFechaUsuario	Método que permite retornar el número de reservas de un usuario en una determinada fecha
findByTotalFranjaUsuario	Método que permite retornar el número de reservas de un usuario en una determinada fecha y franja
findByTotalServicio	Método que permite retornar el número de reservas en las cuales está involucrado un servicio audiovisual
findByIdentificacion	Método que retorna un ArrayList con todas las reservas audiovisuales realizadas por un determinado usuario
createReserva	Método que permite crear una reserva de audiovisuales
updateObservacion	Método que actualiza las observaciones de una determinada Reserva
deleteReservaSancionAudiovisual	Método que elimina todas las reservas de Audiovisuales de un usuario en una fecha determinada
deleteReserva	Método que elimina una Reserva Audiovisual
<b>Observaciones:</b>	

#### 11.1.1.17 Elemento Audiovisual Reservado Value Object

<b>Nombre de la Clase:</b> ElementoAudiovisualReservadoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los elementos de inventario que han sido asignados a una Reserva Audiovisual		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
String str_codElemento	Cadena que almacena el código del elemento de inventario o el computador audiovisual asignado a la reserva	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaAudiovisualVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ReservaAudiovisualVO	Método constructor sin parámetros	
getReservaAudiovisualVO	Retorna un objeto de tipo ReservaAudiovisualVO	
getCodigoElemento	Retorna el código del elemento asignado a la reserva	
setReservaAudiovisualVO	Permite asignar o cambiar un objeto de tipo ReservaAudiovisualVO	
setCodigoElemento	Permite asignar el código del elemento de la reserva	
<b>Observaciones:</b> El código del elemento permitirá almacenar un código de un elemento de inventario o un código de un computador destinado para prestar el servicio de audiovisuales.		

### 11.1.1.18 Elemento Audiovisual Reservado Visitor

<b>Nombre de la Clase:</b> ElementoAudiovisualReservadoVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los elementos que han sido asignados a una Reserva Audiovisual, hace referencia directa con cada uno de los campos de la tabla elementosaudiovisualesreservados	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codReserva	Identifica de manera única una Reserva de Audiovisuales
String str_codElemento	Cadena que indica el Código del elemento audiovisual involucrado en la Reserva
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ElementoAudiovisualReservadoVisitor	Método constructor sin parámetros
getCodigoReserva	Retorna el código de la reserva
getCodigoElemento	Retorna el Código del Elemento involucrado en la Reserva
setCodigoReserva	Permite asignar el código de la Reserva
setCodigoElemento	Permite asignar el código del elemento asignado a la reserva
<b>Observaciones:</b>	

### 11.1.1.19 Elemento Audiovisual Reservado Manager

<b>Nombre de la Clase:</b> ElementoAudiovisualReservadoMgr		
<b>Descripción de la clase</b> Clase que permite la administración de los elementos asignados a las reservas audiovisuales. Tiene toda la lógica de la clase ElementoAudiovisualReservadoVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_elementoReserva	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoAudiovisualReservadoVO	retorna	Asociación
ReservaAudiovisualVO	utiliza	Asociación
ElementoAudiovisualReservadoVisitor	utiliza	Asociación
ReservaAudiovisualMgr	utiliza	Asociación

ElementoAudiovisualReservadoDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoReserva	Método que permite establecer los atributos de un objeto de tipo ElementoAudiovisualReservadoVisitor para acceder a la consulta de los elementos audiovisuales asignados a una reserva	
findByCodigoElemento	Método que permite acceder a la consulta de las reservas que involucran un determinado elemento	
createReservaElemento	Método que permite establecer los atributos de un objeto de tipo ElementoAudiovisualReservadoVisitor para la asignación de un elemento a una reserva audiovisual	
updateReservaElemento	Método que permite establecer los atributos de un objeto de tipo ElementoAudiovisualReservadoVisitor para la actualización de un elemento asignado a una reserva audiovisual	
deleteReserva	Método que elimina todas las reservas de Elementos Audiovisuales de acuerdo a un Código de Reserva	
<b>Observaciones:</b>		

#### 11.1.1.20 Elemento Audiovisual Reservado Data Access Object

<b>Nombre de la Clase:</b> ElementoAudiovisualReservadoDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los elementos involucrados en una reserva audiovisual		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoAudiovisualReservadoVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	

findByCodigoReserva	Método que retorna un ArrayList de objetos de tipo ElementoAudiovisualReservadoVisitor con todos sus atributos de acuerdo a una determinada reserva
findByCodigoElemento	Método que retorna un ArrayList de objetos de tipo ElementoAudiovisualReservadoVisitor con todos sus atributos de acuerdo a un determinado elemento
createReservaElemento	Método que registra los elementos audiovisuales de una determinada reserva
updateReservaElemento	Método que actualiza una reserva de Elementos Audiovisuales de acuerdo un Código de Reserva
deleteReserva	Método que elimina todas las reservas de Elementos Audiovisuales de acuerdo un Código de Reserva
<b>Observaciones:</b>	

## 11.1.2 Administración Informática

### 11.1.2.1 Horario Monitor Informática Value Object

<b>Nombre de la Clase:</b> HorarioMonitorInformaticaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica el horario de monitoría informática		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codHorario	Identifica de manera única un objeto de tipo HorarioMonitorInformaticaVO	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	asignadoA	Asociación
FechaVO	asignadoA	Asociación
UsuarioVO	aCargo	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
HorarioMonitorInformaticaVO	Método constructor sin parámetros	
getCodigoHorario	Retorna el código del registro de horario	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getFechaVO	Retorna un objeto de tipo FechaVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
setCodigoHorario	Permite asignar el código del horario	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
setFechaVO	Permite asignar o cambiar un objeto de tipo FechaVO	

setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO
<b>Observaciones:</b> El código del elemento permitirá almacenar un código de un elemento de inventario o un código de un computador destinado para prestar el servicio de audiovisuales.	

### 11.1.2.2 Horario Monitor Informática Visitor

<b>Nombre de la Clase:</b> HorarioMonitorInformaticaVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica el horario de monitoría informática, hace referencia directa con cada uno de los campos de la tabla horariomonitoresinformatica	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codHorario	Identifica de manera única un registro del horario de monitoría informática
int int_codFranja	Identifica el código de la franja asignada al horario
int int_codDia	Identifica el código de l día de la semana
String str_ideUsuario	Identifica la identificación del usuario a cargo de una franja del horario
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
HorarioMonitorInformaticaVisitor	Método constructor sin parámetros
getCodigoHorario	Retorna el código del horario
getCodigoFranja	Retorna el Código de la franja asignada al horario
getCodigoDia	Retorna el código del día de la semana
getIdentificacionUsuario	Retorna la identificación del usuario
setCodigoHorario	Permite asignar el código del horario
setCodigoFranja	Permite asignar el código de la franja
setCodigoDia	Permite asignar el código del día de la semana
setIdentificacionUsuario	Permite asignar la identificación del usuario
<b>Observaciones:</b>	

### 11.1.2.3 Horario Monitor Informática Manager

<b>Nombre de la Clase:</b> HorarioMonitorInformaticaMgr	
<b>Descripción de la clase</b> Clase que permite la administración de las franjas del horario de monitoría informática. Tiene toda la lógica de la clase HorarioMonitorInformaticaVO a nivel de modelo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

ArrayList al_horario	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioMonitorInformaticaVO	retorna	Asociación
FranjaVO	utiliza	Asociación
FechaVO	utiliza	Asociación
UsuarioVO	utiliza	Asociación
EstadoAulaInformaticaVO	utiliza	Asociación
HorarioMonitorInformaticaVisitor	utiliza	Asociación
FranjaMgr	utiliza	Asociación
FechaMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
EstadoAulaInformaticaMgr	utiliza	Asociación
HorarioMonitorInformaticaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo HorarioMonitorInformaticaVisitor para la consulta del horario de acuerdo a un código	
findByFranjaDia	Método que permite establecer los atributos de un objeto de tipo HorarioMonitorInformaticaVisitor para la consulta del horario de acuerdo a una franja y día determinado	
findByFranja	Método que permite acceder a la consulta del número de coincidencias de una franja en el horario	
findByDia	Método que permite acceder a la consulta de los datos del horario en un día determinado	
findByUsuario	Método que permite acceder a la consulta de los datos del horario de un determinado monitor	
createFranjaHorario	Método que permite establecer los atributos de un objeto de tipo HorarioMonitorInformaticaVisitor para el registro de una franja del horario	
updateFranjaHorario	Método que permite establecer los atributos de un objeto de tipo HorarioMonitorInformaticaVisitor para la actualización de una franja del horario	
deleteHorario	Método que permite la eliminación de los datos del horario	
<b>Observaciones:</b>		

#### 11.1.2.4 Horario Monitor Informática Data Access Object

<b>Nombre de la Clase:</b> HorarioMonitorInformaticaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto al horario de monitoría informática		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
Connection conn		Recibe la conexión creada
ArrayList al_aux		Arreglo que almacena los resultados de las consultas
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioMonitorInformaticaVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>		<b>descripción</b>
findByPrimaryKey		Método que permite retornar los datos del horario en un determinado día y franja de acuerdo al código de horario
findByFranjaDia		Método que permite retornar los datos del horario en una franja y día determinado
findByFranja		Método que permite retornar el número de coincidencias de una franja
findByDia		Método que permite retornar un ArrayList (Colección de objetos) de tipo HorarioMonitorInformaticaVisitor con todos los datos del horario en un determinado día
findByUsuario		Método que permite retornar un ArrayList(Colección de objetos) de tipo HorarioMonitorInformaticaVisitor con los datos del horario de acuerdo a la identificación del monitor
createFranjaHorario		Método que permite registrar los datos del horario de monitoría informática en un día y franja determinada
updateFranjaHorario		Método que permite actualizar los datos del horario de monitoría informática en un día y franja determinada
deleteHorario		Método que permite inicializar el horario de monitoría informática borrando sus datos
<b>Observaciones:</b>		

#### 11.1.2.5 Horario Aula Informática Value Object

<b>Nombre de la Clase:</b> HorarioAulaInformativaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica un horario del aula de Informática		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codHorario		Identifica de manera única un Horario de Aulas de Informática
String str_usoInternet		Cadena que permite identificar si el aula es utilizada o no para el servicio de internet
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	asignadoA	Asociación
FechaVO	asignadoA	Asociación
ProgramaVO	asignadoA	Asociación
UsuarioVO	aCargo	Asociación
SemestreVO	asignadoA	Asociación
<b>Servicio o Responsabilidad</b>		<b>Descripción</b>
HorarioAulaInformativaVO		Método constructor sin parámetros
getCodigoHorario		Retorna el código del horario
getFranjaVO		Retorna un objeto de tipo FranjaVO
getFechaVO		Retorna un objeto de tipo FechaVO
getUbicacionVO		Retorna un objeto de tipo UbicacionVO
getProgramaVO		Retorna un objeto de tipo ProgramaVO
getUsuarioVO		Retorna un objeto de tipo UsuarioVO
getUsoInternet		Retorna el valor del uso de Internet en el horario del aula
getSemestreVO		Retorna un objeto de tipo SemestreVO
setCodigoHorario		Permite asignar el código del horario
setFranjaVO		Permite asignar o cambiar un objeto de tipo FranjaVO
setFechaVO		Permite asignar o cambiar un objeto de tipo FechaVO
setUbicacionVO		Permite asignar o cambiar un objeto de tipo UbicacionVO
setProgramaVO		Permite asignar o cambiar un objeto de tipo ProgramaVO
setUsuarioVO		Permite asignar o cambiar un objeto de tipo UsuarioVO
setUsoInternet		Permite asignar o cambiar el uso del servicio de internet en la franja del horario
setSemestreVO		Permite asignar o cambiar un objeto de tipo SemestreVO
<b>Observaciones:</b>		

### 11.1.2.6 Horario Aula Informática Visitor



<b>Nombre de la Clase:</b> HorarioAulaInformaticaVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica un horario del aula de Informática, hace referencia directa con cada uno de los campos de la tabla horariosaulasinformatica	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codHorario	Identifica de manera única un registro del horario
int int_codFranja	Identifica el código de la franja del horario
int int_codDia	Identifica el código del día del horario
int int_codAula	Identifica el código del aula de informática para la que se crea el horario
String str_codPrograma	Identifica el código del programa que hace uso de la franja del horario
String str_ideUsuario	Determina la identificación del usuario a cargo del aula en una franja del horario
String str_usoInternet	Identifica si el aula utiliza el servicio de internet en una franja del horario
int int_codSemestre	Identifica el código del semestre que hace uso de la franja del horario
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
HorarioAulaInformaticaVisitor	Método constructor sin parámetros
getCodigoHorario	Retorna el código del horario
getCodigoFranja	Retorna el código de la franja del horario
getCodigoDia	Retorna el código del día del horario
getCodigoAula	Retorna el código del aula del horario
getCodigoPrograma	Retorna el código del programa
getIdentificacionUsuario	Retorna la identificación del usuario
getUsoInternet	Retorna el uso de internet en el aula
getCodigoSemestre	Retorna el código del semestre
setCodigoHorario	Permite asignar el código del registro de horario
setCodigoFranja	Permite asignar o cambiar el código de la franja
setCodigoDia	Permite asignar o cambiar el código del día
setCodigoAula	Permite asignar o cambiar el código del aula
setCodigoPrograma	Permite asignar o cambiar el código del programa
setIdentificacionUsuario	Permite asignar o cambiar la identificación del usuario
setUsoInternet	Permite asignar o cambiar el uso de internet en el aula
setCodigoSemestre	Permite asignar o cambiar el código de semestre
<b>Observaciones:</b>	

### 11.1.2.7 Horario Aula Informática Manager

<b>Nombre de la Clase:</b> HorarioAulaInformaticaMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las reservas de internet realizadas. Tiene toda la lógica de la clase ReservainternetVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_horario	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioAulaInformaticaVO	retorna	Asociación
FranjaVO	utiliza	Asociación
FechaVO	utiliza	Asociación
UbicacionVO	utiliza	Asociación
UsuarioVO	utiliza	Asociación
SemestreVO	utiliza	Asociación
ProgramaVO	utiliza	Asociación
EstadoAulaInformaticaVO	utiliza	Asociación
HorarioAulaInformaticaVisitor	utiliza	Asociación
FranjaMgr	utiliza	Asociación
FechaMgr	utiliza	Asociación
UbicacionMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
SemestreMgr	utiliza	Asociación
ProgramaMgr	utiliza	Asociación
EstadoAulaInformaticaMgr	utiliza	Asociación
HorarioAulaInformaticaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo HorarioAulaInformaticaVisitor para la consulta del horario de acuerdo a un código	
findByFranjaDiaAula	Método que permite establecer los atributos de un objeto de tipo HorarioAulaInformaticaVisitor para la consulta del horario en un determinado día, franja y aula	
findByFranjaDia	Método que permite establecer los atributos de un objeto de tipo HorarioAulaInformaticaVisitor para la consulta del horario en una determinada franja y día	
findByDiaAula	Método que permite acceder a la consulta de los datos del horario en un día y aula determinada	
findByUsuario	Método que permite acceder a la consulta de los datos del horario de acuerdo al usuario responsable del aula	
findByFranja	Método que permite acceder a la consulta del número de coincidencias de una franja en el horario	

createFranjaHorario	Método que permite establecer los atributos de un objeto de tipo HorarioAulaInformaticaVisitor para el registro de una franja del horario
updateFranjaHorario	Método que permite establecer los atributos de un objeto de tipo HorarioAulaInformaticaVisitor para la actualización de una franja del horario
deleteHorarioAula	Método que permite establecer los atributos de un objeto de tipo HorarioAulaInformaticaVisitor para la eliminación de los datos de horario de una determinada aula
<b>Observaciones:</b>	

### 11.1.2.8 Horario Aula Informática Data Access Object

<b>Nombre de la Clase:</b> HorarioAulaInformaticaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las franjas de los horarios de aulas de informática		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
Connection conn		Recibe la conexión creada
ArrayList al_aux		Arreglo que almacena los resultados de las consultas
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioAulaInformaticaVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que permite retornar los datos del horario en un determinado día y franja de acuerdo al código de horario	
findByFranjaDiaAula	Método que permite retornar los datos del horario en un determinado día, franja y aula	
findByFranjaDia	Método que permite retornar los datos del horario en un determinado día y franja	
findByDiaAula	Método que permite retornar los datos del horario en un determinado día	
findByFranja	Método que permite retornar el número de coincidencias de una franja	
findByUsuario	Método que permite retornar los datos del horario a cargo de un determinado usuario	
createFranjaHorario	Método que permite registrar los datos del horario de aulas de informática en un día y franja determinada	

updateFranjaHorario	Método que permite actualizar los datos del horario de aulas de informática en un día y franja determinada
deleteHorarioAula	Método que permite inicializar el horario de aulas de informática borrando sus datos
<b>Observaciones:</b>	

### 11.1.2.9 Estado Aula Informática Value Object

<b>Nombre de la Clase:</b> EstadoAulaInformaticaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los estados que pueden tener un aula de informática		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
String str_codEstado	Identifica de manera única un Estado de Aulas de Informática	
tring str_desEstado	Cadena que permite dar una descripción del estado de aulas de informática	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
EstadoAulaInformaticaVO	Método constructor sin parámetros	
getCodigo	Retorna el código del estado	
getDescripcion	Retorna la descripción del estado	
setCodigo	Permite asignar el código del estado	
setDescripcion	Permite asignar la descripción del estado	
<b>Observaciones:</b> El estado permitirá determinar si un aula esta disponible con internet, disponible sin internet o no se presta servicio en ella.		

### 11.1.2.10 Estado Aula Informática Manager

<b>Nombre de la Clase:</b> EstadoAulaInformaticaMgr	
<b>Descripción de la clase</b> Clase que permite acceder a los estados de las aulas de informática existentes. Tiene toda la lógica de la clase EstadoAulaInformaticaVO a nivel de modelo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EstadoAulaInformativaVO	retorna	Asociación
EstadoAulaInformativaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo EstadoAulaInformativaVO para la consulta de un estado de acuerdo a un código	
findByAll	Método que permite acceder a la consulta de todos los estados existentes	
<b>Observaciones:</b>		

#### 11.1.2.11 Estado Aula Informática Data Access Object

<b>Nombre de la Clase:</b> EstadoAulaInformativaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los estados de las aulas de informática		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo EstadoAulaInformativaVO con todos sus atributos	
findByAll	Método que retorna un ArrayList de objetos de tipo EstadoAulaInformativaVO con los datos de los posibles estados de las aulas de informática	
<b>Observaciones:</b>		

#### 11.1.2.12 Reserva Aula Informática Value Object

<b>Nombre de la Clase:</b> ReservaAulaInformativaVO
--

<b>Descripción de la clase</b>		
Clase que representa de forma lógica las reservas de aulas de informática realizadas por un usuario		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codReserva	Identifica de manera única una Reserva de Aulas de Informática	
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva	
String str_usoInternet	Cadena que permite identificar si la reserva utilizará el servicio de internet	
String str_observaciones	Cadena que permite hacer anotaciones sobre alguna eventualidad o requerimiento de la reserva	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UbicacionVO	asignadaA	Asociación
FranjaVO	registradaEn	Asociación
UsuarioVO	aCargo	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ReservaAulaInformaticaVO	Método constructor sin parámetros	
getCodigoReserva	Retorna el código de la reserva	
getFechaReserva	Retorna la fecha en que se hizo la reserva	
getUbicacionVO	Retorna un objeto de tipo UbicacionVO	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getUsolInternet	Retorna el uso del servicio de internet en la reserva	
getObservaciones	Retorna las observaciones realizadas a la reserva	
setCodigoReserva	Permite asignar el código de la reserva	
setFechaReserva	Permite asignar la fecha de la reserva	
setUbicacionVO	Permite asignar o cambiar un objeto de tipo UbicacionVO	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setUsolInternet	Permite asignar el uso de internet en la reserva	
setObservaciones	Permite asignar o cambiar las observaciones de la reserva	
<b>Observaciones:</b>		

### 11.1.2.13 Reserva Aula Informática Visitor

<b>Nombre de la Clase:</b> ReservaAulaInformaticaVisitor
---

<b>Descripción de la clase</b>	
Clase que representa de forma lógica los datos de las reservas de aulas de informática realizadas por un usuario, hace referencia directa con cada uno de los campos de la tabla reservasaulasinformatica	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codReserva	Identifica de manera única una Reserva de Aulas de Informática
String str_fecReserva	Cadena que almacena la fecha en la cual se realiza una reserva
int int_codAula	Identifica el código del aula de informática reservada
int int_codFranja	Identifica el código de la franja en la cual se hizo la reserva
String str_usoInternet	Determina el uso del servicio de internet en la reserva
String str_observaciones	permite almacenar anotaciones de alguna eventualidad que se presente
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ReservaAulaInformaticaVisitor	Método constructor sin parámetros
getCodigoReserva	Retorna el código de la reserva
getFechaReserva	Retorna la fecha en que se hizo la reserva
getCodigoAula	Retorna el código del aula
getCodigoFranja	Retorna el código de la franja en que se hizo la reserva
getIdentificacionUsuario	Retorna la identificación del Usuario dueño de la reserva
getUsolInternet	Retorna el uso del servicio de internet en la reserva
getObservaciones	Retorna las observaciones hechas para la reserva
setCodigoReserva	Permite asignar el código de la reserva
setFechaReserva	Permite asignar la fecha en que se hizo la reserva
setCodigoAula	Permite asignar el código del aula
setCodigoFranja	Permite asignar el código de la franja
setIdentificacionUsuario	Permite asignar la identificación del usuario
setUsolInternet	Permite asignar el uso de internet en la reserva
setObservaciones	Permite asignar las observaciones de la reserva
<b>Observaciones:</b>	

#### 11.1.2.14 Reserva Aula Informática Manager

<b>Nombre de la Clase:</b> ReservaAulaInformaticaMgr
---

<b>Descripción de la clase</b>		
Clase que permite la administración de las reservas de aulas de informática realizadas. Tiene toda la lógica de la clase ReservaAulaInformaticaVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_reserva	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaAulaInformaticaVO	retorna	Asociación
FranjaVO	utiliza	Asociación
UsuarioVO	utiliza	Asociación
UbicacionVO	utiliza	Asociación
ReservaAulaInformaticaVisitor	utiliza	Asociación
FranjaMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
UbicacionMgr	utiliza	Asociación
ReservaAulaInformaticaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByFechaAulaFranja	Método que permite establecer los atributos de un objeto de tipo ReservaAulaInformaticaVisitor para la consulta de una reserva de acuerdo a una fecha, aula y franja determinada	
findByFecha	Método que permite acceder a la consulta de las reservas de aulas de informática en una fecha determinada	
findByUsuario	Método que permite acceder a la consulta de las reservas de aulas de informática de acuerdo a la identificación del usuario	
createReserva	Método que permite establecer los atributos de un objeto de tipo ReservaAulaInformaticaVisitor para el registro de una reserva de aulas de informática	
deleteReserva	Método que permite establecer los atributos de un objeto de tipo ReservaAulaInformaticaVisitor para la eliminación de una reserva de aulas de informática	
<b>Observaciones:</b>		

#### 11.1.2.15 Reserva Aula Informática Data Access Object

<b>Nombre de la Clase:</b> ReservaAulaInformaticaDAO
---



<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las reservas de aulas de informática		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ReservaInternetVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByFechaAulaFranja	Retorna un objeto de tipo ReservaAulaInformaticaVisitor de acuerdo a una fecha, franja y aula determinada	
findByFecha	Retorna un ArrayList(colección de objetos) de tipo ReservaAulaInformaticaVisitor con todas las reservas realizadas en una fecha determinada	
findByUsuario	Método que permite retornar lo datos de las reservas de aulas de informática realizadas por un determinado usuario	
createReserva	Método que permite crear una reserva de aulas de informática	
deleteReserva	Método que elimina una Reserva de Aulas de Informática	
<b>Observaciones:</b>		

### 11.1.2.16 Servicio Audiovisual Value Object

<b>Nombre de la Clase:</b> ServicioAudiovisualVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los Servicios Audiovisuales establecidos en la Universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codigo	Identifica de manera única un Servicio Audiovisual	
String str_descripcion	Cadena que permite hacer una descripción del servicio audiovisual	
int int_estado	Indica si el servicio se encuentra activo o inactivo	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ServicioAudiovisualVO	Método constructor sin parámetros	

getCodigo	Retorna el código del servicio
getDescripcion	Retorna la descripción del servicio
getEstado	Retorna el estado del servicio
setCodigo	Permite asignar el código del servicio
setDescripcion	Permite asignar o cambiar la descripción del servicio
setEstado	Permite asignar o cambiar el estado del servicio
<b>Observaciones:</b>	

### 11.1.2.17 Servicio Audiovisual Manager

<b>Nombre de la Clase:</b> ServicioAudiovisualMgr		
<b>Descripción de la clase</b> Clase que permite administrar los Servicios Audiovisuales existentes. Tiene toda la lógica de la clase ServicioAudiovisualVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioAudiovisualVO	retorna	Asociación
ServicioAudiovisualDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ServicioAudiovisualVO para la consulta de un servicio de acuerdo a un código	
findByAll	Método que permite acceder a la consulta de todos los Servicios Audiovisuales en estado activo	
findByAllTotal	Método que permite acceder a la consulta de todos los Servicios Audiovisuales en estado activo o inactivo	
createServicio	Método que permite establecer los atributos de un objeto de tipo ServicioAudiovisualVO para la creación de un Servicio Audiovisual determinado	
updateServicio	Método que permite establecer los atributos de un objeto de tipo ServicioAudiovisualVO para la actualización de un Servicio Audiovisual determinado	
deleteServicio	Método que permite establecer los atributos de un objeto de tipo ServicioAudiovisualVO para la eliminación de un Servicio Audiovisual determinado	

**Observaciones:**

### 11.1.2.18 Servicio Audiovisual Data Access Object

<b>Nombre de la Clase:</b> ServicioAudiovisualDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los Servicios Audiovisuales		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioAudiovisualVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ServicioAudiovisualVO con los datos del Servicio de acuerdo a un código determinado	
findByAll	Método que retorna un ArrayList(colección de objetos) de tipo ServicioAudiovisualVO con todos los servicios audiovisuales ofrecidos en la Universidad, que se encuentren en estado Activo	
findByAllTotal	Método que retorna un ArrayList(colección de objetos) de tipo ServicioAudiovisualVO con todos los servicios audiovisuales ofrecidos en la Universidad en estado Activo o Inactivo	
createServicio	Método que crea un nuevo Servicio Audiovisual	
updateServicio	Método que actualiza un Servicio Audiovisual	
deleteServicio	Método que elimina un Servicio Audiovisual	
<b>Observaciones:</b>		

### 11.1.2.19 Elemento Servicio Audiovisual Value Object

<b>Nombre de la Clase:</b> ElementoServicioAudiovisualVO
<b>Descripción de la clase</b> Clase que representa de forma lógica los tipos de elementos asociados a un Servicio Audiovisual

Propiedad		Descripción del modelaje, restricciones, información para diseño, suposiciones
<b>Clase Asociada</b>		<b>Nombre Asociación</b>
ServicioAudiovisualVO	registradaEn	Asociación
TipoElementoVO	pertenece	Asociación
<b>Servicio o Responsabilidad</b>		<b>Descripción</b>
ElementoServicioAudiovisualVO		Método constructor sin parámetros
getServicioAudiovisualVO		Retorna un objeto de tipo ServicioAudiovisualVO
getTipoElementoVO		Retorna un objeto de tipo TipoElementoVO
setServicioAudiovisualVO		Permite asignar o cambiar un objeto de tipo ServicioAudiovisualVO
setTipoElementoVO		Permite asignar o cambiar un objeto de tipo TipoElementoVO
<b>Observaciones:</b>		

#### 11.1.2.20 Elemento Servicio Audiovisual Visitor

<b>Nombre de la Clase:</b> ElementoServicioAudiovisualVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los elementos que forman parte de un Servicio Audiovisual, hace referencia directa con cada uno de los campos de la tabla elementosserviciosaudiovisuales	
Propiedad	Descripción del modelaje, restricciones, información para diseño, suposiciones
int int_codServicio	Identifica el código del servicio audiovisual
int int_codTipo	Identifica el código del tipo de elemento asignado al servicio
Servicio o Responsabilidad	Descripción
ElementoServicioAudiovisualVisitor	Método constructor sin parámetros
getCodigoServicio	Retorna el código del servicio audiovisual
getCodigoTipo	Retorna el código del tipo de elemento
setCodigoServicio	Permite asignar el código del servicio audiovisual
setCodigoTipo	Permite asignar o cambiar el código del tipo de elemento
<b>Observaciones:</b>	

#### 11.1.2.21 Elemento Servicio Audiovisual Manager

<b>Nombre de la Clase:</b> ElementoServicioAudiovisualMgr
--

<b>Descripción de la clase</b>		
Clase que permite la administración de los tipos de elementos asignados a un servicio audiovisual. Tiene toda la lógica de la clase ElementoServicioAudiovisualVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_elementoServicio	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoServicioAudiovisualVO	retorna	Asociación
ServicioAudiovisualVO	utiliza	Asociación
TipoElementoVO	utiliza	Asociación
ElementoServicioAudiovisualVisitor	utiliza	Asociación
ServicioAudiovisualMgr	utiliza	Asociación
TipoElementoMgr	utiliza	Asociación
ElementoServicioAudiovisualDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoServicio	Método que permite acceder a la consulta de todos los tipos de elementos asignados a un Servicio Audiovisual de acuerdo a un código determinado	
findByCodigoTipo	Método que permite acceder a la consulta de todos los Servicios Audiovisuales que tengan asignado un tipo de elemento determinado	
findByServicioTipo	Método que permite establecer los atributos de un objeto de tipo elementoServicioAudiovisualVisitor para la consulta de acuerdo a un servicio y tipo determinado	
createServicioTipo	Método que permite establecer los atributos de un objeto de tipo elementoServicioAudiovisualVisitor para la asignación de un tipo de elemento a un servicio audiovisual determinado	
deleteServicioTipo	Método que permite establecer los atributos de un objeto de tipo elementoServicioAudiovisualVisitor para desvincular un tipo de elemento de un servicio audiovisual determinado	
deleteServicio	Método que permite establecer los atributos de un objeto de tipo elementoServicioAudiovisualVisitor para eliminar las asignaciones de tipos de elementos de un servicio audiovisual determinado	
<b>Observaciones:</b>		

### 11.1.2.22 Elemento Servicio Audiovisual Data Access Object

<b>Nombre de la Clase:</b> ElementoServicioAudiovisualDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los tipos de elementos involucrados en un Servicio Audiovisual		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
Connection conn		Recibe la conexión creada
ArrayList al_aux		Arreglo que almacena los resultados de las consultas
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoServicioAudiovisualVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>		<b>descripción</b>
findByCodigoServicio		Retorna un ArrayList(colección de objetos) de tipo ElementoServicioAudiovisualVisitor con todos sus atributos de acuerdo a un determinado Código de Servicio
findByCodigoTipo		Retorna un ArrayList(colección de objetos) de tipo ElementoServicioAudiovisualVisitor con todos sus atributos de acuerdo a un determinado Código de Tipo de Elemento
findByServicioTipo		Método que retorna un objeto de tipo ElementoServicioAudiovisualVisitor con todos sus atributos de acuerdo a un código de Servicio y un código de Tipo de Elemento
createServicioTipo		Método que asigna un Tipo de Elemento a un Servicio Audiovisual
deleteServicioTipo		Método que elimina un Tipo de Elemento a un Servicio Audiovisual
deleteServicio		Método que elimina las asignaciones de tipos de elementos a un determinado Servicio Audiovisual
<b>Observaciones:</b>		

### 11.1.3 Información

#### 11.1.3.1 Novedad Value Object

<b>Nombre de la Clase:</b> NovedadVO
---

<b>Descripción de la clase</b>		
Clase que representa de forma lógica las novedades existentes para darlas a conocer a los usuarios		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codNovedad	Identifica de manera única una novedad	
int int_estNovedad	Indica el estado de la novedad, si esta activa o inactiva	
String str_titNovedad	Cadena que identifica el título de la novedad	
String str_conNovedad	Cadena que identifica el contenido de la novedad	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
NovedadVO	Método constructor sin parámetros	
getCodigo	Retorna el código de la novedad	
getTitulo	Retorna el título de la novedad	
getContenido	Retorna el contenido de la novedad	
getEstado	Retorna el estado de la novedad	
setCodigo	Permite asignar el código de la novedad	
setTitulo	Permite asignar o cambiar el título de la novedad	
setContenido	Permite asignar o cambiar el contenido de la novedad	
setEstado	Permite asignar o cambiar el estado de la novedad	
<b>Observaciones:</b>		

### 11.1.3.2 Novedad Manager

<b>Nombre de la Clase:</b>		
NovedadMgr		
<b>Descripción de la clase</b>		
Clase que permite la administración de las novedades existentes. Tiene toda la lógica de la clase NovedadVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
NovedadVO	retorna	Asociación
NovedadDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo NovedadVO para la consulta de una Novedad de acuerdo a un código	

findByAllTitulo	Método que permite acceder a la consulta de el código y el título de las Novedades existentes
findByAllTituloActivo	Método que permite acceder a la consulta de los títulos de las Novedades cuyo estado es Activo(1)
findByAll	Método que retorna un ArrayList(Colección de objetos) de tipo NovedadVO con los datos de todas las novedades existentes.
createNovedad	Método que permite establecer los atributos de un objeto de tipo NovedadVO para la creación de una novedad
updateNovedad	Método que permite establecer los atributos de un objeto de tipo NovedadVO para la actualización de una novedad
deleteNovedad	Método que permite establecer los atributos de un objeto de tipo NovedadVO para la eliminación de una novedad
<b>Observaciones:</b>	

### 11.1.3.3 Novedad Data Access Object

<b>Nombre de la Clase:</b> NovedadDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las novedades existentes		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
NovedadVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo NovedadVO con todos sus atributos de acuerdo al código de la novedad	
findByAllTitulo	Método que retorna un ArrayList(Colección de objetos) de tipo NovedadVO con el código y el título de las Novedades existentes	
findByAllTituloActivo	Método que retorna un ArrayList(Colección de objetos) de tipo NovedadVO con los títulos de las Novedades cuyo estado es Activo(1)	



findByAll	Método que retorna un ArrayList(Colección de objetos) de tipo NovedadVO con los datos de todas las novedades existentes
createNovedad	Método que permite crear una Novedad
updateNovedad	Método que permite actualizar una determinada Novedad
deleteNovedad	Método que permite eliminar una novedad
<b>Observaciones:</b>	

#### 11.1.3.4 Servicio Recreativo Value Object

<b>Nombre de la Clase:</b> ServicioRecreativoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los Servicios Recreativos que ofrece la Universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codServicio	Identifica de manera única un servicio recreativo	
String str_titServicio	Cadena que identifica el título del servicio	
String str_desServicio	Cadena que identifica la descripción del servicio	
String str_imgServicio	Cadena que identifica el nombre de la imagen asociada al servicio	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ServicioRecreativoVO	Método constructor sin parámetros	
getCodigo	Retorna el código del servicio recreativo	
getTitulo	Retorna el título del servicio recreativo	
getDescripcion	Retorna la descripción del servicio recreativo	
getImagen	Retorna el nombre de la imagen asociada al servicio	
setCodigo	Permite asignar el código del servicio recreativo	
setTitulo	Permite asignar o cambiar el título del servicio recreativo	
setDescripcion	Permite asignar o cambiar la descripción del servicio recreativo	
setImagen	Permite asignar o cambiar la imagen asociada al servicio	
<b>Observaciones:</b>		

#### 11.1.3.5 Servicio Recreativo Manager

<b>Nombre de la Clase:</b> ServicioRecreativoMgr		
<b>Descripción de la clase</b> Clase que permite la administración de los Servicios Recreativos existentes. Tiene toda la lógica de la clase ServicioRecreativoVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioRecreativoVO	retorna	Asociación
ServicioRecreativoDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ServicioRecreativoVO para la consulta de un servicio recreativo de acuerdo a un código	
findByAll	Método que permite acceder a la consulta de todos los servicios recreativos existentes	
createServicioRecreativo	Método que permite establecer los atributos de un objeto de tipo ServicioRecreativoVO para la creación de un determinado servicio recreativo	
updateServicioRecreativo	Método que permite establecer los atributos de un objeto de tipo ServicioRecreativoVO para la actualización de un determinado servicio recreativo	
deleteServicioRecreativo	Método que permite establecer los atributos de un objeto de tipo ServicioRecreativoVO para la eliminación de un determinado servicio recreativo	
<b>Observaciones:</b>		

### 11.1.3.6 Servicio Recreativo Data Access Object

<b>Nombre de la Clase:</b> ServicioRecreativoDAO	
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los Servicios Recreativos ofrecidos en la Universidad	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
Connection conn	Recibe la conexión creada
ArrayList al_aux	Arreglo que almacena los resultados de las consultas

<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioRecreativoVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ServicioRecreativoVO con todos sus atributos	
findByAll	Método que retorna un ArrayList(Colección de objetos) de tipo ServicioRecreativoVO con todos los atributos de los Servicios Recreativos	
createServicioRecreativo	Método que permite crear un Servicio Recreativo	
updateServicioRecreativo	Método que permite actualizar un Servicio Recreativo	
createNovedad	Método que permite crear una Novedad	
deleteServicioRecreativo	Método que permite eliminar un Servicio Recreativo	
<b>Observaciones:</b>		

### 11.1.3.7 Opción Urna Value Object

<b>Nombre de la Clase:</b> OpcionUrnaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las preguntas publicadas en la urna virtual		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codOpcion	Identifica de manera única una opción de la urna virtual	
String str_desOpcion	Cadena que identifica la descripción de la opción	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
OpcionUrnaVO	Método constructor sin parámetros	
getCodigo	Retorna el código de la opción	
getDescripcion	Retorna la descripción de la opción	
setCodigo	Permite asignar el código de la opción	
setDescripcion	Permite asignar o cambiar la descripción de la opción	
<b>Observaciones:</b>		

### 11.1.3.8 Opción Urna Manager

<b>Nombre de la Clase:</b> OpcionUrnaMgr
---

<b>Descripción de la clase</b> Clase que permite la administración de las opciones de la urna virtual. Tiene toda la lógica de la clase OpcionUrnaVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
OpcionUrnaVO	retorna	Asociación
OpcionUrnaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo OpcionUrnaVO para la consulta de una opción de la urna virtual de acuerdo a un código	
findByDescripcion	Método que permite establecer los atributos de un objeto de tipo OpcionUrnaVO para la consulta de una opción de la urna virtual de acuerdo a una descripción	
findByAll	Método que permite acceder a la consulta de todas las opciones de la urna virtual	
createOpcion	Método que permite establecer los atributos de un objeto de tipo OpcionUrnaVO para la creación de una opción determinada	
updateOpcion	Método que permite establecer los atributos de un objeto de tipo OpcionUrnaVO para la actualización de una opción determinada	
deleteOpcion	Método que permite establecer los atributos de un objeto de tipo OpcionUrnaVO para la eliminación de una opción determinada	
<b>Observaciones:</b>		

### 11.1.3.9 Opción Urna Data Access Object

<b>Nombre de la Clase:</b> OpcionUrnaDAO	
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las opciones de respuesta para las preguntas publicadas en la Urna Virtual	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioRecreativoVO	Retorna	Asociación
DBCConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo OpcionUrnaVO con todos sus datos	
findByDescripcion	Método que retorna un objeto de tipo OpcionUrnaVO con la descripción de la opción	
findByAll	Método que retorna un ArrayList(Colección de objetos) de tipo OpcionUrnaVO con todas las opciones de las preguntas de la urna virtual	
createOpcion	Método que crea una opción para las preguntas de la urna virtual	
updateOpcion	Método que actualiza una opción para las preguntas de la urna virtual	
deleteOpcion	Método que elimina una opción de las preguntas de la urna virtual	
<b>Observaciones:</b>		

### 11.1.3.10 Pregunta Urna Value Object

<b>Nombre de la Clase:</b> PreguntaUrnaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las preguntas publicadas en la urna virtual		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codPregunta	Identifica de manera única una pregunta de la urna virtual	
String str_desPregunta	Cadena que identifica la descripción de la pregunta	
String str_estPregunta	Indica el estado de la pregunta, si esta activa o inactiva	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
PreguntaUrnaVO	Método constructor sin parámetros	
getCodigo	Retorna el código de la pregunta	
getDescripcion	Retorna la descripción de la pregunta	
getEstado	Retorna el estado de la pregunta	
setCodigo	Permite asignar el código de la pregunta	
setDescripcion	Permite asignar o cambiar la descripción de la pregunta	

setEstado	Permite asignar o cambiar el estado de la pregunta
<b>Observaciones:</b>	

### 11.1.3.11 Pregunta Urna Manager

<b>Nombre de la Clase:</b> PreguntaUrnaMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las preguntas de la urna virtual. Tiene toda la lógica de la clase PreguntaUrnaVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
PreguntaUrnaVO	retorna	Asociación
PreguntaUrnaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo PreguntaUrnaVO para la consulta de una pregunta de la urna virtual de acuerdo a un código	
findByEstado	Método que retorna un objeto de tipo PreguntaUrnaVO para la pregunta cuyo estado sea activa	
findByAll	Método que permite acceder a la consulta de todas las preguntas de la urna virtual	
createPregunta	Método que permite establecer los atributos de un objeto de tipo PreguntaUrnaVO para la creación de una pregunta determinada	
updatePregunta	Método que permite establecer los atributos de un objeto de tipo PreguntaUrnaVO para la actualización de una pregunta determinada	
deletePregunta	Método que permite establecer los atributos de un objeto de tipo PreguntaUrnaVO para la eliminación de una pregunta determinada	
<b>Observaciones:</b>		

### 11.1.3.12 Pregunta Urna Data Access Object

<b>Nombre de la Clase:</b> PreguntaUrnaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las preguntas publicadas en la Urna Virtual		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
PreguntaUrnaVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo PreguntaUrnaVO con la descripción de la pregunta activa de acuerdo a un Código determinado	
findByEstado	Método que retorna un objeto de tipo PreguntaUrnaVO para la pregunta cuyo estado sea activa	
findByAll	Método que retorna un ArrayList(Colección de objetos) de tipo PreguntaUrnaVO con todas las preguntas de la Urna Virtual	
createPregunta	Método que crea una pregunta para la urna virtual	
updatePregunta	Método que actualiza una pregunta para la urna virtual	
deletePregunta	Método que elimina una pregunta de la urna virtual	
<b>Observaciones:</b>		

### 11.1.3.13 Opinion Urna Value Object

<b>Nombre de la Clase:</b> OpinionUrnaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las opciones asignadas a cada pregunta y su respectivo total de votos		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_total	Identifica el total de votos de una determinada opción	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
PreguntaUrnaVO	conformadaPor	Asociación
OpcionUrnaVO	tieneAsignada	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
OpinionUrnaVO	Método constructor sin parámetros	

getPreguntaUrnaVO	Retorna un objeto de tipo PreguntaUrnaVO
getOpcionUrnaVO	Retorna un objeto de tipo OpcionUrnaVO
getTotal	Retorna el total de votos por una determinada opción
setPreguntaUrnaVO	Permite asignar un objeto de tipo PreguntaUrnaVO
setOpcionUrnaVO	Permite asignar un objeto de tipo OpcionUrnaVO
setTotal	Retorna el total de votos por una opción
<b>Observaciones:</b>	

#### 11.1.3.14 Opinion Urna Visitor

<b>Nombre de la Clase:</b> OpinionUrnaVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica las opciones asignadas a cada pregunta y su respectivo total de votos, hace referencia directa con cada uno de los campos de la tabla opinionurna	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codPregunta	Identifica el código de la pregunta de la urna virtual
int int_codOpcion	Identifica el código de la opción de la urna virtual asignada a la pregunta
int int_total	Identifica el total de votos de la opción
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
OpinionUrnaVisitor	Método constructor sin parámetros
getCodigoPregunta	Retorna el código de la pregunta
getCodigoOpcion	Retorna el código de la opción
getTotal	Retorna el total de votos de la opción
setCodigoPregunta	Permite asignar el código de la pregunta
setCodigoOpcion	Permite asignar el código de la opción
setTotal	Permite asignar el total de votos de la opción
<b>Observaciones:</b>	

#### 11.1.3.15 Opinion Urna Manager

<b>Nombre de la Clase:</b> OpinionUrnaMgr
<b>Descripción de la clase</b> Clase que permite la administración de la asignación de opciones de respuesta a las preguntas de la Urna Virtual. Tiene toda la lógica de la clase OpinionUrnaVO a nivel de modelo



<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_opinionUrna		ArrayList (colección de objetos) que almacena los resultados de las consultas
String v_str_mensaje		Cadena que almacena el mensaje que retorna una determinada operación
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
OpinionUrnaVO	retorna	Asociación
PreguntaUrnaVO	utiliza	Asociación
OpcionUrnaVO	utiliza	Asociación
OpinionUrnaVisitor	utiliza	Asociación
PreguntaUrnaMgr	utiliza	Asociación
OpcionUrnaMgr	utiliza	Asociación
OpinionUrnaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoPregunta	Método que retorna un ArrayList de objetos OpinionUrnaVO de acuerdo a un código de pregunta	
findByPreguntaOpcion	Método que permite establecer los atributos de un objeto de tipo OpinionUrnaVisitor para la consulta de una pregunta y opción determinada	
createPreguntaOpcion	Método que permite establecer los atributos de un objeto de tipo OpinionUrnaVisitor para la asignación de una opción a una pregunta determinada	
updateOpinionUrna	Método que permite establecer los atributos de un objeto de tipo OpinionUrnaVisitor para la actualización de las opciones asignadas a una pregunta determinada	
deletePregunta	Método que permite establecer los atributos de un objeto de tipo OpinionUrnaVisitor para la eliminación de una pregunta determinada y las opciones asignadas a ella	
deleteOpcion	Método que permite establecer los atributos de un objeto de tipo OpinionUrnaVisitor para la eliminación de todos los registros asociados a una determina opción	
deletePreguntaOpcion	Método que permite establecer los atributos de un objeto de tipo OpinionUrnaVisitor para la eliminación de una opción asignada a una pregunta	
<b>Observaciones:</b>		

### 11.1.3.16 Opinion Urna Data Access Object

<b>Nombre de la Clase:</b> OpinionUrnaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las opciones de respuesta y el total de votos para las preguntas publicadas en la Urna Virtual		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
OpinionUrnaVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByCodigoPregunta	Método que retorna un ArrayList de objetos de tipo OpinionUrnaVisitor de acuerdo a una pregunta determinada	
findByPreguntaOpcion	Método que retorna un objeto de tipo OpinionUrnaVisitor de acuerdo a un código de pregunta y de opción determinado	
createPreguntaOpcion	Método que permite crear un registro de acuerdo a un determinado código de pregunta y código de opción	
updateOpinionUrna	Método que permite actualizar el total de votos de acuerdo a un determinado código de pregunta y código de opción	
deletePregunta	Método que permite eliminar todos los registros que tengan asociada una determinada pregunta	
deleteOpcion	Método que permite eliminar todos los registros que tengan asociada una determinada opción	
deletePreguntaOpcion	Método que permite eliminar un registro de acuerdo a un determinado código de pregunta y código de opción	
<b>Observaciones:</b>		

## 11.1.4 Usuarios

### 11.1.4.1 Usuario Value Object

<b>Nombre de la Clase:</b> UsuarioVO
---

<b>Descripción de la clase</b>		
Clase que representa de forma lógica los datos personales de los usuarios		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
String str_ideUsuario	Identifica de manera única a un usuario	
String str_nom1Usuario	Cadena que almacena el primer nombre del usuario	
String str_nom2Usuario	Cadena que almacena el segundo nombre del usuario	
String str_ape1Usuario	Cadena que almacena el primer apellido del usuario	
String str_ape2Usuario	Cadena que almacena el segundo apellido del usuario	
String str_dirUsuario	Cadena que almacena la dirección del usuario	
String str_telUsuario	Cadena que almacena el teléfono del usuario	
String str_logUsuario	Cadena que almacena el login del usuario	
String str_pasUsuario	Cadena que almacena el password del usuario	
String str_sexUsuario	Cadena que almacena el sexo del usuario	
String str_estCivil	Cadena que almacena el estado civil del usuario	
int int_falInternet	Identifica el número de faltas del usuario para el servicio de internet	
int int_falComputadores	Identifica el número de faltas del usuario para el servicio de computadores	
int int_falAudiovisuales	Identifica el número de faltas del usuario para el servicio de audiovisuales	
String str_fecSancionInternet	Cadena que almacena la fecha en la cual se sanciona al usuario para el servicio de internet	
Str_fecSancionComputadores	Cadena que almacena la fecha en la cual se sanciona al usuario para el servicio de computadores	
String str_fecSancionAudiovisuales	Cadena que almacena la fecha en la cual se sanciona al usuario para el servicio de audiovisuales	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
LugarVO	pertenece	Asociación
TipoidentificacionVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
UsuarioVO	Método constructor con parámetros y sin parámetros	
getIdentificacion	Retorna la identificación del usuario	
getNombreUno	Retorna el primer nombre del usuario	
getNombreDos	Retorna el segundo nombre del usuario	
getApellidoUno	Retorna el primer apellido del usuario	
getApellidoDos	Retorna segundo apellido del usuario	
getDireccion	Retorna la dirección del usuario	

getTelefono	Retorna el teléfono del usuario
getLogin	Retorna el login del usuario
getPassword	Retorna el password del usuario
getLugarNacimientoVO	Retorna un objeto de tipo LugarVO
getLugarResidenciaVO	Retorna un objeto de tipo LugarVO
getTipoidentificacionVO	Retorna un objeto de tipo TipoidentificacionVO
getLugarIdentificacionVO	Retorna un objeto de tipo LugarVO
getSexo	Retorna el sexo del usuario
getEstadoCivil	Retorna el estado Civil del usuario
getFaltaInternet	Retorna el número del faltas del usuario para el servicio de Internet
getFaltaComputador	Retorna el número del faltas del usuario para el servicio de computadores
getFaltaAudiovisual	Retorna el número del faltas del usuario para el servicio de audiovisual
getFechaSancionInternet	Retorna la fecha en que el usuario quedó sancionado para el servicio de Internet
getFechaSancionComputador	Retorna la fecha en que el usuario quedó sancionado para el servicio de computadores
getFechaSancionAudiovisual	Retorna la fecha en que el usuario quedó sancionado para el servicio audiovisual
setIdenticacion	Permite asignar o cambiar la identificación del usuario
setNombreUno	Permite asignar o cambiar el primer nombre del usuario
setNombreDos	Permite asignar o cambiar el segundo nombre del usuario
setApellidoUno	Permite asignar o cambiar el primer apellido del usuario
setApellidoDos	Permite asignar o cambiar el segundo apellido del usuario
setDireccion	Permite asignar o cambiar la dirección del usuario
setTelefono	Permite asignar o cambiar el teléfono del usuario
setLogin	Permite asignar o cambiar el login del usuario
setPassword	Permite asignar o cambiar el password del usuario
setLugarNacimientoVO	Permite asignar o cambiar un objeto de tipo LugarVO
setLugarResidenciaVO	Permite asignar o cambiar un objeto de tipo LugarVO
setTipoidentificacionVO	Permite asignar o cambiar un objeto de tipo TipoidentificacionVO
setLugarIdentificacionVO	Permite asignar o cambiar un objeto de tipo LugarVO
setSexo	Permite asignar o cambiar el sexo del usuario

setEstadoCivil	Permite asignar o cambiar el estado civil del usuario
setFaltaInternet	Permite asignar o cambiar el número de faltas para el servicio de internet
setFaltaComputador	Permite asignar o cambiar el número de faltas para el servicio de computadores
setFaltaAudiovisual	Permite asignar o cambiar el número de faltas para el servicio audiovisual
setFechaSancionInternet	Permite asignar o cambiar la fecha de sanción del usuario para el servicio de internet
setFechaSancionComputador	Permite asignar o cambiar la fecha de sanción del usuario para el servicio de computadores
setFechaSancionAudiovisual	Permite asignar o cambiar la fecha de sanción del usuario para el servicio audiovisual
<b>Observaciones:</b>	

#### 11.1.4.2 Usuario Visitor

<b>Nombre de la Clase:</b> UsuarioVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del usuario. Hace referencia directa con cada uno de los campos de la tabla usuarios	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_ideUsuario	Identifica de manera única un usuario
String str_nom1Usuario	Cadena que almacena el primer nombre del usuario
String str_nom2Usuario	Identifica el segundo nombre del usuario
String str_ape1Usuario	Determina el primer apellido del usuario
String str_ape2Usuario	Determina el segundo apellido del usuario
String str_dirUsuario	permite almacenar la dirección del usuario
String str_telUsuario	Determina el teléfono del usuario
String str_logUsuario	Determina el login del usuario
String str_pasUsuario	Determina el password del usuario
int_lugNacimiento	Determina el código de la ciudad de nacimiento
int_lugResidencia	Determina el código de la ciudad de residencia
int_tiplIdentificacion	Determina el código del tipo de identificación del usuario
int_lugIdentificacion	Determina el código de la ciudad de expedición del documento de identidad
String str_sexUsuario	Determina el sexo del usuario
String str_estCivil	Identifica el estado civil del usuario

int int_falInternet	Identifica el número de faltas del usuario para el servicio de internet
int int_falComputadores	Identifica el número de faltas del usuario para el servicio de computadores
int int_falAudiovisuales	Identifica el número de faltas del usuario para el servicio audiovisual
String str_fecSancionInternet	Identifica la fecha de sanción del usuario para el servicio de internet
String str_fecSancionComputadores	Identifica la fecha de sanción del usuario para el servicio de computadores
String str_fecSancionAudiovisuales	Identifica la fecha de sanción del usuario para el servicio de audiovisuales
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
UsuarioVisitor	Método constructor con parámetros y sin parámetros
getIdentificacion	Retorna la identificación del usuario
getNombreUno	Retorna el primer nombre del usuario
getNombreDos	Retorna el segundo nombre del usuario
getApellidoUno	Retorna el primer apellido del usuario
getApellidoDos	Retorna el segundo apellido del usuario
getDireccion	Retorna la dirección del usuario
getTelefono	Retorna el teléfono del usuario
getLogin	Retorna el login del usuario
getPassword	Retorna el password del usuario
getLugarNacimiento	Retorna el código de la ciudad de nacimiento
getLugarResidencia	Retorna el código de la ciudad de residencia
getTipoidentificacion	Retorna el código del tipo de identificación
getLugarIdentificacion	Retorna el código de la ciudad de expedición del documento de identidad
getSexo	Retorna el sexo del usuario
getEstadoCivil	Retorna el estado civil del usuario
getFaltaInternet	Retorna el número de faltas del usuario para el servicio de internet
getFaltaComputador	Retorna el número de faltas del usuario para el servicio de computadores
getFaltaAudiovisual	Retorna el número de faltas del usuario para el servicio audiovisual
getFechaSancionInternet	Retorna la fecha de sanción del usuario para el servicio de internet
getFechaSancionComputador	Retorna la fecha de sanción del usuario para el servicio de computadores
getFechaSancionAudiovisual	Retorna la fecha de sanción del usuario para el servicio audiovisual
setIdentificacion	Permite asignar o cambiar la identificación del usuario
setNombreUno	Permite asignar o cambiar el primer nombre del usuario

setNombreDos	Permite asignar o cambiar el segundo nombre del usuario
setApellidoUno	Permite asignar o cambiar el primer apellido del usuario
setApellidoDos	Permite asignar o cambiar el segundo apellido del usuario
setDireccion	Permite asignar o cambiar la dirección del usuario
setTelefono	Permite asignar o cambiar el teléfono del usuario
setLogin	Permite asignar o cambiar el login del usuario
setPassword	Permite asignar o cambiar el password del usuario
setLugarNacimiento	Permite asignar o cambiar el código de la ciudad de nacimiento
setLugarResidencia	Permite asignar o cambiar el código de la ciudad de residencia del usuario
setTipoidentificacion	Permite asignar o cambiar el código de la ciudad de identificación del usuario
setLugarIdentificacion	Permite asignar o cambiar el código de la ciudad de expedición del documento de identidad
setSexo	Permite asignar o cambiar el sexo del usuario
setEstadoCivil	Permite asignar o cambiar el estado civil del usuario
setFaltaInternet	Permite asignar o cambiar el número de faltas del usuario para el servicio de internet
setFaltaComputador	Permite asignar o cambiar el número de faltas del usuario para el servicio de computadores
setFaltaAudiovisual	Permite asignar o cambiar el número de faltas del usuario para el servicio audiovisual
setFechaSancionInternet	Permite asignar o cambiar la fecha de sanción del usuario para el servicio de internet
setFechaSancionComputador	Permite asignar o cambiar la fecha de sanción del usuario para el servicio de computadores
setFechaSancionAudiovisual	Permite asignar o cambiar la fecha de sanción del usuario para el servicio audiovisual
<b>Observaciones:</b>	

### 11.1.4.3 Usuario Manager

<b>Nombre de la Clase:</b> UsuarioMgr
--

<b>Descripción de la clase</b>		
Clase que permite la administración de los datos de los usuarios. Tiene toda la lógica de la clase UsuarioVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_usuario	Arraylist (colección de obj) que almacena los datos del usuario como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UsuarioVO	<i>retorna</i>	Asociación
LugarVO	<i>utiliza</i>	Asociación
TipoidentificacionVO	<i>utiliza</i>	Asociación
ComputadorVO	<i>utiliza</i>	Asociación
UsuarioVisitor	<i>utiliza</i>	Asociación
TipoidentificacionMgr	<i>utiliza</i>	Asociación
LugarMgr	<i>utiliza</i>	Asociación
UsuarioDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la consulta de un Usuario de acuerdo a una identificación.	
findByNombreUno	Método que permite acceder a la consulta de todos los usuarios de acuerdo a su primer nombre.	
findByApellidoUno	Método que permite acceder a la consulta de todos los usuarios de acuerdo a su primer apellido.	
findByNombreApellido	Método que permite acceder a la consulta de todos los usuarios de acuerdo a su primer nombre y primer apellido.	
findByLugar	Método que permite acceder a la consulta de todos los usuarios de acuerdo a un determinado lugar de nacimiento o residencia.	
findByAllSancionInternet	Método que permite acceder a la consulta de todos los usuarios que tienen faltas para el Servicio de Internet.	
findByAllSancionAudiovisual	Método que permite acceder a la consulta de todos los usuarios que tienen faltas para el Servicio de Audiovisuales.	
findByAllSancionComputador	Método que permite acceder a la consulta de todos los usuarios que tienen faltas para el Servicio de Computadores.	
findByFechaIdentificacion	Método que permite acceder a la consulta de todas las reservas de internet de un usuario en una fecha determinada	
findByLoginPassword	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la consulta de un Usuario de acuerdo a un Login y Password.	



findByIdentificacion	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la consulta de un Usuario de acuerdo a una Identificación.
findByLogin	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la consulta de un Usuario de acuerdo a un Login.
findByIdentificacionPassword	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la consulta de un Usuario de acuerdo a una Identificación y a un Password.
createUsuario	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la creación de un Usuario determinado.
updateFechaFaltaComputador	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la actualización de las faltas y fecha de sanción para el Servicio de Computadores de un Usuario de acuerdo a una identificación.
updateLoginPassword	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la actualización del Login y el Password de un Usuario de acuerdo a una identificación.
updateUsuario	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la actualización de los datos personales de un Usuario de acuerdo a una identificación.
updatePassword	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la actualización del Password de un Usuario de acuerdo a una identificación.
deleteUsuario	Método que permite eliminar un usuario de acuerdo a la identificación.
updateFechaFaltaInternet	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la actualización de las faltas y fecha de sanción para el Servicio de Internet de un Usuario de acuerdo a una identificación.
updateFechaFaltaAudiovisual	Método que permite establecer los atributos de un objeto de tipo UsuarioVisitor para la actualización de las faltas y fecha de sanción para el Servicio de de Audiovisuales de un Usuario de acuerdo a una identificación.
<b>Observaciones:</b>	

#### 11.1.4.4 Usuario Data Access Object

<b>Nombre de la Clase:</b> UsuarioDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los datos personales del usuario		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
Connection conn		Recibe la conexión creada
ArrayList al_aux		Arreglo que almacena los resultados de las consultas
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UsuarioVisitor	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo UsuarioVisitor con todos sus parámetros de acuerdo a una determinada identificación.	
findByNombreUno	Método que retorna un ArrayList de objetos de tipo UsuarioVisitor de acuerdo al primer nombre del usuario.	
findByNombreApellido	Método que retorna un ArrayList de objetos de tipo UsuarioVisitor de acuerdo al primer nombre y primer apellido del usuario.	
findByLugar	Método que retorna un ArrayList de objetos de tipo UsuarioVisitor de acuerdo un a un determinado Lugar de Nacimiento o Residencia.	
findByAllSancionInternet	Método que retorna un ArrayList con todos los usuarios que tienen faltas para el Servicio de Internet.	
findByAllSancionAudiovisual	Método que retorna un ArrayList con todos los usuarios que tienen faltas para el Servicio de Audiovisuales.	
findByAllSancionComputador	Método que retorna un ArrayList con todos los usuarios que tienen faltas para el Servicio de Computadores.	
findByLoginPassword	Método que retorna un objeto de tipo UsuarioVisitor con los datos de un usuario de acuerdo a su Login y Password.	
findByIdentificacion	Método que retorna un objeto de tipo UsuarioVisitor con los datos de un usuario de acuerdo a su identificación.	
UsuarioVisitor	Método que retorna un objeto de tipo UsuarioVisitor con los datos de un usuario de acuerdo a su login.	
findByIdentificacionPassword	Método que retorna un objeto de tipo UsuarioVisitor con los datos de un usuario de acuerdo a su identificación y password.	
createUsuario	Método que permite registrar un Usuario	

updateFechaFaltaComputador	Método que actualiza los atributos correspondientes a las faltas y a la fecha de sanción del servicio de computadores del objeto de tipo UsuarioVisitor.
updateLoginPassword	Método que actualiza los atributos correspondientes al login y password de un objeto de tipo UsuarioVisitor.
updatePassword	Método que actualiza el atributo correspondiente al password de un objeto de tipo UsuarioVisitor.
updateFechaFaltaInternet	Método que actualiza los atributos correspondientes a las faltas y a la fecha de sanción del servicio de internet de un objeto de tipo UsuarioVisitor.
updateFechaFaltaAudiovisual	Método que actualiza los atributos correspondientes a las faltas y a la fecha de sanción del servicio de audiovisuales de un objeto de tipo UsuarioVisitor.
updateUsuario	Método que permite actualizar un Usuario.
deleteUsuario	Método que elimina un usuario.
<b>Observaciones:</b>	

#### 11.1.4.5 Tipo Identificación Value Object

<b>Nombre de la Clase:</b> TipoidentificacionVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los tipos de identificación		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codTipo	Identifica de manera única a un tipo de identificación	
String str_nomTipo	Cadena que almacena el nombre del tipo de identificación	
String str_desTipo	Cadena que almacena el nombre del tipo de identificación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>RoI Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
TipoidentificacionVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código del tipo de identificación	
getNombre	Retorna el nombre del tipo de identificación	
getDescripcion	Retorna la descripción del tipo de identificación	

setCodigo	Permite asignar el código del tipo de identificación
setNombre	Permite asignar el nombre del tipo de identificación
setDescripcion	Permite asignar la descripción del tipo de identificación
<b>Observaciones:</b>	

#### 11.1.4.6 Tipo de Identificación Manager

<b>Nombre de la Clase</b> TipoidentificacionMgr		
<b>Descripción de la clase</b> Clase que acceder a los tipos de identificación. Tiene toda la lógica de la clase TipoidentificacionVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de los tipos de identificación como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoidentificacionVO	<i>retorna</i>	Asociación
TipoidentificacionDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo TipoidentificacionVO de acuerdo a un código determinado	
findByNombre	Método que permite establecer los atributos de un objeto de tipo TipoidentificacionVO de acuerdo a un código determinado.	
<b>Observaciones:</b>		

#### 11.1.4.7 Tipo Identificación Data Access Object

<b>Nombre de la Clase:</b> TipoidentificacionDAO
---

<b>Descripción de la clase</b>		
Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a los tipos de identificación		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoidentificacionVO	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo TipoidentificacionVO de acuerdo a un código determinado.	
findByNombre	Método que retorna un objeto de tipo TipoidentificacionVO de acuerdo a un nombre determinado	
<b>Observaciones:</b>		

#### 11.1.4.8 Estado Usuario Value Object

<b>Nombre de la Clase:</b>		
estadoUsuarioVO		
<b>Descripción de la clase</b>		
Clase que representa de forma lógica los estados de los usuarios		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codEstado	Identifica de manera única un objeto de tipo EstadoUsuarioVO	
String str_nomEstado	Indica el nombre de un estado de usuario	
String str_desEstado	Indica la descripción de un estado de usuario	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
EstadoUsuarioVO	Método constructor con parámetros y sin parámetros	
getCodigoEstado	Retorna el código del estado	
getNombreEstado	Retorna el nombre del estado	
getDescripcionEstado	Retorna la descripción del estado	
setCodigoEstado	Permite asignar el código del estado	
setNombreEstado	Permite asignar el nombre del estado	
setDescripcionEstado	Permite asignar la descripción del estado	

**Observaciones:**

#### 11.1.4.9 Tipo de Identificación Manager

<b>Nombre de la Clase</b> EstadoUsuarioMgr		
<b>Descripción de la clase</b> Clase que permite acceder a los estados de los usuarios. Tiene toda la lógica de la clase EstadoUsuarioVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de los estados de los usuarios como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EstadoUsuarioVO	<i>retorna</i>	Asociación
EstadoUsuarioDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo EstadoUsuarioVO para la consulta de un estado de acuerdo a un código.	
findByAll	Método que permite acceder a la consulta de todos los estados de los usuarios.	
<b>Observaciones:</b>		

#### 11.1.4.10 Tipo Identificación Data Access Object

<b>Nombre de la Clase:</b> EstadoUsuarioDAO		
<b>Descripción de la clase</b> Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a los estados de los usuarios		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EstadoUsuarioVO	<i>Retorna</i>	Asociación

DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo EstadoUsuarioVO de acuerdo al código de estado determinado.	
findByAll	Método que retorna un ArrayList de objetos con todos los atributos de los estados de los usuarios.	
<b>Observaciones:</b>		

#### 11.1.4.11 Cargo Value Object

<b>Nombre de la Clase:</b> CargoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los cargos laborales de los usuarios		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codCargo	Identifica de manera única un objeto de tipo CargoVO	
String str_nomCargo	Indica el nombre de un cargo	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
CargoVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código del cargo	
getNombre	Retorna el nombre del cargo	
setCodigo	Permite asignar o cambiar el código del cargo	
setNombre	Permite asignar o cambiar el nombre del cargo	
<b>Observaciones:</b>		

#### 11.1.4.12 Cargo Manager

<b>Nombre de la Clase</b> CargoMgr	
<b>Descripción de la clase</b> Clase que permite acceder a los cargos laborales de los usuarios. Tiene toda la lógica de la clase CargoVO a nivel de modelo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de los cargos laborales de los usuarios como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CargoVO	<i>retorna</i>	Asociación
CargoDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo Cargoso para la consulta de un cargo de acuerdo a un código.	
findByNombre	Método que permite establecer los atributos de un objeto de tipo CargoVO para la consulta de un cargo de acuerdo a una descripción.	
findByAll	Método que permite acceder a la consulta de todos los cargos de la Universidad.	
createCargo	Método que permite establecer los atributos de un objeto de tipo CargoVO para la creación de un cargo determinado.	
updateCargo	Método que permite establecer los atributos de un objeto de tipo CargoVO para la actualización de un cargo determinado.	
deleteCargo	Método que permite establecer los atributos de un objeto de tipo CargoVO para la eliminación de un cargo determinado	
<b>Observaciones:</b>		

#### 11.1.4.13 Cargo Data Access Object

<b>Nombre de la Clase:</b> CargoDAO		
<b>Descripción de la clase</b> Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a los cargos laborales de los usuarios		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CargoVO	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	



findByPrimaryKey	Método que retorna un objeto de tipo CargoVO con todos sus atributos.
findByNombre	Método que retorna un objeto tipo CargoVO.
findByAll	Método que permite crear un Cargo.
updateCargo	Método que permite actualizar un determinado Cargo.
deleteCargo	Método que permite eliminar un determinado Cargo
<b>Observaciones:</b>	

#### 11.1.4.14 Estudiante Value Object

<b>Nombre de la Clase:</b> EstudianteVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de los estudiantes de la Universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
String str_codEstudiante	Cadena que identifica el código estudiantil asignado por la universidad	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UsuarioVO	es	Asociación
ProgramaVO	pertenece	Asociación
SemestreVO	estaEn	Asociación
EstadoUsuarioVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
EstudianteVO	Método constructor sin parámetros	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getCodigoEstudiante	Retorna el código del estudiante	
getProgramaVO	Retorna un objeto de tipo ProgramaVO	
getSemestreVO	Retorna un objeto de tipo SemestreVO	
getEstadoUsuarioVO	Retorna un objeto de tipo EstadoUsuarioVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setCodigoEstudiante	Permite asignar el código del estudiante	
setProgramaVO	Permite asignar o cambiar un objeto de tipo ProgramaVO	
setSemestreVO	Permite asignar o cambiar un objeto de tipo SemestreVO	
setEstadoUsuarioVO	Permite asignar o cambiar un objeto de tipo EstadoUsuarioVO	
<b>Observaciones:</b>		

#### 11.1.4.15 Estudiante Visitor

<b>Nombre de la Clase:</b> EstudianteVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de los estudiantes de la Universidad, hace referencia directa con cada uno de los campos de la tabla estudiantes	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_ideUsuario	Cadena que almacena la identificación del usuario
String str_codEstudiante	Cadena que almacena el código del estudiante
String str_codPrograma	Identifica el código del programa del estudiante
int int_codSemestre	Determina el código del semestre en el que está un estudiante
int int_codEstado	Identifica el código del estado del estudiante(Activo, inactivo,..)
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
EstudianteVisitor	Método constructor sin parámetros
getIdentificacionUsuario	Retorna la identificación del usuario
getCodigoEstudiante	Retorna el código del estudiante
getCodigoPrograma	Retorna el código del programa al que pertenece el estudiante
getCodigoSemestre	Retorna el semestre en el que se encuentra el estudiante
getCodigoEstado	Retorna el código del estado
setIdentificacionUsuario	Permite asignar la identificación del usuario
setCodigoEstudiante	Permite asignar el código del estudiante
setCodigoPrograma	Permite asignar o cambiar el código del programa
setCodigoSemestre	Permite asignar o cambiar el código del semestre
setCodigoEstado	Permite asignar o cambiar el código del estado
<b>Observaciones:</b>	

#### 11.1.4.16 Estudiante Manager

<b>Nombre de la Clase:</b> EstudianteMgr	
<b>Descripción de la clase</b> Clase que permite el acceso a los datos de los estudiantes de la universidad. Tiene toda la lógica de la clase EstudianteVO a nivel de modelo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_estudiante	ArrayList (colección de objetos) que almacena los resultados de las consultas
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación

<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EstudianteVO	retorna	Asociación
UsuarioVO	utiliza	Asociación
EstadoUsuarioVO	utiliza	Asociación
ProgramaVO	utiliza	Asociación
SemestreVO	utiliza	Asociación
EstudianteVisitor	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
ProgramaMgr	utiliza	Asociación
SemestreMgr	utiliza	Asociación
EstadoUsuarioMgr	utiliza	Asociación
EstudianteDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo EstudianteVisitor para la consulta de un Estudiante de acuerdo a una identificación	
findByCodigoEstudiante	Método que permite establecer los atributos de un objeto de tipo EstudianteVisitor para la consulta de un Estudiante de acuerdo a un código estudiantil	
findByProgramaSemestreEstado	Método que permite establecer los atributos de un objeto de tipo EstudianteVisitor para la consulta de un estudiantes de acuerdo a un código de programa, semestre y estado	
findByNombreApellidoEstado	Método que permite establecer los atributos de un objeto de tipo EstudianteVisitor para la consulta de un Estudiante de acuerdo a su primer nombre, primer apellido y estado	
createEstudiante	Método que permite establecer los atributos de un objeto de tipo EstudianteVisitor para la creación de un Estudiante determinado	
updateEstudiante	Método que permite establecer los atributos de un objeto de tipo EstudianteVisitor para la actualización de un Estudiante determinado	
<b>Observaciones:</b>		

#### 11.1.4.17 Estudiante Data Access Object

<b>Nombre de la Clase:</b> EstudianteDAO	
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los datos de los estudiantes	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EstudianteVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo EstudianteVisitor con todos sus atributos de acuerdo a una identificación determinada	
findByCodigoEstudiante	Método que retorna un objeto de tipo EstudianteVisitor con todos sus atributos de acuerdo a un código determinado	
findByProgramaSemestreEstado	Método que retorna un objeto de tipo EstudianteVisitor con todos sus atributos de acuerdo a un código de programa, semestre y estado determinado	
createEstudiante	Método que permite registrar un Estudiante	
updateEstudiante	Método que permite actualizar los datos de un Estudiante	
<b>Observaciones:</b>		

#### 11.1.4.18 Estudiante Base Value Object

<b>Nombre de la Clase:</b> EstudianteBaseVO	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de los estudiantes de la Universidad obtenidos desde la sede principal de Pasto	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_ideEstudiante	Cadena que determina la identificación del estudiante
String str_apeEstudiante	Cadena que identifica los apellidos del estudiante
String str_nomEstudiante	Cadena que identifica los nombres del estudiante
String str_dirEstudiante	Cadena que identifica la dirección del estudiante
String str_barDireccion	Cadena que identifica el barrio donde reside el estudiante
String str_telEstudiante	Cadena que identifica el teléfono del estudiante
String str_ciuNacimiento	Cadena que identifica la ciudad de nacimiento del estudiante
String str_depNacimiento	Cadena que identifica el departamento de nacimiento del estudiante
String str_paiNacimiento	Cadena que identifica el país del nacimiento del estudiante

String str_ciuResidencia	Cadena que identifica la ciudad de residencia del estudiante	
String str_depResidencia	Cadena que identifica el departamento de residencia del estudiante	
String str_paiResidencia	Cadena que identifica el país de residencia del estudiante	
String str_tiplIdentificacion	Cadena que identifica el tipo de identificación del estudiante	
String str_ciuidentificacion	Cadena que identifica la ciudad de expedición de la identificación del estudiante	
String str_deplIdentificacion	Cadena que identifica el departamento de expedición de la identificación del estudiante	
String str_paiIdentificacion	Cadena que identifica el país de expedición de la identificación del estudiante	
String str_sexEstudiante	Cadena que identifica el género o sexo del estudiante	
String str_estCivil	Cadena que identifica el estado civil del estudiante	
String str_codEstudiante	Cadena que identifica el código del estudiante	
String str_codFacultad	Cadena que identifica el código de la facultad a la que pertenece el estudiante	
String str_nomFacultad	Cadena que identifica el nombre de la facultad a la que pertenece el estudiante	
String str_codPrograma	Cadena que identifica el código del programa al que pertenece el estudiante	
String str_nomPrograma	Cadena que identifica el nombre del programa al que pertenece el estudiante	
String str_semEstudiante	Cadena que identifica el semestre al que pertenece el estudiante	
String str_estEstudiante	Cadena que identifica el estado del estudiante	
String str_dirPermanente	Cadena que identifica la dirección permanente del estudiante	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
EstudianteBaseVO	Método constructor sin parámetros	
getIdentificacion	Retorna la identificación del estudiante	
getApellido	Retorna los apellidos del estudiante	
getNombre	Retorna los nombres del estudiante	
getDireccion	Retorna la dirección del estudiante	
getBarrio	Retorna el barrio del estudiante	
getTelefono	Retorna el teléfono del estudiante	
getCiudadNacimiento	Retorna la ciudad de nacimiento del estudiante	
getDepartamentoNacimiento	Retorna el departamento de nacimiento del estudiante	
getPaisNacimiento	Retorna el país de nacimiento del estudiante	
getCiudadResidencia	Retorna la ciudad de residencia del estudiante	
getDepartamentoResidencia	Retorna el departamento de residencia del estudiante	
getPaisResidencia	Retorna el país de residencia del estudiante	
getTipoidentificacion	Retorna el tipo de identificación del estudiante	

getCiudadIdentificacion	Retorna la ciudad de expedición de la identificación del estudiante
getDepartamentoIdentificacion	Retorna el departamento de expedición de la identificación del estudiante
getPaisIdentificacion	Retorna el país de expedición de la identificación del estudiante
getSexo	Retorna el sexo del estudiante
getEstadoCivil	Retorna el estado civil del estudiante
getCodigoEstudiante	Retorna el código del estudiante
getCodigoFacultad	Retorna el código de la facultad del estudiante
getNombreFacultad	Retorna el nombre de la facultad del estudiante
getCodigoPrograma	Retorna el código del programa del estudiante
getNombrePrograma	Retorna el nombre del programa del estudiante
getSemestre	Retorna el semestre del estudiante
getEstadoEstudiante	Retorna el estado del estudiante
getDireccionPermanente	Retorna la dirección permanente del estudiante
setIdentificacion	Permite asignar la identificación del estudiante
setApellido	Permite asignar los apellidos del estudiante
setNombre	Permite asignar los nombres del estudiante
setDireccionEstudiante	Permite asignar la dirección del estudiante
setBarrio	Permite asignar el barrio del estudiante
setTelefono	Permite asignar el teléfono del estudiante
setCiudadNacimiento	Permite asignar la ciudad de nacimiento del estudiante
setDepartamentoNacimiento	Permite asignar el departamento de nacimiento del estudiante
setPaisNacimiento	Permite asignar el país de nacimiento del estudiante
setCiudadResidencia	Permite asignar la ciudad de residencia del estudiante
setDepartamentoResidencia	Permite asignar el departamento de residencia del estudiante
setPaisResidencia	Permite asignar el país de residencia del estudiante
setTipoidentificacion	Permite asignar el tipo de identificación del estudiante
setCiudadIdentificacion	Permite asignar la ciudad de expedición de la identificación del estudiante
setDepartamentoIdentificacion	Permite asignar el departamento de expedición de la identificación del estudiante
setPaisIdentificacion	Permite asignar el país de expedición de la identificación del estudiante
setSexo	Permite asignar el sexo del estudiante
setEstadoCivil	Permite asignar el estado civil del estudiante
setCodigoEstudiante	Permite asignar el código del estudiante
setCodigoFacultad	Permite asignar el código de la facultad del estudiante
setNombreFacultad	Permite asignar el nombre de la facultad del estudiante
setCodigoPrograma	Permite asignar el código del programa del estudiante
setNombrePrograma	Permite asignar el nombre del programa del estudiante
setSemestre	Permite asignar el semestre del estudiante
setEstadoEstudiante	Permite asignar el estado del estudiante
setDireccionPermanente	Permite asignar la dirección permanente del estudiante

**Observaciones:**

#### 11.1.4.19 Estudiante Base Manager

<b>Nombre de la Clase:</b> EstudianteBaseMgr		
<b>Descripción de la clase</b> Clase que permite el acceso a los datos de los estudiantes de la universidad descargados desde la sede principal de Pasto. Tiene toda la lógica de la clase EstudianteBaseVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_estudiante	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EstudianteBaseVO	retorna	Asociación
EstudianteBaseDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
loadDataInfile	Método que permite efectuar la carga de los datos de los estudiantes a través de un archivo suministrado por la sede principal de Pasto	
findByAll	Método que permite acceder a la consulta de todos los datos de los estudiantes descargados desde Pasto	
findByTotalIdentificacion	Método que permite acceder a la consulta del número de coincidencias de una determinada identificación	
deletelInformacion	Método que permite acceder a la inicialización de la información descargada desde Pasto	
<b>Observaciones:</b>		

#### 11.1.4.20 Estudiante Base Data Access Object

<b>Nombre de la Clase:</b> EstudianteDAO
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los datos de los estudiantes descargados desde Pasto

Propiedad		Descripción del modelaje, restricciones, información para diseño, suposiciones	
Connection conn		Recibe la conexión creada	
ArrayList al_aux		Arreglo que almacena los resultados de las consultas	
Clase Asociada	Nombre Asociación	Rol Clase Asociada	
EstudianteBaseVO	Retorna	Asociación	
DBConnection	Solicita	Asociación	
Servicio o Responsabilidad	Descripción		
loadDataInfile	Método que carga los datos de los estudiantes en la base de datos		
findByAll	Método que retorna un ArrayList(Colección de objetos) de tipo EstudianteBaseVO con todos los atributos de los estudiantes		
findByTotalIdentificacion	Método que permite retornar el número de coincidencias de una determinada identificación		
deletelInformacion	Método que permite borrar la información descargada desde Pasto de todos los estudiantes		
<b>Observaciones:</b>			

#### 11.1.4.21 Programa Value Object

<b>Nombre de la Clase:</b> ProgramaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los programas existentes en la Universidad		
Propiedad		Descripción del modelaje, restricciones, información para diseño, suposiciones
String str_codPrograma		Cadena que identifica de manera única un programa
String str_nomPrograma		Cadena que identifica el nombre de un programa
String str_abrPrograma		Cadena que identifica el nombre abreviado de un programa
Clase Asociada	Nombre Asociación	Rol Clase Asociada
FacultadVO	pertenece	Asociación
Servicio o Responsabilidad	Descripción	
ProgramaVO	Método constructor sin parámetros	
getCodigo	Retorna el código del programa	
getNombre	Retorna el nombre del programa	
getNombreAbreviado	Retorna el nombre abreviado del programa	
getFacultadVO	Retorna un objeto de tipo FacultadVO	
setCodigo	Permite asignar el código del programa	
setNombre	Permite asignar o cambiar el nombre del programa	
setNombreAbreviado	Permite asignar o cambiar el nombre abreviado del programa	



setFacultadVO	Permite asignar o cambiar un objeto de tipo FacultadVO
<b>Observaciones:</b>	

#### 11.1.4.22 Programa Visitor

<b>Nombre de la Clase:</b> ProgramaVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los programas de la Universidad, hace referencia directa con cada uno de los campos de la tabla programas	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_codPrograma	Cadena que almacena el código del programa
String str_nomPrograma	Cadena que almacena el nombre del programa
String str_abrPrograma	Cadena que almacena el nombre abreviado del programa
String str_codFacultad	Cadena que almacena el código de la facultad a la que pertenece programa
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ProgramaVisitor	Método constructor sin parámetros
getCodigoPrograma	Retorna el código del programa
getNombre	Retorna el nombre del programa
getNombreAbreviado	Retorna el nombre abreviado del programa
getCodigoFacultad	Retorna el código de la facultad
setCodigoPrograma	Permite asignar el código del programa
setNombre	Permite asignar el nombre del programa
setNombreAbreviado	Permite asignar el nombre abreviado del programa
setCodigoFacultad	Permite asignar el código de la facultad
<b>Observaciones:</b>	

#### 11.1.4.23 Programa Manager

<b>Nombre de la Clase:</b> ProgramaMgr	
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los programas existentes en la universidad. Tiene toda la lógica de la clase ProgramaVO a nivel de modelo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

ArrayList al_programa	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ProgramaVO	retorna	Asociación
FacultadVO	utiliza	Asociación
ProgramaVisitor	utiliza	Asociación
FacultadMgr	utiliza	Asociación
ProgramaDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ProgramaVisitor para la consulta de un Programa de acuerdo a un código	
findByAll	Método que permite acceder a la consulta de todos los programas existentes	
createPrograma	Método que permite establecer los atributos de un objeto de tipo ProgramaVisitor para la creación de un programa	
<b>Observaciones:</b>		

#### 11.1.4.24 Programa Data Access Object

<b>Nombre de la Clase:</b> EstudianteDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los programas ofrecidos en la universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ProgramaVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ProgramaVisitor con todos los datos de un programa de acuerdo a un código determinado	
findByAll	Método que retorna un ArrayList de objetos de tipo ProgramaVisitor con los datos de todos los programas ofrecidos	
createPrograma	Método que permite crear un programa	

**Observaciones:**

#### 11.1.4.25 Semestre Value Object

<b>Nombre de la Clase:</b> SemestreVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los semestres vigentes en la Universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codSemestre	Identifica de manera única un semestre de la universidad	
String str_desSemestre	Cadena que identifica la descripción del semestre	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
SemestreVO	Método constructor sin parámetros	
getCodigo	Retorna el código del semestre	
getDescripcion	Retorna la descripción del semestre	
setCodigo	Permite asignar el código del semestre	
setDescripcion	Permite asignar la descripción del semestre	
<b>Observaciones:</b>		

#### 11.1.4.26 Semestre Manager

<b>Nombre de la Clase:</b> SemestreMgr		
<b>Descripción de la clase</b> Clase que accede de forma lógica las operaciones realizadas a la base de datos con respecto a los semestres de la Universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
SemestreVO	retorna	Asociación
SemestreDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	

findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo SemestreVO para la consulta de un Semestre de acuerdo a un código
findByAll	Método que permite acceder a la consulta de todos los semestres existentes
<b>Observaciones:</b>	

#### 11.1.4.27 Semestre Data Access Object

<b>Nombre de la Clase:</b> SemestreDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los semestres existentes en la Universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
SemestreVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo SemestreVO con la descripción del semestre de acuerdo a un Código determinado	
findByAll	Método que retorna un ArrayList(Colección de objetos) de objetos de tipo SemestreVO con los datos de todos los semestres ofrecidos	
<b>Observaciones:</b>		

#### 11.1.4.28 Facultad Value Object

<b>Nombre de la Clase:</b> FacultadVO	
<b>Descripción de la clase</b> Clase que representa de forma lógica las facultades de la universidad	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_codFacultad	Identifica de manera única una facultad de la universidad

String str_nomFacultad	Cadena que identifica el nombre de la facultad	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
FacultadVO	Método constructor sin parámetros	
getCodigo	Retorna el código de la facultad	
getNombre	Retorna el nombre de la facultad	
setCodigo	Permite asignar el código de la facultad	
setNombre	Permite asignar el nombre de la facultad	
<b>Observaciones:</b>		

#### 11.1.4.29 Facultad Manager

<b>Nombre de la Clase:</b> FacultadMgr		
<b>Descripción de la clase</b> Clase que accede de forma lógica las operaciones realizadas a la base de datos con respecto a las facultades de la universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FacultadVO	retorna	Asociación
FacultadDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo FacultadVO para la consulta de un Rol de acuerdo a un código	
createFacultad	Método que permite establecer los atributos de un objeto de tipo FacultadVO para la creación de una facultad determinada	
<b>Observaciones:</b>		

#### 11.1.4.30 Facultad Data Access Object

<b>Nombre de la Clase:</b> FacultadDAO
---

<b>Descripción de la clase</b>		
Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las facultades		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FacultadVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo FacultadVO de acuerdo a un código determinado	
createFacultad	Método que crea una facultad	
<b>Observaciones:</b>		

#### 11.1.4.31 Lugar Value Object

<b>Nombre de la Clase:</b>		
LugarVO		
<b>Descripción de la clase</b>		
Clase que representa de forma lógica los lugares( Países, Departamentos y Ciudades)		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codLugar	Identifica de manera única un objeto de tipo LugarVO	
String str_desLugar	Indica el nombre del lugar	
int int_raizLugar	Indica la raíz del lugar o el código del país o departamento al que pertenece	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
LugarVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código del lugar	
getDescripcion	Retorna el nombre del lugar	
getRaiz	Permite retornar el código del lugar	
setCodigo	Permite asignar o cambiar el código del lugar	
setDescripcion	Permite asignar o cambiar el nombre del lugar	
setRaiz	Permite asignar o cambiar la raíz del lugar	
<b>Observaciones:</b>		

### 11.1.4.32 Lugar Manager

<b>Nombre de la Clase</b> LugarMgr		
<b>Descripción de la clase</b> Clase que permite acceder a los datos de los lugares. Tiene toda la lógica de la clase LugarVO a nivel de modelo		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_aux		Arraylist (colección de obj) que almacena los datos de los lugares como resultado de las consultas
String v_str_mensaje		Cadena que almacena el mensaje que retorna una determinada operación
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
LugarVO	retorna	Asociación
LugarDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>		<b>Descripción</b>
findByPrimaryKey		Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo a un código
findByDescripcion		Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo a una descripción.
findByCodigoPais		Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo al código del país.
findByRaiz		Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo al código de la raíz.
administracionPais		Método que realiza la búsqueda de un país por su descripción y si no lo encuentra lo crea y devuelve los datos del país. Si el país viene vacío se determina si ya existe por defecto el país Colombia, si no existe lo crea y devuelve los datos del país.

administracionDepartamento	Método que realiza la búsqueda de un departamento por su descripción y como hijo de un país determinado, si no lo encuentra lo crea como hijo de ese país y devuelve los datos del departamento. Si el departamento viene vacío se determina si ya existe por defecto el departamento Nariño, si no existe lo crea como hijo del país de referencia y devuelve los datos del departamento.
administracionCiudad	Método que realiza la búsqueda de una ciudad por su descripción y como hijo de un departamento determinado, si no lo encuentra lo crea como hijo de ese departamento y devuelve los datos de la ciudad. Si la ciudad está vacía se determina si ya existe por defecto la ciudad Ipiales, si no existe lo crea como hijo del departamento de referencia y devuelve los datos de la ciudad.
administracionAutomaticaCiudad	Método que realiza la búsqueda de una ciudad por su descripción y si no la encuentra, la crea como un hija del departamento Nariño y el país Colombia
administracionAutomaticaDepartamento	Método que realiza la búsqueda de un departamento por su descripción y si no lo encuentra, lo crea como un hijo del país Colombia.
findByDescripcionCiudad	Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo a su descripción y que sea una ciudad.
findByDescripcionDepartamento	Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo a su descripción y que sea un departamento
findByDescripcionPais	Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo a una descripción y que sea un país.
findByDescripcionRaiz	Método que permite establecer los atributos de un objeto de tipo LugarVO para la consulta de un lugar de acuerdo a una descripción y una raíz determinada.



createLugar	Método que permite establecer los atributos de un objeto de tipo LugarVO para la creación de un lugar determinado.
updateLugar	Método que permite establecer los atributos de un objeto de tipo LugarVO para la actualización de un Lugar determinado.
deleteLugar	Método que permite eliminar un determinado Lugar.
<b>Observaciones:</b>	

### 11.1.4.33 Lugar Data Access Object

<b>Nombre de la Clase:</b> LugarDAO		
<b>Descripción de la clase</b> Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a los lugares		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
LugarVO	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo LugarVO con todos sus atributos.	
findByDescripcion	Método que retorna un objeto tipo LugarVO de acuerdo a la descripción.	
findByDescripcionGeneral	Método que retorna un ArrayList de objetos de tipo LugarVO de acuerdo a una descripción.	
findByDescripcionPais	Método que retorna un bjecto de tipo LugarVO de acuerdo a una descripción y que sea un país.	
findByDescripcionRaiz	Método que retorna un objeto de tipo LugarVO de acuerdo a una descripción y raíz determinada.	
findByCodigoPais	Método que retorna un ArrayList con los departamentos de un determinado País.	
findByRaiz	Método que retorna un ArrayList de objetos de tipo LugarVO de acuerdo a la raíz	
createLugar	Método que permite crear un Lugar	
updateLugar	Método que permite actualizar un determinado Lugar.	

deleteLugar	Método que permite eliminar un determinado Lugar
<b>Observaciones:</b>	

#### 11.1.4.34 Dependencia Value Object

<b>Nombre de la Clase:</b> DependenciaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las dependencias existentes en la universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codDependencia	Identifica de manera única un objeto de tipo DependenciaVO	
String str_nomDependencia	Cadena que almacena el nombre de una dependencia	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
DependenciaVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código de la dependencia	
getNombre	Retorna el nombre de la dependencia	
setCodigo	Permite asignar o cambiar el código de la dependencia	
setNombre	Permite asignar o cambiar el nombre de la dependencia	
<b>Observaciones:</b>		

#### 11.1.4.35 Dependencia Manager

<b>Nombre de la Clase</b> DependenciaMgr	
<b>Descripción de la clase</b> Clase que permite acceder a las dependencias de la universidad. Tiene toda la lógica de la clase DependenciaVO a nivel de modelo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de las dependencias como resultado de las consultas

String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
DependenciaVO	<i>retorna</i>	Asociación
DependenciaDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo DependenciaVO para la consulta de una dependencia de acuerdo al código.	
findByAll	Método que permite acceder a la consulta de todas las dependencias de la Universidad	
<b>Observaciones:</b>		

#### 11.1.4.36 Dependencia Data Access Object

<b>Nombre de la Clase:</b> DependenciaDAO		
<b>Descripción de la clase</b> Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a las dependencias		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
DependenciaVO	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo DependenciaVO de acuerdo a un Código de dependencia determinado.	
findByAll	Método que retorna un ArrayList de objetos con todos los atributos de las dependencias de la Universidad.	
<b>Observaciones:</b>		

#### 11.1.4.37 Empleado Value Object

<b>Nombre de la Clase:</b> EmpleadoVO
--

<b>Descripción de la clase</b>		
Clase que representa de forma lógica los datos de los empleados		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UsuarioVO	es	Asociación
DependenciaVO	pertenece	Asociación
CargoVO	tiene	Asociación
EstadoUsuarioVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
EmpleadoVO	Método constructor con parámetros y sin parámetros	
getUsuarioVO	Método analizador, que permite retornar un objeto de tipo UsuarioVO	
getDependenciaVO	Método analizador, que permite retornar un objeto de tipo DependenciaVO.	
getCargoVO	Método analizador, que permite retornar un objeto de tipo CargoVO	
getEstadoUsuarioVO	Método analizador, que permite retornar un objeto de tipo EstadoUsuarioVO.	
setUsuarioVO	Permite asignar o cambiar la identificación del usuario	
setDependenciaVO	Permite asignar o cambiar la dependencia del usuario	
setCargoVO	Método modificador que asigna un objeto de tipo CargoVO.	
setEstadoUsuarioVO	Método modificador que asigna un objeto de tipo EstadoUsuarioVO.	
<b>Observaciones:</b>		

#### 11.1.4.38 Empleado Visitor

<b>Nombre de la Clase:</b>	
EmpleadoVisitor	
<b>Descripción de la clase</b>	
Clase que representa de forma lógica los datos de los empleados. Hace referencia directa con cada uno de los campos de la tabla empleados	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_ideUsuario	Identifica de manera única un empleado
int_codDependencia	Identifica el primer código de la dependencia
int_int_codCargo	Identifica el código del cargo
int_int_codEstado	Indica el código del estado del empleado
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>

EmpleadoVisitor	Método constructor con parámetros y sin parámetros
getIdentificacionUsuario	Retorna la identificación del empleado.
getCodigoDependencia	Retorna el código de la dependencia a la que pertenece el empleado
getCodigoCargo	Retorna el código del cargo del empleado
getCodigoEstado	Retorna el código del estado del empleado
setIdentificacionUsuario	Permite asignar un valor al atributo Identificación del objeto EmpleadoVisitor.
setCodigoDependencia	Permite asignar un valor al atributo Código de Dependencia del objeto EmpleadoVisitor.
setCodigoCargo	Permite asignar un valor al atributo Código del Cargo del objeto EmpleadoVisitor
setCodigoEstado	Permite asignar un valor al atributo Código Estado del objeto EmpleadoVisitor.
<b>Observaciones:</b>	

#### 11.1.4.39 Empleado Manager

<b>Nombre de la Clase:</b> EmpleadoMgr		
<b>Descripción de la clase</b> Clase que permite la administración de los datos de los empleados. Tiene toda la lógica de la clase EmpleadoVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_empleado	Arraylist (colección de obj) que almacena los datos del empleado como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EmpleadoVO	<i>retorna</i>	Asociación
UsuarioVO	<i>utiliza</i>	Asociación
DependenciaVO	<i>utiliza</i>	Asociación
CargoVO	<i>utiliza</i>	Asociación
EstadoUsuarioVO	<i>utiliza</i>	Asociación
UsuarioMgr	<i>utiliza</i>	Asociación
DependenciaMgr	<i>utiliza</i>	Asociación
CargoMgr	<i>utiliza</i>	Asociación
EstadoUsuarioMgr	<i>utiliza</i>	Asociación
EmpleadoVisitor	<i>utiliza</i>	Asociación
EmpleadoDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	

findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo EmpleadoVisitor para la consulta de los datos de un empleado de acuerdo a una determinada identificación.
findByAll	Método que permite acceder a la consulta de todos los empleados de la Universidad.
findByCargo	Método que permite acceder a la consulta de todos los empleados de la Universidad.
findByTotalCargo	Método que permite acceder a la consulta del número de empleados que tiene asignado un determinado cargo.
createEmpleado	Método que permite establecer los atributos de un objeto de tipo EmpleadoVisitor para la creación un empleado determinado.
updateEmpleado	Método que permite establecer los atributos de un objeto de tipo EmpleadoVisitor para la actualización de un empleado determinado.
deleteEmpleado	Método que permite establecer los atributos de un objeto de tipo EmpleadoVisitor para la eliminación de un empleado determinado.
<b>Observaciones:</b>	

#### 11.1.4.40 Empleado Data Access Object

<b>Nombre de la Clase:</b> EmpleadoDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los empleados		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EmpleadoVisitor	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo EmpleadoVisitor de acuerdo a una identificación de usuario determinada	
findByAll	Método que retorna un ArrayList de objetos de tipo EmpleadoVisitor con los datos de todos los empleados de la universidad	

findByCargo	Método que retorna un ArrayList de objetos de tipo EmpleadoVisitor con los datos de los empleados de acuerdo a un cargo determinado.
findByTotalCargo	Retorna el número de empleados que tienen un determinado cargo.
createEmpleado	Método que permite registrar un empleado.
updateEmpleado	Método que permite actualizar un empleado.
deleteEmpleado	Método que permite eliminar un empleado.
<b>Observaciones:</b>	

#### 11.1.4.41 Médico Value Object

<b>Nombre de la Clase:</b> MedicoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del personal de salud(Médico, Odontólogo, Psicólogo)		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
String str_dirConsultorio	Cadena que almacena la dirección del consultorio	
String str_telConsultorio	Cadena que almacena el teléfono del consultorio	
String str_email	Cadena que almacena el correo electrónico	
int int_codServicio	Identifica el código del servicio que presta el personal de salud	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UsuarioVO	es	Asociación
EstadoUsuarioVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
MedicoVO	Método constructor con parámetros y sin parámetros	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getDireccion	Retorna la dirección del consultorio	
getTelefono	Retorna el teléfono del consultorio	
getEstadoUsuarioVO	Retorna un objeto de tipo EstadoUsuarioVO.	
getEmail	Retorna el correo electrónico del personal de salud.	
getServicio	Retorna el código del servicio que presta el personal de salud.	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setEstadoUsuarioVO	Permite asignar o cambiar un objeto de tipo EstadoUsuarioVO	

**Observaciones:**

#### 11.1.4.42 Médico Visitor

<b>Nombre de la Clase:</b> MedicoVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del personal de salud(médicos, odontólogos, psicólogos). Hace referencia directa con cada uno de los campos de la tabla médicos	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_ideUsuario	Identifica de manera única un médico
String str_dirConsultorio	Cadena que almacena la dirección del consultorio
String str_telConsultorio	Cadena que almacena el teléfono del consultorio
int int_codEstado	Indica el código del estado del empleado
String str_email	Cadena que almacena el correo electrónico
int int_codServicio	Identifica el código del servicio que presta el personal de salud.
int int_codEstado	Identifica el código del estado del personal de salud.
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
MedicoVisitor	Método constructor con parámetros y sin parámetros
getIdentificacion	Retorna la identificación del médico
getDireccion	Retorna la dirección del consultorio
getTelefono	Retorna el teléfono del consultorio
getEmail	Retorna la dirección del correo electrónico
getServicio	Retorna el servicio que presta el personal de salud
detestado	Retorna el servicio que presta el personal de salud
setIdentificacion	Permite asignar o cambiar la identificación
setDireccion	Permite asignar o cambiar la dirección del consultorio
setTelefono	Permite asignar o cambiar el teléfono del consultorio
setEmail	Permite asignar o cambiar la dirección del correo electrónico.
setServicio	Permite asignar o cambiar el código del servicio que presta el personal de salud



detestado	Permite asignar o cambiar el código del estado del personal de salud
<b>Observaciones:</b>	

#### 11.1.4.43 Médico Manager

<b>Nombre de la Clase:</b> MedicoMgr		
<b>Descripción de la clase</b> Clase que permite la administración de los datos del personal de salud. Tiene toda la lógica de la clase MedicoVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_medico	Arraylist (colección de obj) que almacena los datos del médico como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
MedicoVO	<i>retorna</i>	Asociación
UsuarioVO	<i>utiliza</i>	Asociación
EstadoUsuarioVO	<i>utiliza</i>	Asociación
UsuarioMgr	<i>utiliza</i>	Asociación
EstadoUsuarioMgr	<i>utiliza</i>	Asociación
MedicoVisitor	<i>utiliza</i>	Asociación
MedicoDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo MedicoVO para la consulta de los datos de un médico de acuerdo a una determinada identificación.	
findByServicio	Método que permite retornar los datos del médico que presta un determinado servicio.	
findByServicioEstado	Método que permite retornar los datos del médico que presta un determinado servicio y tenga un estado determinado.	
findByAll	Método que permite retornar un ArrayList de objetos de tipo MedicoVO.	
createMedico	Método que permite registrar el personal de Salud.	
updateMedico	Método que permite actualizar los datos profesionales del personal de Salud.	
deleteMedico	Método que permite eliminar los datos profesionales del personal de Salud.	
<b>Observaciones:</b>		

#### 11.1.4.44 Médico Data Access Object

<b>Nombre de la Clase:</b> MedicoDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto al personal de salud(médicos, odontólogos, psicólogos)		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
MedicoVisitor	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo MedicoVisitor con todos los datos del personal de salud de acuerdo a una identificación determinado.	
findByServicio	Método que permite retornar los datos del médico que presta un determinado servicio.	
findByServicioEstado	Método que permite retornar los datos del médico que presta un determinado servicio y estado (activo, inactivo etc)	
findByAll	Método que permite retornar un ArrayList de objetos de tipo MedicoVO.	
createMedico	Método que permite registrar el personal de Salud.	
updateMedico	Método que permite actualizar el personal de Salud.	
deleteMedico	Método que permite eliminar el personal de Salud.	
<b>Observaciones:</b>		

#### 11.1.4.45 Profesor Value Object

<b>Nombre de la Clase:</b> ProfesorVO	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del personal docente de la universidad	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_titProfesor	Cadena que almacena el título del profesor
String str_email	Cadena que almacena la dirección del correo electrónico

Clase Asociada	Nombre Asociación	Rol Clase Asociada
UsuarioVO	es	Asociación
EstadoUsuarioVO	tiene	Asociación
Servicio o Responsabilidad	Descripción	
ProfesorVO	Método constructor con parámetros y sin parámetros	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getTituloProfesor	Retorna el título del profesor	
getEmail	Retorna la dirección del correo	
getEstadoUsuarioVO	Retorna un objeto de tipo EstadoUsuarioVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	
setTituloProfesor	Permite asignar o cambiar el título del profesor	
setEmail	Permite asignar o cambiar la dirección del correo electrónico	
setEstadoUsuarioVO	Permite asignar o cambiar un objeto de tipo EstadoUsuarioVO	
<b>Observaciones:</b>		

#### 11.1.4.46 Profesor Visitor

<b>Nombre de la Clase:</b> ProfesorVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del personal docente. Hace referencia directa con cada uno de los campos de la tabla profesores	
Propiedad	Descripción del modelaje, restricciones, información para diseño, suposiciones
String str_ideUsuario	Identifica de manera única un profesor
String str_titProfesor	Cadena que almacena el título del profesor
String str_email	Cadena que almacena la dirección del correo electrónico
int int_codEstado	Indica el código del estado del profesor
Servicio o Responsabilidad	Descripción
ProfesorVisitor	Método constructor con parámetros y sin parámetros
getIdentificacionUsuario	Retorna la identificación del profesor
getTituloProfesor	Retorna el título del profesor
getEmail	Retorna la dirección del correo electrónico
getCodigoEstado	Retorna el código del estado del profesor
setIdentificacionUsuario	Permite asignar o cambiar la identificación del profesor
setTituloProfesor	Permite asignar o cambiar el título del profesor

setEmail	Permite asignar o cambiar la dirección del correo electrónico
setCodigoEstado	Permite asignar o cambiar el código del estado
<b>Observaciones:</b>	

#### 11.1.4.47 Profesor Manager

<b>Nombre de la Clase:</b> ProfesorMgr		
<b>Descripción de la clase</b> Clase que permite la administración de los datos del personal docente. Tiene toda la lógica de la clase ProfesorVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_profesor	Arraylist (colección de obj) que almacena los datos de los profesores como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ProfesorVO	<i>retorna</i>	Asociación
UsuarioVO	<i>utiliza</i>	Asociación
EstadoUsuarioVO	<i>utiliza</i>	Asociación
UsuarioMgr	<i>utiliza</i>	Asociación
EstadoUsuarioMgr	<i>utiliza</i>	Asociación
ProfesorVisitor	<i>utiliza</i>	Asociación
ProfesorDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ProfesorVisitor para la consulta de los datos de un profesor de acuerdo a una determinada identificación.	
findByAll	Método que permite acceder a la consulta de todo el personal docente de la Universidad	
createProfesor	Método que permite establecer los atributos de un objeto de tipo ProfesorVisitor para la creación de un profesor determinado.	
updateProfesor	Método que permite establecer los atributos de un objeto de tipo ProfesorVisitor para la actualización de un profesor determinado.	
deleteProfesor	Método que permite establecer los atributos de un objeto de tipo ProfesorVisitor para la eliminación de un profesor determinado.	
<b>Observaciones:</b>		

#### 11.1.4.48 Profesor Data Access Object

<b>Nombre de la Clase:</b> ProfesorDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto al personal docente.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ProfesorVisitor	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ProfesorVisitor de acuerdo a una identificación de usuario determinada.	
findByAll	Método que retorna un ArrayList de objetos de tipo ProfesorVisitor con los datos de los profesores asignados a la universidad.	
createProfesor	Método que permite registrar un profesor.	
updateProfesor	Método que permite actualizar los datos de un profesor	
deleteProfesor	Método que permite eliminar un profesor	
<b>Observaciones:</b>		

#### 11.1.5 Secretaria

##### 11.1.5.1 Horario Médico Value Object

<b>Nombre de la Clase:</b> HorarioMedicoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del horario para el servicio Médico.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FechaVO	esAsignada	Asociación
FranjaVO	esAsignada	Asociación

<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
HorarioMedicoVO	Método constructor con parámetros y sin parámetros
getFechaVO	Retorna un objeto de tipo FechaVO
getFranjaVO	Retorna un objeto de tipo FranjaVO
setFechaVO	Permite asignar o cambiar un objeto de tipo FechaVO
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO
<b>Observaciones:</b>	

### 11.1.5.2 Horario Médico Visitor

<b>Nombre de la Clase:</b> HorarioMedicoVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del horario Médico. Hace referencia directa con cada uno de los campos de la tabla horariosmedico	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int_codDia	Identifica el código de cualquier día de la semana
int_codFranja	Identifica el código de la franja.
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
HorarioMedicoVisitor	Método constructor con parámetros y sin parámetros
getCodigoDia	Retorna el código del día
getCodigoFranja	Retorna el código de la franja
setCodigoDia	Permite asignar o cambiar el código del día
setCodigoFranja	Permite asignar o cambiar el código de la franja
<b>Observaciones:</b>	

### 11.1.5.3 Horario Médico Manager

<b>Nombre de la Clase:</b> HorarioMedicoMgr	
<b>Descripción de la clase</b> Clase que permite la administración del horario para el servicio médico. Tiene toda la lógica de la clase HorarioMedicoVO a nivel de modelo	

Propiedad		Descripción del modelaje, restricciones, información para diseño, suposiciones
ArrayList al_horario		Arraylist (colección de obj) que almacena los datos del horario como resultado de las consultas
String v_str_mensaje		Cadena que almacena el mensaje que retorna una determinada operación
Clase Asociada	Nombre Asociación	Rol Clase Asociada
HorarioMedicoVO	<i>retorna</i>	Asociación
FechaVO	<i>utiliza</i>	Asociación
FranjaVO	<i>utiliza</i>	Asociación
FechaMgr	<i>utiliza</i>	Asociación
FranjaMgr	<i>utiliza</i>	Asociación
HorarioMedicoVisitor	<i>utiliza</i>	Asociación
HorarioMedicoDAO	<i>utiliza</i>	Asociación
Servicio o Responsabilidad	Descripción	
findByCodigoDia	Método que permite acceder a la consulta del horario médico de acuerdo a un determinado día	
findByDiaFranja	Método que permite establecer los atributos de un objeto de tipo HorarioMedicoVisitor para la consulta de un horario médico de acuerdo a un determinado día y franja	
findByFranja	Método que permite acceder a la consulta del número de coincidencias de una determinada franja	
findByFranjaDia	Método que permite acceder a la consulta del número de coincidencias de una franja y día determinados	
createHorarioMedico	Método que permite establecer los atributos de un objeto de tipo HorarioMedicoVisitor para registrar un horario	
deleteHorarioMedico	Método que permite establecer los atributos de un objeto de tipo HorarioMedicoVisitor para eliminar un horario	
<b>Observaciones:</b>		

#### 11.1.5.4 Horario Médico Data Access Object

<b>Nombre de la Clase:</b> HorarioMedicoDAO	
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto al horario médico.	
Propiedad	Descripción del modelaje, restricciones, información para diseño, suposiciones
Connection conn	Recibe la conexión creada

ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioMedicoVisitor	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoDia	Método que retorna un ArrayList de objetos con todos los atributos de los horarios médicos de acuerdo a un determinado día	
findByDiaFranja	Método que retorna un ArrayList de objetos con todos los atributos de los horarios médicos de acuerdo a una determinada franja y día.	
findByFranja	Método que permite retornar el número de coincidencias de una franja	
findByFranjaDia	Método que permite retornar el número de coincidencias de acuerdo a una determinada franja y día.	
createHorarioMedico	Método que permite crear un registro del Horario Médico de acuerdo a un determinado día y franja.	
deleteHorarioMedico	Método que elimina un registro del Horario Médico de acuerdo a un determinado día y franja.	
<b>Observaciones:</b>		

### 11.1.5.5 Horario Odontólogo Value Object

<b>Nombre de la Clase:</b> HorarioOdontologoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del horario para el servicio Odontológico.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FechaVO	esAsignada	Asociación
FranjaVO	esAsignada	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
HorarioOdontologoVO	Método constructor con parámetros y sin parámetros	
getFechaVO	Retorna un objeto de tipo FechaVO	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
setFechaVO	Permite asignar o cambiar un objeto de tipo FechaVO	



setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO
<b>Observaciones:</b>	

### 11.1.5.6 Horario Odontólogo Visitor

<b>Nombre de la Clase:</b> HorarioOdontologoVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del horario para el servicio Odontológico. Hace referencia directa con cada uno de los campos de la tabla horariosodontologo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int_codDia	Identifica el código de cualquier día de la semana
int_codFranja	Identifica el código de la franja.
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
HorarioOdontologoVisitor	Método constructor con parámetros y sin parámetros
getCodigoDia	Retorna el código del día
getCodigoFranja	Retorna el código de la franja
setCodigoDia	Permite asignar o cambiar el código del día
setCodigoFranja	Permite asignar o cambiar el código de la franja
<b>Observaciones:</b>	

### 11.1.5.7 Horario Odontólogo Manager

<b>Nombre de la Clase:</b> HorarioOdontologoMgr		
<b>Descripción de la clase</b> Clase que permite la administración del horario para el servicio odontológico. Tiene toda la lógica de la clase HorarioOdontologoVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_horario	Arraylist (colección de obj) que almacena los datos del horario como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>

HorarioOdontologoVO	<b>retorna</b>	Asociación
FechaVO	<b>utiliza</b>	Asociación
FranjaVO	<b>utiliza</b>	Asociación
FechaMgr	<b>utiliza</b>	Asociación
FranjaMgr	<b>utiliza</b>	Asociación
HorarioOdontologoVisitor	<b>utiliza</b>	Asociación
HorarioOdontologoDAO	<b>utiliza</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoDia	Método que permite acceder a la consulta del horario odontológico de acuerdo a un determinado día	
findByDiaFranja	Método que permite establecer los atributos de un objeto de tipo HorarioOdontologoVisitor para la consulta de un horario odontológico de acuerdo a un determinado día y franja	
findByFranja	Método que permite acceder a la consulta del número de coincidencias de una determinada franja	
findByFranjaDia	Método que permite acceder a la consulta del número de coincidencias de una franja y día determinados	
createHorarioOdontologo	Método que permite establecer los atributos de un objeto de tipo HorarioOdontologoVisitor para registrar un horario	
deleteHorarioOdontologo	Método que permite establecer los atributos de un objeto de tipo HorarioOdontologoVisitor para eliminar un horario	
<b>Observaciones:</b>		

### 11.1.5.8 Horario Odontólogo Data Access Object

<b>Nombre de la Clase:</b> HorarioOdontologoDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto al horario odontológico.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioOdontologoVisitor	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	

findByCodigoDia	Método que retorna un ArrayList de objetos con todos los atributos del horario odontológico de acuerdo a un determinado día
findByDiaFranja	Método que retorna un ArrayList de objetos con todos los atributos de los horarios odontológico de acuerdo a una determinada franja y día.
findByFranja	Método que permite retornar el número de coincidencias de una franja
findByFranjaDia	Método que permite retornar el número de coincidencias de acuerdo a una determinada franja y día.
createHorarioOdontologo	Método que permite crear un registro del Horario Odontológico de acuerdo a un determinado día y franja.
deleteHorarioOdontologo	Método que elimina un registro del Horario Odontológico de acuerdo a un determinado día y franja.
<b>Observaciones:</b>	

### 11.1.5.9 Horario Psicólogo Value Object

<b>Nombre de la Clase:</b> HorarioPsicologoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del horario para el servicio Psicológico.		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FechaVO	esAsignada	Asociación
FranjaVO	esAsignada	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
HorarioPsicologoVO	Método constructor con parámetros y sin parámetros	
getFechaVO	Retorna un objeto de tipo FechaVO	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
setFechaVO	Permite asignar o cambiar un objeto de tipo FechaVO	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
<b>Observaciones:</b>		

### 11.1.5.10 Horario Psicólogo Visitor

<b>Nombre de la Clase:</b> HorarioPsicologoVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del horario para el servicio Psicológico. Hace referencia directa con cada uno de los campos de la tabla horariospsicologo	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int_codDia	Identifica el código de cualquier día de la semana
int_codFranja	Identifica el código de la franja.
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
HorarioPsicologoVisitor	Método constructor con parámetros y sin parámetros
getCodigoDia	Retorna el código del día
getCodigoFranja	Retorna el código de la franja
setCodigoDia	Permite asignar o cambiar el código del día
setCodigoFranja	Permite asignar o cambiar el código de la franja
<b>Observaciones:</b>	

### 11.1.5.11 Horario Psicólogo Manager

<b>Nombre de la Clase:</b> HorarioPsicologoMgr		
<b>Descripción de la clase</b> Clase que permite la administración del horario para el servicio psicológico. Tiene toda la lógica de la clase HorarioPsicologoVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_horario	Arraylist (colección de obj) que almacena los datos del horario como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioPsicologoVO	<i>retorna</i>	Asociación
FechaVO	<i>utiliza</i>	Asociación
FranjaVO	<i>utiliza</i>	Asociación
FechaMgr	<i>utiliza</i>	Asociación
FranjaMgr	<i>utiliza</i>	Asociación
HorarioPsicologoVisitor	<i>utiliza</i>	Asociación

HorarioPsicologoAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoDia	Método que permite acceder a la consulta del horario psicológico de acuerdo a un determinado día	
findByDiaFranja	Método que permite establecer los atributos de un objeto de tipo HorarioPsicologoVisitor para la consulta de un horario psicológico de acuerdo a un determinado día y franja	
findByFranja	Método que permite acceder a la consulta del número de coincidencias de una determinada franja	
findByFranjaDia	Método que permite acceder a la consulta del número de coincidencias de una franja y día determinados	
createHorarioPsicologo	Método que permite establecer los atributos de un objeto de tipo HorarioPsicologoVisitor para registrar un horario	
deleteHorarioPsicologo	Método que permite establecer los atributos de un objeto de tipo HorarioPsicologoVisitor para eliminar un horario	
<b>Observaciones:</b>		

### 11.1.5.12 Horario Psicólogo Data Access Object

<b>Nombre de la Clase:</b> HorarioPsicologoDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto al horario psicológico.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
HorarioPsicologoVisitor	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoDia	Método que retorna un ArrayList de objetos con todos los atributos del horario psicológico de acuerdo a un determinado día	

findByDiaFranja	Método que retorna un ArrayList de objetos con todos los atributos de los horarios psicológico de acuerdo a una determinada franja y día.
findByFranja	Método que permite retornar el número de coincidencias de una franja
findByFranjaDia	Método que permite retornar el número de coincidencias de acuerdo a una determinada franja y día.
createHorarioPsicologo	Método que permite crear un registro del Horario Psicológico de acuerdo a un determinado día y franja.
deleteHorarioPsicologo	Método que elimina un registro del Horario Psicológico de acuerdo a un determinado día y franja.
<b>Observaciones:</b>	

### 11.1.5.13 Cita Médica Value Object

<b>Nombre de la Clase:</b> CitaMedicaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las citas Médicas.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codCita	Identifica de forma única un cita médica	
String str_fecCita	Cadena que almacena la fecha de la cita médica	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	tieneRegistrada	Asociación
usuarioVO	esAsignada	Asociación
medicoVO	esAtendida	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
CitaMedicaVO	Método constructor con parámetros y sin parámetros	
getCodigoCita	Retorna el código de la cita médica	
getFechaCita	Retorna la fecha de la cita médica	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getMedicoVO	Retorna un objeto de tipo MedicoVO	
setCodigoCita	Permite asignar o cambiar el código de la cita médica	
setFechaCita	Permite asignar o cambiar la fecha de la cita médica	

setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO
setMedicoVO	Permite asignar o cambiar un objeto de tipo MedicoVO
<b>Observaciones:</b>	

#### 11.1.5.14 Cita Médica Visitor

<b>Nombre de la Clase:</b> CitaMedicaVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de las citas médicas. Hace referencia directa con cada uno de los campos de la tabla citasmedicas	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codCita	Identifica el código de cualquier día de la semana
String str_fecCita	Identifica la fecha de cita médica
int int_codFranja	Identifica el código de la franja
String str_ideUsuario	Determina la identificación del usuario
String str_ideMedico	Identifica la identificación del médico
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
CitaMedicaVisitor	Método constructor con parámetros y sin parámetros
getCodigoCita	Retorna el código de la cita
getFechaCita	Retorna la fecha de la cita
getCodigoFranja	Retorna el código de la franja
getIdentificacionUsuario	Retorna la identificación del usuario
getIdentificacionMedico	Retorna la identificación del médico
setCodigoCita	Permite asignar o cambiar el código de la cita médica
setFechaCita	Permite asignar o cambiar la fecha de la cita médica
setCodigoFranja	Permite asignar o cambiar el código de la franja
setIdentificacionUsuario	Permite asignar o cambiar la identificación del usuario
setIdentificacionMedico	Permite asignar o cambiar la identificación del médico
<b>Observaciones:</b>	

### 11.1.5.15 Cita Medica Manager

<b>Nombre de la Clase:</b> CitaMedicaMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las citas médicas. Tiene toda la lógica de la clase CitaMedicaVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_citaMedica	Arraylist (colección de obj) que almacena los datos de las citas médicas como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CitaMedicaVO	<i>retorna</i>	Asociación
UsuarioVO	<i>utiliza</i>	Asociación
FranjaVO	<i>utiliza</i>	Asociación
MedicoVO	<i>utiliza</i>	Asociación
MedicoMgr	<i>utiliza</i>	Asociación
FranjaMgr	<i>utiliza</i>	Asociación
UsuarioMgr	<i>utiliza</i>	Asociación
CitaMedicaVisitor	<i>utiliza</i>	Asociación
CitaMedicaDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByFechaFranja	Método que permite consultar el total de citas médicas realizadas de acuerdo a una fecha y franja determinada.	
findByFranjaFechaIdentificacion	Método que permite acceder a la consulta de todas las citas médicas existentes de un usuario en una determinada fecha y franja.	
findByFechaIdentificacion	Método que permite acceder a la consulta de todas las citas médicas de un usuario en una determinada fecha.	
findByIdentificacion	Método que permite acceder a la consulta de todas las citas médicas de un usuario determinado	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo CitaMedicaVO para la consulta de una cita de acuerdo al código.	
findByFecha	Método que permite acceder a la consulta de todas las citas médicas en una determinada fecha.	
createCita	Método que permite establecer los atributos de un objeto de tipo CitaMedicaVisitor para el registro de una determinada cita.	



updateCitaMedica	Método que permite establecer los atributos de un objeto de tipo CitaMedicaVisitor para la actualización de una cita determinada.
deleteCita	Método que permite establecer los atributos de un objeto de tipo CitaMedicaVisitor para la eliminación de una cita determinada.
deleteCita	Método que permite establecer los atributos de un objeto de tipo CitaMedicaVisitor para la eliminación de una cita determinada de acuerdo a la fecha y franja.
<b>Observaciones:</b>	

### 11.1.5.16 Cita Médica Data Access Object

<b>Nombre de la Clase:</b> CitaMedicaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las citas médicas.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CitaMedicaVisitor	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByFechaFranja	Método que retorna el total de citas médicas realizadas de acuerdo a una fecha y franja determinada.	
findByFranjaFechaIdentificacion	Método que retorna un ArrayList de objetos de tipo CitaMedicaVO de acuerdo de un Usuario en una determinada franja en una determinada fecha.	
findByFechaIdentificacion	Método que permite retornar un ArrayList de objetos de tipo CitaMedicaVO de un determinado usuario en una fecha determinada.	
findByPrimaryKey	Método que permite retornar un objeto de tipo CitaMedicaVisitor de acuerdo al código de una determinada cita médica.	
findByFecha	Método que permite retornar un ArrayList de objetos de tipo CitaMedicaVisitor de acuerdo a una fecha determinada.	

findByIdentificacion	Método que permite retornar un ArrayList de objetos de tipo CitaMedicaVO de acuerdo a un determinado usuario.
findByTotalCitas	Método que permite retornar el número de citas en una determinada fecha y franja.
createCita	Método que permite crear una Cita Médica.
updateCitaMedica	Método que permite actualizar una Cita Médica.
deleteCita	Método que permite eliminar una Cita Médica
deleteCitaFechaFranja	Método que permite eliminar una Cita Médica en una fecha y franja determinada.
<b>Observaciones:</b>	

### 11.1.5.17 Cita Odontológica Value Object

<b>Nombre de la Clase:</b> CitaOdontologicaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las citas Odontológicas.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codCita	Identifica de forma única un cita odontológica	
String str_fecCita	Cadena que almacena la fecha de la cita odontológica	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	tieneRegistrada	Asociación
usuarioVO	esAsignada	Asociación
medicoVO	esAtendida	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
CitaOdontologicaVO	Método constructor con parámetros y sin parámetros	
getCodigoCita	Retorna el código de la cita odontológica	
getFechaCita	Retorna la fecha de la cita odontológica	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getMedicoVO	Retorna un objeto de tipo MedicoVO	
setCodigoCita	Permite asignar o cambiar el código de la cita odontológica	
setFechaCita	Permite asignar o cambiar la fecha de la cita odontológica	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	
setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO	

setMedicoVO	Permite asignar o cambiar un objeto de tipo MedicoVO
<b>Observaciones:</b>	

### 11.1.5.18 Cita Odontológica Visitor

<b>Nombre de la Clase:</b> CitaOdontologicaVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de las citas odontológicas. Hace referencia directa con cada uno de los campos de la tabla citasodontológicas.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codCita	Identifica el código de cualquier día de la semana
String str_fecCita	Identifica la fecha de cita odontológica
int int_codFranja	Identifica el código de la franja
String str_ideUsuario	Determina la identificación del usuario
String str_ideMedico	Identifica la identificación del médico
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
CitaOdontologicaVisitor	Método constructor con parámetros y sin parámetros
getCodigoCita	Retorna el código de la cita
getFechaCita	Retorna la fecha de la cita
getCodigoFranja	Retorna el código de la franja
getIdentificacionUsuario	Retorna la identificación del usuario
getIdentificacionMedico	Retorna la identificación del médico
setCodigoCita	Permite asignar o cambiar el código de la cita odontológica
setFechaCita	Permite asignar o cambiar la fecha de la cita odontológica
setCodigoFranja	Permite asignar o cambiar el código de la franja
setIdentificacionUsuario	Permite asignar o cambiar la identificación del usuario
setIdentificacionMedico	Permite asignar o cambiar la identificación del médico
<b>Observaciones:</b>	

### 11.1.5.19 Cita Odontológica Manager

<b>Nombre de la Clase:</b> CitaOdontologicaMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las citas odontológicas. Tiene toda la lógica de la clase CitaOdontologicaVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_citaOdontologica	Arraylist (colección de obj) que almacena los datos de las citas odontológicas como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CitaOdontologicaVO	<i>retorna</i>	Asociación
UsuarioVO	<i>utiliza</i>	Asociación
FranjaVO	<i>utiliza</i>	Asociación
MedicoVO	<i>utiliza</i>	Asociación
MedicoMgr	<i>utiliza</i>	Asociación
FranjaMgr	<i>utiliza</i>	Asociación
UsuarioMgr	<i>utiliza</i>	Asociación
CitaOdontologicaVisitor	<i>utiliza</i>	Asociación
CitaOdontologicaDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByFechaFranja	Método que permite consultar el total de citas odontológicas realizadas de acuerdo a una fecha y franja determinada.	
findByFranjaFechaIdentificacion	Método que permite acceder a la consulta de todas las citas odontológicas existentes de un usuario en una determinada fecha y franja.	
findByFechaIdentificacion	Método que permite acceder a la consulta de todas las citas odontológicas de un usuario en una determinada fecha.	
findByIdentificacion	Método que permite acceder a la consulta de todas las citas odontológicas de un usuario determinado	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo CitaOdontologicaVO para la consulta de una cita de acuerdo al código.	
findByFecha	Método que permite acceder a la consulta de todas las citas odontológicas en una determinada fecha.	
createCita	Método que permite establecer los atributos de un objeto de tipo CitaOdontologicaVisitor para el registro de una determinada cita.	

updateCitaMedica	Método que permite establecer los atributos de un objeto de tipo CitaOdontologicaVisitor para la actualización de una cita determinada.
deleteCita	Método que permite establecer los atributos de un objeto de tipo CitaOdontologicaVisitor para la eliminación de una cita determinada.
deleteCita	Método que permite establecer los atributos de un objeto de tipo CitaOdontologicaVisitor para la eliminación de una cita determinada de acuerdo a la fecha y franja.
<b>Observaciones:</b>	

### 11.1.5.20 Cita Odontológica Data Access Object

<b>Nombre de la Clase:</b> CitaOdontologicaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las citas odontológicas.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CitaOdontologicaVisitor	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByFechaFranja	Método que retorna el total de citas odontológicas realizadas de acuerdo a una fecha y franja determinada.	
findByFranjaFechaIdentificacion	Método que retorna un ArrayList de objetos de tipo CitaOdontologicaVO de acuerdo de un Usuario en una determinada franja en una determinada fecha.	
findByFechaIdentificacion	Método que permite retornar un ArrayList de objetos de tipo CitaOdontologicaVO de un determinado usuario en una fecha determinada.	
findByPrimaryKey	Método que permite retornar un objeto de tipo CitaOdontologicaVisitor de acuerdo al código de una determinada cita odontológica.	
findByFecha	Método que permite retornar un ArrayList de objetos de tipo CitaOdontologicaVisitor de acuerdo a una fecha determinada.	

findByIdentificacion	Método que permite retornar un ArrayList de objetos de tipo CitaOdontologicaVO de acuerdo a un determinado usuario.
findByTotalCitas	Método que permite retornar el número de citas en una determinada fecha y franja.
createCita	Método que permite crear una Cita odontológica.
updateCitaMedica	Método que permite actualizar una Cita odontológica.
deleteCita	Método que permite eliminar una Cita odontológica
deleteCitaFechaFranja	Método que permite eliminar una Cita odontológica en una fecha y franja determinada.
<b>Observaciones:</b>	

### 11.1.5.21 Cita Psicológica Value Object

<b>Nombre de la Clase:</b> CitaPsicologicaVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las citas psicológicas.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codCita	Identifica de forma única un cita psicológica	
String str_fecCita	Cadena que almacena la fecha de la cita psicológica	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	tieneRegistrada	Asociación
usuarioVO	esAsignada	Asociación
medicoVO	esAtendida	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
CitaPsicologicaVO	Método constructor con parámetros y sin parámetros	
getCodigoCita	Retorna el código de la cita psicológica	
getFechaCita	Retorna la fecha de la cita psicológica	
getFranjaVO	Retorna un objeto de tipo FranjaVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getMedicoVO	Retorna un objeto de tipo MedicoVO	
setCodigoCita	Permite asignar o cambiar el código de la cita psicológica	
setFechaCita	Permite asignar o cambiar la fecha de la cita psicológica	
setFranjaVO	Permite asignar o cambiar un objeto de tipo FranjaVO	

setUsuarioVO	Permite asignar o cambiar un objeto de tipo UsuarioVO
setMedicoVO	Permite asignar o cambiar un objeto de tipo MedicoVO
<b>Observaciones:</b>	

### 11.1.5.22 Cita Psicológica Visitor

<b>Nombre de la Clase:</b> CitaPsicologicaVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos de las citas psicológicas. Hace referencia directa con cada uno de los campos de la tabla citaspsicologicas.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codCita	Identifica el código de cualquier día de la semana
String str_fecCita	Identifica la fecha de cita psicológica
int int_codFranja	Identifica el código de la franja
String str_ideUsuario	Determina la identificación del usuario
String str_ideMedico	Identifica la identificación del médico
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
CitaOdontologicaVisitor	Método constructor con parámetros y sin parámetros
getCodigoCita	Retorna el código de la cita
getFechaCita	Retorna la fecha de la cita
getCodigoFranja	Retorna el código de la franja
getIdentificacionUsuario	Retorna la identificación del usuario
getIdentificacionMedico	Retorna la identificación del médico
setCodigoCita	Permite asignar o cambiar el código de la cita psicológica
setFechaCita	Permite asignar o cambiar la fecha de la cita psicológica
setCodigoFranja	Permite asignar o cambiar el código de la franja
setIdentificacionUsuario	Permite asignar o cambiar la identificación del usuario
setIdentificacionMedico	Permite asignar o cambiar la identificación del médico
<b>Observaciones:</b>	

### 11.1.5.23 Cita Psicológica Manager

<b>Nombre de la Clase:</b> CitaPsicologicaMgr		
<b>Descripción de la clase</b> Clase que permite la administración de las citas psicológicas. Tiene toda la lógica de la clase CitaPsicologicaVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_citaPsicologica	Arraylist (colección de obj) que almacena los datos de las citas psicológicas como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CitaPsicologicaVO	<i>retorna</i>	Asociación
UsuarioVO	<i>utiliza</i>	Asociación
FranjaVO	<i>utiliza</i>	Asociación
MedicoVO	<i>utiliza</i>	Asociación
MedicoMgr	<i>utiliza</i>	Asociación
FranjaMgr	<i>utiliza</i>	Asociación
UsuarioMgr	<i>utiliza</i>	Asociación
CitaPsicologicaVisitor	<i>utiliza</i>	Asociación
CitaPsicologicaDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByFechaFranja	Método que permite consultar el total de citas psicológicas realizadas de acuerdo a una fecha y franja determinada.	
findByFranjaFechaIdentificacion	Método que permite acceder a la consulta de todas las citas psicológicas existentes de un usuario en una determinada fecha y franja.	
findByFechaIdentificacion	Método que permite acceder a la consulta de todas las citas psicológicas de un usuario en una determinada fecha.	
findByIdentificacion	Método que permite acceder a la consulta de todas las citas psicológicas de un usuario determinado	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo CitaPsicologicaVO para la consulta de una cita de acuerdo al código.	
findByFecha	Método que permite acceder a la consulta de todas las citas psicológicas en una determinada fecha.	
createCita	Método que permite establecer los atributos de un objeto de tipo CitaPsicologicaVisitor para el registro de una determinada cita.	
updateCitaMedica	Método que permite establecer los atributos de un objeto de tipo CitaPsicologicaVisitor para la actualización de una cita determinada.	



deleteCita	Método que permite establecer los atributos de un objeto de tipo CitaPsicologicaVisitor para la eliminación de una cita determinada.
deleteCita	Método que permite establecer los atributos de un objeto de tipo CitaPsicologicaVisitor para la eliminación de una cita determinada de acuerdo a la fecha y franja.
<b>Observaciones:</b>	

#### 11.1.5.24 Cita Psicológica Data Access Object

<b>Nombre de la Clase:</b> CitaPsicologicaDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las citas psicológicas.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
CitaPsicologicaVisitor	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByFechaFranja	Método que retorna el total de citas psicológicas realizadas de acuerdo a una fecha y franja determinada.	
findByFranjaFechaIdentificacion	Método que retorna un ArrayList de objetos de tipo CitaPsicologicaVO de acuerdo de un Usuario en una determinada franja en una determinada fecha.	
findByFechaIdentificacion	Método que permite retornar un ArrayList de objetos de tipo CitaPsicologicaVO de un determinado usuario en una fecha determinada.	
findByPrimaryKey	Método que permite retornar un objeto de tipo CitaPsicologicaVisitor de acuerdo al código de una determinada cita psicológica.	
findByFecha	Método que permite retornar un ArrayList de objetos de tipo CitaPsicologicaVisitor de acuerdo a una fecha determinada.	
findByIdentificacion	Método que permite retornar un ArrayList de objetos de tipo CitaPsicologicaVO de acuerdo a un determinado usuario.	

findByTotalCitas	Método que permite retornar el número de citas en una determinada fecha y franja.
createCita	Método que permite crear una Cita psicológica.
updateCitaMedica	Método que permite actualizar una Cita odontológica.
deleteCita	Método que permite eliminar una Cita psicológica
deleteCitaFechaFranja	Método que permite eliminar una Cita psicológica en una fecha y franja determinada.
<b>Observaciones:</b>	

## 11.1.6 Inventario

### 11.1.6.1 Inventario Value Object

<b>Nombre de la Clase:</b> InventarioVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los elementos existentes en la Universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codElemento	Identifica de manera única un Elemento de Inventario	
String str_desElemento	Indica la descripción de un elemento de Inventario.	
String str_fechaIngreso	Determina la fecha en que se ingresa un determinado elemento.	
String str_modElemento	Indica el modelo de un elemento de Inventario.	
String str_serElemento	Indica el serial de un elemento de Inventario.	
String str_actEntrega	Indica el número del Acta de Entrega de un elemento de Inventario.	
String str_observaciones	Permite registrar anotaciones propias de un determinado elemento.	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoElementoVO	tieneAsignado	Asociación
EstadoElementoVO	tieneUn	Composición
UbicacionVO	estaEn	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
InventarioVO	Método constructor sin parámetros	
getCodigoElemento	Retorna el código del elemento	
getDescripcionElemento	Retorna la descripción del elemento	
getFechaIngreso	Retorna la fecha en que en la que ingresa un elemento	
getTipoElementoVO	Retorna un objeto de tipo TipoElementoVO	

getMarcaElemento	Retorna la marca del elemento
getModeloElemento	Retorna el modelo del elemento
getSerialElemento	Retorna el serial del elemento
getEstadoElementoVO	Retorna un objeto de tipo EstadoElementoVO
getActaEntrega	Retorna el acta de entrega del elemento
getUbicacionVO	Retorna un objeto de tipo UbicacionVO
getObservaciones	Retorna las observaciones de entrada de un elemento
setCodigoElemento	Permite asignar el código del elemento
setDescripcionElemento	Permite asignar o cambiar la descripción del elemento
setFechaIngreso	Permite asignar o cambiar la fecha de ingreso
setTipoElementoVO	Permite asignar o cambiar un objeto TipoElementoVO
setMarcaElemento	Permite asignar o cambiar la marca del elemento
setModeloElemento	Permite asignar o cambiar el modelo del elemento
setSerialElemento	Permite asignar o cambiar el serial del elemento
setEstadoElementoVO	Permite asignar o cambiar un objeto EstadoElementoVO
setActaEntrega	Permite asignar o cambiar el acta de entrega del elemento
setUbicacionVO	Permite asignar o cambiar un objeto UbicacionVO
setObservaciones	Permite asignar o cambiar las observaciones de un elemento
<b>Observaciones:</b>	

### 11.1.6.2 Inventario Visitor

<b>Nombre de la Clase:</b> InventarioVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica un objeto de tipo InventarioVisitor, hace referencia directa con cada uno de los campos de la tabla inventario	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codElemento	Identifica de manera única un elemento de inventario
String str_desElemento	Determina la descripción de un elemento
String str_fecIngreso	Cadena que almacena la fecha en la cual ingresa un elemento al inventario
int int_codTipo	Identifica el código del tipo de elemento
String str_marElemento	Determina la marca de un elemento
String str_modElemento	Determina el modelo de un elemento
String str_serElemento	Determina el serial de un elemento
int int_codEstado	Identifica el código del estado del elemento

String str_actEntrega	Determina el acta de entrega de un elemento.
int int_codUbicacion	Identifica el código de la ubicación
String str_observaciones	Determina las observaciones realizadas a un elemento
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
InventarioVisitor	Método constructor sin parámetros
getCodigoElemento	Retorna el código del elemento
getDescripcionElemento	Retorna la descripción del elemento
getFechaIngreso	Retorna la fecha en que ingresa un elemento al inventario
getTipoElementoVO	Retorna el código del tipo de elemento asignado
getMarcaElemento	Retorna la marca del elemento
getModeloElemento	Retorna el modelo del elemento
getSerialElemento	Retorna el serial del elemento
getEstadoElementoVO	Retorna el código del estado del elemento
getActaEntrega	Retorna el acta de entrega del elemento
getUbicacionVO	Retorna el código de la ubicación asignada a un elemento de inventario
getObservaciones	Retorna las observaciones
setCodigoElemento	Permite asignar el código del elemento
setDescripcionElemento	Permite asignar o cambiar la descripción del elemento
setFechaIngreso	Permite asignar o cambiar la fecha en que ingresa un elemento
setTipoElementoVO	Permite asignar o cambiar el código del tipo de elemento
setMarcaElemento	Permite asignar o cambiar la marca del elemento
setModeloElemento	Permite asignar o cambiar el modelo del elemento
setSerialElemento	Permite asignar o cambiar el serial del elemento
setEstadoElementoVO	Permite asignar o cambiar el código del estado de elemento
setActaEntrega	Permite asignar o cambiar el acta de entrega del elemento
setUbicacionVO	Permite asignar o cambiar el código de la ubicación asignada a un elemento
setObservaciones	Permite asignar o cambiar las observaciones del elemento
<b>Observaciones:</b>	

### 11.1.6.3 Inventario Manager

<b>Nombre de la Clase:</b> InventarioMgr
---

<b>Descripción de la clase</b>		
Clase que permite la administración de los elementos de inventario. Tiene toda la lógica de la clase InventarioVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_elementoInventario	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
InventarioVO	retorna	Asociación
TipoElementoVO	utiliza	Asociación
EstadoElementoVO	utiliza	Asociación
UbicacionVO	utiliza	Asociación
InventarioVisitor	utiliza	Asociación
TipoElementoMgr	utiliza	Asociación
EstadoElementoMgr	utiliza	Asociación
UbicacionMgr	utiliza	Asociación
InventarioDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo InventarioVO para la consulta de un elemento de acuerdo a un código	
findByTipoEstado	Método que permite establecer la cantidad de elementos existentes de acuerdo a un tipo y estado específico	
FindByTipo	Método que retorna la cantidad de elementos existentes de acuerdo a un tipo específico	
findByTipoEstado	Método que permite acceder a la consulta de todos los elementos de inventario de acuerdo a un determinado tipo y estado de elemento	
findByUbicacionTipo	Método que permite acceder a la consulta de todos los elementos de inventario de acuerdo a una determinada ubicación y estado	
findByTipoEstadoAdecuado	Método que permite acceder a la consulta de todos los elementos de inventario de acuerdo a un determinado tipo de elemento	
FindByUbicacion	Método que retorna un ArrayList de objetos tipo InventarioVO de acuerdo a una determinada Ubicación	
findByTipoElemento	Método que retorna un ArrayList de objetos tipo InventarioVO de acuerdo a un determinado Tipo de Elemento	
findByActaEntrega	Método que retorna un ArrayList de objetos tipo InventarioVO de acuerdo a una determinada Acta de Entrega	

findByFechaIngreso	Método que retorna un ArrayList de objetos tipo InventarioVO de acuerdo a una determinada Fecha de Ingreso
updateUbicacionElemento	Método que permite establecer los atributos de un objeto de tipo InventarioVisitor para la actualización de su ubicación
updateEstadoElemento	Método que permite establecer los atributos de un objeto de tipo InventarioVisitor para la actualización del estado del elemento
CreateElemento	Método que permite establecer los atributos de un objeto de tipo InventarioVisitor para la creación de un elemento de inventario
UpdateElemento	Método que permite establecer los atributos de un objeto de tipo InventarioVisitor para la actualización de un determinado elemento de inventario
DeleteElemento	Método que permite establecer los atributos de un objeto de tipo InventarioVisitor para la eliminación de un determinado elemento de Inventario
<b>Observaciones:</b>	

#### 11.1.6.4 Inventario Data Access Object

<b>Nombre de la Clase:</b> InventarioDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los elementos de inventario		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
InventarioVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Retorna un objeto de tipo InventarioVisitor de acuerdo al código de la reserva	
findByTipoEstadoUtil	Método que retorna la cantidad de elementos existentes de acuerdo a un tipo y estado correspondiente a 1 (bueno) o 2 (regular)	
FindByTipo	Método que retorna la cantidad de elementos existentes de acuerdo a un tipo específico	

findByTipoEstado	Método que retorna la cantidad de elementos existentes de acuerdo a un tipo y estado específico
findByUbicacionTipo	Método que retorna un ArrayList con objetos de InventarioVisitor de acuerdo a una determinada ubicación y tipo de elemento
findByTipoEstadoAdecuado	Método que retorna un ArrayList con objetos de tipo InventarioVisitor de undeterminado tipo de elemento que se encuentren en estado bueno o regular
findByUbicacion	Método que retorna un ArrayList de objetos tipo InventarioVisitor de acuerdo a una determinada Ubicación
findByTipoElemento	Método que retorna un ArrayList de objetos tipo InventarioVisitor de acuerdo a un determinado Tipo de Elemento
findByActaEntrega	Método que retorna un ArrayList de objetos tipo InventarioVisitor de acuerdo a una determinada Acta de Entrega
findByFechaIngreso	Método que retorna un ArrayList de objetos tipo InventarioVisitor de acuerdo a una determinada Fecha de Ingreso
updateUbicacionElemento	Método que permite actualizar la ubicación de un determinado elemento de Inventario.
updateEstadoElemento	Método que permite actualizar el estado de un determinado elemento de Inventario
CreateElemento	Método que permite crear un elemento de Inventario
updateElemento	Método que permite actualizar un determinado elemento de Inventario.
DeleteElemento	Método que permite eliminar un determinado elemento
<b>Observaciones:</b>	

#### 11.1.6.5 Tipo Elemento Value Object

<b>Nombre de la Clase:</b> TipoElementoVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica la clasificación de los elementos existentes en un tipo determinado		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codTipo	Identifica de manera única un tipo de elemento	
String str_desTipo	Determina la descripción de tipo de elemento	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>

<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
TipoElementoVO	Método constructor con parámetros y sin parámetros
GetCodigo	Retorna el código del tipo de elemento
getDescripcion	Retorna la descripción del tipo de elemento
SetCodigo	Permite asignar el código del tipo de elemento
setDescripcion	Permite asignar o cambiar la descripción del tipo de elemento
<b>Observaciones:</b>	

### 11.1.6.6 Tipo Elemento Manager

<b>Nombre de la Clase:</b> TipoElementoMgr		
<b>Descripción de la clase</b> Clase que permite administrar los tipos de elementos existentes. Tiene toda la lógica de la clase TipoElementoVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoElementoVO	Retorna	Asociación
TipoElementoDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo TipoElementoVO para la consulta de un tipo de elemento de acuerdo a un código	
findByDescripcion	Método que permite acceder a la consulta de todos los elementos existentes en inventario de acuerdo a una determinada descripción	
createTipoElemento	Método que permite establecer los atributos de un objeto de tipo TipoElementoVO para la creación de un determinado tipo de elemento	
deleteTipoElemento	Método que permite establecer los atributos de un objeto de tipo TipoElementoVO para la eliminación de un determinado tipo de elemento	
updateTipoElemento	Método que permite establecer los atributos de un objeto de tipo TipoElementoVO para la actualización de un determinado tipo de elemento	
<b>Observaciones:</b>		



### 11.1.6.7 Tipo Elemento Data Access Object

<b>Nombre de la Clase:</b> TipoElementoDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los Tipos de Elementos existentes.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoElementoVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo TipoElementoVO de acuerdo a un determinado código	
findByDescripcion	Método que retorna un ArrayList con objetos de tipo TipoElementoVO	
FindByAll	Método que retorna un ArrayList con todos los objetos de tipo TipoElementoVO	
findByAllAlfabetico	Método que retorna un ArrayList con todos los objetos de tipo TipoElementoVO ordenado alfabéticamente por la descripción del tipo	
createTipoElemento	Método que permite crear un tipo de elemento	
deleteTipoElemento	Método que permite eliminar un determinado Tipo de Elemento	
updateTipoElemento	Método que permite actualizar un determinado Tipo de Elemento	
<b>Observaciones:</b>		

### 11.1.6.8 Estado Elemento Value Object

<b>Nombre de la Clase:</b> EstadoElementoVO	
<b>Descripción de la clase</b> Clase que representa de forma lógica los estados de los elementos registrados en el inventario	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codEstado	Identifica de manera única un estado de elemento
String str_desEstado	Determina la descripción de estado de elemento

Clase Asociada	Nombre Asociación	Rol Clase Asociada
Servicio o Responsabilidad	Descripción	
EstadoElementoVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código del estado de elemento	
getDescripcion	Retorna la descripción del estado de elemento	
setCodigo	Permite asignar el código del estado de elemento	
setDescripcion	Permite asignar o cambiar la descripción del estado de elemento	
<b>Observaciones:</b>		

### 11.1.6.9 Estado Elemento Manager

<b>Nombre de la Clase:</b> EstadoElementoMgr		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los estados elementos Tiene toda la lógica de la clase EstadoElementoVO a nivel de modelo.		
Propiedad	Descripción del modelaje, restricciones, información para diseño, suposiciones	
ArrayList al_aux	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
Clase Asociada	Nombre Asociación	Rol Clase Asociada
EstadoElementoVO	Retorna	Asociación
EstadoElementoDAO	utiliza	Asociación
Servicio o Responsabilidad	Descripción	
findByPrimaryKey	Método que retorna un objeto de tipo EstadoElementoVO	
findAll	Método que retorna un ArrayList con todos los objetos de tipo EstadoElementoVO	
<b>Observaciones:</b>		

### 11.1.6.10 Estado Elemento Data Access Object

<b>Nombre de la Clase:</b> EstadoElementoDAO
---

<b>Descripción de la clase</b>		
Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los estados elementos.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EstadoElementoVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo EstadoElementoVO	
FindByAll	Método que retorna un ArrayList con todos los objetos de tipo EstadoElementoVO	
<b>Observaciones:</b>		

#### 11.1.6.11 Tipo Ubicación Value Object

<b>Nombre de la Clase:</b>		
TipoUbicacionVO		
<b>Descripción de la clase</b>		
Clase que representa de forma lógica los tipos de ubicación o instalaciones físicas		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codTipo	Identifica de manera única un tipo de ubicación	
String str_desTipo	Determina la descripción de un tipo de ubicación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
UbicacionVO	Método constructor con parámetros y sin parámetros	
setCodigo	Retorna el código del tipo de ubicación	
getDescripcion	Retorna la descripción del tipo de ubicación	
setCodigo	Permite asignar el código del tipo de ubicación	
setDescripcion	Permite asignar o cambiar la descripción del tipo de ubicación	
<b>Observaciones:</b>		

#### 11.1.6.12 Tipo Ubicación Manager

<b>Nombre de la Clase:</b> TipoUbicacionMgr		
<b>Descripción de la clase</b> Clase que permite administrar los tipos de ubicación existentes. Tiene toda la lógica de la clase TipoUbicacionVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoUbicacionVO	Retorna	Asociación
TipoUbicacionDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo TipoUbicacionVO para la consulta de un Tipo de Ubicación de acuerdo a un código	
FindByAll	Método que permite acceder a la consulta de todos los tipos de ubicaciones existentes	
<b>Observaciones:</b>		

### 11.1.6.13 Tipo Ubicación Data Access Object

<b>Nombre de la Clase:</b> TipoUbicacionDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los tipos de ubicaciones.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoUbicacionVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo TipoUbicacionVO con la descripción de la ubicación de acuerdo a un código determinado	
FindByAll	étodo que retorna un ArrayList de objetos de tipo TipoUbicacionVO con todos sus atributos	

**Observaciones:**

#### 11.1.6.14 Ubicación Value Object

<b>Nombre de la Clase:</b> UbicacionVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las instalaciones físicas involucradas en los procesos de la aplicación. Ejem. Aulas, almacén, audiovisuales, secretaria, etc...		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codUbicacion	Identifica de manera única una ubicación	
String str_nomUbicacion	Cadena que almacena el nombre de la ubicación	
String str_desUbicacion	Cadena que almacena la descripción de la ubicación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
TipoUbicacionVO	tieneAsignado	Asociación
<b>Servicio o Responsabilidad</b>		<b>Descripción</b>
UbicacionVO		Método constructor sin parámetros
getCodigo		Retorna el código de la ubicación
getNombre		Retorna el nombre de la ubicación
getDescripcion		Retorna la descripción de la ubicación
getTipoUbicacionVO		Retorna un objeto de tipo TipoUbicacionVO
setCodigo		Permite asignar el código de la ubicación
setNombre		Permite asignar o cambiar el nombre de la ubicación
setDescripcion		Permite asignar o cambiar la descripción de la ubicación
setTipoUbicacionVO		Permite asignar o cambiar un objeto de tipo TipoUbicacionVO
<b>Observaciones:</b>		

#### 11.1.6.15 Ubicación Visitor

<b>Nombre de la Clase:</b> UbicacionVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica un objeto de tipo UbicacionVisitor, hace referencia directa con cada uno de los campos de la tabla ubicacion	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>

int int_codUbicacion	Identifica de manera única una ubicación
String str_nomUbicacion	Cadena que almacena el nombre de la ubicación
String str_desUbicacion	Cadena que almacena la descripción de la ubicación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
UbicacionVisitor	Método constructor sin parámetros
getCodigo	Retorna el código de la ubicación
getNombre	Retorna el nombre de la ubicación
getDescripcion	Retorna la descripción de la ubicación
getTipo	Retorna el tipo de ubicación
setCodigo	Permite asignar el código de la ubicación
SetNombre	Permite asignar o cambiar el nombre de la ubicación
setDescripcion	Permite asignar o cambiar la descripción de la ubicación
SetTipo	Permite asignar o cambiar el tipo de ubicación
<b>Observaciones:</b>	

#### 11.1.6.16 Ubicación Manager

<b>Nombre de la Clase:</b> UbicacionMgr		
<b>Descripción de la clase</b> Clase que permite administrar las ubicación existentes. Tiene toda la lógica de la clase UbicacionVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_ubicacion	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UbicacionVO	retorna	Asociación
TipoUbicacionVO	tieneAsignadoUn	Composición
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
FindByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo UbicacionVisitor para la consulta de una ubicación de acuerdo a un código	
FindByTipo	Método que permite acceder a la consulta de todas las ubicaciones de un determinado tipo	
findByTotal	Método que permite acceder a la consulta del número de ubicaciones de un determinado tipo	

findByNombreTipo	Método que permite establecer los atributos de un objeto de tipo UbicacionVisitor para la consulta de una ubicación de acuerdo a un nombre y tipo determinado
findByNombre	Método que permite acceder a la consulta de todas las ubicaciones de acuerdo a un determinado nombre
FindByAll	Método que permite acceder a la consulta de todas las ubicaciones existentes
createUbicacion	Método que permite establecer los atributos de un objeto de tipo UbicacionVisitor para el registro de una ubicación
updateUbicacion	Método que permite establecer los atributos de un objeto de tipo UbicacionVisitor para la actualización de una ubicación
deleteUbicacion	Método que permite establecer los atributos de un objeto de tipo UbicacionVisitor para la eliminación de una ubicación
<b>Observaciones:</b>	

#### 11.1.6.17 Ubicación Data Access Object

<b>Nombre de la Clase:</b> UbicacionDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las ubicaciones		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UbicacionVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
FindByPrimaryKey	Método que retorna un objeto de tipo UbicacionVisitor con todos sus atributos de acuerdo a un código de ubicación	
findByTipo	Método que retorna un ArrayList con todas las ubicaciones existentes de acuerdo a un determinado tipo de ubicación	

findByTotal	Método que devuelve el número de elementos existentes de acuerdo a un determinado tipo de ubicación
findByNombreTipo	Método que retorna un objeto de tipo UbicacionVisitor con todos sus atributos de acuerdo a un determinado nombre y tipo de ubicación
findByNombre	Método que retorna un ArrayList de objetos de tipo UbicacionVisitor con todos sus atributos de acuerdo a un determinado nombre de ubicación
FindByAll	Método que retorna un ArrayList con todos los objetos de tipo UbicacionVisitor.
createUbicacion	Método que permite crear una nueva Ubicación
createReserva	Método que permite crear una reserva de internet
deleteUbicacion	Método que permite eliminar una determinada Ubicación
updateUbicacion	Método que permite actualizar una determinada Ubicación.
<b>Observaciones:</b>	

## 11.1.7 Administración de Computadores

### 11.1.7.1 Computador Value Object

<b>Nombre de la Clase:</b> ComputadorVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica un objeto de tipo ComputadorVO		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
String str_codComputador	Identifica de manera única un computador	
String str_fechaIngreso	Cadena que almacena la fecha en la cual ingresa un computador	
String str_actEntrega	Cadena que permite almacenar el acta de entrega de un computador	
String str_internet	Determina si el computador es para internet	
String str_audiovisual	Determina si el computador es para audiovisuales	
String str_observaciones	Cadena que permite almacenar las observaciones hechas a un computador	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
UbicacionVO	estaEn	Asociación
EstadoElementoVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ComputadorVO	Método constructor con parámetros y sin parámetros	



getCodigoComputador	Retorna el código del computador
getFechaIngreso	Retorna la fecha en que ingresó un computador
getActaEntrega	Retorna el acta de entrega
getUbicacionVO	Retorna un objeto de tipo UbicacionVO
getEstadoElementoVO	Retorna un objeto de tipo EstadoElementoVO
getInternet	Retorna si el uso es internet
getAudiovisual	Retorna si el uso es audiovisual
getObservacion	Retorna las observaciones
setCodigoComputador	Permite asignar el código del computador
setFechaIngreso	Permite asignar o cambiar la fecha de ingreso del computador
setActaEntrega	Permite asignar o cambiar el acta de entrega del computador
setUbicacionVO	Permite asignar o cambiar la un objeto de tipo UbicacionVO
SetEstadoElementoVO	Permite asignar o cambiar la un objeto de tipo EstadoElementoVO
SetInternet	Permite asignar o cambiar el uso de internet
SetAudiovisual	Permite asignar o cambiar el uso de audiovisuales
SetObservacion	Permite asignar o cambiar las observaciones del computador
<b>Observaciones:</b>	

### 11.1.7.2 Computador Visitor

<b>Nombre de la Clase:</b> ComputadorVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica un objeto de tipo ComputadorVisitor. Hace referencia directa con cada uno de los campos de la tabla computador.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_codComputador	Identifica de manera única un computador
String str_fechaIngreso	Cadena que almacena la fecha en la cual ingresa un computador
String str_actEntrega	Determina el acta de entrega de un computador
String str_internet	Determina el uso de internet del computador
String str_audiovisual	Determina el uso audiovisual del computador
String str_observacion	Permite almacenar las observaciones hechas a un computador
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ComputadorVisitor	Método constructor con parámetros y sin parámetros
getCodigoComputador	Retorna el código del computador

getFechaIngreso	Retorna la fecha en que ingresa un computador
getActaEntrega	Retorna el acta de entrega del computador
getCodigoUbicacion	Retorna el código de la ubicación en la que está un computador
getCodigoEstado	Retorna el código del estado asignado al computador
getInternet	Retorna el uso de internet de un computador
getAudiovisual	Retorna el uso audiovisual de un computador
getObservacion	Retorna las observaciones de un computador
setCodigoComputador	Permite asignar el código del computador
setFechaIngreso	Permite asignar o cambiar la fecha de ingreso del computador
setActaEntrega	Permite asignar o cambiar el acta de entrega del computador
setCodigoUbicacion	Permite asignar o cambiar el código de la ubicación
setCodigoEstado	Permite asignar o cambiar el código del estado
setInternet	Permite asignar o cambiar el uso de internet de un computador
setAudiovisual	Permite asignar o cambiar el uso audiovisual de un computador
setObservacion	Permite asignar o cambiar las observaciones de un computador
<b>Observaciones:</b>	

### 11.1.7.3 Computador Manager

<b>Nombre de la Clase:</b> ComputadorMgr		
<b>Descripción de la clase</b> Clase que permite la administración de los computadores existentes. Tiene toda la lógica de la clase ComputadorVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_computador	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ComputadorVO	retorna	Asociación
UbicacionVO	estaEn	Asociación
EstadoElementoVO	tiene	Asociación
ComputadorVisitor	utiliza	Asociación
UbicacionMgr	utiliza	Asociación
EstadoElementoMgr	utiliza	Asociación

ReservaInternetDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ComputadorVisitor para la consulta de un Computador de acuerdo a un código	
findByUbicacionInternet	Método que permite acceder a la consulta de todos los computadores asignados para el servicio de internet que se encuentran en una determinada ubicación	
findByEstado	Método que permite acceder a la consulta de todos los computadores de acuerdo a un determinado estado de elemento	
findByEstadoUbicacion	Método que permite acceder a la consulta de todos los computadores de acuerdo a una determinada ubicación y un determinado estado	
findByUbicacionEstado	Método que permite acceder a la consulta de todos los computadores que se encuentran en una determinada ubicación y cuyo estado sea bueno o regular	
findByAudiovisual	Método que permite acceder a la consulta del número de computadores asignados para el servicio audiovisual cuyo estado sea bueno o regular	
findByTrabajoGeneral	Método que permite acceder a la consulta de todos los computadores asignados para el servicio de trabajos varios cuyo estado sea bueno o regular	
findByEquipoInternet	Método que permite acceder a la consulta de todos los computadores asignados para el servicio de internet cuyo estado sea bueno o regular	
findByEquipoAudiovisual	Método que permite acceder a la consulta de todos los computadores asignados para el servicio audiovisual cuyo estado sea bueno o regular	
findByUbicacionInternetEstado	Método que permite acceder a la consulta de todos los computadores que no estén asignados para internet cuyo estado sea bueno o regular y se encuentren en una ubicación determinada	
findByUbicacion	Método que permite acceder a la consulta de todos los computadores que se encuentran en una ubicación determinada	
findByFechaIngreso	Método que retorna un ArrayList de objetos tipo ComputadorVO de acuerdo a una determinada Fecha de Ingreso	
findByActaEntrega	Método que retorna un ArrayList de objetos tipo ComputadorVisitor de acuerdo a un determinado número de Acta de Entrega	
createComputador	Método que permite establecer los atributos de un objeto de tipo ComputadorVisitor para el registro de un computador	

updateComputador	Método que permite establecer los atributos de un objeto de tipo ComputadorVisitor para la actualización de un computador
deleteComputador	Método que permite establecer los atributos de un objeto de tipo ComputadorVisitor para la eliminación de un computador
<b>Observaciones:</b>	

#### 11.1.7.4 Computador Data Access Object

<b>Nombre de la Clase:</b> ComputadorDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los computadores existentes.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ComputadorVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ComputadorVisitor de acuerdo a un código de computador determinado	
findByUbicacionInternet	Método que retorna un ArrayList con objetos de ComputadorVisitor de a una determinada ubicación y cuyo uso sea internet	
findByEstado	Método que retorna un ArrayList de objetos tipo ComputadorVisitor de acuerdo a un determinado estado de elemento	
findByEstadoUbicacion	Método que retorna un ArrayList con objetos de ComputadorVisitor de a una determinada ubicación y un determinado estado	
findByUbicacionEstado	Método que retorna un ArrayList con objetos de ComputadorVisitor de a una determinada ubicación y un determinado estado	
findByAudiovisual	Método que permite determinar la cantidad de computadores asignados para el servicio de audiovisuales y que esten en buen estado	

findByTrabajoGeneral	Método que permite retornar un ArrayList con todos los atributos de los computadores asignados para trabajos varios
findByEquipoInternet	Método que permite retornar un ArrayList con todos los atributos de los computadores asignados para el servicio de internet y que estén en estado adecuado para prestar el servicio
findByEquipoAudiovisual	Método que permite retornar un ArrayList con todos los atributos de los computadores asignados para el servicio audiovisual y que estén en estado adecuado para prestar el servicio
findByUbicacionInternetEstado	Método que permite retornar un ArrayList con todos los atributos de los computadores que se encuentren en una determinada ubicación y que estén en estado bueno o regular
findByUbicacion	Método que retorna un ArrayList de objetos tipo ComputadorVisitor de una determinada Ubicación
findByFechaIngreso	Método que retorna un ArrayList de objetos tipo ComputadorVisitor de acuerdo a una determinada Fecha de Ingreso
findByActaEntrega	Método que retorna un ArrayList de objetos tipo ComputadorVisitor de acuerdo a un determinado número de Acta de Entrega
createComputador	Método que permite crear un Computador
updateComputador	Método que permite actualizar la información de un determinado Computador
deleteComputador	Método que permite eliminar un determinado Computador
<b>Observaciones:</b>	

#### 11.1.7.5 Elemento Computador Value Object

<b>Nombre de la Clase:</b> ElementoComputadorVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica un objeto de tipo ElementoComputadorVO		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ComputadorVO	pertenece	Asociación
InventarioVO	pertenece	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ElementoComputadorVO	Método constructor sin parámetros	

getComputadorVO	Retorna un objeto de tipo ComputadorVO
getInventarioVO	Retorna un objeto de tipo InventarioVO
setComputadorVO	Permite asignar o cambiar la un objeto de tipo ComputadorVO
setInventarioVO	Permite asignar o cambiar la fecha de ingreso del InventarioVO
<b>Observaciones:</b>	

### 11.1.7.6 Elemento Computador Visitor

<b>Nombre de la Clase:</b> ElementoComputadorVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica un objeto de tipo ElementoComputadorVisitor.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
String str_codComputador	Identifica el computador al que se le asigna un elemento
Int int_codElemento	Determina un elemento de inventario asignado a un computador
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
ElementoComputadorVisitor	Método constructor sin parámetros
getCodigoComputador	Retorna el código del computador al que se le asigna un elemento
getCodigoElemento	Retorna el código del elemento asignado a un computador
setCodigoComputador	Permite asignar o cambiar el código del computador
setCodigoElemento	Permite asignar o cambiar el código del elemento
<b>Observaciones:</b>	

### 11.1.7.7 Elemento Computador Inicial Manager

<b>Nombre de la Clase:</b> ElementoComputadorInicialMgr	
<b>Descripción de la clase</b> Clase que permite la administración de los componentes iniciales de un computador. Tiene toda la lógica de la clase ElementoComputadorVO a nivel de modelo.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_elementoComputador	ArrayList (colección de obj) que almacena los resultados de las consultas

String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoComputadorVO	retorna	Asociación
ComputadorVO	pertenece	Asociación
InventarioVO	pertenece	Asociación
ElementoComputadorVisitor	utiliza	Asociación
ComputadorMgr	utiliza	Asociación
InventarioMgr	utiliza	Asociación
ElementoComputadorInicialDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoComputador	Método que permite acceder a la consulta de los elementos que componen un determinado computador	
findByCodigoElemento	Método que permite establecer los atributos de un objeto de tipo ElementoComputadorVisitor para consultar a que computador se encuentra asignado un determinado elemento	
createElementoComputador	Método que permite establecer los atributos de un objeto de tipo ElementoComputadorVisitor para la asignación de los componentes iniciales de un determinado computador	
deleteComponente	Método que permite establecer los atributos de un objeto de tipo ElementoComputadorVO para la eliminación de un determinado componente	
<b>Observaciones:</b>		

### 11.1.7.8 Elemento Computador Actual Manager

<b>Nombre de la Clase:</b> ElementoComputadorActualMgr		
<b>Descripción de la clase</b> Clase que permite la administración de los componentes Actuales de un computador. Tiene toda la lógica de la clase ElementoComputadorVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_elementoComputador	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoComputadorVO	retorna	Asociación
ComputadorVO	pertenece	Asociación
InventarioVO	pertenece	Asociación

ElementoComputadorVisitor	utiliza	Asociación
ComputadorMgr	utiliza	Asociación
InventarioMgr	utiliza	Asociación
ElementoComputadorActualDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoComputador	Método que permite acceder a la consulta de los elementos que componen un determinado computador	
findByCodigoElemento	Método que permite establecer los atributos de un objeto de tipo ElementoComputadorVisitor para consultar a que computador se encuentra asignado un determinado elemento	
createElementoComputador	Método que permite establecer los atributos de un objeto de tipo ElementoComputadorVisitor para la asignación de los componentes actuales de un determinado computador	
deleteComponente	Método que permite establecer los atributos de un objeto de tipo ElementoComputadorVO para la eliminación de un determinado componente y su reubicación en Monitoría	
<b>Observaciones:</b>		

#### 11.1.7.9 Elemento Computador Inicial Data Access Object

<b>Nombre de la Clase:</b> ElementoComputadorInicialDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los elementos iniciales que componen un Computador		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoComputadorVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByCodigoComputador	Método que retorna un ArrayList de objetos de tipo ElementoComputadorVisitor con todos sus atributos de acuerdo a un determinado Código de Computador	
createElementoComputador	Método que permite asignar componentes iniciales a un Computador	



findByCodigoElemento	Método que retorna un ArrayList de objetos de tipo ElementoComponenteVisitor con todos sus atributos de acuerdo a un determinado Código de Elemento
deleteComponente	Método que elimina una componente asignado a un computador
<b>Observaciones:</b>	

#### 11.1.7.10 Elemento Computador Actual Data Access Object

<b>Nombre de la Clase:</b> ElementoComputadorActualDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los elementos actuales que componen un Computador		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ElementoComputadorVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByCodigoComputador	Método que retorna un ArrayList de objetos de tipo ElementoComputadorVisitor con todos sus atributos de acuerdo a un determinado Código de Computador	
findByCodigoElemento	Método que retorna un ArrayList de objetos de tipo ElementoComponenteVisitor con todos sus atributos de acuerdo a un determinado Código de Elemento	
createElementoComputador	Método que permite asignar componentes actuales a un Computador	
deleteComponente	Método que elimina una componente asignado a un computador	
<b>Observaciones:</b>		

#### 11.1.7.11 Registro Cambio Computador Value Object

<b>Nombre de la Clase:</b> InventarioVO
--

<b>Descripción de la clase</b>		
Clase que permite realizar un seguimiento en cuanto a los cambios efectuados en un determinado computador		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codRegistro	Identifica de manera única un registro de cambio	
String str_fecCambio	Determina la fecha en que se ingresa un determinado elemento.	
String str_observaciones	Permite registrar anotaciones propias de un determinado elemento.	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ComputadorVO	generaUn	Asociación
UsuarioVO	esRealizadoPor	Asociación
EstadoElementoVO	tiene	Asociación
UbicacionVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
RegistroCambioComputadorVO	Método constructor sin parámetros	
getCodigoRegistro	Retorna el código del registro	
getComputadorVO	Retorna un objeto de tipo ComputadorVO	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getFechaCambio	Retorna la fecha en que en la que se realiza un cambio	
getEstadoInicialVO	Retorna un objeto de tipo EstadoElementoVO	
getEstadoActualVO	Retorna un objeto de tipo EstadoElementoVO	
getUbicacionInicialVO	Retorna un objeto de tipo UbicacionVO	
getUbicacionActualVO	Retorna un objeto de tipo UbicacionVO	
getObservacion	Retorna las anotaciones realizadas con relación al cambio	
setCodigoRegistro	Permite asignar el código del registro	
setComputadorVO	Permite asignar un objeto ComputadorVO	
setUsuarioVO	Permite asignar un objeto UsuarioVO	
setFechaCambio	Permite asignar la fecha en la que se realiza el cambio	
setEstadoInicialVO	Permite asignar un objeto EstadoElementoVO	
setEstadoActualVO	Permite asignar un objeto EstadoElementoVO	
setUbicacionInicialVO	Permite asignar un objeto UbicacionVO	
setUbicacionActualVO	Permite asignar un objeto UbicacionVO	
setObservacion	Permite asignar las observaciones pertinentes al cambio efectuado	
<b>Observaciones:</b>		

### 11.1.7.12 Registro Cambio Computador Visitor

<b>Nombre de la Clase:</b> RegistroCambioComputadorVisitor
---

<b>Descripción de la clase</b>	
Clase que representa de forma lógica un objeto de tipo ComputadorVisitor, hace referencia directa con cada uno de los campos de la tabla registrocambioscomputador.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codRegistro	Identifica de manera única un registro de cambio
String str_codComputador	Identifica el código del computador
String str_ideUsuario	Determina la identificación del usuario que realiza el cambio
String str_fecCambio	Determina la fecha en la que realiza el cambio
int int_codEstadoInicial	Determina el estado del computador antes de realizar el cambio
int int_codEstadoActual	Determina el estado del computador después de realizar el cambio
int int_codUbicacionInicial	Determina la ubicación del computador antes de realizar el cambio
int int_codUbicacionActual	Determina la ubicación del computador después de realizar el cambio
String str_observaciones	Identifica las anotaciones pertinentes al cambio efectuado
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
RegistroCambioComputadorVisitor	Método constructor sin parámetros
getCodigoRegistro	Retorna el código del registro
getCodigoComputador	Retorna el código del computador
getIdentificacionUsuario	Retorna la identificación del usuario que realiza el cambio
getFechaCambio	Retorna la fecha en la se realiza el cambio
getCodigoEstadoInicial	Retorna el código del estado inicial del computador
getCodigoEstadoActual	Retorna el código del estado actual del computador
getCodigoUbicacionInicial	Retorna el código de la ubicación inicial del computador
getCodigoUbicacionActual	Retorna el código de la ubicación actual del computador
getObservacion	Retorna las anotaciones del cambio
setCodigoRegistro	Permite asignar el código del elemento
setCodigoComputador	Permite asignar el código del computador
setIdentificacionUsuario	Permite asignar la identificación del usuario que realiza el cambio
setFechaCambio	Permite asignar la fecha en la que se realiza el cambio
setCodigoEstadoInicial	Permite asignar el estado inicial del computador
setCodigoEstadoActual	Permite asignar el estado actual del computador
setCodigoUbicacionInicial	Permite asignar la ubicación inicial del computador
setCodigoUbicacionActual	Permite asignar la ubicación actual del computador
setObservacion	Permite asignar las observaciones relacionadas al cambio efectuado
<b>Observaciones:</b>	

### 11.1.7.13 Registro Cambio Computador Manager

<b>Nombre de la Clase:</b> RegistroCambioComputadorMgr		
<b>Descripción de la clase</b> Clase que permite llevar un registro de los cambios realizados a un computador. Tiene toda la lógica de la clase RegistroCambioComputadorVO a nivel de modelo.		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_registroCambio		ArrayList (colección de obj) que almacena los resultados de las consultas
String v_str_mensaje		Cadena que almacena el mensaje que retorna una determinada operación
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RegistroCambioComputadorVO	retorna	Asociación
ComputadorVO	tiene	Asociación
UsuarioVO	tiene	Asociación
UbicacionVO	tiene	Asociación
EstadoElementoVO	tiene	Asociación
RegistroCambioComputadorVisitor	utiliza	Asociación
ComputadorMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
UbicacionMgr	utiliza	Asociación
EstadoElementoMgr	utiliza	Asociación
RegistroCambioComputadorDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>		<b>Descripción</b>
findByCodigoComputador		Método que permite acceder a la consulta de todos los cambios efectuados a un determinado computador
findByAll		Método que permite acceder a la consulta de todos los cambios efectuados
findByFecha		Método que permite acceder a la consulta de todos los cambios efectuados a los computadores en un intervalo de fecha dado
createRegistroCambioComputador		Método que permite establecer los atributos de un objeto de tipo RegistroCambioComputadorVisitor para el registro de un cambio efectuado
deleteRegistroCambioComputador		Método que permite establecer los atributos de un objeto de tipo RegistroCambioComputadorVisitor para la eliminación de un cambio efectuado en un computador
<b>Observaciones:</b>		

#### 11.1.7.14 Registro Cambio Computador Data Access Object

<b>Nombre de la Clase:</b> RegistroCambioComputadorDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto al registro de cambio efectuados sobre un computador.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RegistroCambioComputadorVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByCodigoComputador	Método que retorna un ArrayList de objetos de tipo RegistroCambioComputadorVisitor con todos sus atributos de acuerdo a un código	
findByAll	Método que retorna un ArrayList de objetos de tipo RegistroCambioComputadorVO con todos sus atributos	
findByFecha	Método que retorna un ArrayList de objetos de tipo RegistroCambioComputadorVisitor con todos sus atributos de acuerdo a una determinada fecha	
createRegistroCambioComputador	Método que permite registrar los cambios efectuados en un Computador	
deleteRegistroCambioComputador	Método que permite eliminar un determinado registro de un cambio efectuado a un computador	
<b>Observaciones:</b>		

#### 11.1.7.15 Registro Cambio Componente Value Object

<b>Nombre de la Clase:</b> RegistroCambioComponenteVO
--

<b>Descripción de la clase</b>		
Clase que permite realizar un seguimiento en cuanto a los cambios de componentes efectuados a un determinado computador		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codRegistro	Identifica de manera única un registro de cambio	
String str_fecCambio	Determina la fecha en que se ingresa un determinado elemento.	
String str_observaciones	Permite registrar anotaciones propias de un determinado elemento.	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
InventarioVO	tiene	Asociación
UbicacionVO	estaEn	Asociación
ComputadorVO	tiene	Asociación
UsuarioVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
RegistroCambioComponenteVO	Método constructor sin parámetros	
getCodigoRegistro	Retorna el código del registro	
getInventarioVO		
getUbicacionOriginalVO		
getUbicacionActualVO		
getComputadorOriginalVO	Retorna un objeto de tipo ComputadorVO	
getComputadorActualVO		
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getFechaCambio	Retorna la fecha en que en la que se realiza un cambio	
getObservacion	Retorna las anotaciones realizadas con relación al cambio	
setCodigoRegistro	Permite asignar el código del registro	
setInventarioVO	Permite asignar un objeto InventarioVO	
setUbicacionOriginalVO	Permite asignar un objeto UbicacionVO	
setUbicacionActualVO	Permite asignar un objeto UbicacionVO	
setComputadorOriginalVO	Permite asignar un objeto ComputadorVO	
setComputadorActualVO	Permite asignar un objeto ComputadorVO	
setUsuarioVO	Permite asignar un objeto UsuarioVO	
setFechaCambio	Permite asignar la fecha en la que se realiza el cambio	
setObservacion	Permite asignar las observaciones pertinentes al cambio efectuado	
<b>Observaciones:</b>		

#### 11.1.7.16 Registro Cambio Componente Visitor

<b>Nombre de la Clase:</b> RegistroCambioComputadorVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica un objeto de tipo ComputadorVisitor, hace referencia directa con cada uno de los campos de la tabla registrocambioscomputador.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codRegistro	Identifica de manera única un registro de cambio
String str_codComputador	Identifica el código del computador
String str_ideUsuario	Determina la identificación del usuario que realiza el cambio
String str_fecCambio	Determina la fecha en la que realiza el cambio
int int_codEstadoInicial	Determina el estado del computador antes de realizar el cambio
int int_codEstadoActual	Determina el estado del computador después de realizar el cambio
int int_codUbicacionInicial	Determina la ubicación del computador antes de realizar el cambio
int int_codUbicacionActual	Determina la ubicación del computador después de realizar el cambio
String str_observaciones	Identifica las anotaciones pertinentes al cambio efectuado
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
RegistroCambioComputadorVisitor	Método constructor sin parámetros
getCodigoRegistro	Retorna el código del registro
getCodigoComputador	Retorna el código del computador
getIdentificacionUsuario	Retorna la identificación del usuario que realiza el cambio
getFechaCambio	Retorna la fecha en la se realiza el cambio
getCodigoEstadoInicial	Retorna el código del estado inicial del computador
getCodigoEstadoActual	Retorna el código del estado actual del computador
getCodigoUbicacionInicial	Retorna el código de la ubicación inicial del computador
getCodigoUbicacionActual	Retorna el código de la ubicación actual del computador
getObservacion	Retorna las anotaciones del cambio
setCodigoRegistro	Permite asignar el código del elemento
setCodigoComputador	Permite asignar el código del computador
setIdentificacionUsuario	Permite asignar la identificación del usuario que realiza el cambio
setFechaCambio	Permite asignar la fecha en la que se realiza el cambio
setCodigoEstadoInicial	Permite asignar el estado inicial del computador
setCodigoEstadoActual	Permite asignar el estado actual del computador
setCodigoUbicacionInicial	Permite asignar la ubicación inicial del computador
setCodigoUbicacionActual	Permite asignar la ubicación actual del computador

setObservacion	Permite asignar las observaciones relacionadas al cambio efectuado
<b>Observaciones:</b>	

### 11.1.7.17 Registro Cambio Componente Manager

<b>Nombre de la Clase:</b> RegistroCambioComponenteMgr		
<b>Descripción de la clase</b> Clase que permite llevar un registro de los cambios de componentes realizados a un computador. Tiene toda la lógica de la clase RegistroCambioComponenteVO a nivel de modelo.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_registroComponente	ArrayList (colección de obj) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RegistroCambioComponenteVO	retorna	Asociación
ComputadorVO	tiene	Asociación
UsuarioVO	tiene	Asociación
InventarioVO	tiene	Asociación
UbicacionVO	tiene	Asociación
RegistroCambioComponenteVisitor	utiliza	Asociación
ComputadorMgr	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
InventarioMgr	utiliza	Asociación
UbicacionMgr	utiliza	Asociación
RegistroCambioComponenteDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoComputador	Método que permite acceder a la consulta de todos los cambios de componentes efectuados a un determinado computador	
findByUsuario	Método que permite acceder a la consulta de todos los cambios de los elementos de inventario efectuados por un determinado usuario	
findByCodigoElemento	Método que permite acceder a la consulta de todos los cambios efectuados de un determinado elemento de inventario	



findByFecha	Método que permite acceder a la consulta de todos los cambios de componentes efectuados en un intervalo de fechas dado
createRegistroCambioComponente	Método que permite establecer los atributos de un objeto de tipo RegistroCambioComponenteVisitor para el registro de un cambio efectuado
<b>Observaciones:</b>	

### 11.1.7.18 Registro Cambio Componente Data Access Object

<b>Nombre de la Clase:</b> RegistroCambioComponenteDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto al registro de cambio de componentes efectuados a un computador.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RegistroCambioComponenteVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByCodigoComputador	Método que retorna un ArrayList de objetos de tipo RegistroCambioComponenteVisitor con todos sus atributos de acuerdo a un determinado código de computador	
findByUsuario	Método que retorna un ArrayList de objetos de tipo RegistroCambioComponenteVisitor con todos sus atributos de acuerdo al usuario que realizó los cambios	
findByCodigoElemento	Método que retorna un ArrayList de objetos de tipo RegistroCambioComponenteVisitor con todos sus atributos de acuerdo a un determinado código de elemento de inventario	

findByFecha	Método que retorna un ArrayList de objetos de tipo RegistroCambioComponenteVisitor con todos sus atributos de acuerdo a un determinado intervalo de fechas
createRegistroCambioComponente	Método que permite registrar los cambios efectuados en un Computador
<b>Observaciones:</b>	

## 11.1.8 Administración

### 11.1.8.1 Servicio General Value Object

<b>Nombre de la Clase:</b> ServicioGeneralVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las restricciones de los servicios generales		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codServicio	Identifica de forma única el código del servicio.	
String str_desServicio	Cadena que almacena la descripción del servicio	
int int_intServicio	Indica el período de anticipación para poder crear una cita de salud	
int int_numCitas	Identifica el número de citas que un usuario puede efectuar en un día.	
int int_numPacientes	Identifica el número de Pacientes que se pueden atender en una franja.	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ServicioGeneralVO	Método constructor con parámetros y sin parámetros	
getCodigoServicio	Retorna el código del servicio	
getDescripcionServicio	Retorna la descripción del servicio	
getIntervaloServicio	Retorna el periodo de anticipación para reservar una cita de salud	
getNumeroCita	Retorna el número de citas que un usuario puede efectuar en un día	
getNumeroPaciente	Retorna el número de Pacientes que se pueden atender en una franja.	
setCodigoServicio	Permite asignar el código del servicio	
setDescripcionServicio	Permite asignar la descripción del servicio	

setIntervaloServicio	Permite asignar o cambiar el periodo de anticipación para reservar una cita de salud
setNumeroCita	Permite asignar o cambiar el número de citas que un usuario puede efectuar en un día
setNumeroPaciente	Permite asignar o cambiar el número de Pacientes que se pueden atender en una franja.
<b>Observaciones:</b>	

### 11.1.8.2 Servicio General Manager

<b>Nombre de la Clase</b> ServicioGeneralMgr		
<b>Descripción de la clase</b> Clase que permite administrar las restricciones para los servicios generales. Tiene toda la lógica de la clase ServicioGeneralVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de los servicios generales como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioGeneralVO	<i>retorna</i>	Asociación
ServicioGeneralDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ServicioGeneralVO para la consulta de una franja de acuerdo a un código	
updateIntervaloCitasPacientes	Método que permite establecer los atributos de un objeto de tipo servicioGeneralVO para la actualización del intervalo y el período de sanción establecido para un servicio determinado.	
<b>Observaciones:</b>		

### 11.1.8.3 Servicio General Data Access Object

<b>Nombre de la Clase:</b> ServicioGeneralDAO
--

<b>Descripción de la clase</b>		
Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a los servicios generales		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioGeneralVO	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo ServicioGeneralVO con todos sus atributos.	
updateIntervaloCitasPacientes	Método que actualiza los valores correspondientes al intervalo del servicio y número de reservas que un usuario puede hacer por día, cantidad de pacientes que un médico atiende por franja.	
<b>Observaciones:</b>		

#### 11.1.8.4 Franja Value Object

<b>Nombre de la Clase:</b>		
FranjaVO		
<b>Descripción de la clase</b>		
Clase que representa de forma lógica las franjas horarias establecidas para los servicios ofrecidos por la universidad		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codFranja	Identifica de forma única una franja.	
String str_desFranja	Cadena que almacena la descripción de la franja	
int int_estFranja	Identifica el estado de una franja	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
FranjaVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código de la franja	
getDescripcion	Retorna la descripción de la franja	
getEstado	Retorna el estado de la franja	
setCodigo	Permite asignar o cambiar el código del servicio	
setDescripcion	Permite asignar o cambiar la descripción de la franja	

setEstado	Permite asignar o cambiar el estado de una franja
<b>Observaciones:</b> El estado de una franja puede ser activa o inactiva	

### 11.1.8.5 Franja Manager

<b>Nombre de la Clase</b> FranjaMgr		
<b>Descripción de la clase</b> Clase que permite administrar las franjas horarias. Tiene toda la lógica de la clase FranjaVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de las franjas como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	<i>retorna</i>	Asociación
FranjaDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo FranjaVO con la descripción de la franja horaria de acuerdo a un Código determinado.	
findByAll	Método que permite acceder a la consulta de todos las franjas existentes.	
findByAllInactiva	Método que permite acceder a la consulta de todas las franjas inactivas.	
findByDescripcion	Método que permite establecer los atributos de un objeto de tipo FranjaVO para la consulta de una franja de acuerdo a la descripción.	
findByDescripcion	Método que permite establecer los atributos de un objeto de tipo FranjaVO para la consulta de una franja de acuerdo a una determinada descripción y estado.	
findByTotal	Método que permite acceder a la consulta del número de franjas activas.	
updateEstado	Método que permite actualizar el estado de una determinada franja.	
<b>Observaciones:</b>		

### 11.1.8.6 Franja Data Access Object

<b>Nombre de la Clase:</b> FranjaDAO		
<b>Descripción de la clase</b> Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a las franjas horarias		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
FranjaVO	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo FranjaVO con la descripción de la franja horaria de acuerdo a un Código determinado.	
findByAll	Método que retorna un ArrayList de objetos con todos los atributos de las franjas horarias activas.	
findByAllInactiva	Método que retorna un ArrayList de objetos con todos los atributos de las franjas horarias inactivas.	
findByDescripcion	Método que retorna un objeto de tipo FranjaVO con el Código de la franja horaria de acuerdo a una descripción determinada.	
findByDescripcionEstado	Método que retorna un objeto de tipo FranjaVO con el Código de la franja horaria de acuerdo a una descripción determinada.	
findByTotal	Método que retorna el total de las franjas horarias existentes.	
updateEstado	Método que permite actualizar el estado de una determinada franja.	
<b>Observaciones:</b>		

#### 11.1.8.7 Servicio Informático Value Object

<b>Nombre de la Clase:</b> ServicioInformaticoVO	
<b>Descripción de la clase</b> Clase que representa de forma lógica las restricciones para el uso de los servicios informáticos.	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codServicio	Identifica de forma única un servicio

String str_desServicio	Cadena que almacena la descripción del servicio	
int int_intServicio	Identifica el periodo de tiempo en días durante el cual se pueden efectuar las reservas.	
int int_numReserva	Identifica el número de reservas que un usuario puede efectuar en un día.	
int int_numFalta	Indica el número de faltas a partir del cual se establece una sanción.	
int int_perSancion	Indica el periodo de tiempo durante el cual un usuario estará sancionado.	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
ServicioInformativoVO	Método constructor con parámetros y sin parámetros	
getCodigoServicio	Retorna el código del servicio	
getDescripcionServicio	Retorna la descripción del servicio	
getIntervaloServicio	Retorna el periodo de tiempo en días durante el cual se pueden efectuar las reservas.	
getNumeroReserva	Retorna el número de reservas que un usuario puede efectuar en un día	
getNumeroFalta	Retorna el número de faltas a partir del cual se establece una sanción	
getPeriodoSancion	Retorna el periodo de tiempo durante el cual un usuario estará sancionado	
setCodigoServicio	Permite asignar o cambiar el código del servicio	
setDescripcionServicio	Permite asignar un valor al atributo Descripción del Servicio	
setIntervaloServicio	Permite asignar el periodo de tiempo en días durante el cual se pueden efectuar las reservas.	
setNumeroReserva	Permite asignar el número de reservas que un usuario puede efectuar en un día	
setNumeroFalta	Permite asignar el número de faltas a partir del cual se establece una sanción	
setPeriodoSancion	Permite asignar el periodo de tiempo durante el cual un usuario estará sancionado	
<b>Observaciones:</b>		

### 11.1.8.8 Servicio Informático Manager

<b>Nombre de la Clase</b> ServicioInformativoMgr
Clase que permite administrar las restricciones de los servicios informáticos existentes. Tiene toda la lógica de la clase ServicioInformativoVO a nivel de modelo

<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_aux		Arraylist (colección de obj) que almacena los datos de los servicios informáticos como resultado de las consultas
String v_str_mensaje		Cadena que almacena el mensaje que retorna una determinada operación
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioInformaticoVO	<i>retorna</i>	Asociación
ServicioInformaticoDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo ServicioInformaticoVO para la consulta de Servicio de acuerdo a un código.	
findByMaximoIntervalo	Método que permite acceder a la consulta del máximo valor de los intervalos establecidos para los Servicios Informáticos.	
updateFaltaPeriodo	Método que permite establecer los atributos de un objeto de tipo servicioInformaticoVO para la actualización del número de faltas y el periodo de sanción establecido para un servicio determinado.	
updateIntervaloNumeroReserva	Método que permite establecer los atributos de un objeto de tipo servicioInformaticoVO para la actualización del intervalo y el número de reservas diarias establecidas para un Servicio determinado.	
findByAll	Método que permite acceder a la consulta de todos los servicios informáticos existentes.	
<b>Observaciones:</b>		

#### 11.1.8.9 Servicio Informático Data Access Object

<b>Nombre de la Clase:</b> ServicioInformaticoDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a las restricciones establecidos para los servicios informáticos.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
ServicioInformaticoVO	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación



<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
findByPrimaryKey	Método que retorna un objeto de tipo ServicioInformaticoVO con todos sus atributos
findByMaximoIntervalo	Método que retorna el máximo valor de los intervalos establecidos para los Servicios Informáticos
updateFaltaPeriodo	Método que actualiza los atributos correspondientes al número de faltas y al período de Sanción de un objeto de tipo ServicioInformaticoVO
updateIntervaloNumeroReserva	Método que actualiza los atributos correspondientes al intervalo del servicio y el número de reservas de un objeto de tipo ServicioInformaticoVO.
findByAll	Método que retorna un ArrayList de objetos de tipo ServicioInformaticoVO con todos sus atributos
<b>Observaciones:</b>	

#### 11.1.8.10 Privilegio Value Object

<b>Nombre de la Clase:</b> PrivilegioVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los privilegios establecidos para un determinado rol.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codPrivilegio	Identifica de forma única un privilegio	
String str_desPrivilegio	Cadena que almacena la descripción de un privilegio	
String str_urlPrivilegio	Cadena que almacena la ruta de acceso par un determinado módulo	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
PrivilegioVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código del privilegio	
getDescripcion	Retorna la descripción del privilegio	
getUrl	Retorna la ruta de acceso par un determinado módulo	
setCodigo	Permite asignar o cambiar el código del privilegio	
setDescripcion	Permite asignar o cambiar la descripción del privilegio	
setUrl	Permite asignar o cambiar la ruta de acceso par un determinado módulo	

**Observaciones:**

### 11.1.8.11 Privilegio Manager

<b>Nombre de la Clase</b> PrivilegioMgr		
Clase que administra de forma lógica las operaciones de los privilegios establecidos para un determinado rol. Tiene toda la lógica de la clase PrivilegioVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de los privilegios como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
PrivilegioVO	<i>retorna</i>	Asociación
PrivilegioDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo PrivilegioVO con todos sus atributos de acuerdo a un código determinado.	
findByDescripcion	Método que retorna un objeto de tipo PrivilegioVO con todos sus atributos de acuerdo a una descripción determinada.	
findByUrl	Método que retorna un objeto de tipo PrivilegioVO con todos sus atributos de acuerdo a una ruta determinada.	
findByAll	Método que retorna un Arreglo (ArrayList) con objetos de tipo PrivilegioVO con todos sus atributos.	
createPrivilegio	Método que permite adicionar un nuevo Privilegio	
deletePrivilegio	Método que permite eliminar un Privilegio	
updatePrivilegio	Método que permite actualizar un Privilegio	
<b>Observaciones:</b>		

### 11.1.8.12 Privilegio Data Access Object

**Nombre de la Clase:**  
PrivilegioDAO

<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los privilegios.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_privilegio	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
PrivilegioVO	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo PrivilegioVO con todos sus atributos de acuerdo a un código determinado.	
findByDescripcion	Método que retorna un objeto de tipo PrivilegioVO con todos sus atributos de acuerdo a una descripción determinada.	
findByUrl	Método que retorna un objeto de tipo PrivilegioVO con todos sus atributos de acuerdo a una url determinada.	
findByAll	Método que retorna un Arreglo (ArrayList) con objetos de tipo PrivilegioVO con todos sus atributos.	
createPrivilegio	Método que permite adicionar un nuevo Privilegio	
deletePrivilegio	Método que permite eliminar un Privilegio	
updatePrivilegio	Método que permite actualizar un Privilegio.	
<b>Observaciones:</b>		

### 11.1.8.13 Rol Value Object

<b>Nombre de la Clase:</b> RolVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los roles establecidos para los usuarios		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codRol	Identifica de forma única un rol	
String str_desRol	Cadena que almacena la descripción de un rol	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
RolVO	Método constructor con parámetros y sin parámetros	

getCodigo	Retorna el código del rol
getDescripcion	Retorna la descripción del rol
setCodigo	Permite asignar o cambiar el código del rol
setDescripcion	Permite asignar o cambiar la descripción del rol
<b>Observaciones:</b>	

#### 11.1.8.14 Rol Manager

<b>Nombre de la Clase</b>		
RolMgr		
Clase que administra de forma lógica las operaciones los roles establecidos para los usuarios. Tiene toda la lógica de la clase RolVO a nivel de modelo		
<b>Propiedad</b>		<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_aux		Arraylist (colección de obj) que almacena los datos de los roles como resultado de las consultas
String v_str_mensaje		Cadena que almacena el mensaje que retorna una determinada operación
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RolVO	<i>retorna</i>	Asociación
RolDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo RolVO para la consulta de un Rol de acuerdo a un código.	
findByAll	Método que permite acceder a la consulta de todos los roles existentes.	
findByDescripcion	Método que permite establecer los atributos de un objeto de tipo RolVO para la consulta de un Rol de acuerdo a la descripción.	
createRol	Método que permite establecer los atributos de un objeto de tipo RolVO para la creación de un Rol determinado.	
updateRol	Método que permite establecer los atributos de un objeto de tipo RolVO para la actualización de un Rol determinado.	
deleteRol	Método que permite establecer los atributos de un objeto de tipo RolVO para la eliminación de un Rol determinado.	
<b>Observaciones:</b>		

#### 11.1.8.15 Rol Data Access Object

<b>Nombre de la Clase:</b> RolDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los roles.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_privilegio	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RolVO	<i>Retorna</i>	Asociación
DBConnection	<i>Solicita</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo RolVO de acuerdo a un código determinado.	
findByAll	Método que retorna un ArrayList de objetos de tipo RolVO con los datos de todos los roles existentes.	
findByDescripcion	Método que retorna un objeto de tipo RolVO de acuerdo a una descripción determinada.	
createRol	Método que permite la creación de un Rol	
updateRol	Método que permite la actualización de un Rol	
deleteRol	Método que permite la eliminación de un Rol	
<b>Observaciones:</b>		

#### 11.1.8.16 Rol Privilegio Value Object

<b>Nombre de la Clase:</b> RolPrivilegioVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los privilegios asignados a un determinado rol.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RolVO	pertenece	Asociación
PrivilegioVO	está	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
RolPrivilegioVO	Método constructor con parámetros y sin parámetros	
getRolVO	Retorna un objeto de tipo RolVO	
getPrivilegioVO	Retorna un objeto de tipo PrivilegioVO	

setRolVO	Permite asignar o modificar un objeto o de tipo RolVO
setPrivilegioVO	Permite asignar o modificar un objeto o de tipo PrivilegioVO
<b>Observaciones:</b>	

### 11.1.8.17 Rol Privilegio Visitor

<b>Nombre de la Clase:</b> RolPrivilegioVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los privilegios asignados a un determinado rol. Hace referencia directa con cada uno de los campos de la tabla rolesprivilegios	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codRol	Identifica el código del rol
int int_codPrivilegio	Identifica el código del privilegio
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
RolPrivilegioVisitor	Método constructor con parámetros y sin parámetros
getCodigoRol	Retorna el código del rol
getCodigoPrivilegio	Retorna el código del privilegio
setCodigoRol	Permite asignar o cambiar el código del rol
setCodigoPrivilegio	Permite asignar o cambiar el código del privilegio
<b>Observaciones:</b>	

### 11.1.8.18 Rol Privilegio Manager

<b>Nombre de la Clase:</b> RolPrivilegioMgr		
<b>Descripción de la clase</b> Clase que permite administrar de forma lógica los privilegios asignados a los roles. Tiene toda la lógica de la clase RolPrivilegioVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_rolPrivilegio	Arraylist (colección de obj) que almacena los privilegios asignados a un determinado rol.	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>

RolPrivilegioVO	<b>retorna</b>	Asociación
RolVO	<b>utiliza</b>	Asociación
PrivilegioVO	<b>utiliza</b>	Asociación
RolMgr	<b>utiliza</b>	Asociación
PrivilegioMgr	<b>utiliza</b>	Asociación
RolPrivilegioVisitor	<b>utiliza</b>	Asociación
RolPrivilegioDAO	<b>utiliza</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByCodigoRol	Método que permite acceder a la consulta de todos los privilegios asignados a un Rol de acuerdo a un código determinado.	
findByTotalCodigoPrivilegio	Método que permite acceder a la consulta del número de veces que se encuentra asignado un determinado privilegio.	
createRolPrivilegio	Método que permite establecer los atributos de un objeto de tipo rolPrivilegioVisitor para la asignación de un privilegio a un Rol determinado.	
deleteRol	Método que permite establecer los atributos de un objeto de tipo RolPrivilegioVisitor para la eliminación de todos los privilegios asignados a un Rol determinado.	
deleteRolPrivilegio	Método que permite establecer los atributos de un objeto de tipo RolPrivilegioVisitor para la eliminación de un privilegio asignado a un Rol determinado.	
<b>Observaciones:</b>		

#### 11.1.8.19 Rol Privilegio Data Access Object

<b>Nombre de la Clase:</b> RolPrivilegioDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los privilegios asignados a un rol.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RolPrivilegioVisitor	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	

findByCodigoRol	Método que retorna un ArrayList de objetos de tipo RolPrivilegioVisitor con todos sus atributos de acuerdo a un Código de rol determinado.
findByTotalCodigoPrivilegio	Método que retorna el número de registros en los cuales está involucrado un determinado privilegio.
createRolPrivilegio	Método que permite la asignación de un Privilegio a un determinado Rol
deleteRol	Método que elimina el registro de la tabla rolesprivilegios de todos los privilegios asignados a un determinado rol.
deleteRolPrivilegio	Método que elimina el registro de asignación de un Privilegio a un determinado Rol.
<b>Observaciones:</b>	

### 11.1.8.20 Rol Usuario Value Object

<b>Nombre de la Clase:</b> RolUsuarioVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los roles asignados a un determinado usuario.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RolVO	pertenece	Asociación
UsuarioVO	tiene	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
RolUsuarioVO	Método constructor con parámetros y sin parámetros	
getUsuarioVO	Retorna un objeto de tipo UsuarioVO	
getRolVO	Retorna un objeto de tipo RolVO	
setUsuarioVO	Permite asignar o modificar un objeto o de tipo UsuarioVO	
setRolVO	Permite asignar o modificar un objeto o de tipo RolVO	
<b>Observaciones:</b>		

### 11.1.8.21 Rol Usuario Visitor

<b>Nombre de la Clase:</b> RolUsuarioVisitor
---



<b>Descripción de la clase</b>	
Clase que representa de forma lógica los roles asignados a un determinado usuario. Hace referencia directa con cada uno de los campos de la tabla rolesusuarios	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
str_ideUsuario	Determina la identificación del usuario
int int_codRol	Identifica el código del rol
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
RolUsuarioVisitor	Método constructor con parámetros y sin parámetros
getIdentificacion	Retorna la identificación del usuario
getCodigo	Retorna el código del rol
setIdentificacion	Permite asignar o cambiar la identificación del usuario
setCodigo	Permite asignar o cambiar el código del rol
<b>Observaciones:</b>	

#### 11.1.8.22 Rol Usuario Manager

<b>Nombre de la Clase:</b>		
RolUsuarioMgr		
<b>Descripción de la clase</b>		
Clase que permite administrar de forma lógica los roles asignados a un determinado usuario. Tiene toda la lógica de la clase RolUsuarioVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_rolUsuario	Arraylist (colección de obj) que almacena los roles asignados a un determinado usuario.	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RolUsuarioVO	<i>retorna</i>	Asociación
UsuarioVO	<i>utiliza</i>	Asociación
RolVO	<i>utiliza</i>	Asociación
UsuarioMgr	<i>utiliza</i>	Asociación
RolMgr	<i>utiliza</i>	Asociación
RolUsuarioVisitor	<i>utiliza</i>	Asociación
RolUsuarioDAO	<i>utiliza</i>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByIdentificacion	Método que permite acceder a la consulta de todos los roles asignados a un Usuario de acuerdo a una identificación.	

findByTotalCodigo	Método que permite establecer los atributos de un objeto de tipo rolUsuarioVisitor para obtener el número de usuarios que tiene un rol determinado.
findByTotalRolUsuario	Método que permite establecer los atributos de un objeto de tipo rolUsuarioVisitor para obtener el número de roles que tiene un determinado usuario.
createRolUsuario	Método que permite establecer los atributos de un objeto de tipo rolUsuarioVisitor para la asignación de un rol a un usuario determinado.
deleteRolUsuario	Método que permite establecer los atributos de un objeto de tipo RolUsuarioVisitor para la eliminación de un rol asignado a un Usuario determinado.
deleteUsuario	Método que permite establecer los atributos de un objeto de tipo RolUsuarioVisitor para la eliminación de un registro rolUsuario de acuerdo a un usuario determinado.
<b>Observaciones:</b>	

### 11.1.8.23 Rol Usuario Data Access Object

<b>Nombre de la Clase:</b> RolUsuarioDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la Base de Datos con respecto a los roles asignados a un usuario.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RolUsuarioVisitor	<b>Retorna</b>	Asociación
DBConnection	<b>Solicita</b>	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByIdentificacion	Método que retorna un ArrayList de objetos de tipo RolUsuarioVisitor con todos sus atributos de acuerdo a una Identificación determinada.	
findByCodigo	Método que retorna un ArrayList de objetos de acuerdo a un código de rol determinado.	
findByTotalCodigo	Método que retorna el número de usuarios que tienen un determinado rol.	

findByTotalRolUsuario	Método que retorna el número de roles que tiene un determinado usuario.
createRolUsuario	Método que permite asignar un rol a un determinado Usuario.
deleteRolUsuario	Método que elimina el registro de asignación de un Rol a un determinado Usuario.
deleteUsuario	Método que elimina el registro de asignación de un Rol de acuerdo a un determinado Usuario.
<b>Observaciones:</b>	

## 11.1.9 Almacén

### 11.1.9.1 Entrada Almacén Value Object

<b>Nombre de la Clase:</b> EntradaAlmacenVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las entradas de elementos a la sección de Almacén		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codEntrada	Identifica de manera única un registro de entrada de un elemento a Almacén	
String str_fecEntrada	Cadena que almacena la fecha en la cual se realiza la entrada	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
InventarioVO	registra	Asociación
EstadoElementoVO	posee	Asociación
UsuarioVO	aCargo	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
EntradaAlmacenVO	Método constructor sin parámetros	
getCodigoEntrada	Retorna el código de la entrada	
getInventarioVO	Retorna un objeto de tipo InventarioVO	
getFechaEntrada	Retorna la fecha en la que se registra la entrada del elemento	
getEstadoElementoVO	Retorna un objeto de tipo EstadoElementoVO	
getUsuarioRecibeVO	Retorna un objeto de tipo UsuarioVO del usuario que recibe el elemento que se ingresa	
getUsuarioEntregaVO	Retorna un objeto de tipo UsuarioVO del usuario que entrega el elemento que se ingresa	
setCodigoEntrada	Permite asignar el código de la entrada	
setInventarioVO	Permite asignar un objeto de tipo InventarioVO	
setFechaEntrada	Permite asignar la fecha en la que se registra la entrada del elemento	

setEstadoElementoVO	Permite asignar un objeto de tipo EstadoElementoVO
setUsuarioRecibeVO	Permite asignar un objeto de tipo UsuarioVO del usuario que recibe el elemento que se ingresa
setUsuarioEntregaVO	Permite asignar un objeto de tipo UsuarioVO del usuario que entrega el elemento que se ingresa
<b>Observaciones:</b>	

### 11.1.9.2 Entrada Almacén Visitor

<b>Nombre de la Clase:</b> EntradaAlmacenVisitor	
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del registro de entrada de un elemento de inventario a la sección de Almacén	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codEntrada	Identifica de manera única una Entrada a la sección de almacén
int int_codElemento	Indica el código del elemento que se ingresa
String str_fecEntrada	Identifica la fecha en la cual se hace la entrada del elemento
int int_codEstado	Identifica el código del estado con el cual es ingresado un elemento
String str_ideRecibe	Determina la identificación del usuario que recibe el elemento
String str_ideEntrega	Determina la identificación del usuario que entrega el elemento
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
EntradaAlmacenVisitor	Método constructor sin parámetros
getCodigoEntrada	Retorna el código de la entrada
getCodigoElemento	Retorna el código del elemento
getFechaEntrada	Retorna la fecha de entrada
getCodigoEstado	Retorna el código del estado
getIdentificacionRecibe	Retorna la identificación del usuario que recibe el elemento
getIdentificacionEntrega	Retorna la identificación del usuario que entrega el elemento
setCodigoEntrada	Permite asignar el código de la entrada
setCodigoElemento	Permite asignar el código del elemento
setFechaEntrada	Permite asignar la fecha de entrada
setCodigoEstado	Permite asignar el código del estado
setIdentificacionRecibe	Permite asignar la identificación del usuario que recibe el elemento

setIdentificacionEntrega	Permite asignar la identificación del usuario que entrega el elemento
<b>Observaciones:</b>	

### 11.1.9.3 Entrada Almacén Manager

<b>Nombre de la Clase:</b> EntradaAlmacenMgr		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los datos de las entradas de elementos a la sección de almacén, hace referencia directa con cada uno de los campos de la tabla entradasalmacen		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_entradaAlmacen	ArrayList (colección de objetos) que almacena los resultados de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EntradaAlmacenVO	retorna	Asociación
UsuarioVO	utiliza	Asociación
InventarioVO	utiliza	Asociación
EstadoElementoVO	utiliza	Asociación
EntradaAlmacenVisitor	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
InventarioMgr	utiliza	Asociación
EstadoElementoMgr	utiliza	Asociación
EntradaAlmacenDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo EntradaAlmacenVisitor para la consulta de los datos de la entrada de un elemento de acuerdo a un determinado código	
findByAll	Método que permite acceder a la consulta de todos los registros de entrada de elementos a Almacén	
findByCodigoElemento	Método que permite acceder a la consulta de todos los registros de entradas de almacén de un determinado elemento	
findByFechaEntrada	Método que permite acceder a la consulta de todos los registros de entradas de almacén en una determinada fecha	

createEntradaAlmacen	Método que permite establecer los atributos de un objeto de tipo EntradaAlmacenVisitor para el registro de la entrada de un elemento a la sección de almacén
<b>Observaciones:</b>	

#### 11.1.9.4 Entrada Almacén Data Access Object

<b>Nombre de la Clase:</b> EntradaAlmacenDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las entradas de elementos a la sección de almacén		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
EntradaAlmacenVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo EntradaAlmacenVisitor de acuerdo a un código de entrada	
findByAll	Método que retorna un ArrayList(Colección de objetos) de tipo EntradaAlmacenVisitor con los datos de todas las entradas de elementos a la sección de Almacén	
findByCodigoElemento	Método que retorna un ArrayList(Colección de objetos) de tipo EntradaAlmacenVisitor con los datos de las entradas de un determinado elemento	
findByFechaEntrada	Método que retorna un ArrayList(Colección de objetos) de tipo EntradaAlmacenVisitor con los datos de las entradas de elementos en una fecha determinada	
createEntradaAlmacen	Método que permite registrar la entrada de un elemento a almacén	
<b>Observaciones:</b>		

#### 11.1.9.5 Salida Almacén Value Object

<b>Nombre de la Clase:</b> SalidaAlmacenVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica las Salidas de elementos de la sección de Almacén		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
int int_codSalida	Identifica de manera única un registro de salida de un elemento a Almacén	
String str_fecSalida	Cadena que almacena la fecha en la cual se realiza la salida	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
InventarioVO	registra	Asociación
EstadoElementoVO	posee	Asociación
UsuarioVO	aCargo	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
SalidaAlmacenVO	Método constructor sin parámetros	
getCodigoSalida	Retorna el código de la salida	
getInventarioVO	Retorna un objeto de tipo InventarioVO	
getFechaSalida	Retorna la fecha en la que se registra la salida del elemento	
getEstadoElementoVO	Retorna un objeto de tipo EstadoElementoVO	
getUsuarioRecibeVO	Retorna un objeto de tipo UsuarioVO del usuario que recibe el elemento que sale	
getUsuarioEntregaVO	Retorna un objeto de tipo UsuarioVO del usuario que entrega el elemento que sale	
setCodigoSalida	Permite asignar el código de la salida	
setInventarioVO	Permite asignar un objeto de tipo InventarioVO	
setFechaSalida	Permite asignar la fecha en la que se registra la salida del elemento	
setEstadoElementoVO	Permite asignar un objeto de tipo EstadoElementoVO	
setUsuarioRecibeVO	Permite asignar un objeto de tipo UsuarioVO del usuario que recibe el elemento que sale	
setUsuarioEntregaVO	Permite asignar un objeto de tipo UsuarioVO del usuario que entrega el elemento que sale	
<b>Observaciones:</b>		

#### 11.1.9.6 Salida Almacén Visitor

<b>Nombre de la Clase:</b> SalidaAlmacenVisitor
<b>Descripción de la clase</b> Clase que representa de forma lógica los datos del registro de salida de un elemento de inventario

<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
int int_codSalida	Identifica de manera única una Salida de la sección de almacén
int int_codElemento	Indica el código del elemento que sale
String str_fecSalida	Identifica la fecha en la cual se hace la salida del elemento
int int_codEstado	Identifica el código del estado con el cual sale un elemento
String str_ideRecibe	Determina la identificación del usuario que recibe el elemento
String str_ideEntrega	Determina la identificación del usuario que entrega el elemento
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>
SalidaAlmacenVisitor	Método constructor sin parámetros
getCodigoSalida	Retorna el código de la salida
getCodigoElemento	Retorna el código del elemento
getFechaSalida	Retorna la fecha de salida
getCodigoEstado	Retorna el código del estado
getIdentificacionRecibe	Retorna la identificación del usuario que recibe el elemento
getIdentificacionEntrega	Retorna la identificación del usuario que entrega el elemento
setCodigoSalida	Permite asignar el código de la salida
setCodigoElemento	Permite asignar el código del elemento
setFechaSalida	Permite asignar la fecha de salida
setCodigoEstado	Permite asignar el código del estado
setIdentificacionRecibe	Permite asignar la identificación del usuario que recibe el elemento
setIdentificacionEntrega	Permite asignar la identificación del usuario que entrega el elemento
<b>Observaciones:</b>	

### 11.1.9.7 Salida Almacén Manager

<b>Nombre de la Clase:</b> SalidaAlmacenMgr	
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a los datos de las Salidas de elementos de la sección de almacén	
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>
ArrayList al_salidaAlmacen	ArrayList (colección de objetos) que almacena los resultados de las consultas



String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
SalidaAlmacenVO	retorna	Asociación
UsuarioVO	utiliza	Asociación
InventarioVO	utiliza	Asociación
EstadoElementoVO	utiliza	Asociación
SalidaAlmacenVisitor	utiliza	Asociación
UsuarioMgr	utiliza	Asociación
InventarioMgr	utiliza	Asociación
EstadoElementoMgr	utiliza	Asociación
EntradaAlmacenDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo SalidaAlmacenVisitor para la consulta de los datos de la Salida de un elemento de acuerdo a un determinado código	
findByAll	Método que permite acceder a la consulta de todos los registros de Salida de elementos a Almacén	
findByCodigoElemento	Método que permite acceder a la consulta de todos los registros de Salidas de almacén de un determinado elemento	
findByFechaSalida	Método que permite acceder a la consulta de todos los registros de Salidas de almacén en una determinada fecha	
createSalidaAlmacen	Método que permite establecer los atributos de un objeto de tipo SalidaAlmacenVisitor para el registro de la Salida de un elemento de la sección de almacén	
<b>Observaciones:</b>		

### 11.1.9.8 Salida Almacén Data Access Object

<b>Nombre de la Clase:</b> SalidaAlmacenDAO		
<b>Descripción de la clase</b> Clase que administra de forma lógica las operaciones realizadas a la base de datos con respecto a las salidas de elementos de la sección de almacén		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>

SalidaAlmacenVisitor	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo SalidaAlmacenVisitor de acuerdo a un código de Salida	
findAll	Método que retorna un ArrayList(Colección de objetos) de tipo SalidaAlmacenVisitor con los datos de todas las Salidas de elementos a la sección de Almacén	
findByCodigoElemento	Método que retorna un ArrayList(Colección de objetos) de tipo SalidaAlmacenVisitor con los datos de las Salidas de un determinado elemento	
findByFechaSalida	Método que retorna un ArrayList(Colección de objetos) de tipo SalidaAlmacenVisitor con los datos de las Salidas de elementos en una determinada fecha	
createSalidaAlmacen	Método que permite registrar la salida de un elemento a almacén	
<b>Observaciones:</b>		

### 11.1.10 Rubro

#### 11.1.10.1 Rubro Value Object

<b>Nombre de la Clase:</b> RubroVO		
<b>Descripción de la clase</b> Clase que representa de forma lógica los rubros.		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
String str_codRubro	Identifica de manera única un objeto de tipo RubroVO	
String str_desRubro	Indica la descripción de un rubro	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>RoI Clase Asociada</b>
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
RubroVO	Método constructor con parámetros y sin parámetros	
getCodigo	Retorna el código del rubro	
getDescripcion	Retorna la descripción del rubro	
setCodigo	Permite asignar o cambiar el código del rubro	

setDescripcion	Permite asignar o cambiar la descripción del rubro
<b>Observaciones:</b>	

### 11.1.10.2 Rubro Manager

<b>Nombre de la Clase</b> RubroMgr		
<b>Descripción de la clase</b> Clase que permite administrar los rubros manejados en caja menor. Tiene toda la lógica de la clase RubroVO a nivel de modelo		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
ArrayList al_aux	Arraylist (colección de obj) que almacena los datos de los rubros como resultado de las consultas	
String v_str_mensaje	Cadena que almacena el mensaje que retorna una determinada operación	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RubroVO	Retorna	Asociación
RubroDAO	utiliza	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que permite establecer los atributos de un objeto de tipo RubroVO para la consulta de un Rubro de acuerdo a un código.	
findByDescripcion	Método que permite establecer los atributos de un objeto de tipo RubroVO para la consulta de un rubro de acuerdo a una descripción.	
findByAll	Método que permite acceder a la consulta de todos los rubros manejados en caja menor.	
createRubro	Método que permite establecer los atributos de un objeto de tipo RubroVO para la creación de un rubro determinado.	
updateRubro	Método que permite establecer los atributos de un objeto de tipo RubroVO para la actualización de un rubro determinado.	
deleteRubro	Método que permite establecer los atributos de un objeto de tipo RubroVO para la eliminación de un rubro determinado	
<b>Observaciones:</b>		

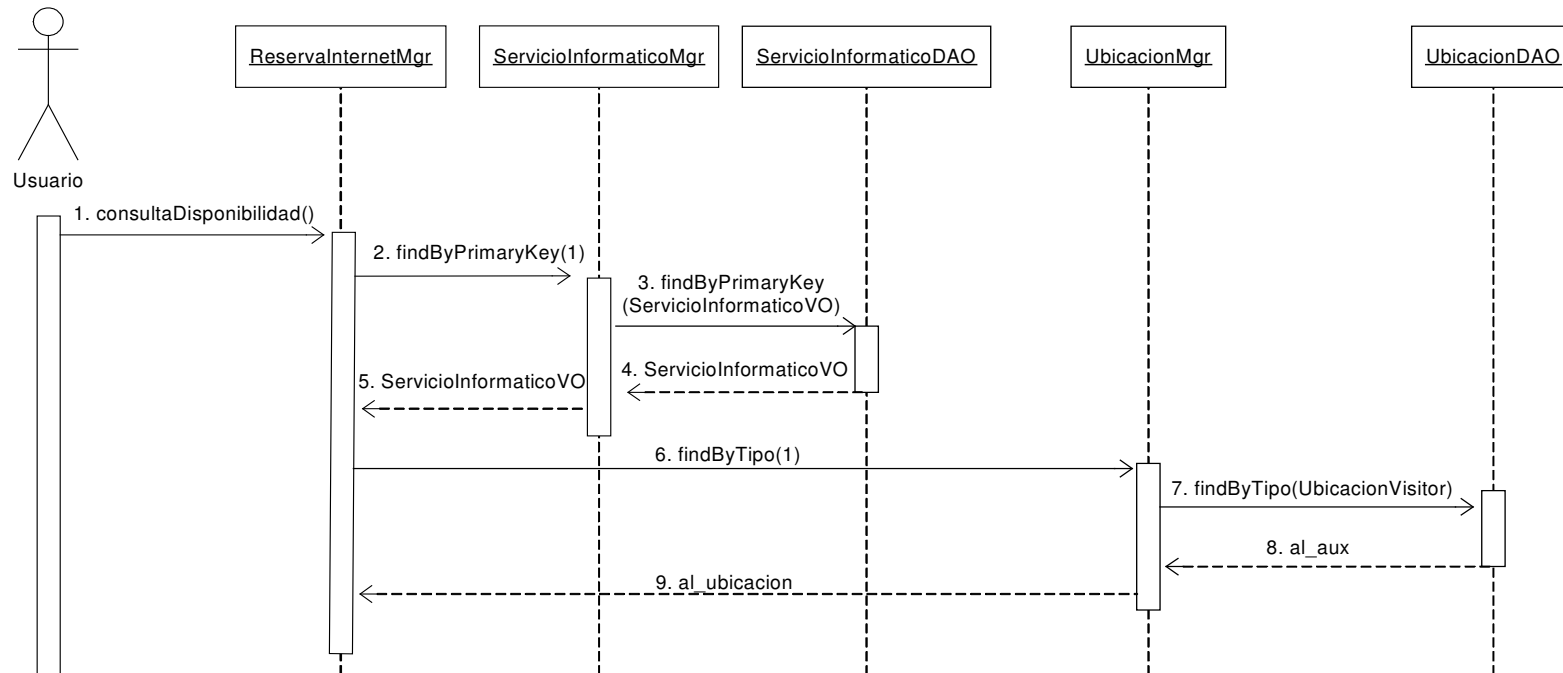
### 11.1.10.3 Rubro Data Access Object

<b>Nombre de la Clase:</b> RubroDAO		
<b>Descripción de la clase</b> Clase que permite acceder de forma lógica a las operaciones realizadas a la Base de Datos con respecto a los rubros		
<b>Propiedad</b>	<b>Descripción del modelaje, restricciones, información para diseño, suposiciones</b>	
Connection conn	Recibe la conexión creada	
ArrayList al_aux	Arreglo que almacena los resultados de las consultas	
<b>Clase Asociada</b>	<b>Nombre Asociación</b>	<b>Rol Clase Asociada</b>
RubroVO	Retorna	Asociación
DBConnection	Solicita	Asociación
<b>Servicio o Responsabilidad</b>	<b>Descripción</b>	
findByPrimaryKey	Método que retorna un objeto de tipo RubroVO con la descripción del rubro de acuerdo a un Código determinado.	
findByDescripcion	Método que retorna un objeto tipo RubroVO de acuerdo a una determinada descripción.	
findByAll	Método que retorna un ArrayList de objetos de tipo RubroVO con los datos de todos los rubros manejados en caja menor.	
createRubro	Método que permite crear un determinado Rubro	
updateRubro	Método que permite actualizar un determinado Rubro	
deleteRubro	Método que permite eliminar un determinado Rubro	
<b>Observaciones:</b>		

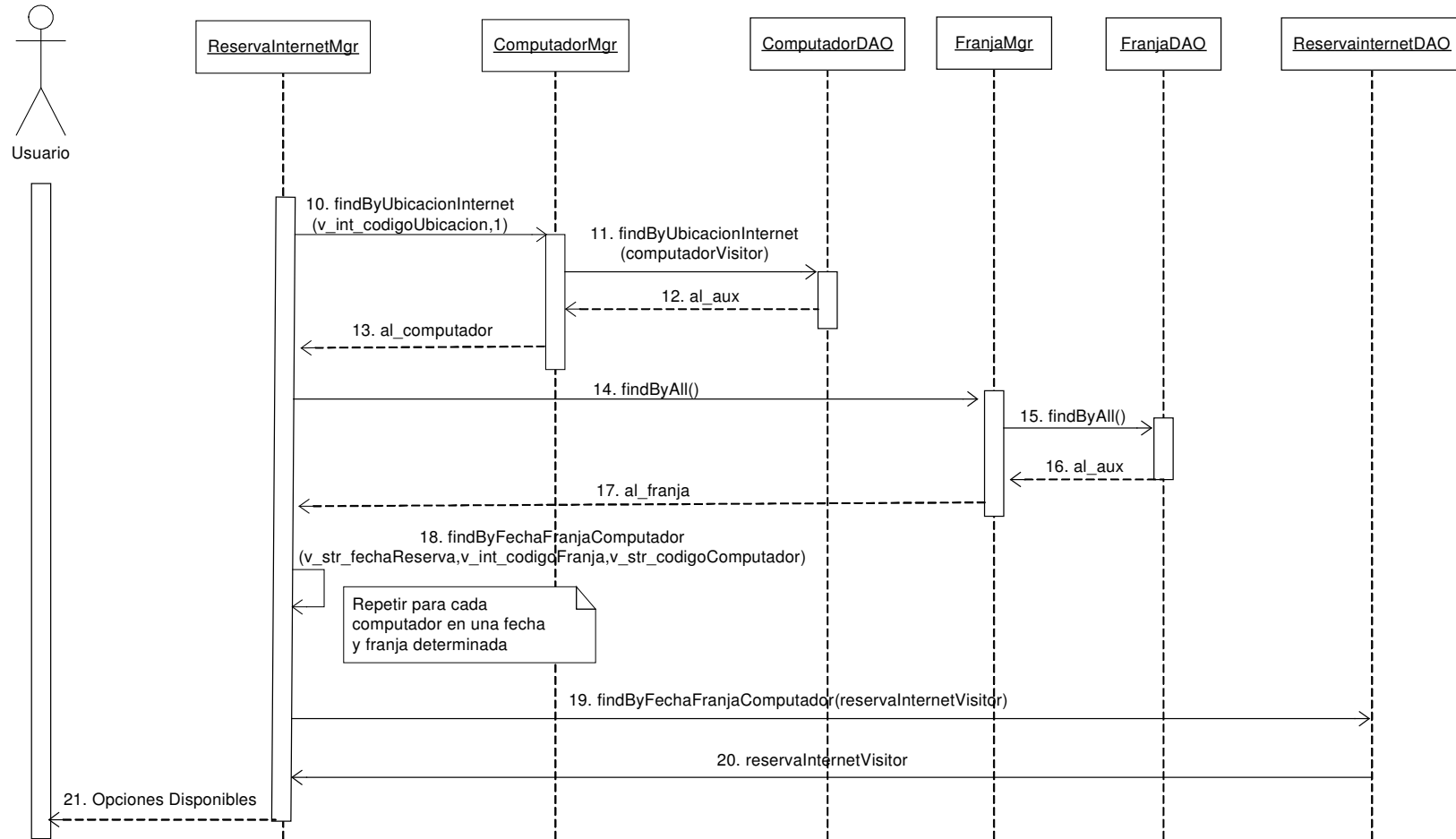
## 11.2 DIAGRAMAS DE SECUENCIA

### 11.2.1 Administración de Reservas de Internet

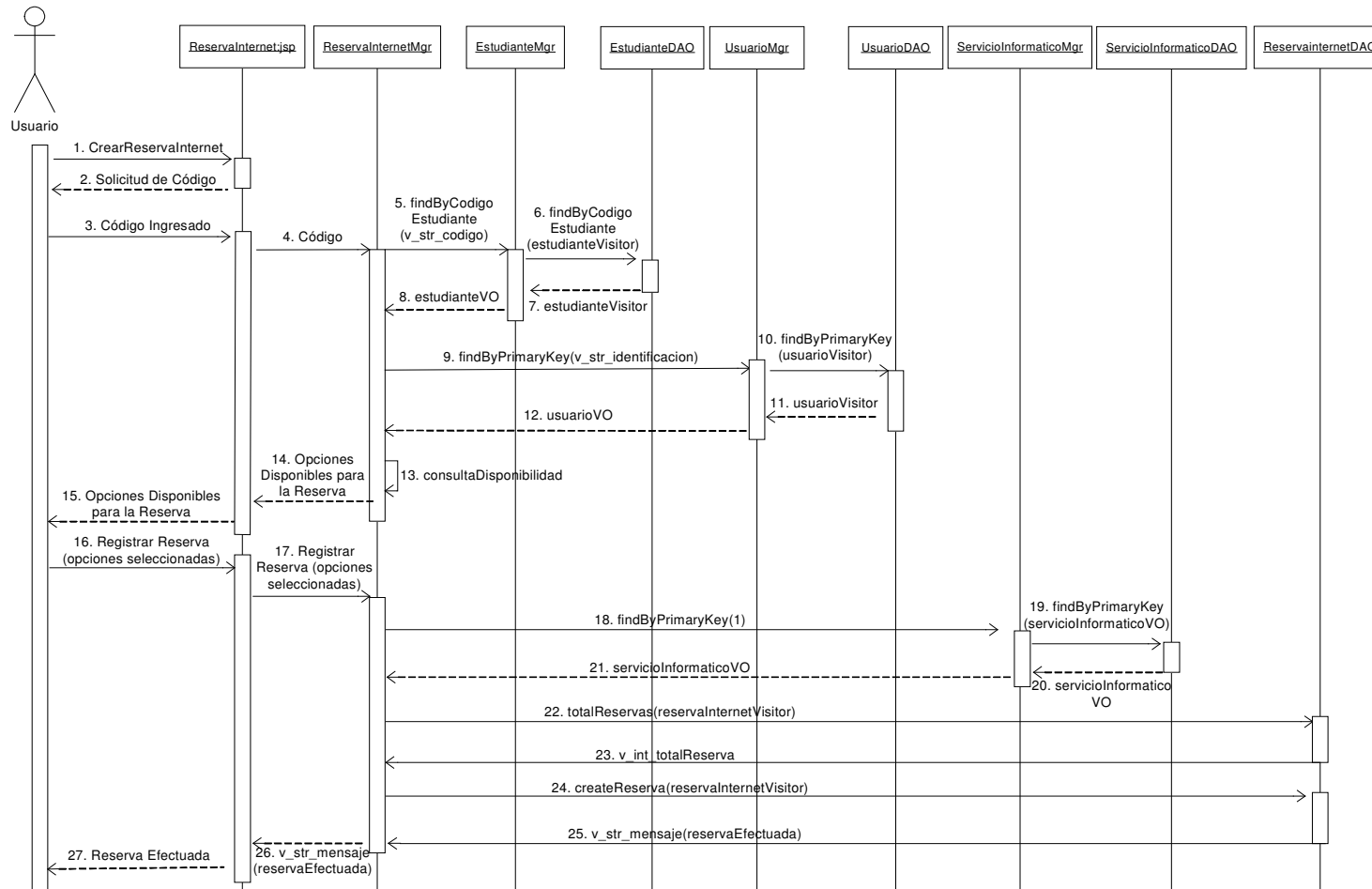
#### 11.2.1.1 Consulta Disponibilidad Reservas de Internet



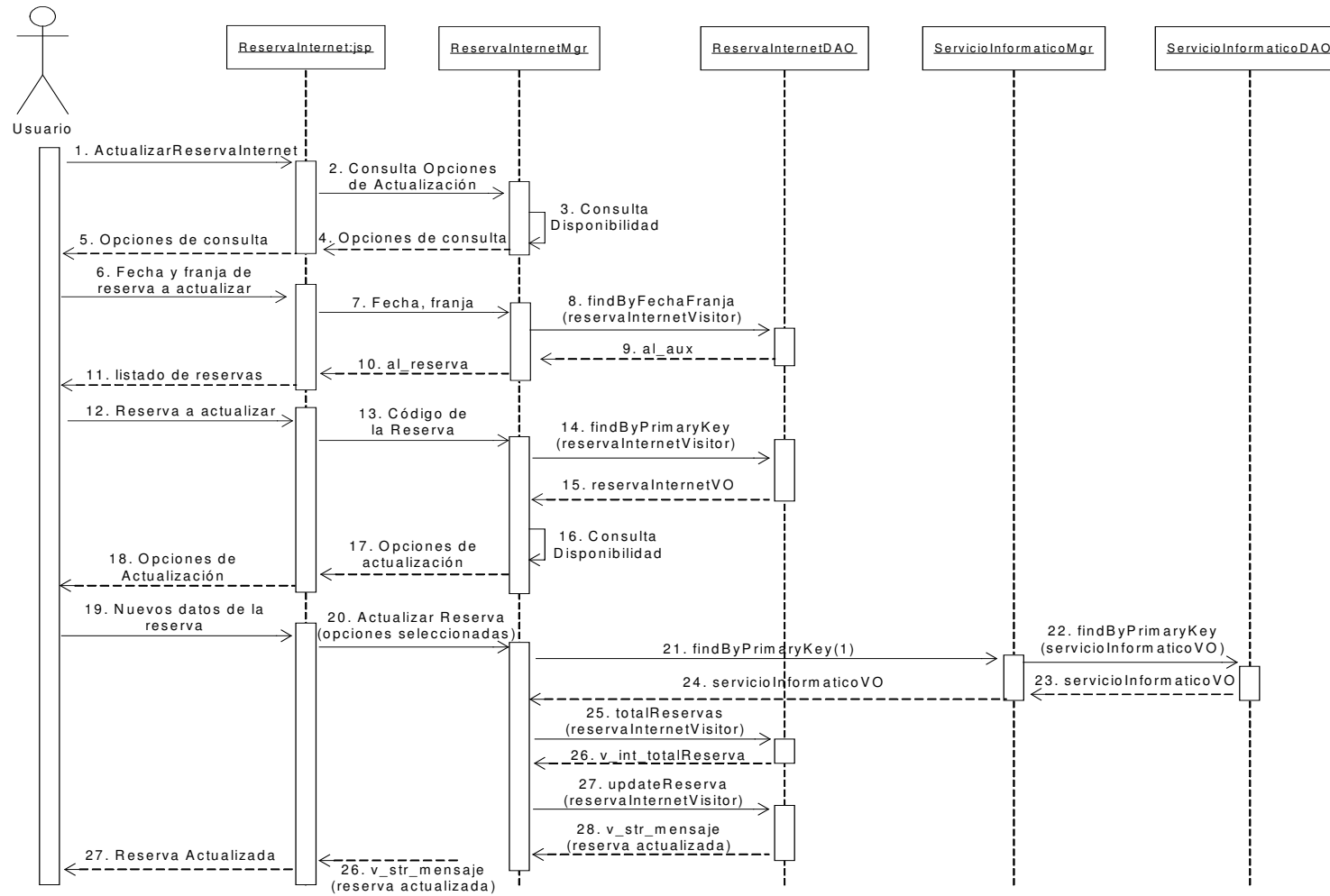
## Consulta Disponibilidad Reservas de Internet - Continuación



### 11.2.1.2 Creación Reserva de Internet

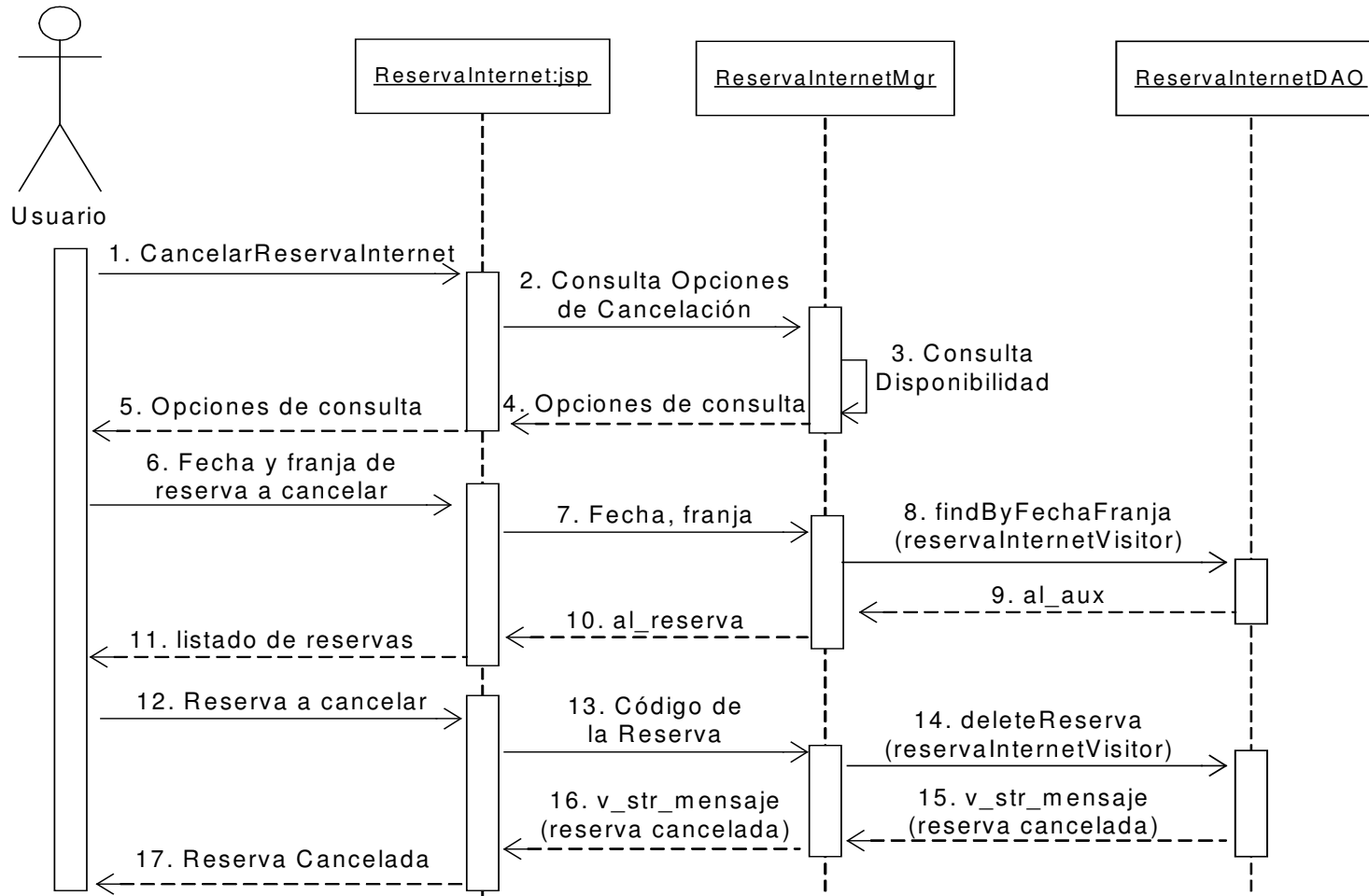


### 11.2.1.3 Actualización Reserva de Internet

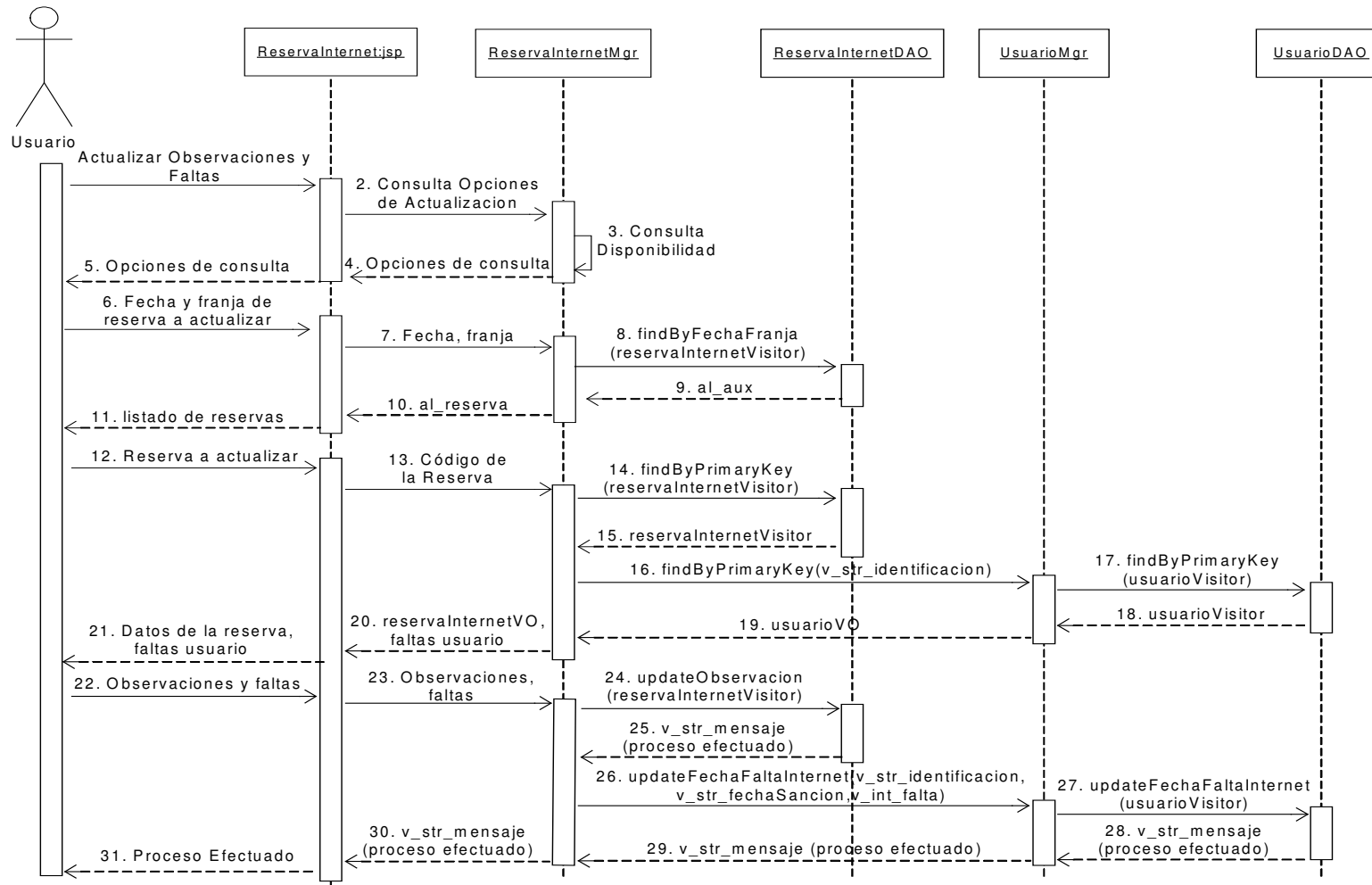




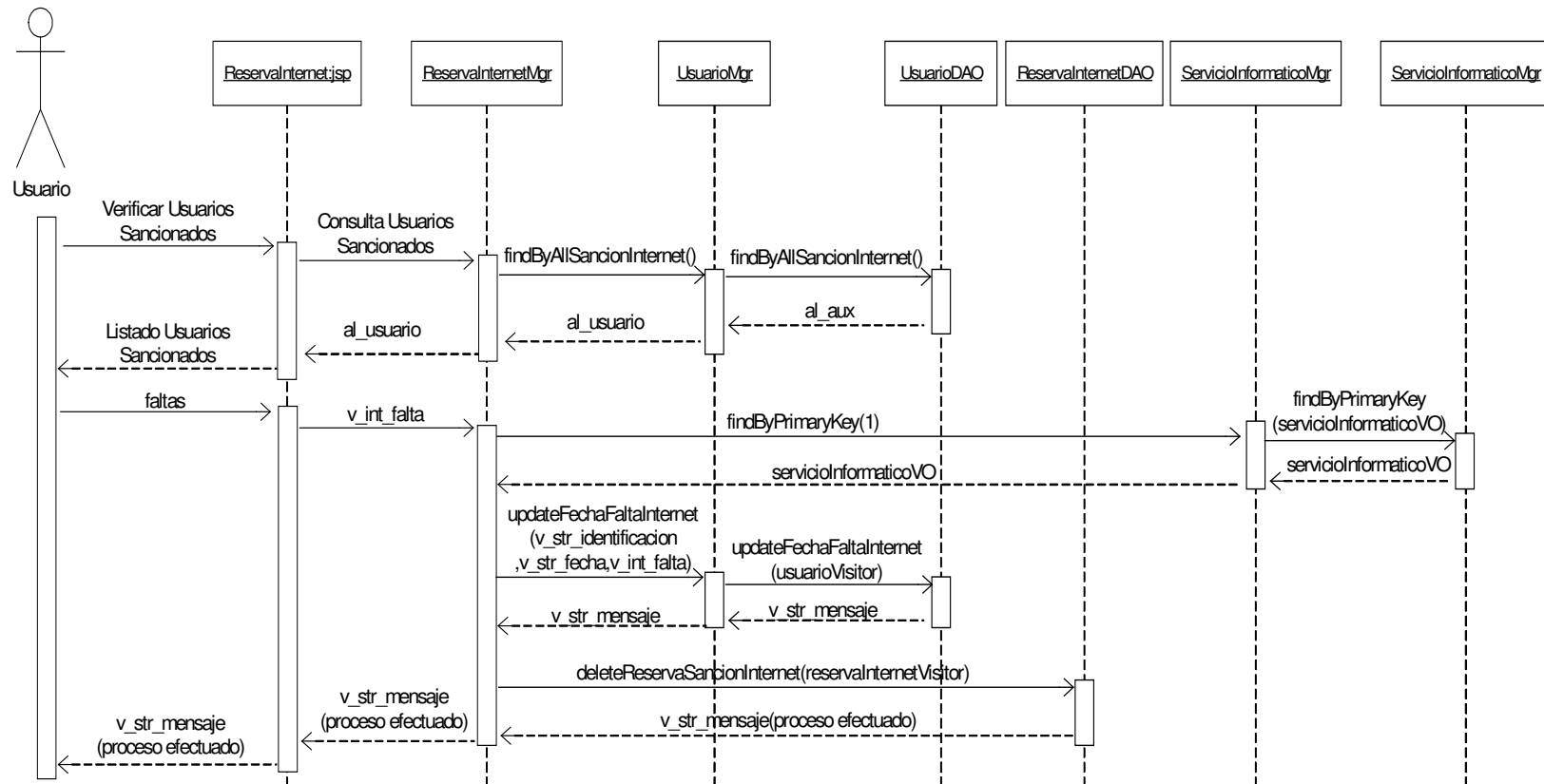
### 11.2.1.4 Eliminación Reserva de Internet



### 11.2.1.5 Actualización Observaciones y Faltas de Reservas de Internet

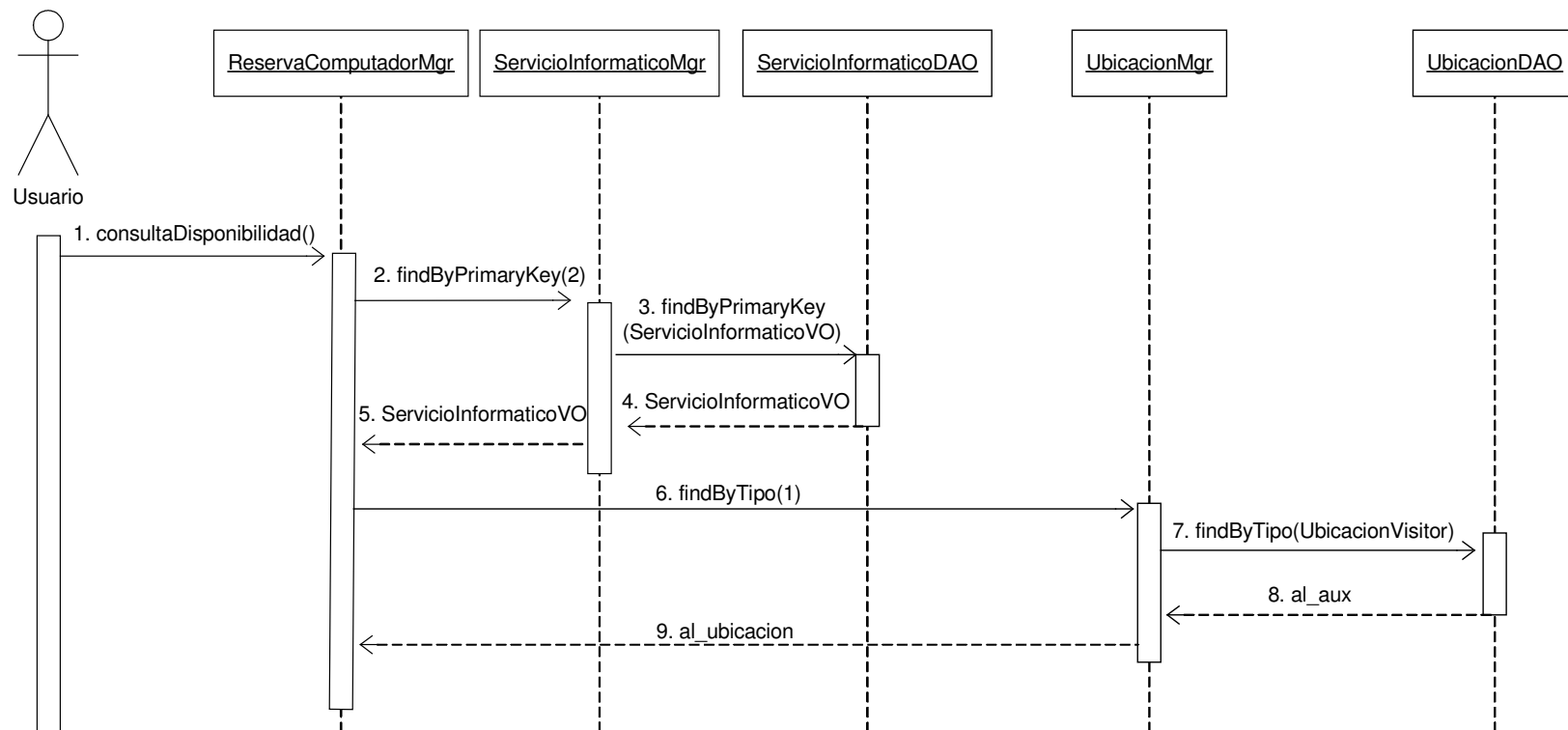


### 11.2.1.6 Verificar Usuarios Sancionados para Reservas de Internet

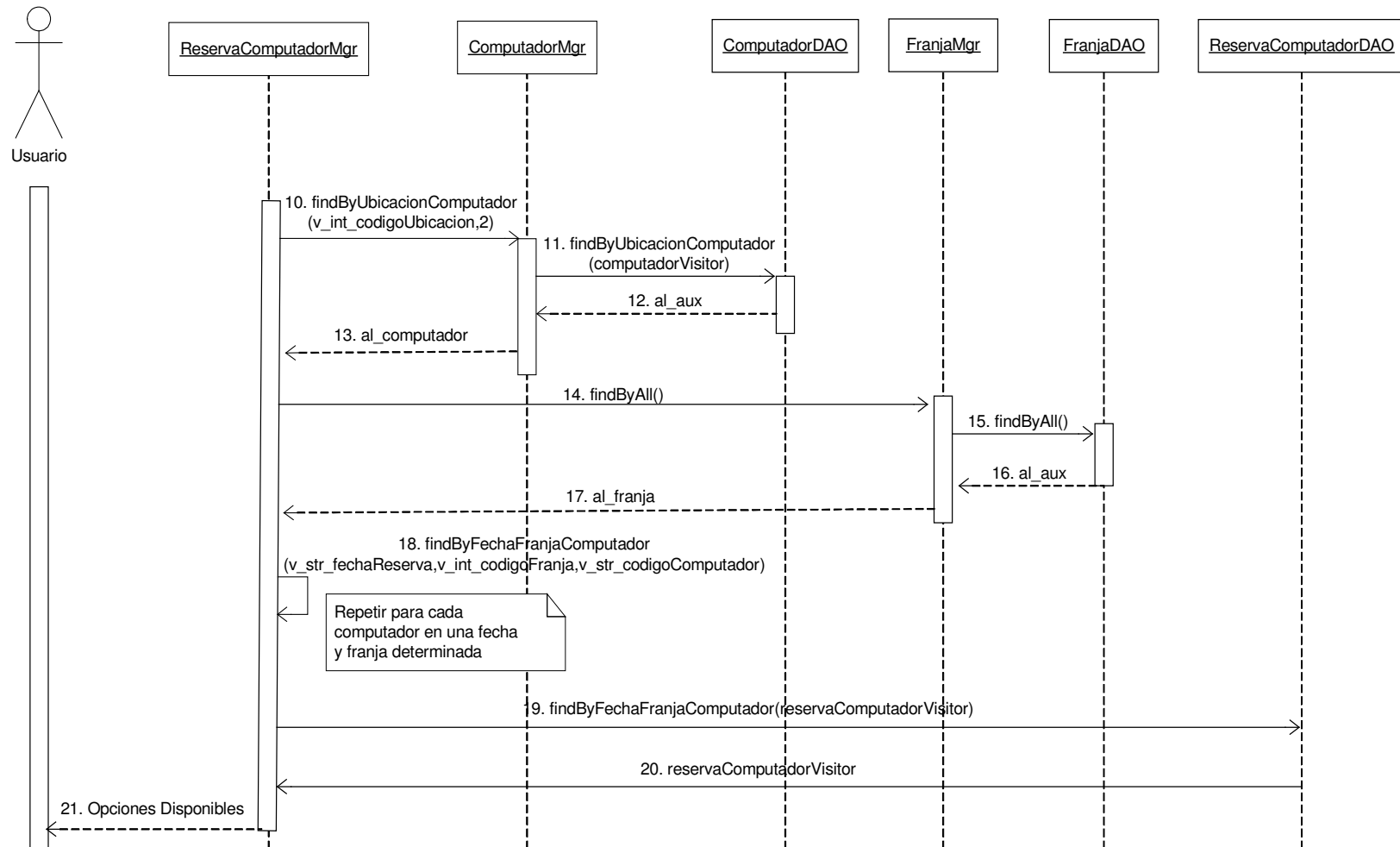


## 11.2.2 Administración de Reservas de Computadores

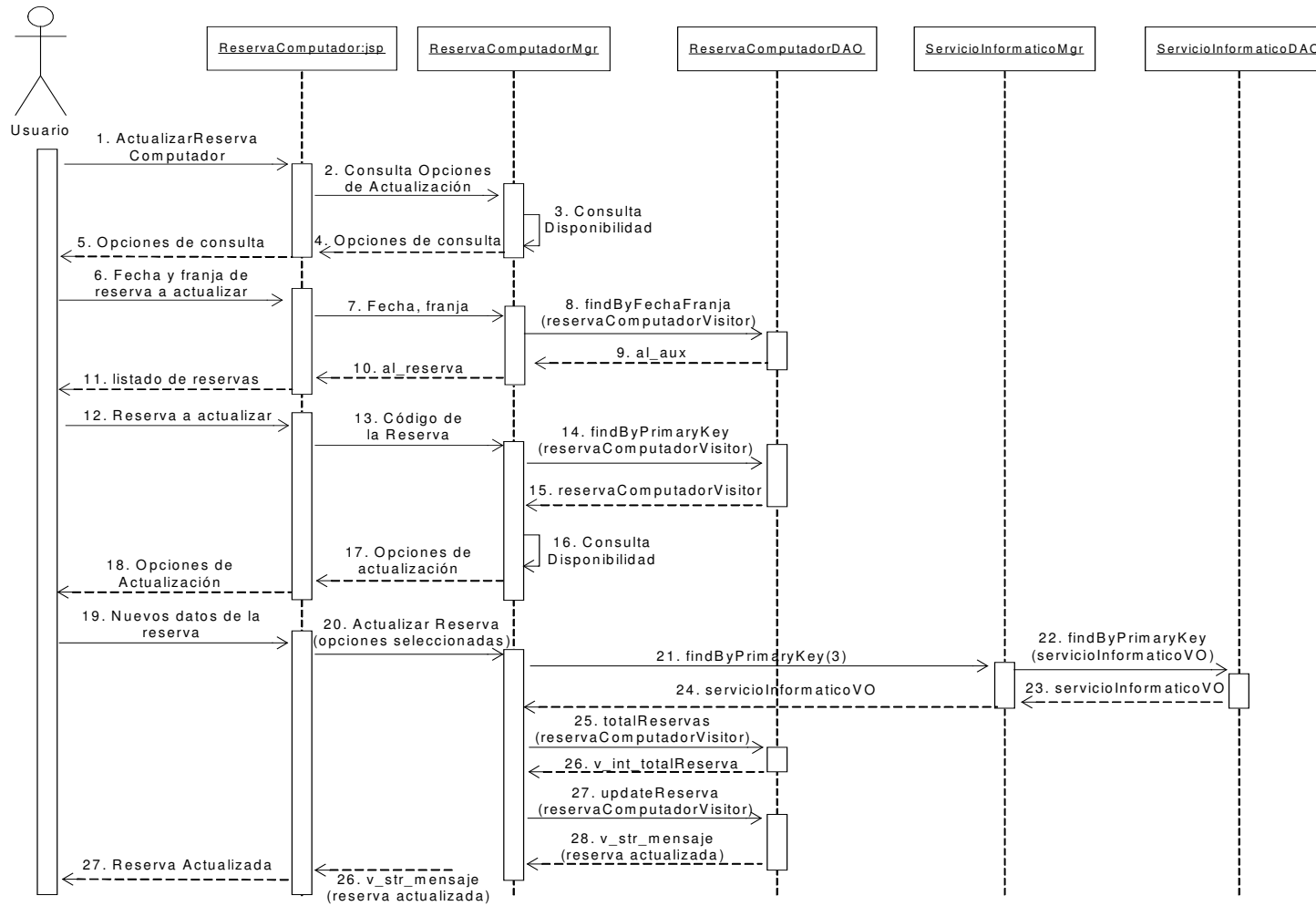
### 11.2.2.1 Consulta Disponibilidad Reservas de Computadores



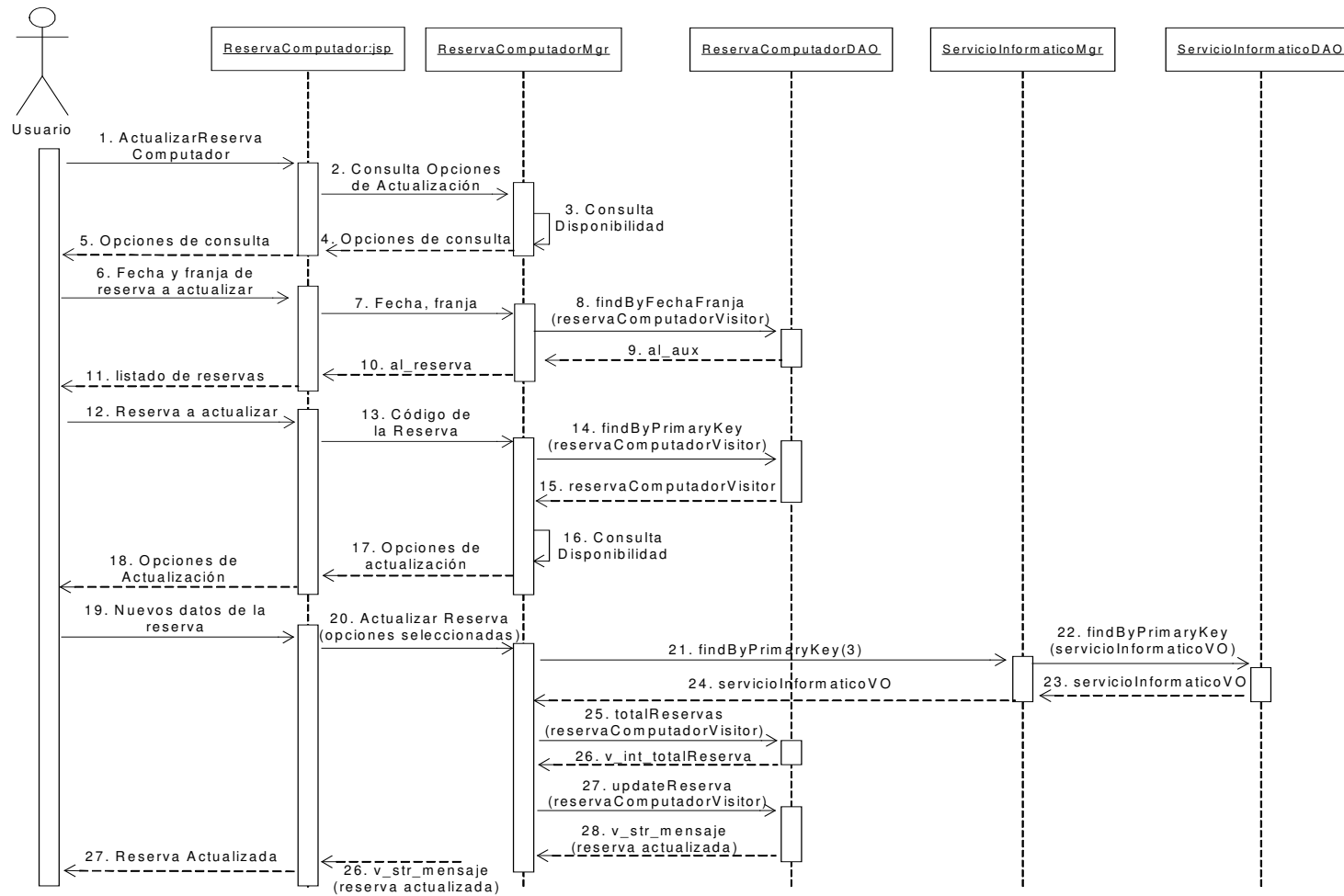
## Consulta Disponibilidad Reservas de Computadores - Continuación



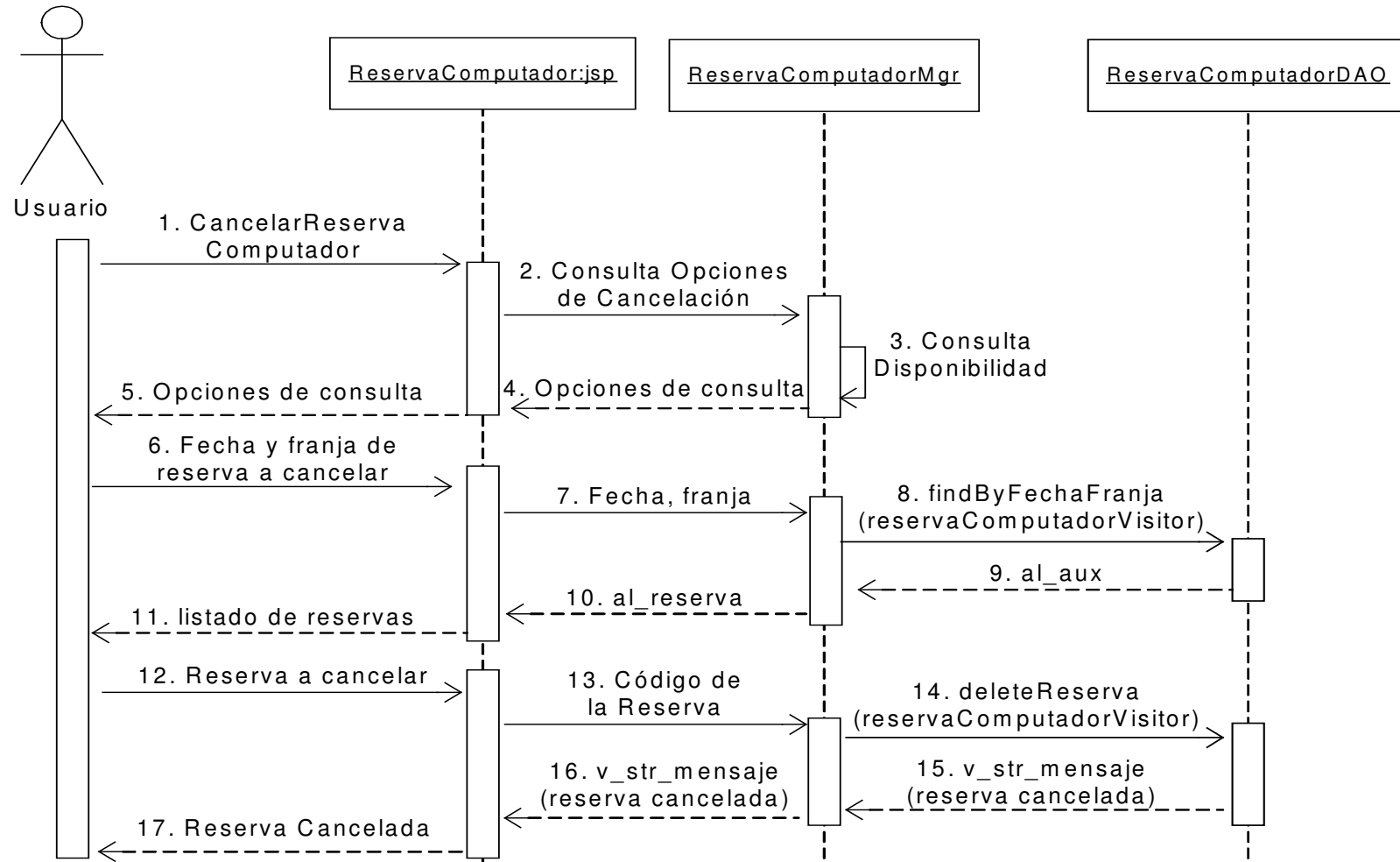
### 11.2.2.2 Creación Reserva de Computadores



### 11.2.2.3 Actualización Reserva de Computadores

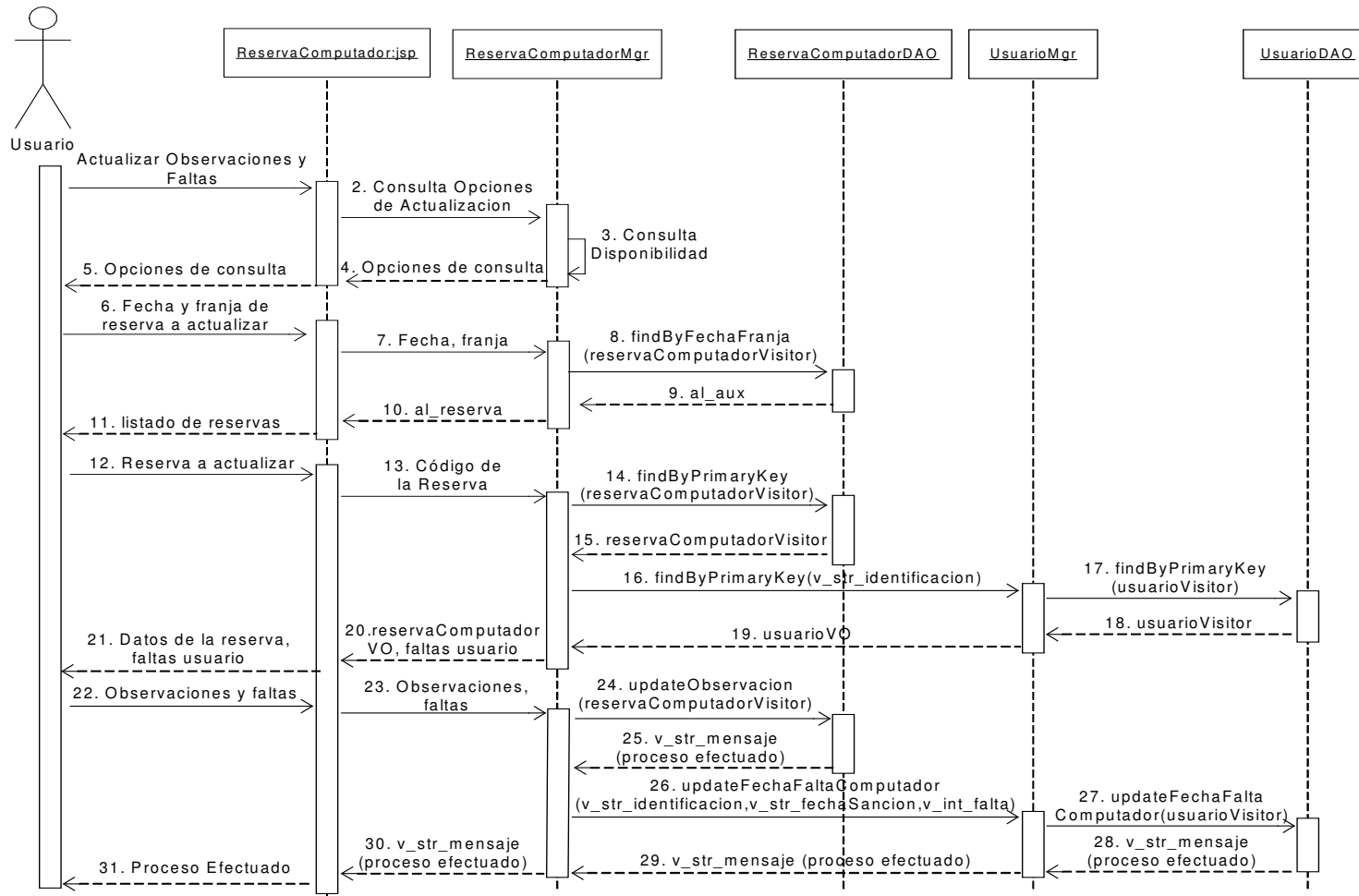


### 11.2.2.4 Eliminación Reserva de Computadores

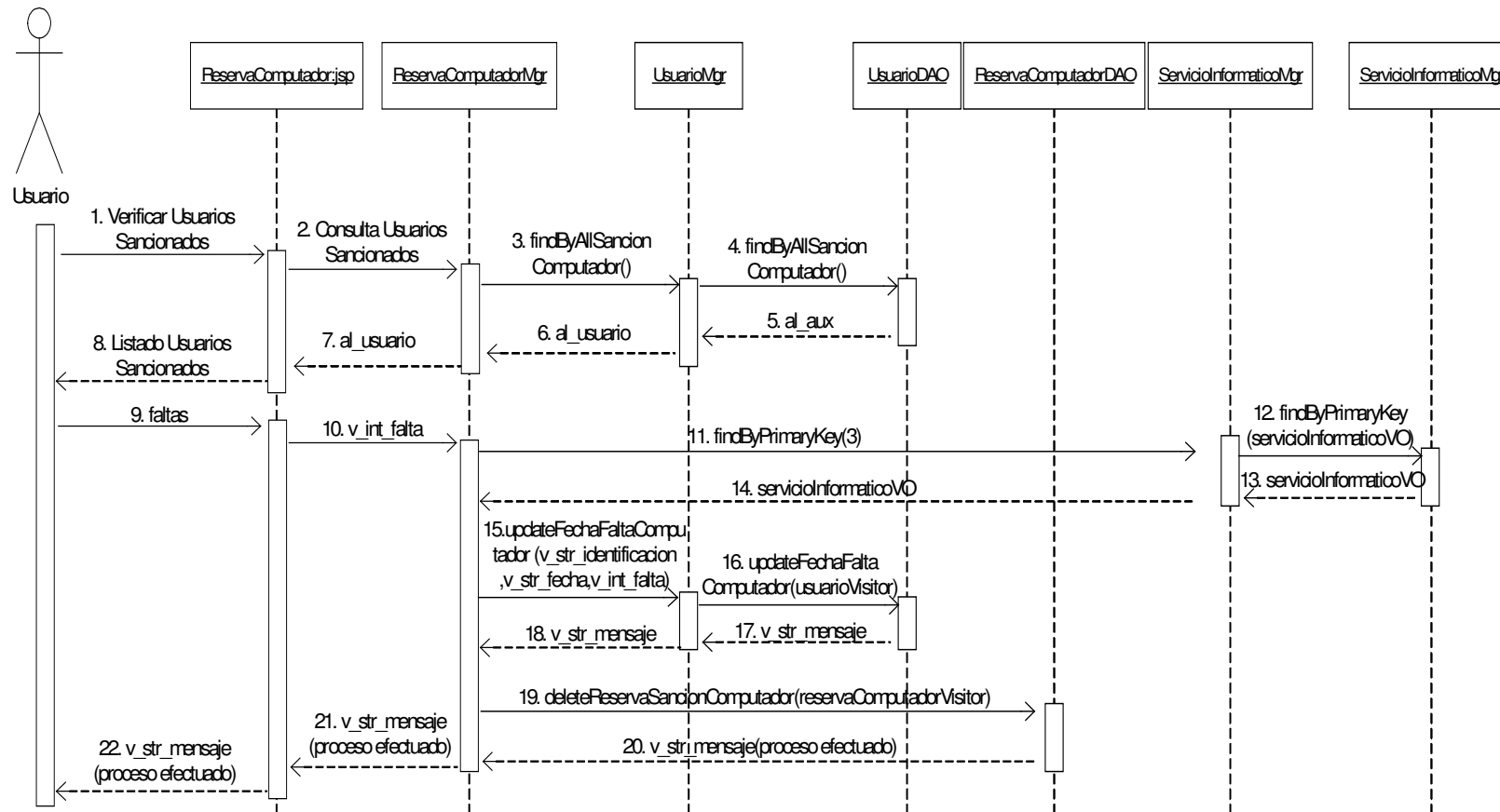




### 11.2.2.5 Actualización Observaciones y Faltas de Reservas de Computadores

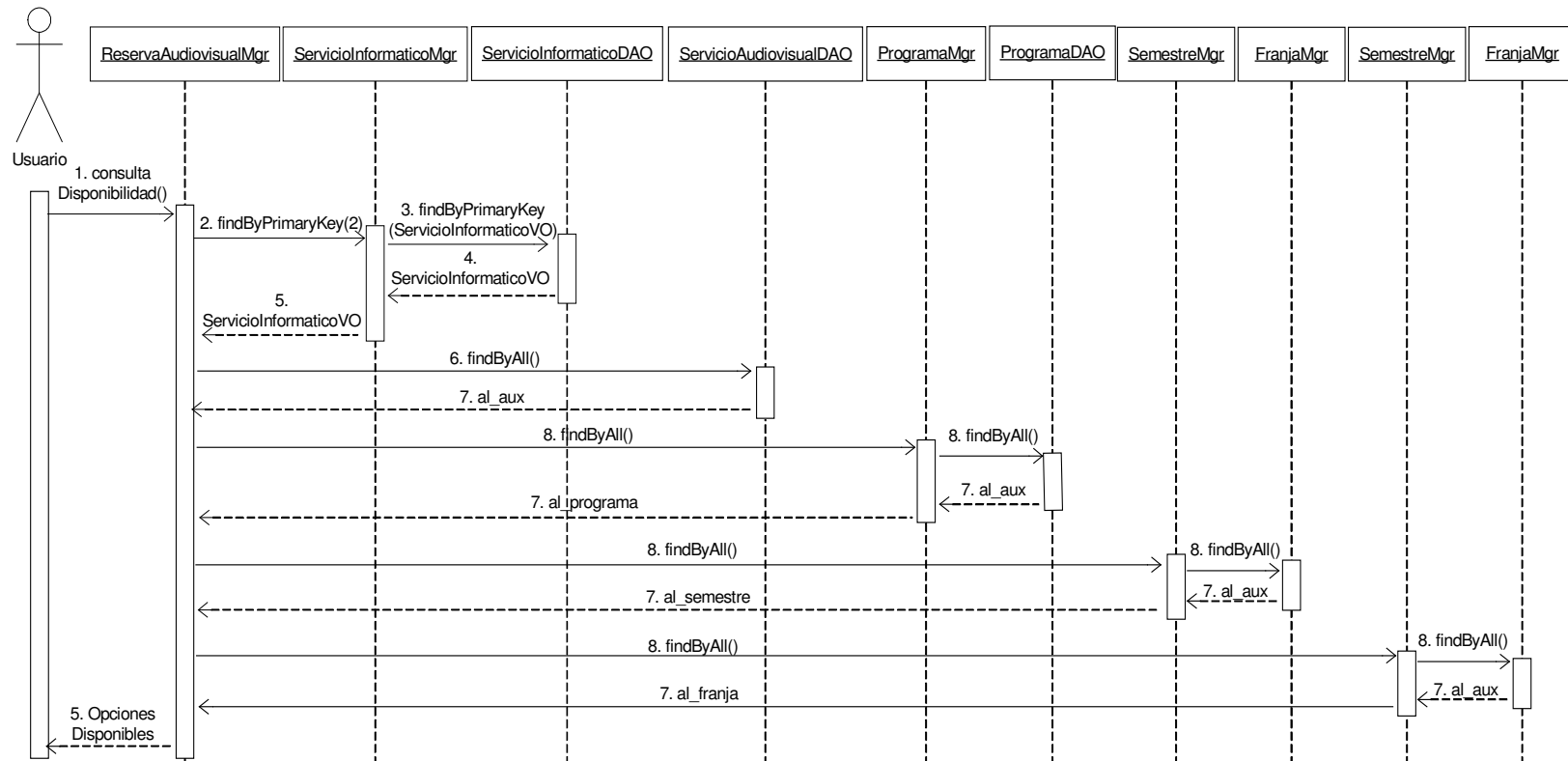


### 11.2.2.6 Verificar Usuarios Sancionados para Reservas de Computadores

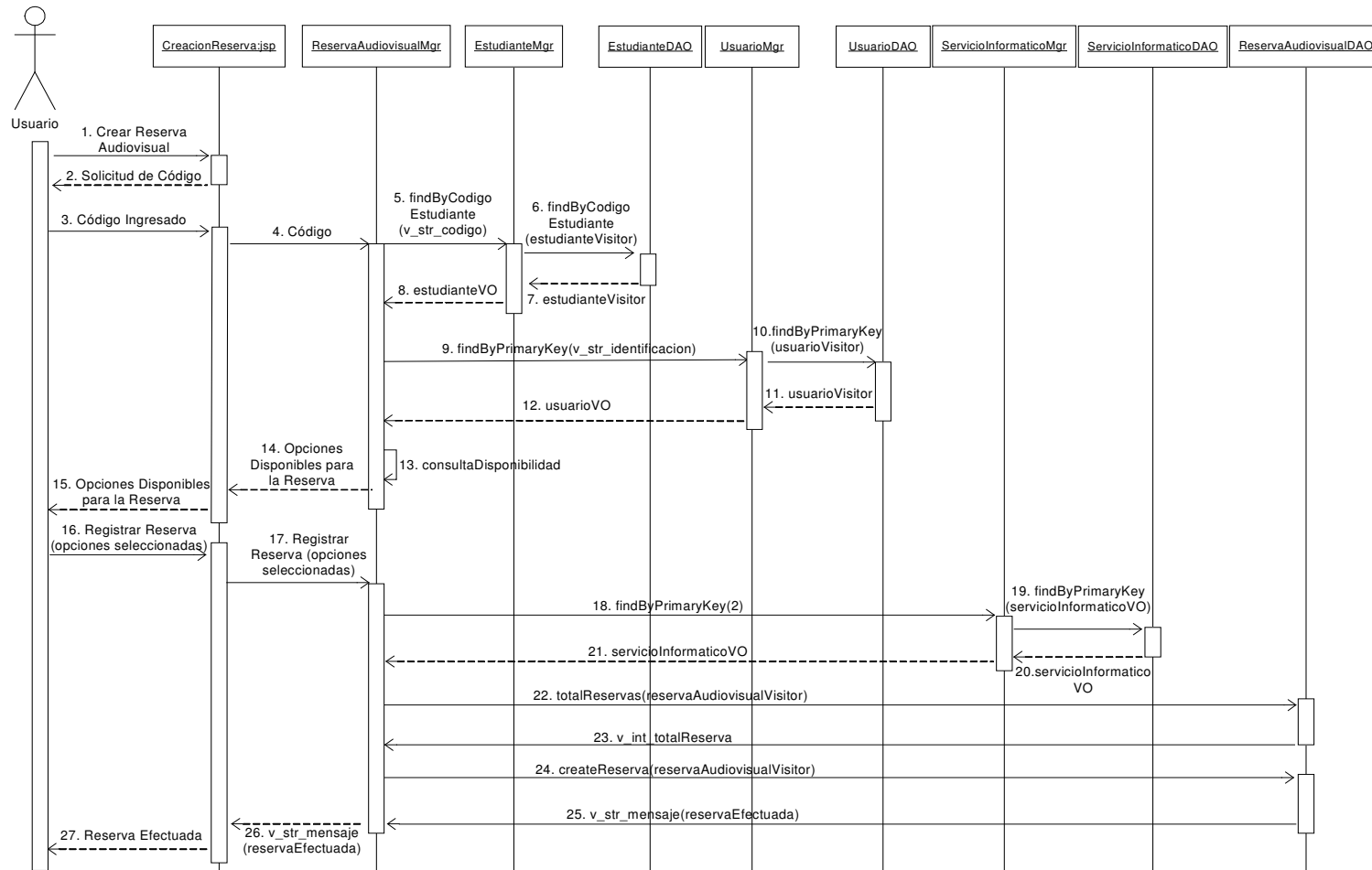


## 11.2.3 Administración de Reservas de Audiovisuales

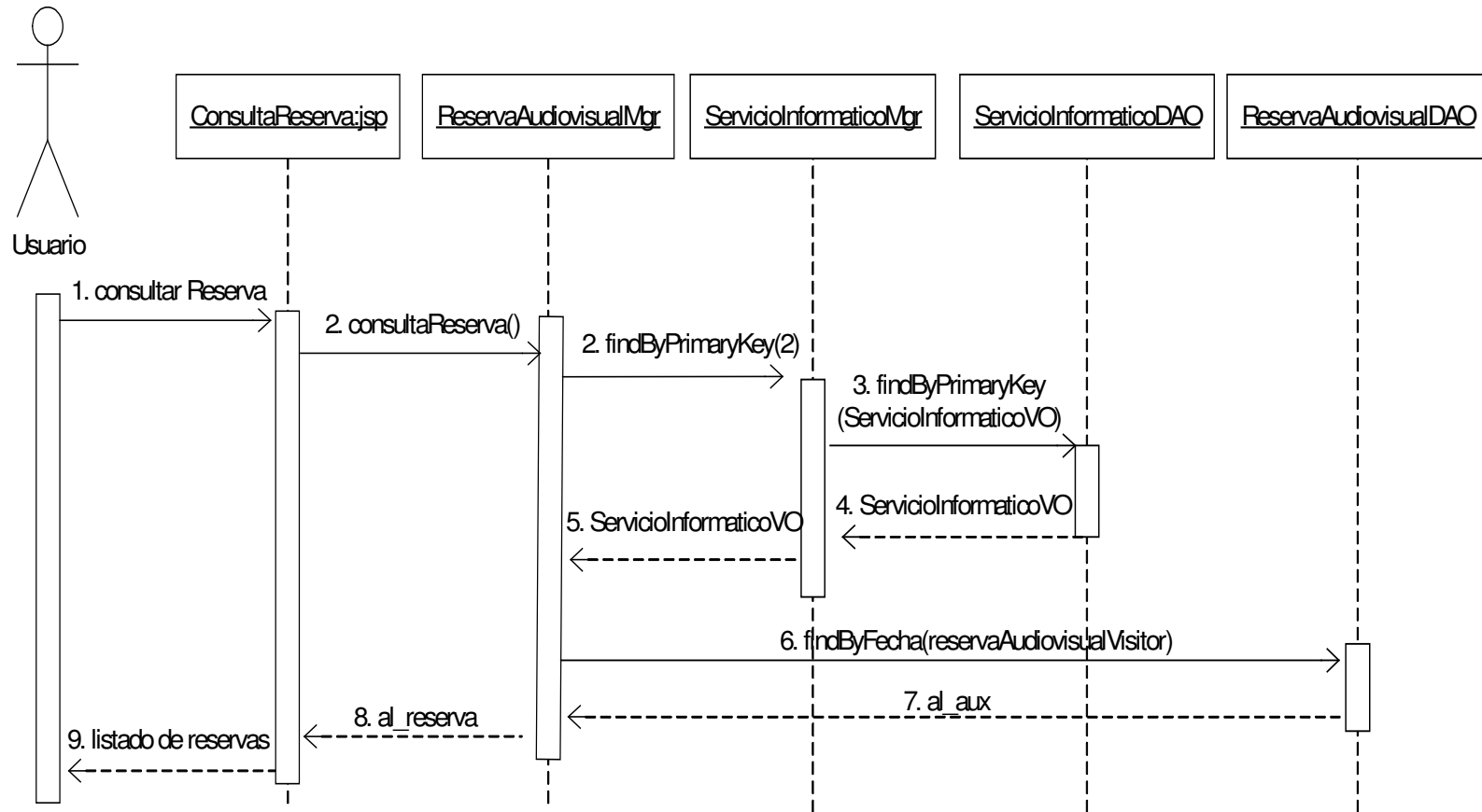
### 11.2.3.1 Consulta Disponibilidad Reservas de Audiovisuales



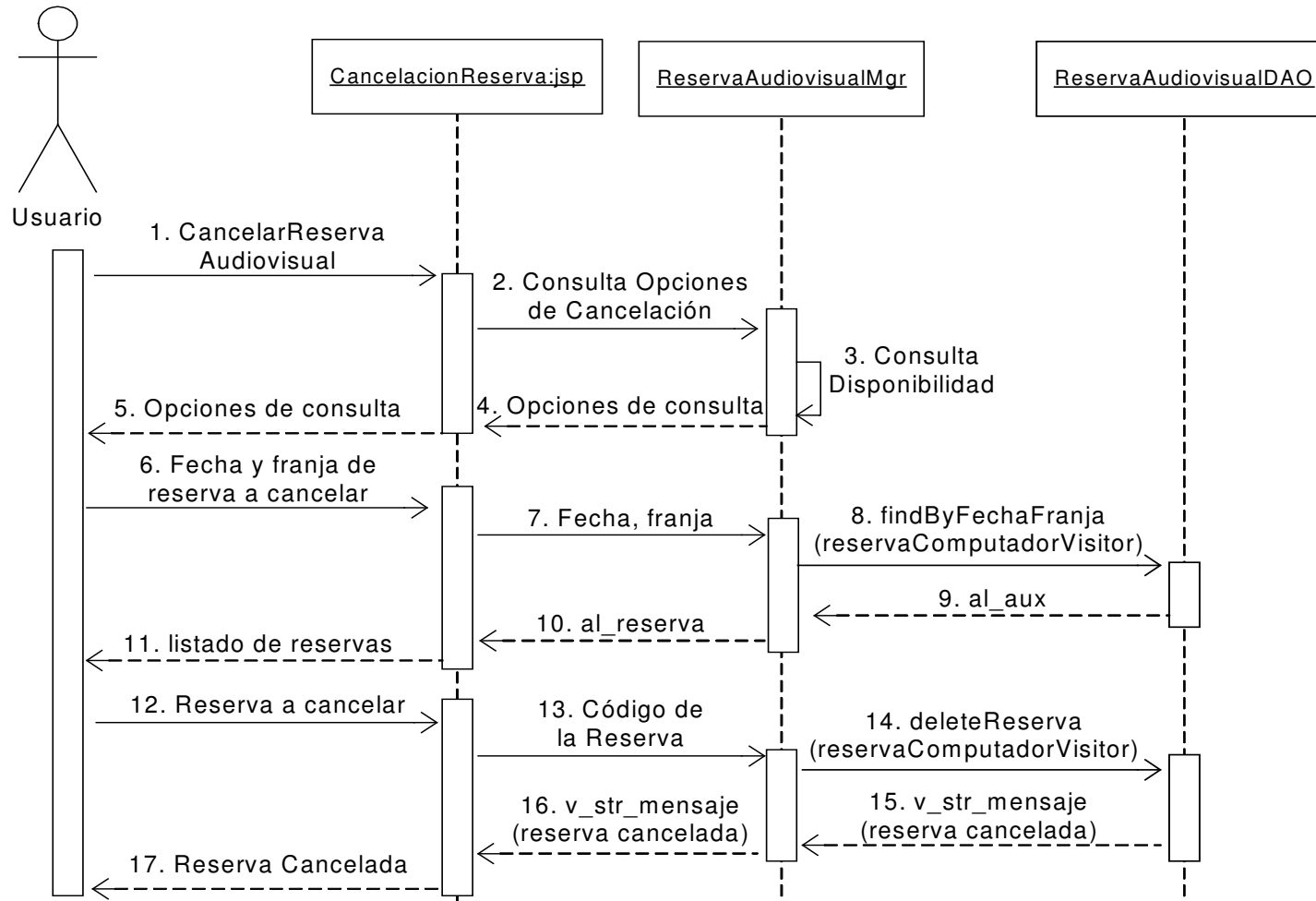
### 11. 2.3.2 Creación Reservas de Audiovisuales



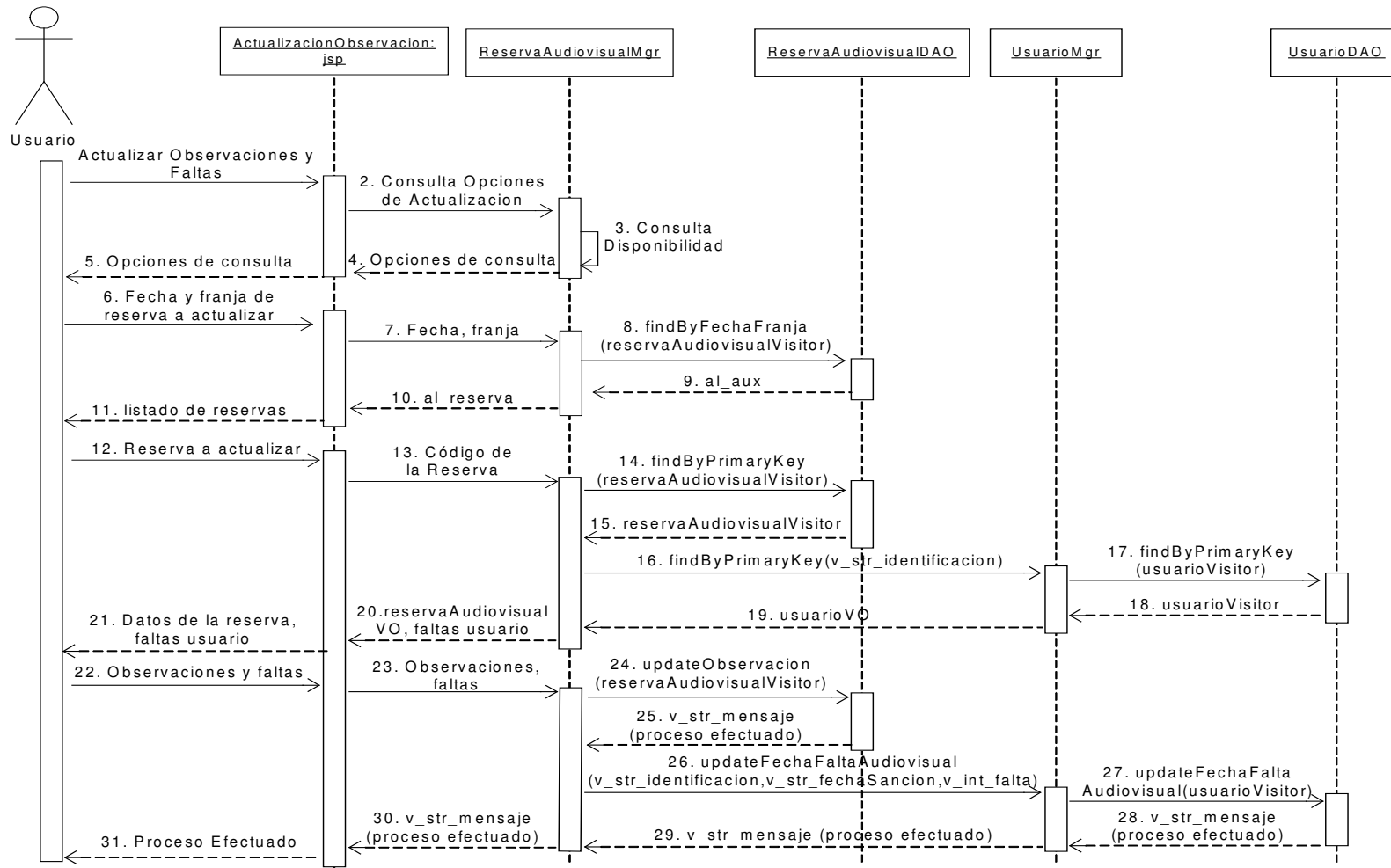
### 11. 2.3.3 Consulta Reservas de Audiovisuales



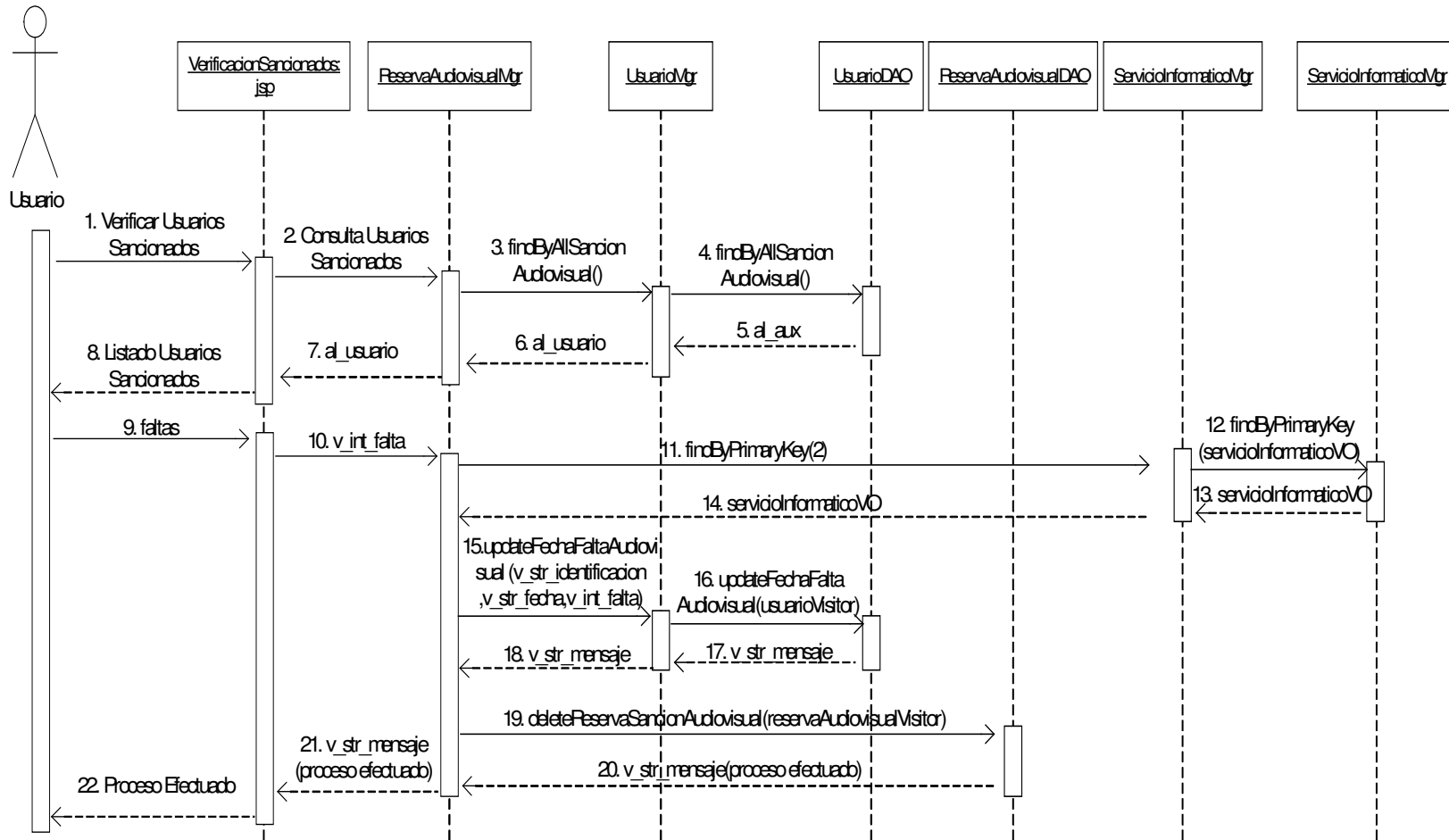
### 11. 2.3.4 Cancelación Reservas de Audiovisuales



### 11. 2.3.5 Actualización Observaciones y Faltas Reservas de Audiovisuales



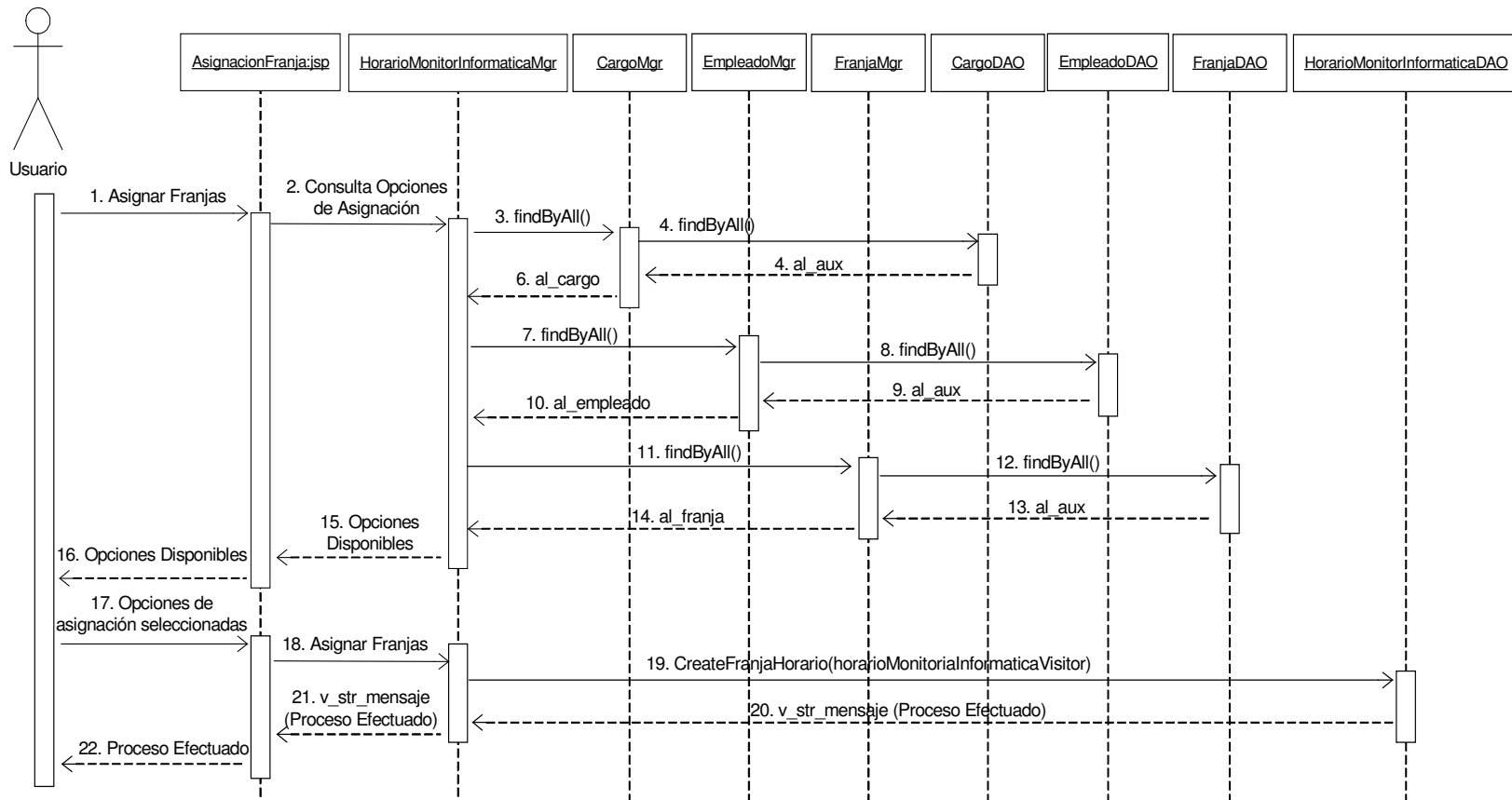
### 11. 2.3.6 Verificar Usuarios Sancionados para Reservas de Audiovisuales



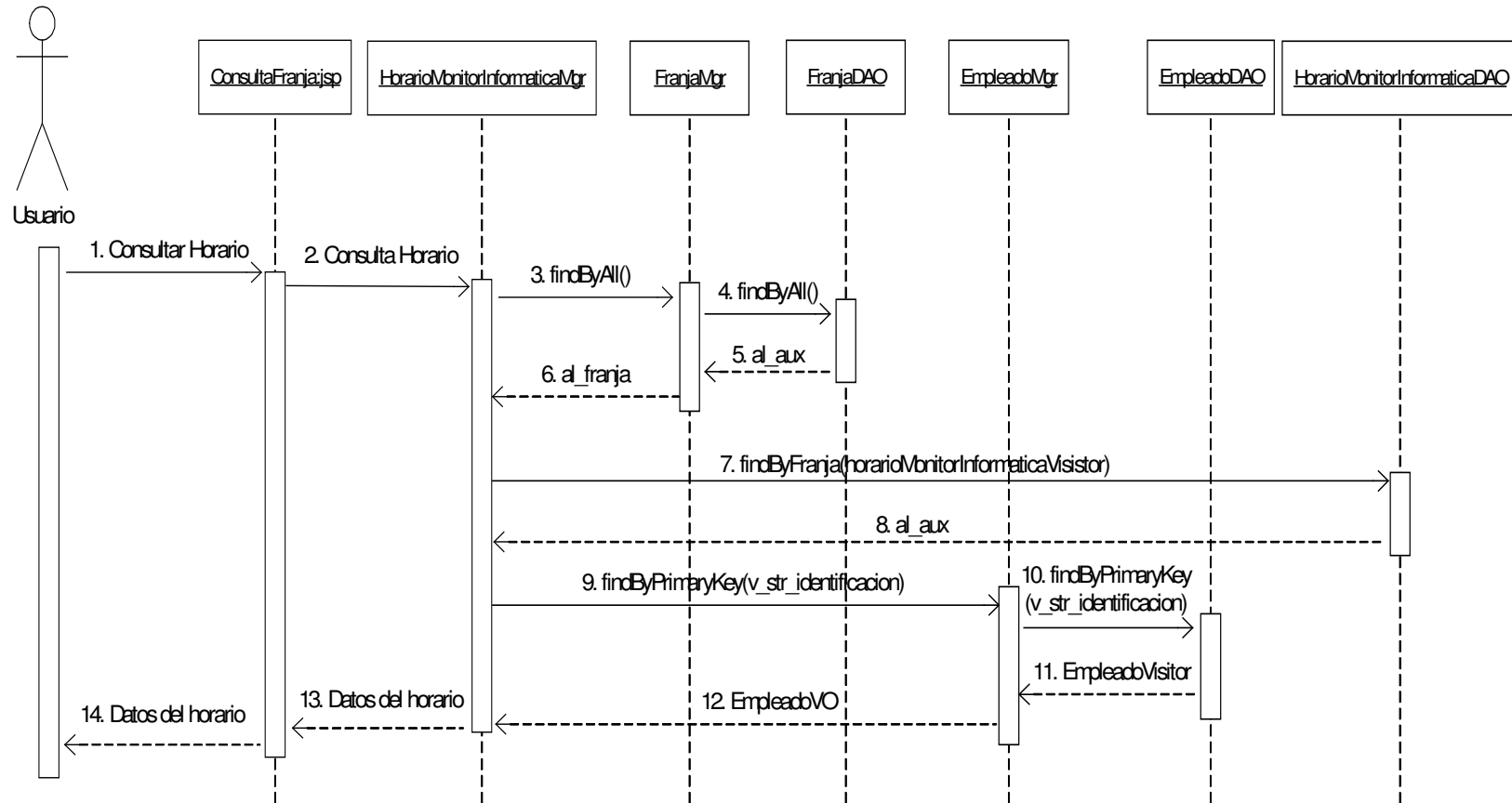


## 11.2.4 Administración de Horarios de Monitoría Informática

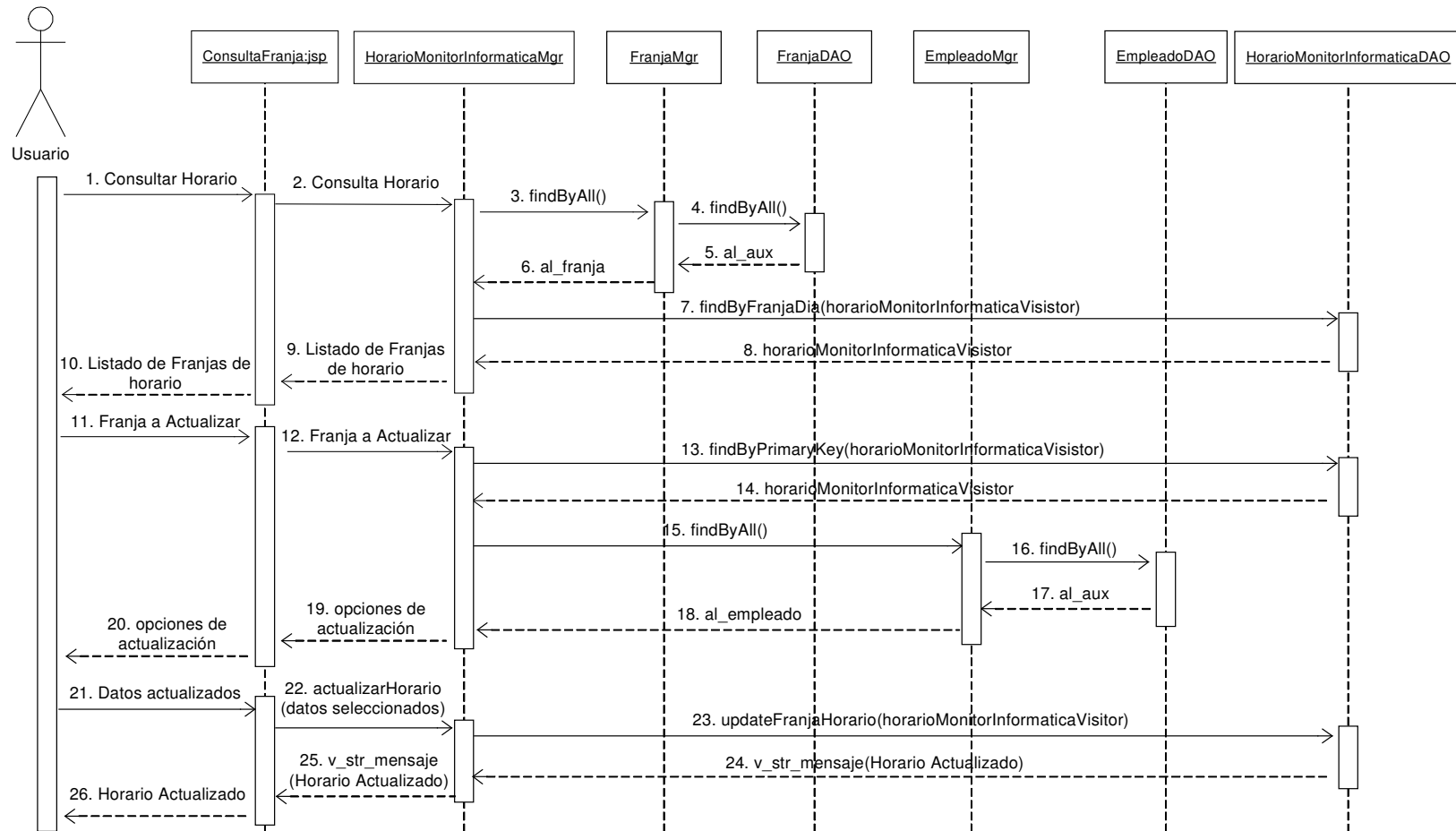
### 11.2.4.1 Asignar Franjas Horario Monitoría Informática



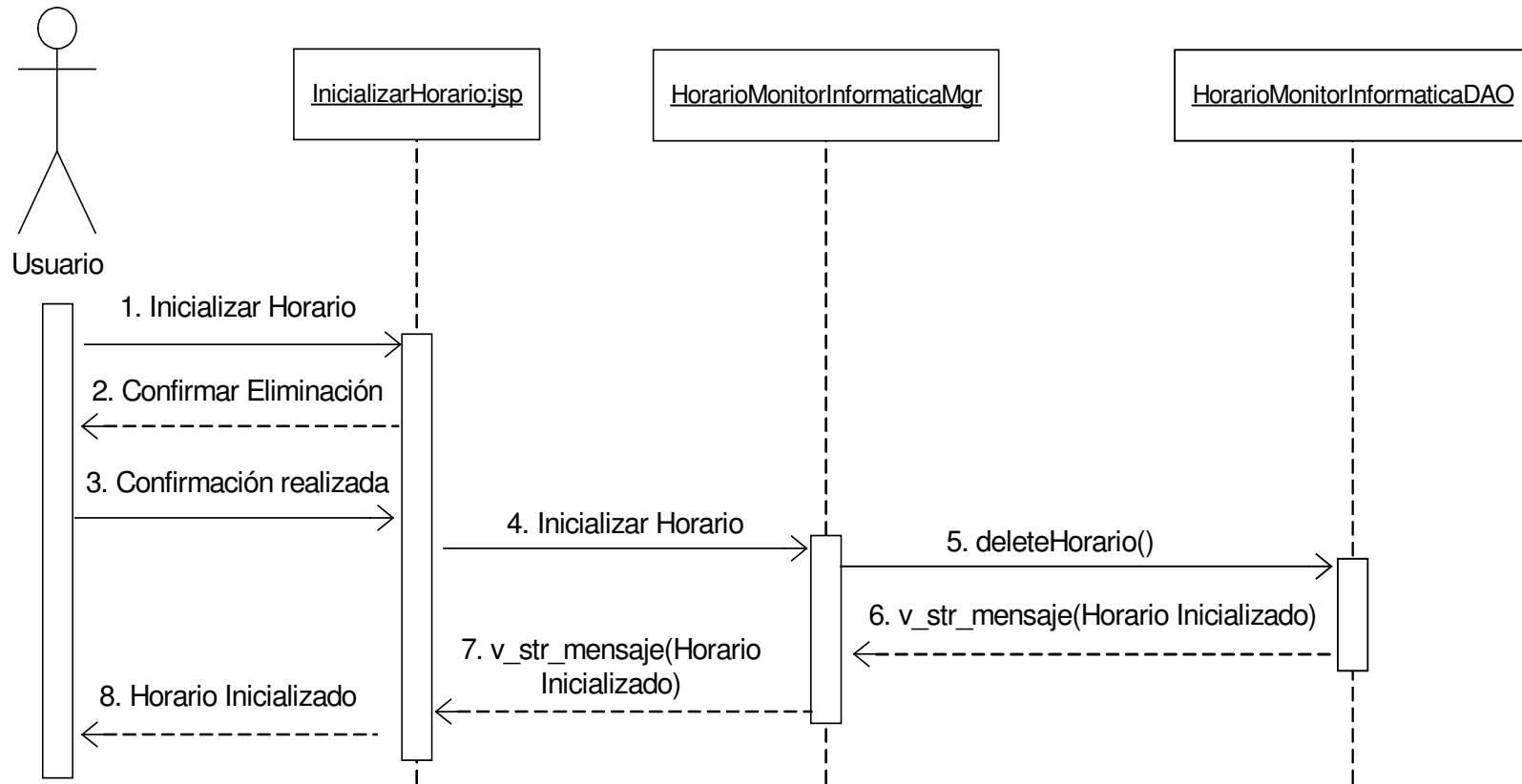
### 11.2.4.2 Consultar Horario Monitoría Informática



### 11.2.4.3 Actualizar Franja Horario Monitoría Informática

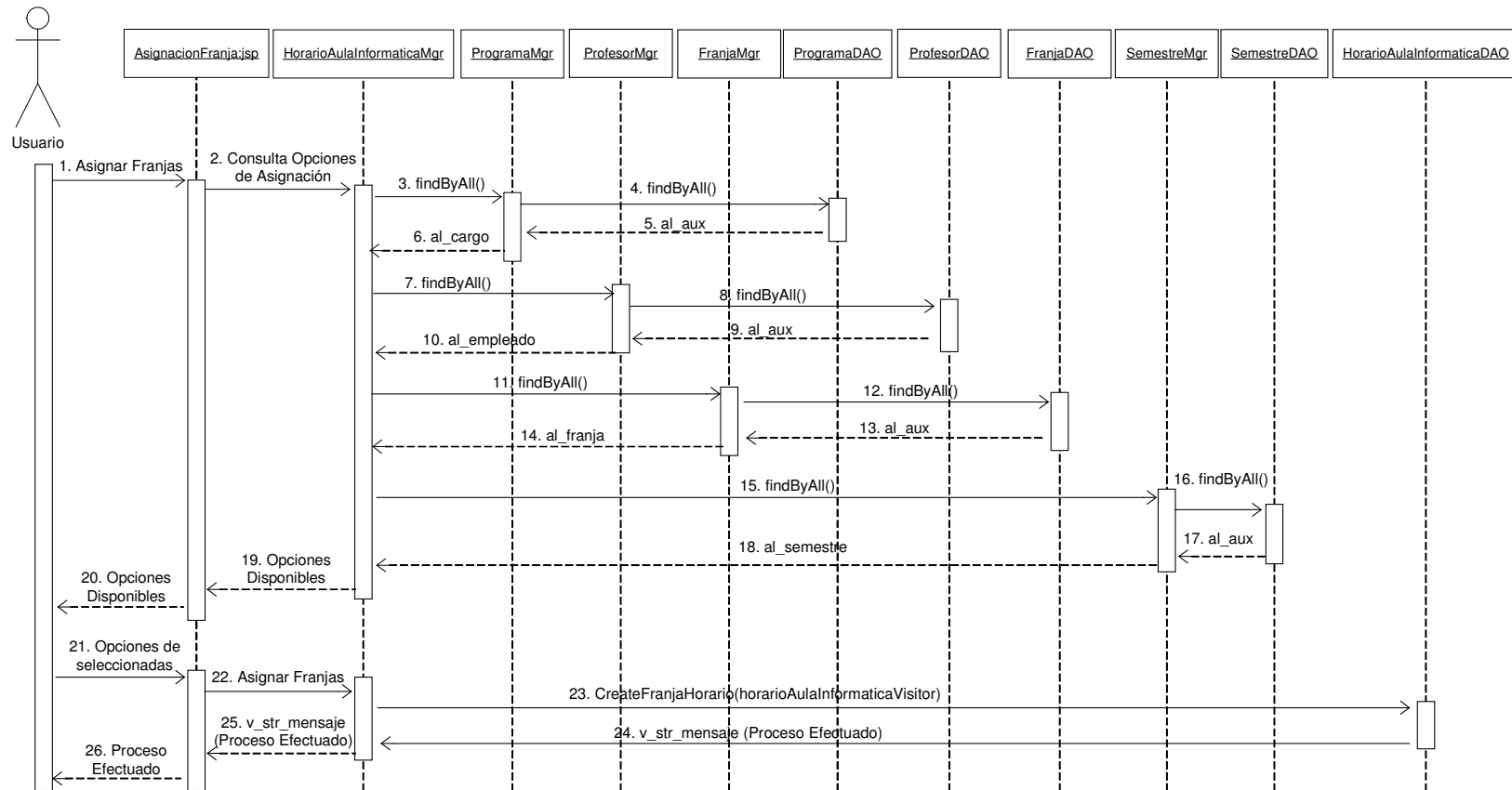


#### 11.2.4.4 Inicializar Horario Monitoría Informática

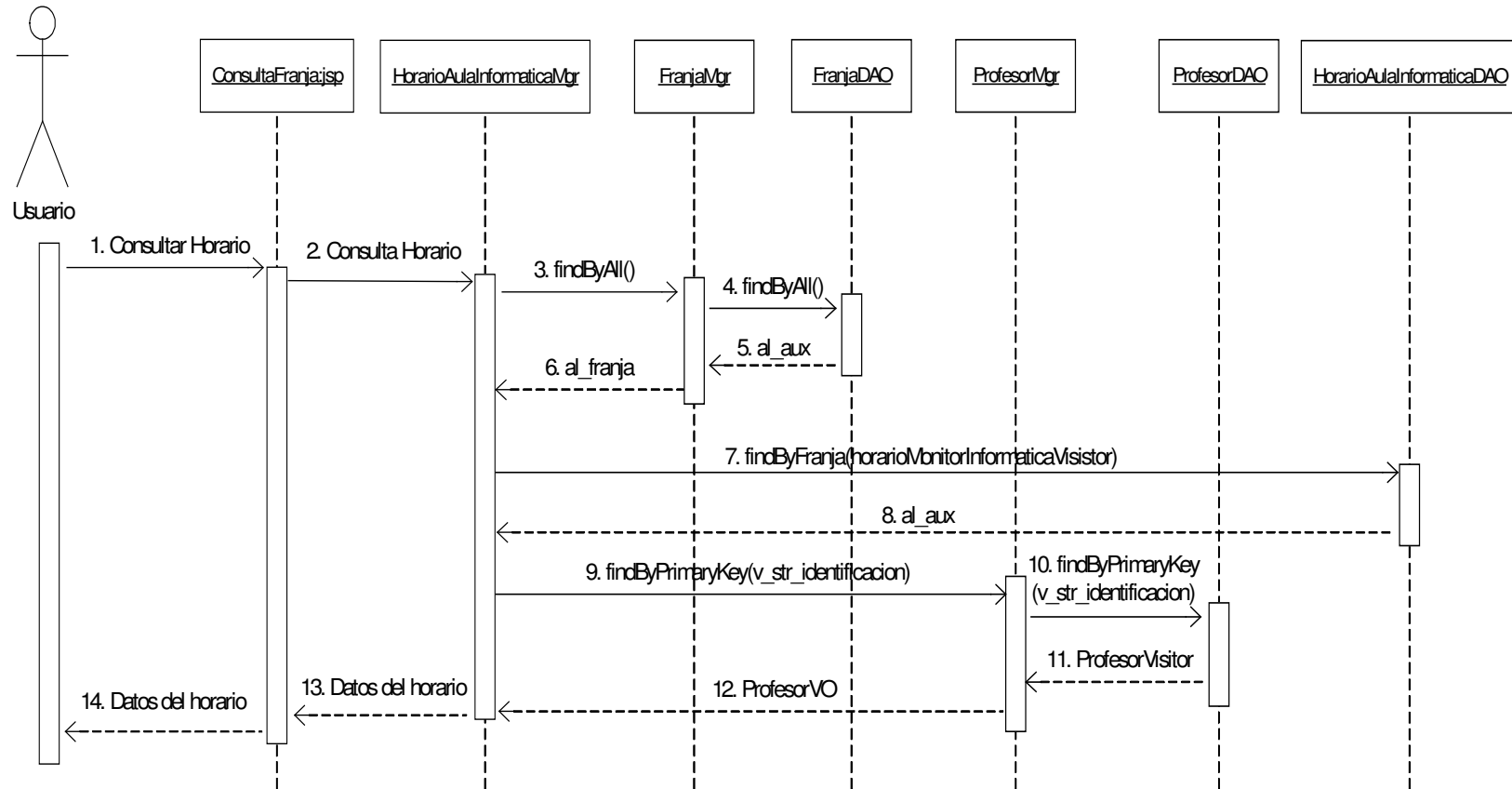


## 11.2.5 Administración de Horarios de Aulas De Informática

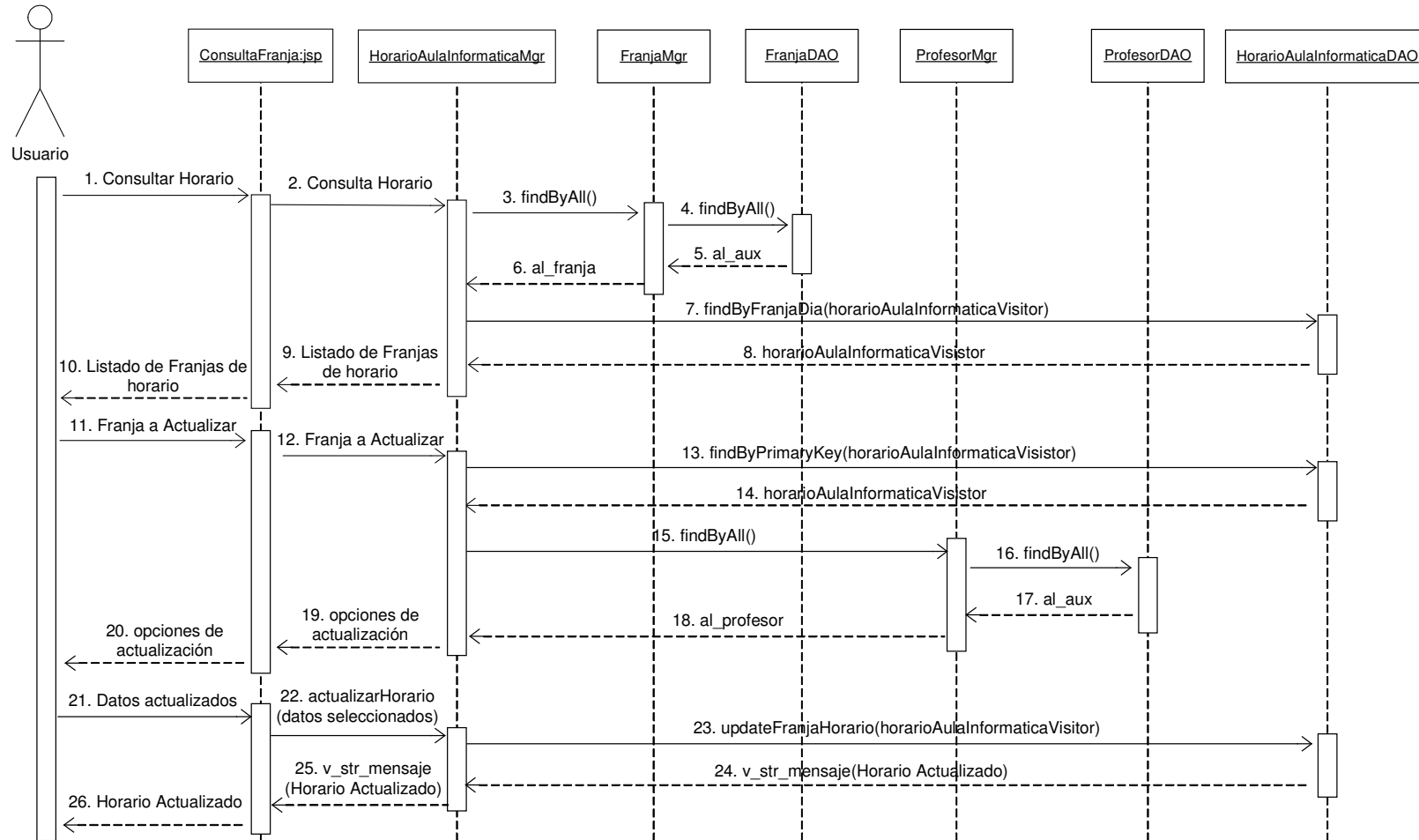
### 11.2.5.1 Asignar Franjas Horario Aula Informática



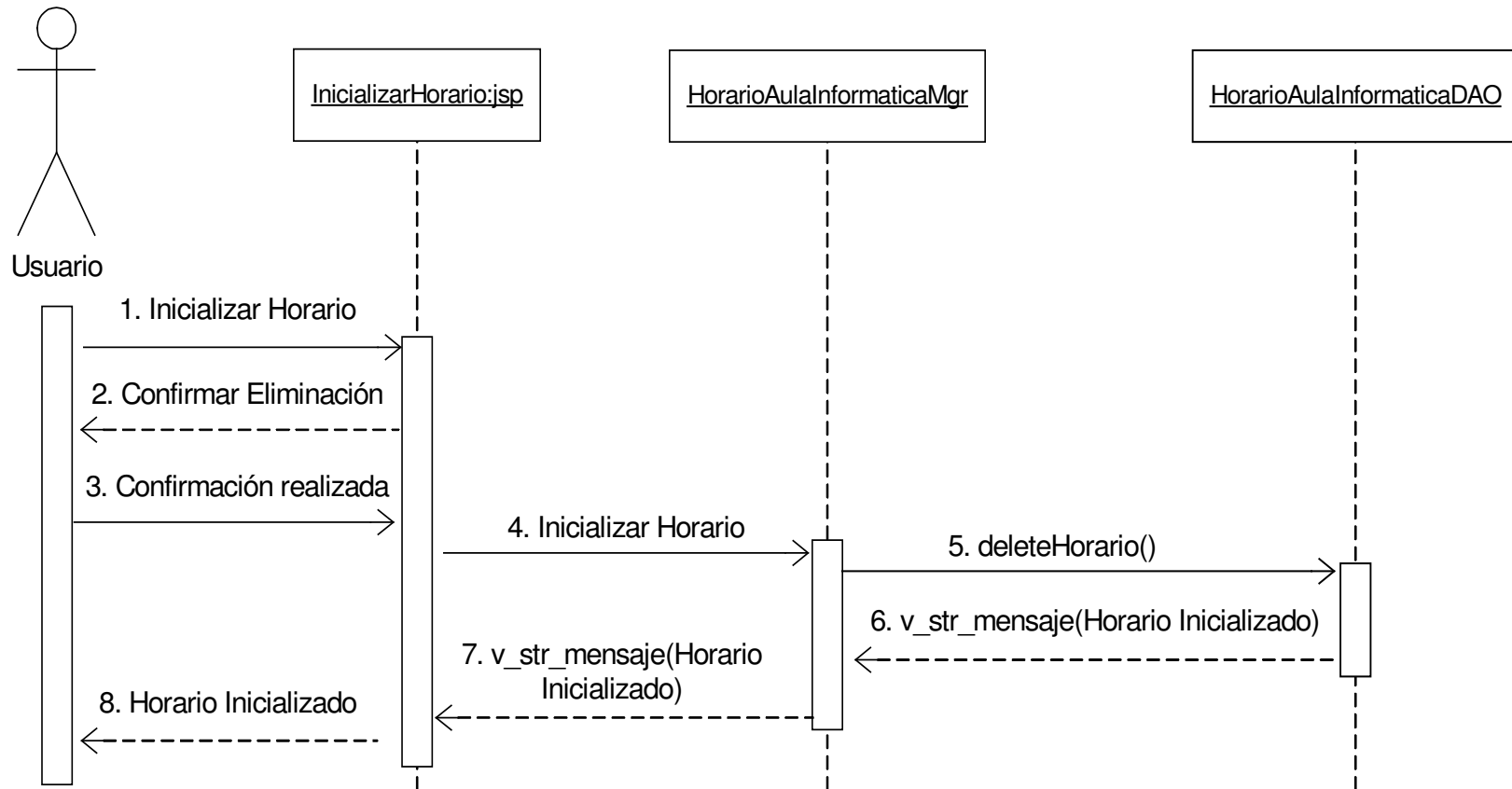
### 11.2.5.2 Consultar Horario Aula Informática



### 11.2.5.3 Actualizar Franjas Horario Aula Informática



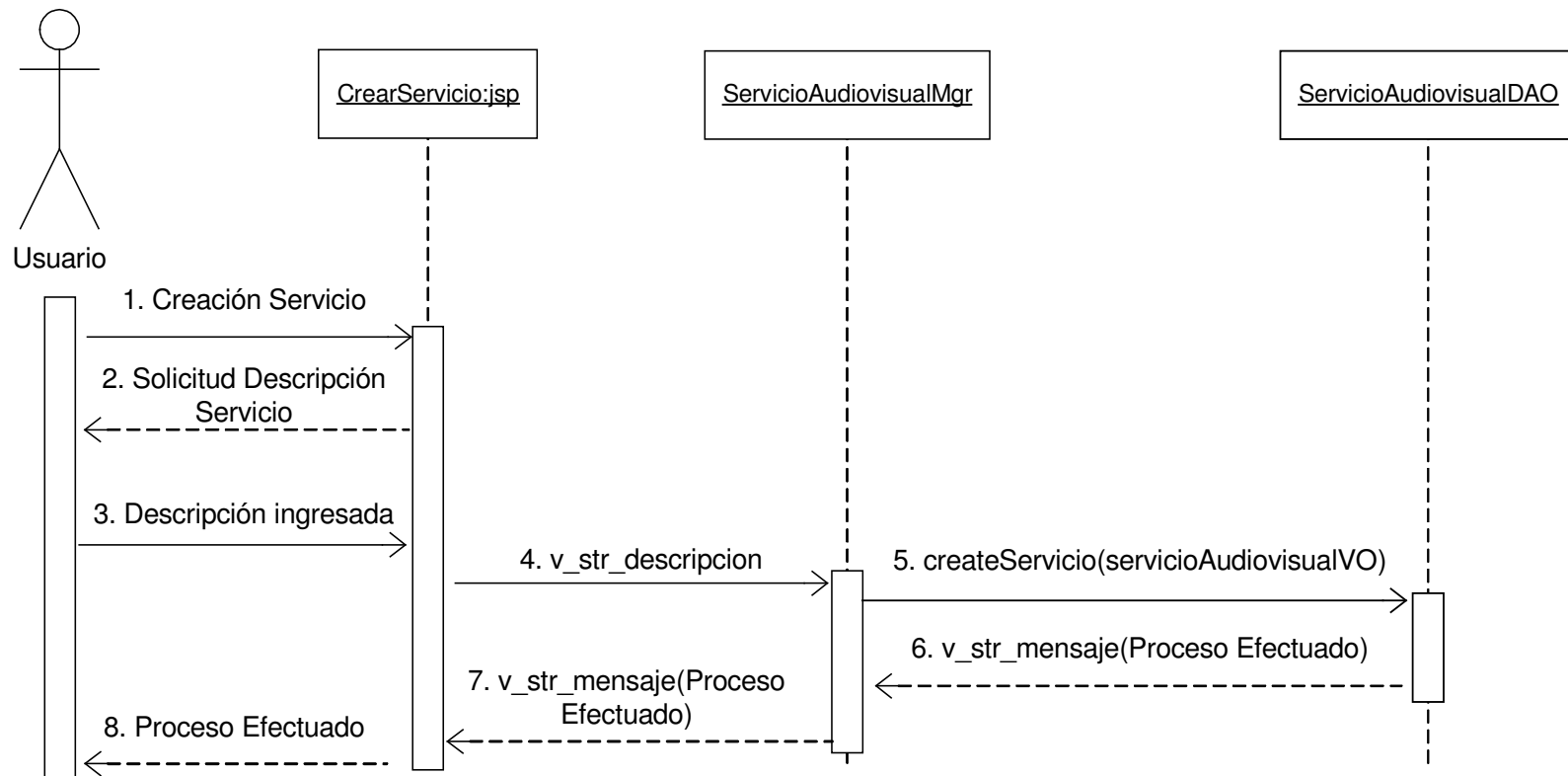
#### 11.2.5.4 Inicializar Horario Aula Informática



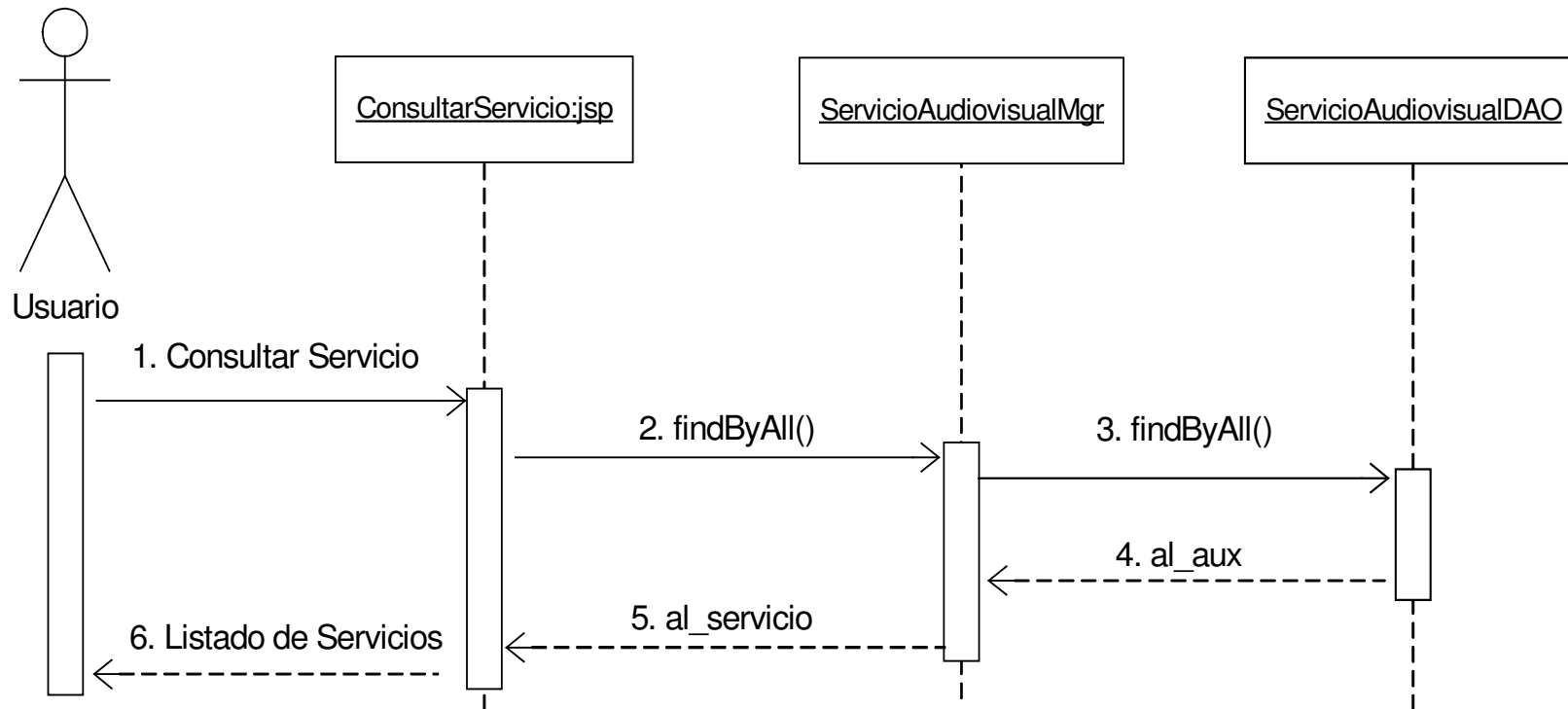


## 11.2.6 Administración de Servicios Audiovisuales

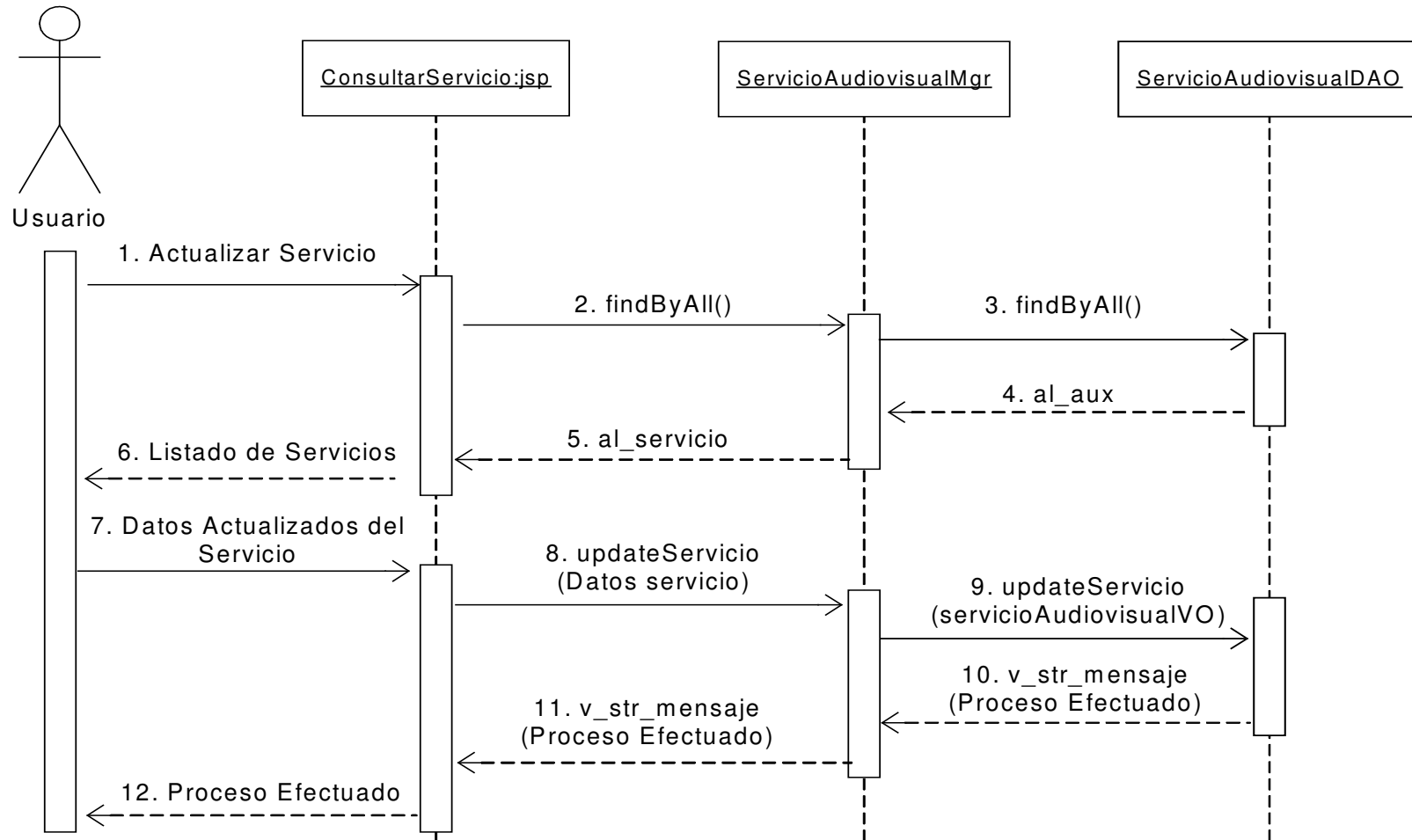
### 11.2.6.1 Creación Servicio Audiovisual



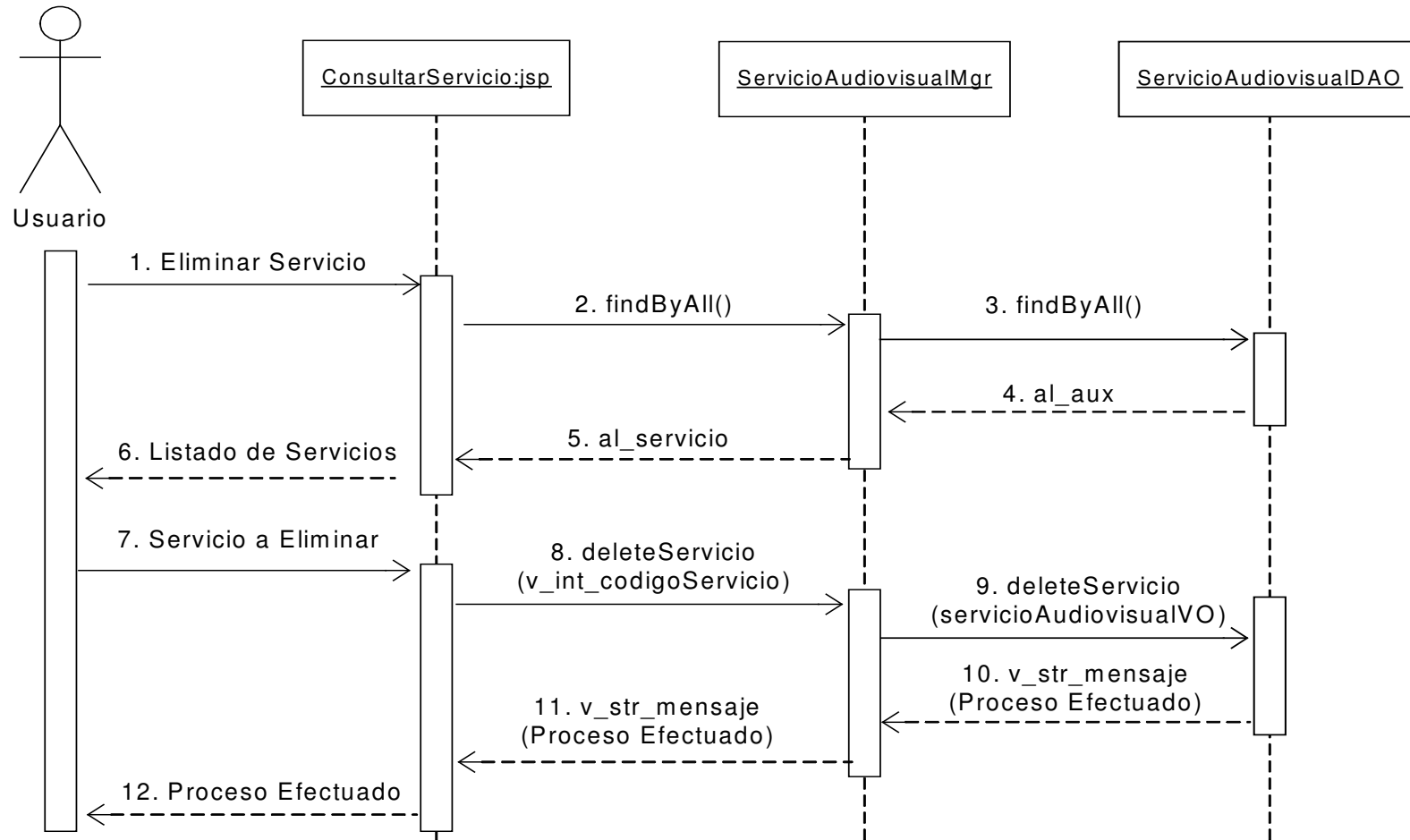
### 11.2.6.2 Consulta Servicio Audiovisual



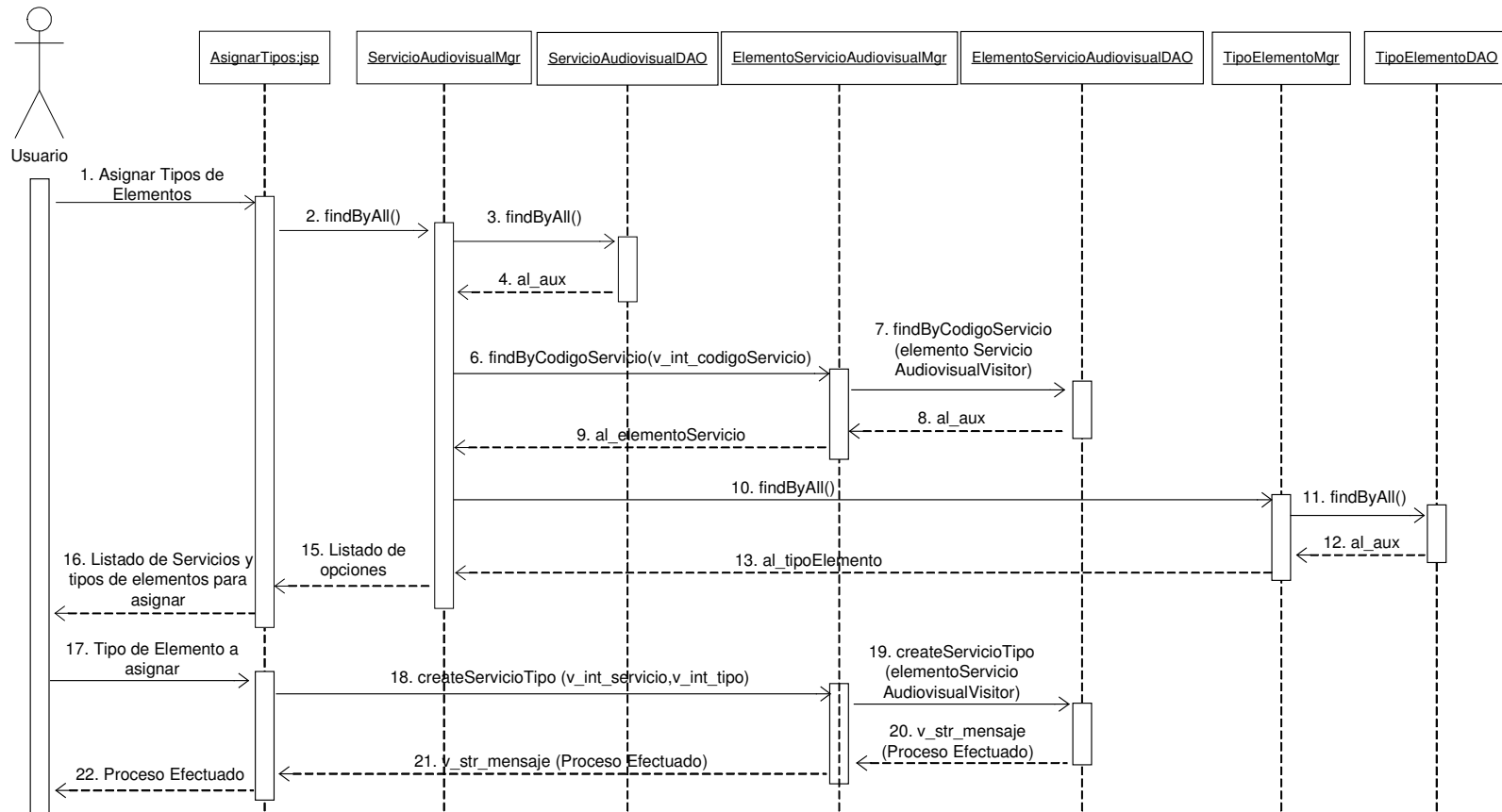
### 11.2.6.3 Actualizar Servicio Audiovisual



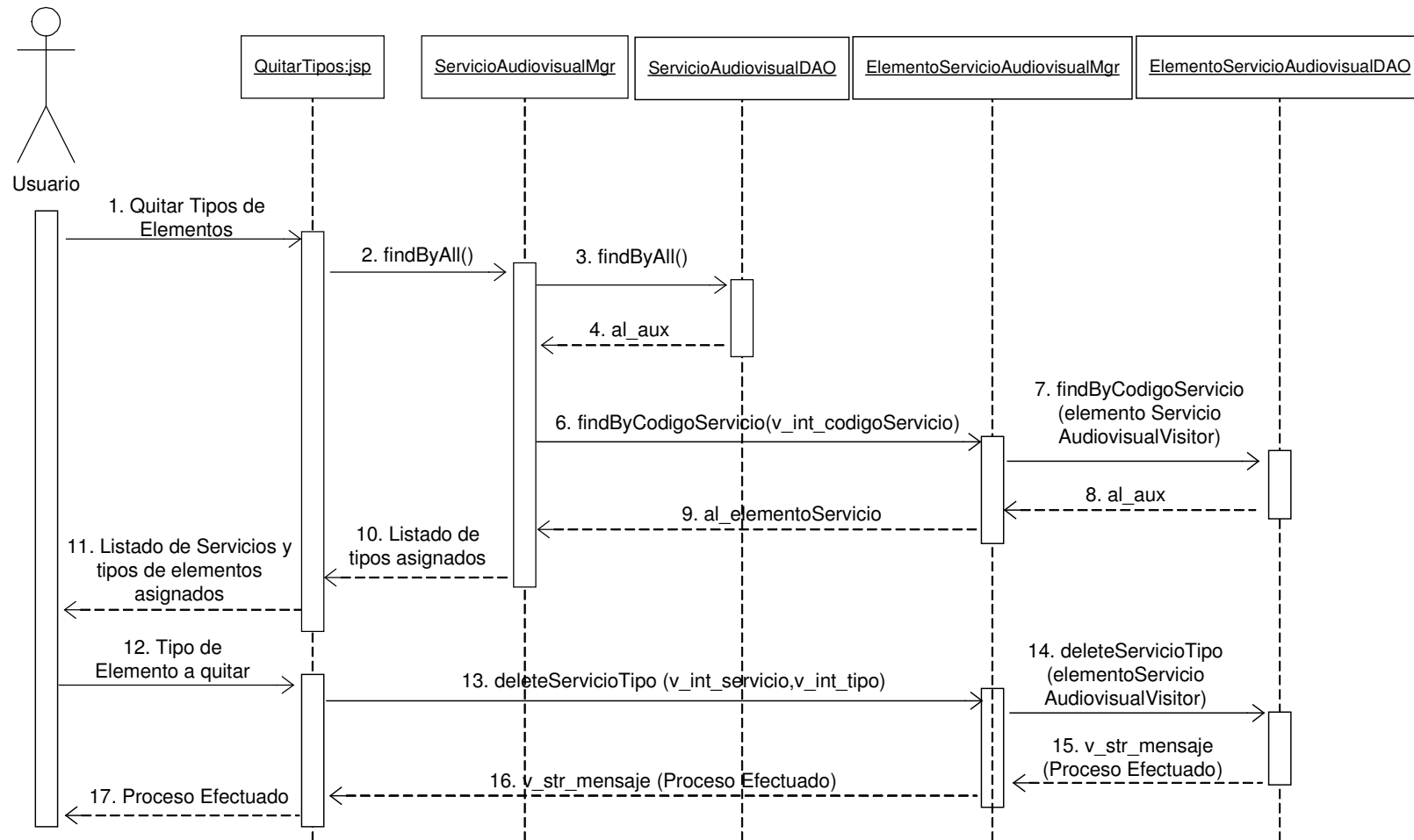
#### 11.2.6.4 Eliminar Servicio Audiovisual



### 11.2.6.5 Asignar Tipos de Elementos

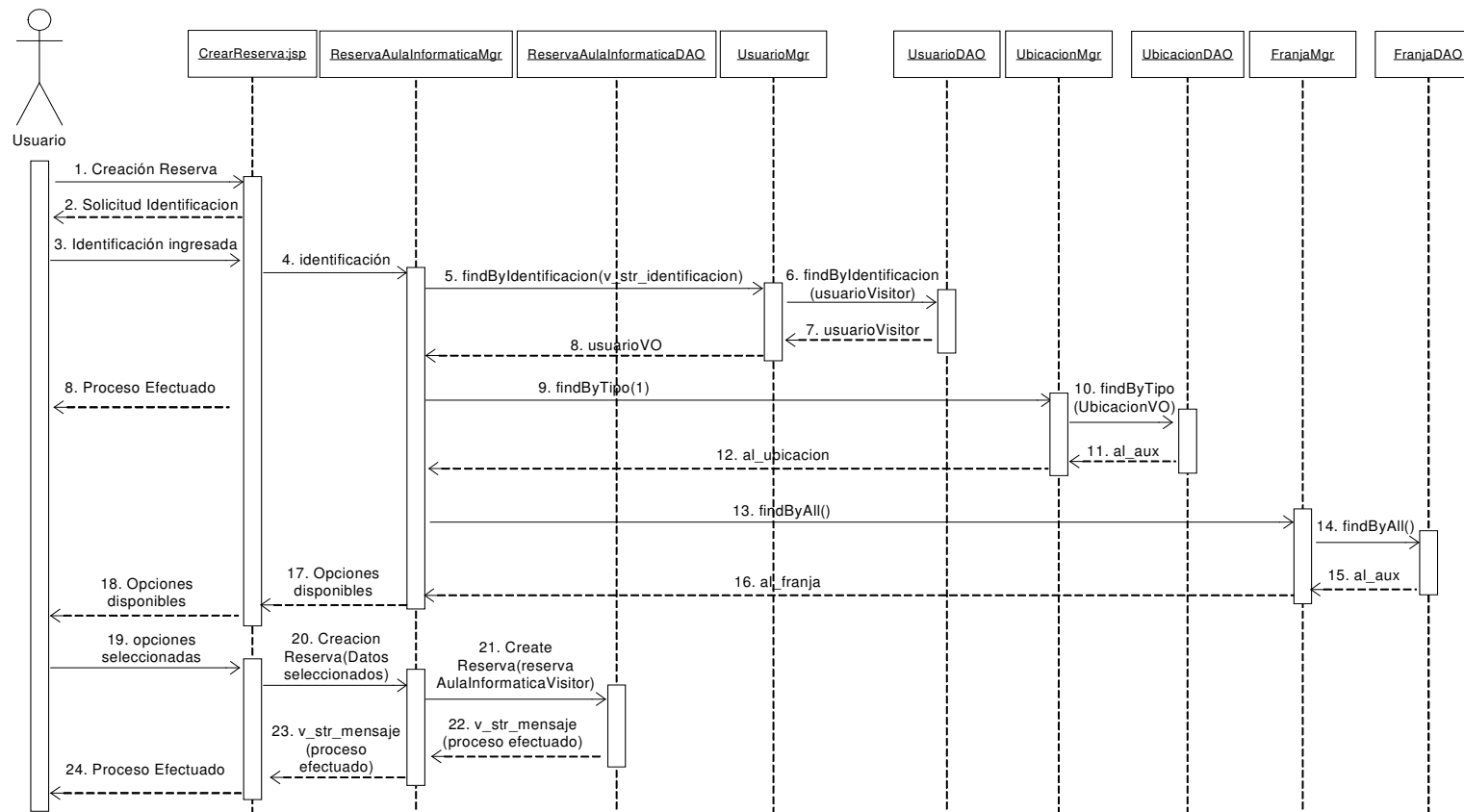


### 11.2.6.6 Quitar Tipos de Elementos

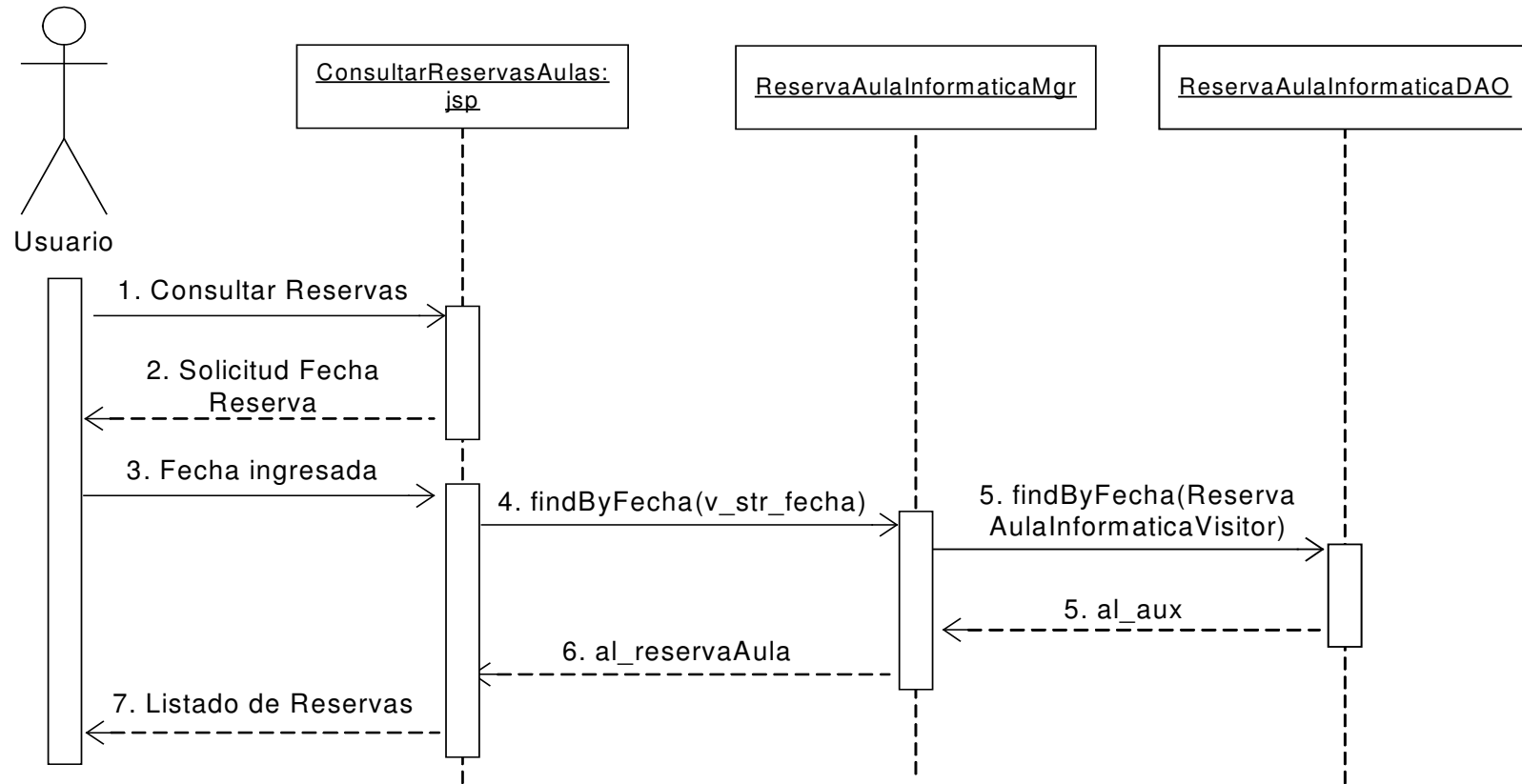


## 11.2.7 Administración de Reservas de Aulas de Informática

### 11.2.7.1 Creación Reserva Aula Informática

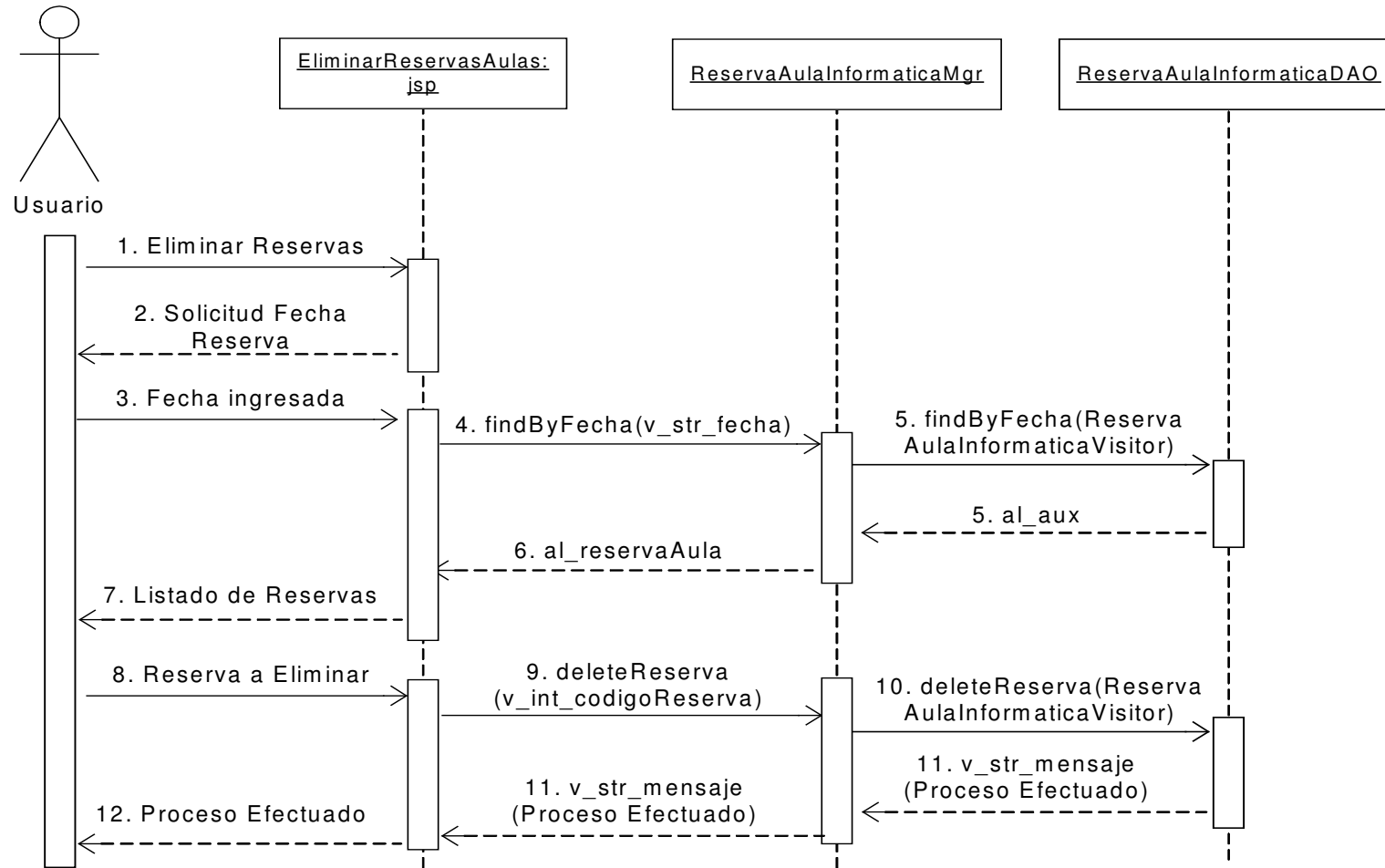


### 11.2.7.2 Consultar Reserva Aula Informática



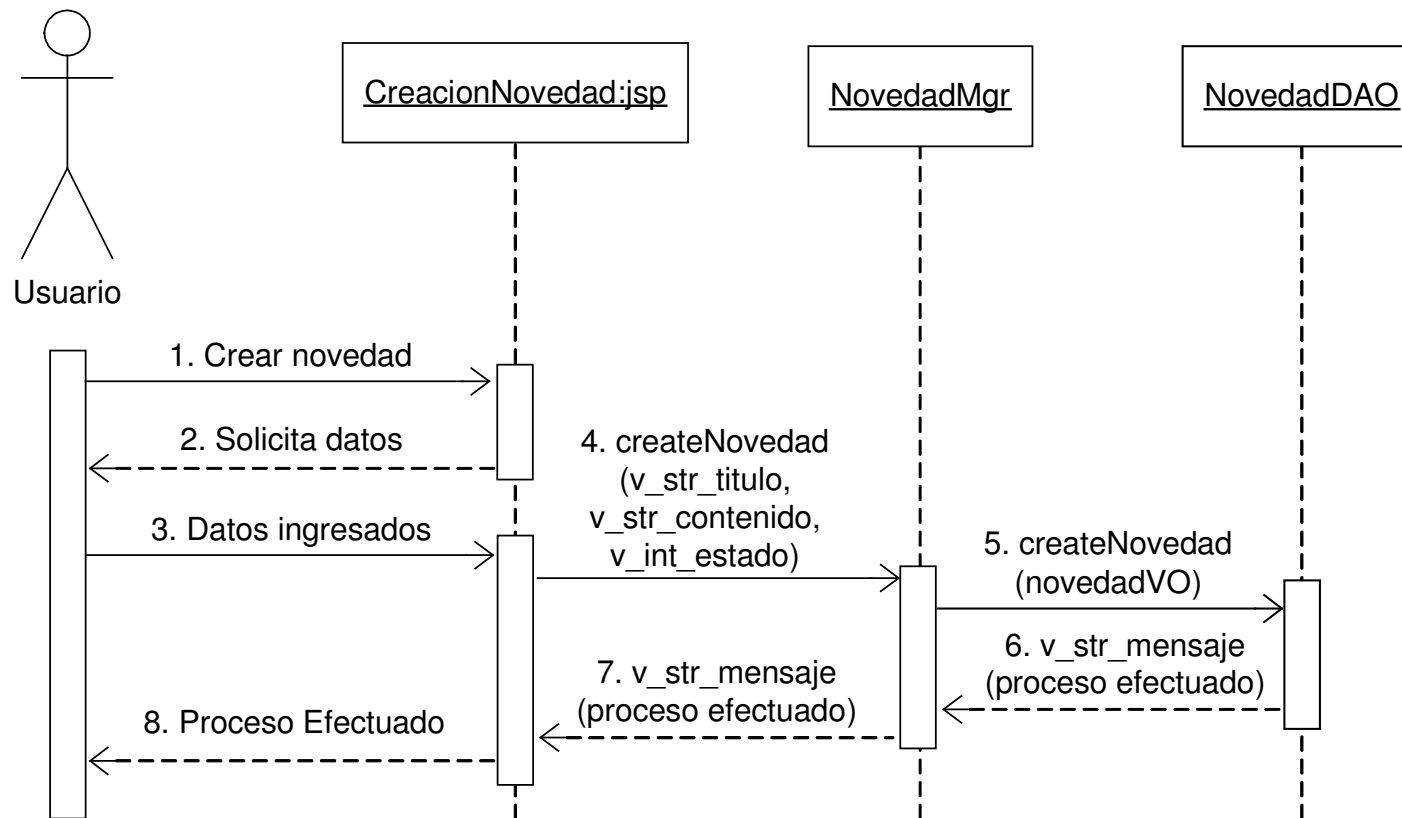


### 11.2.7.3 Cancelar Reserva Aula Informática

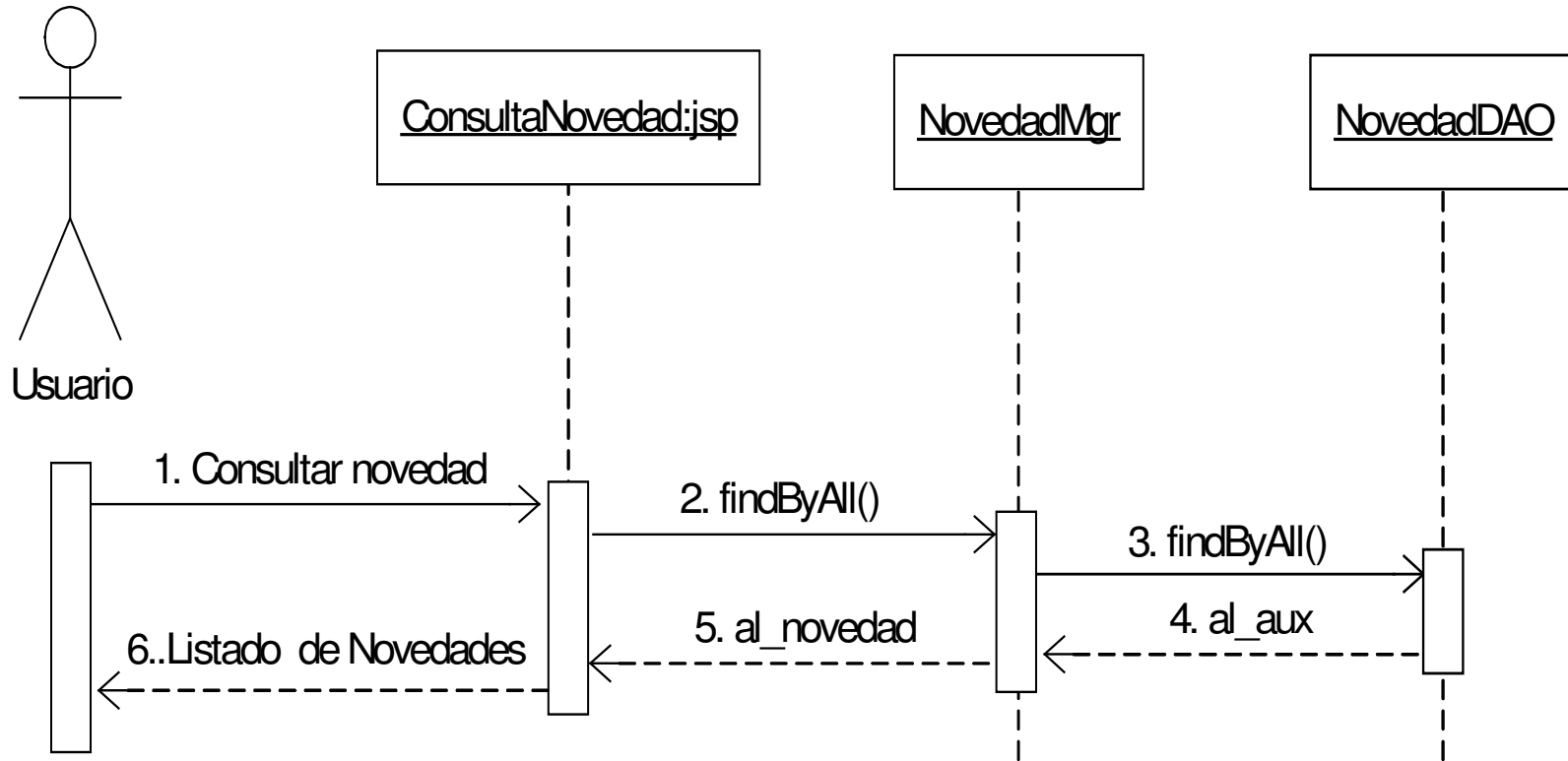


## 11.2.8 Administración Novedades

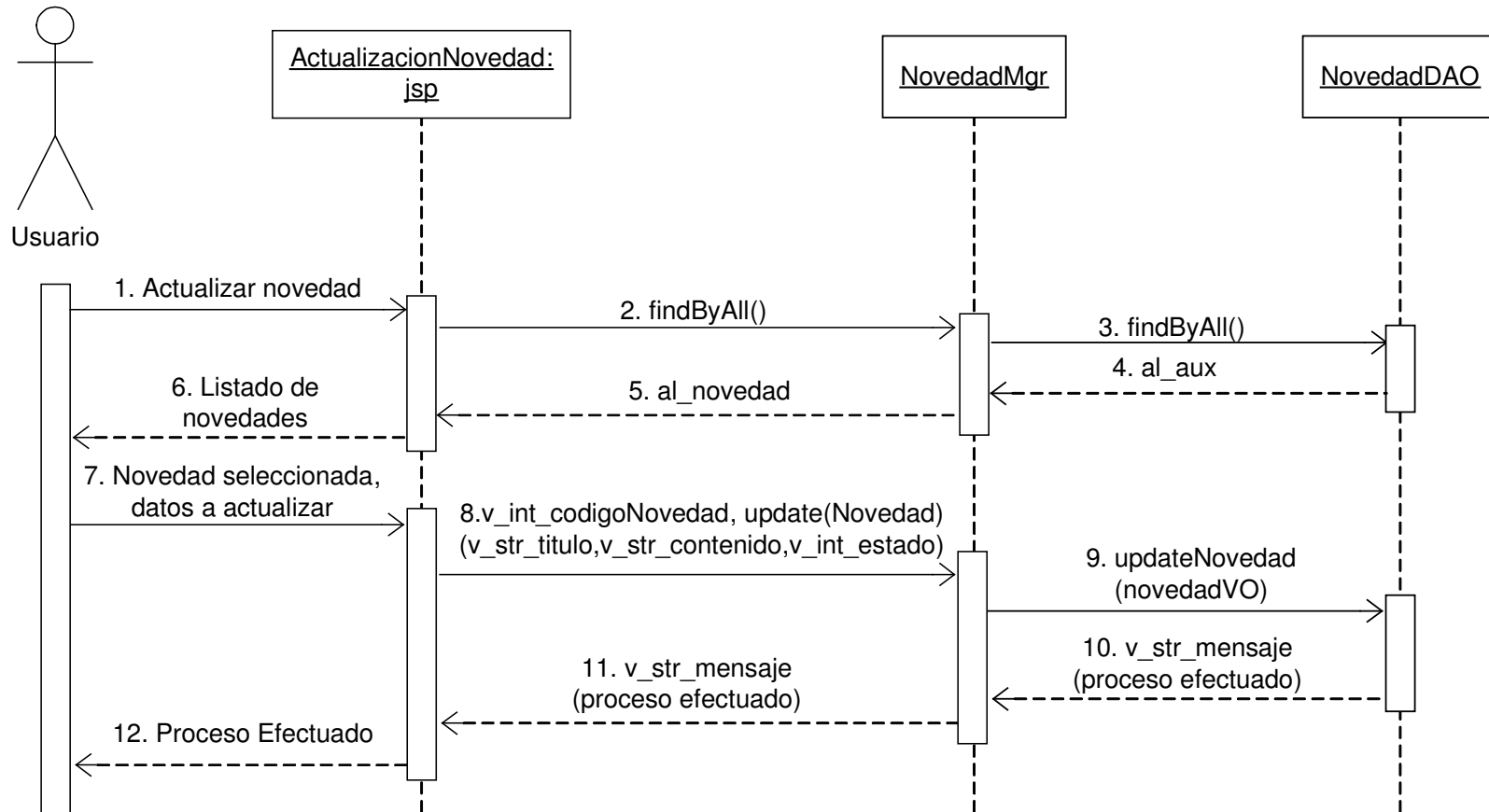
### 11.2.8.1 Crear Novedad



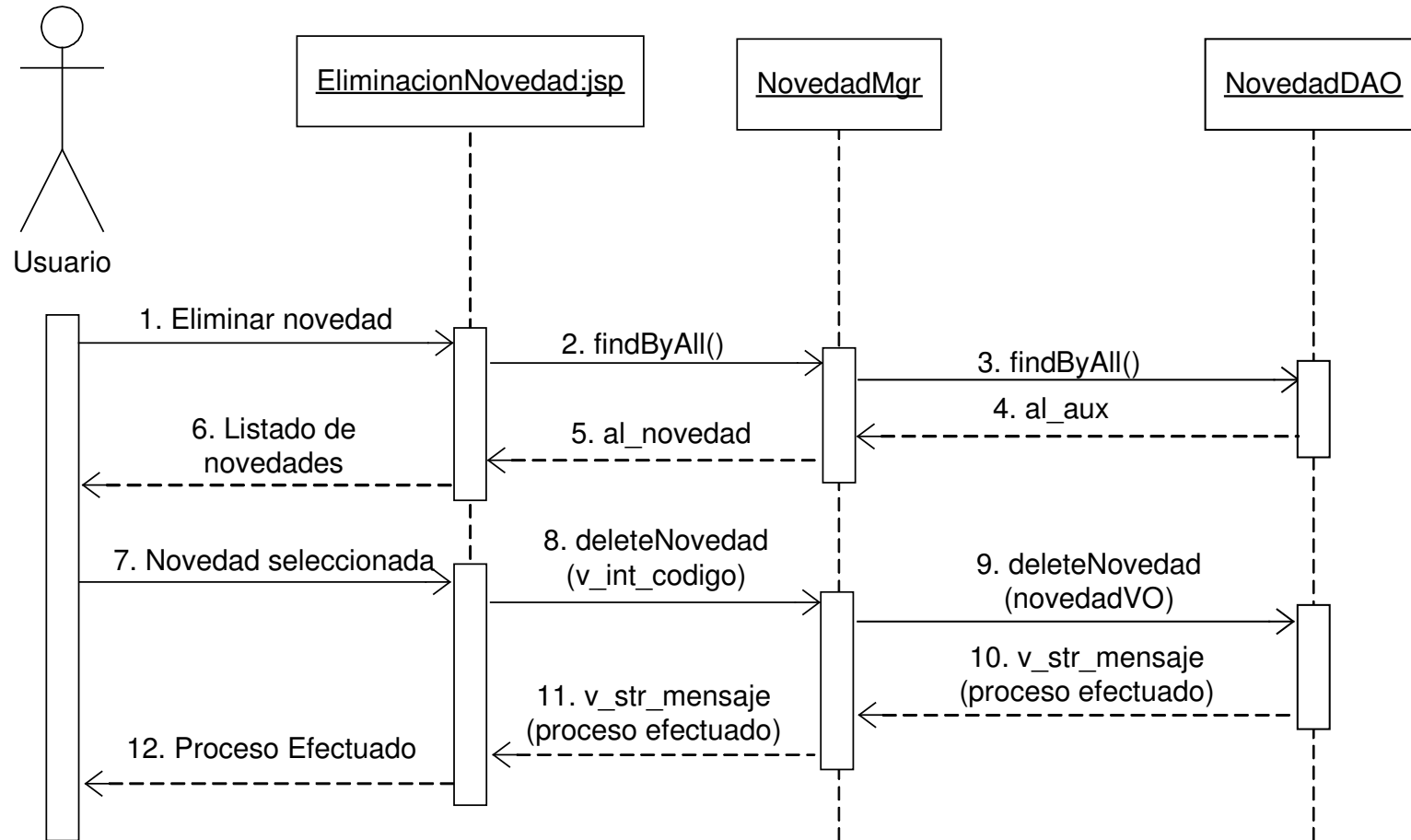
### 11.2.8.2 Consultar Novedad



### 11.2.8.3 Actualizar Novedad

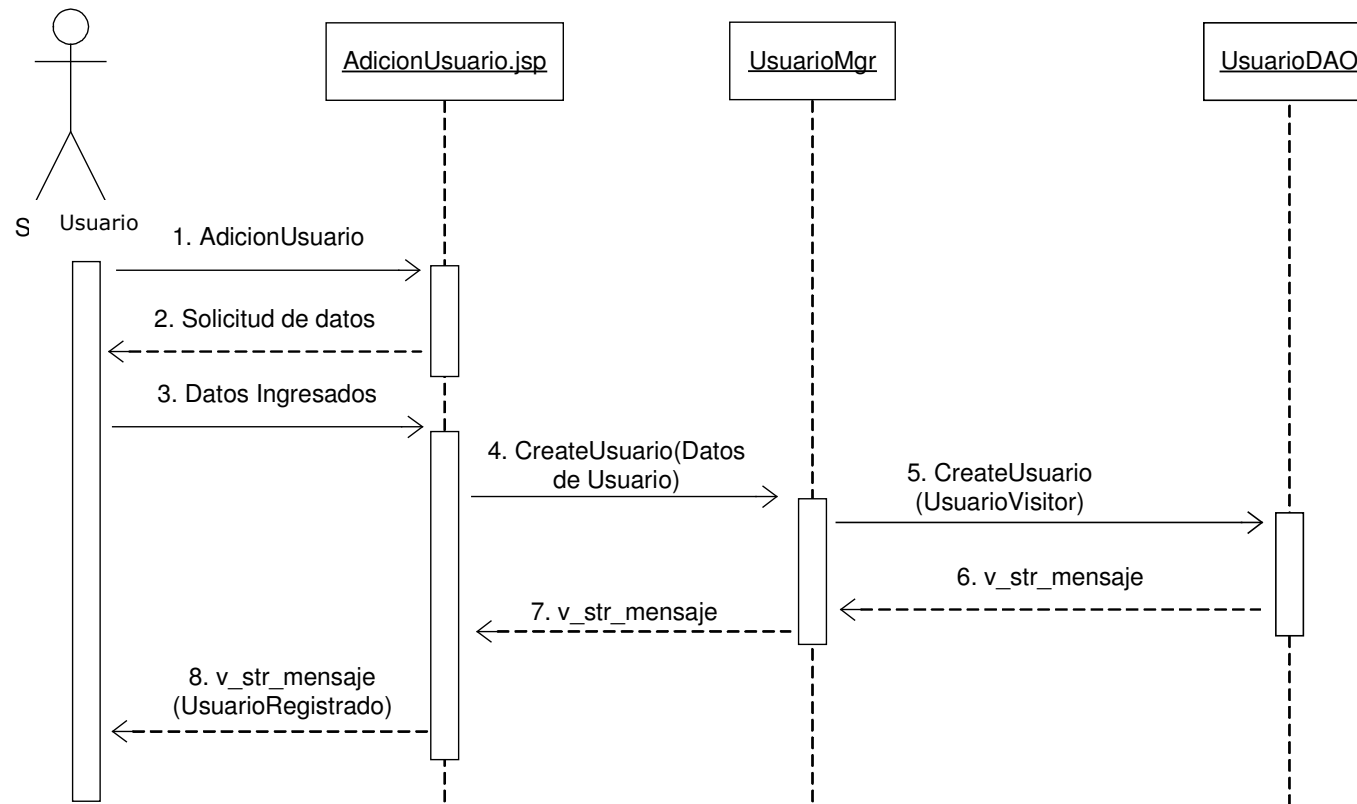


### 11.2.8.4 Eliminar Novedad

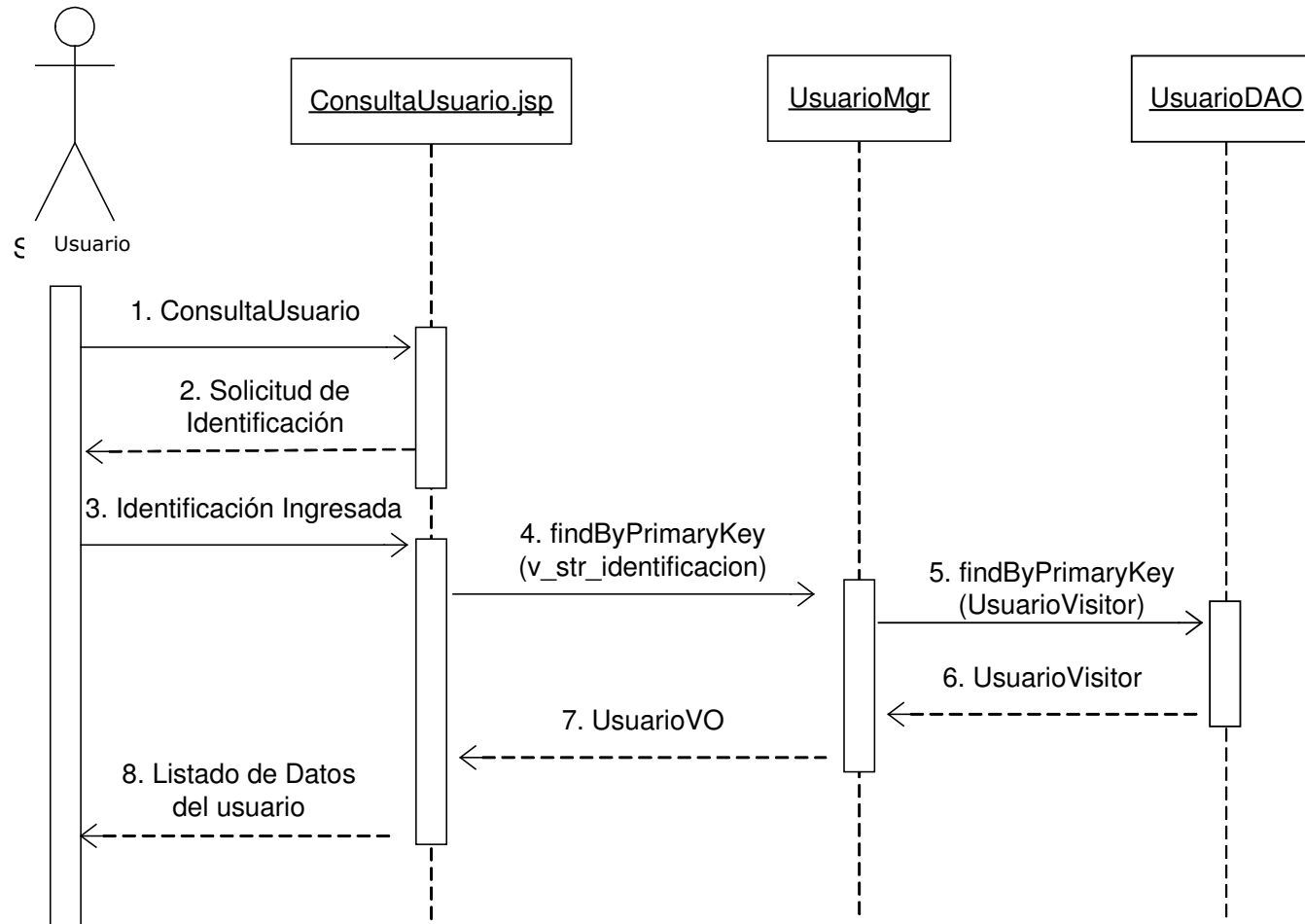


## 11.2.9 Administración de Usuarios

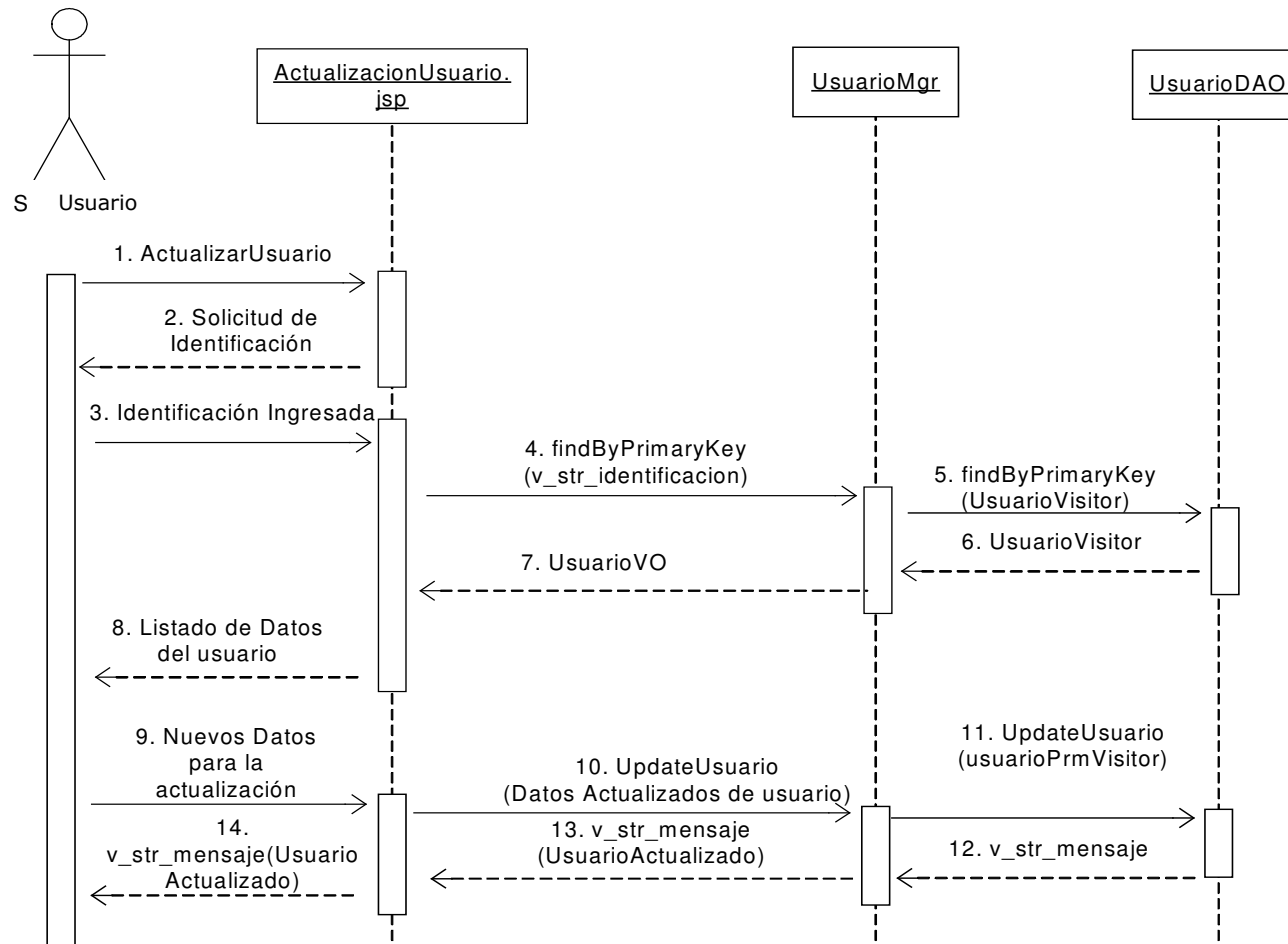
### 11.2.9.1 Registrar Usuario



### 11.2.9.2 Consultar Usuario

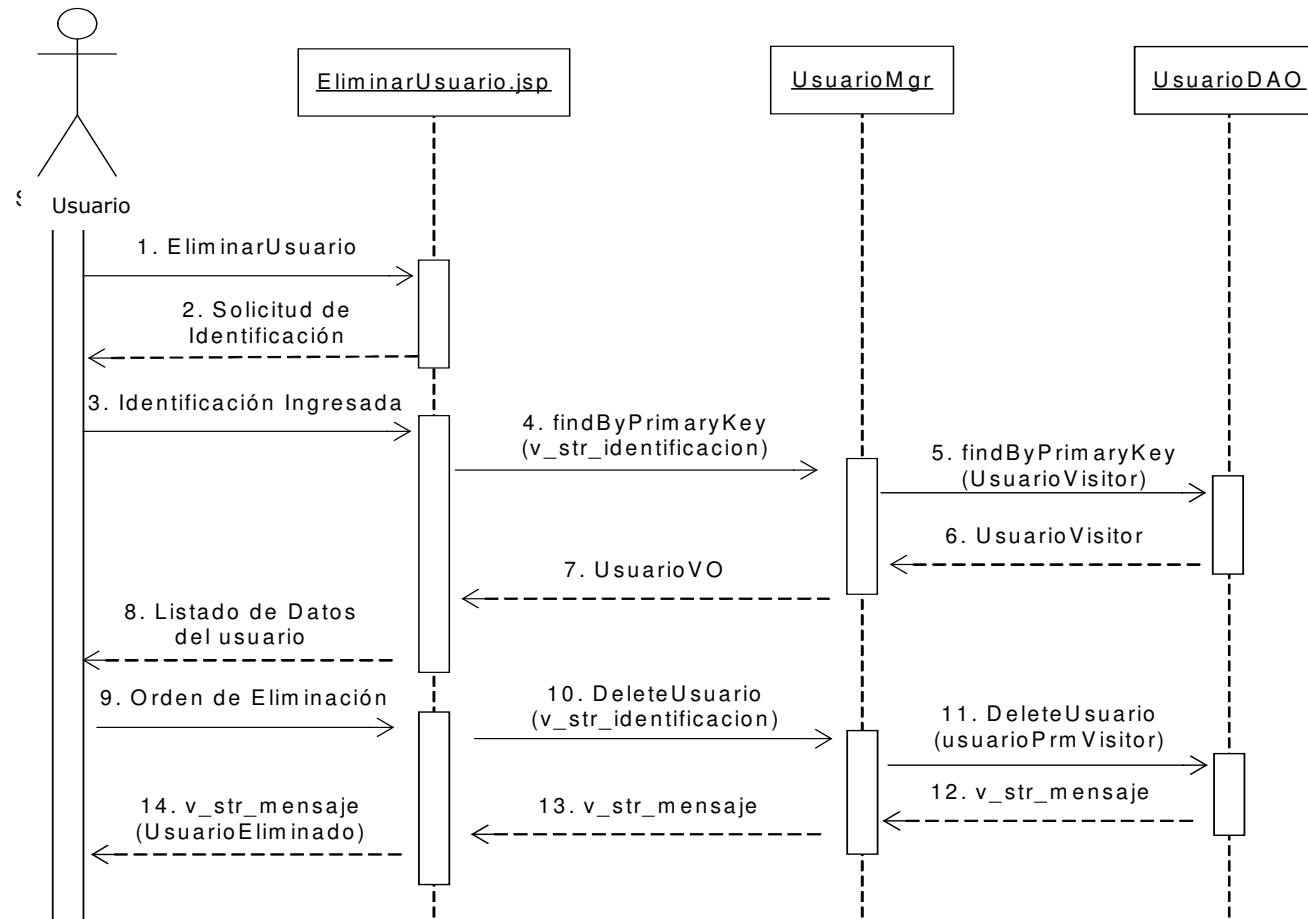


### 11.2.9.3 Actualizar Usuario



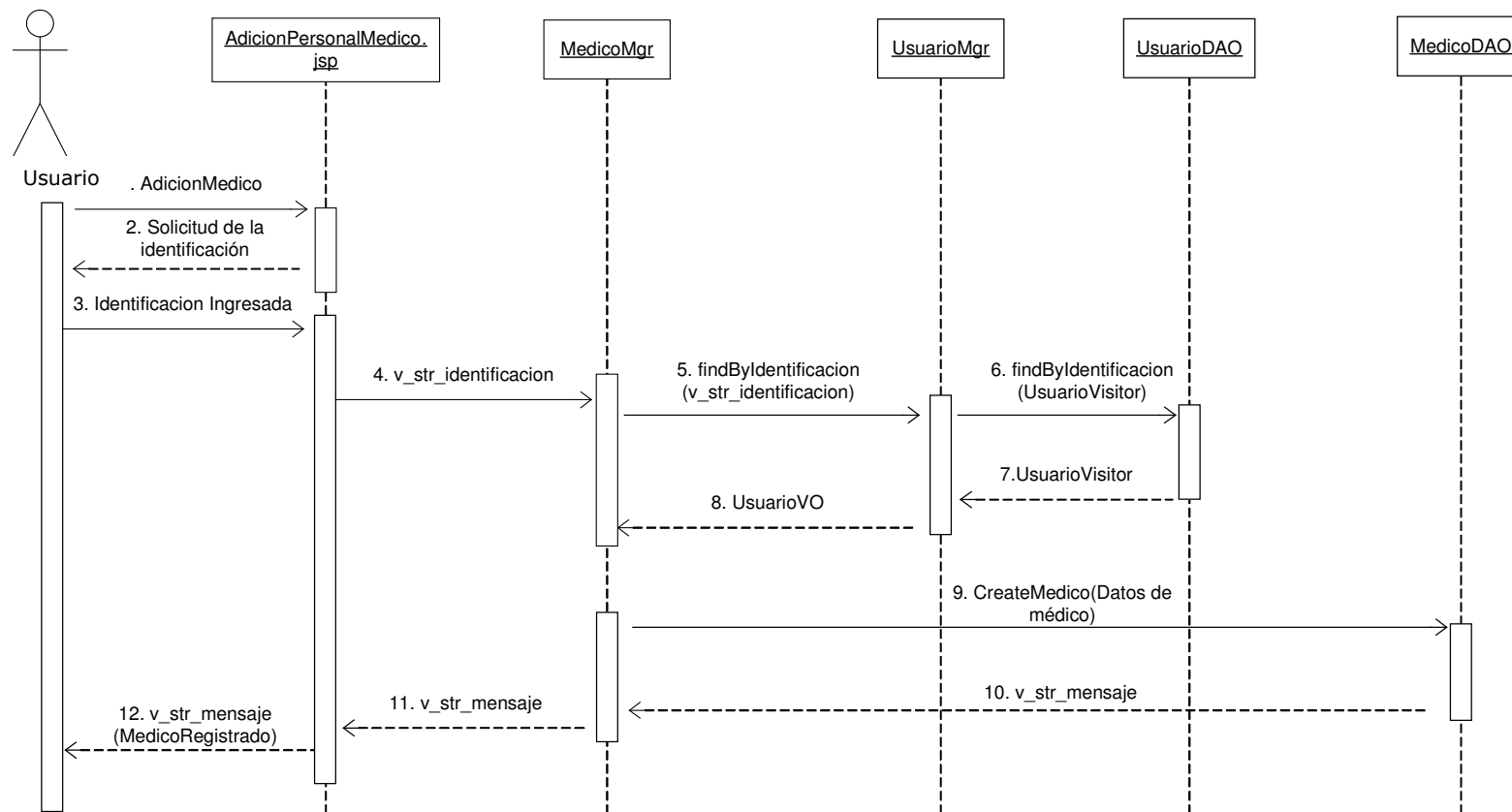


### 11.2.9.4 Eliminar Usuario

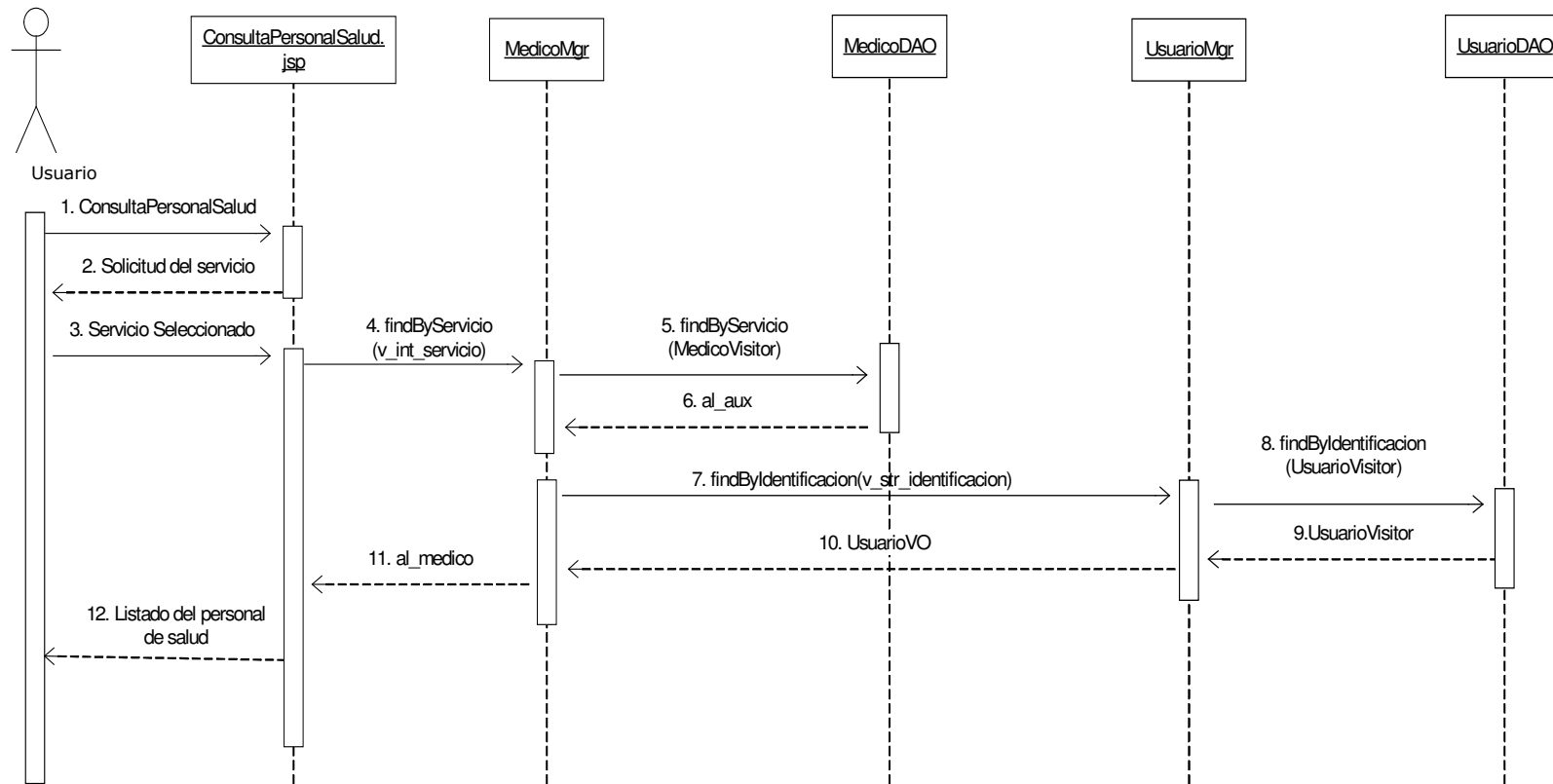


## 11.2.10 Administración del Personal de Salud

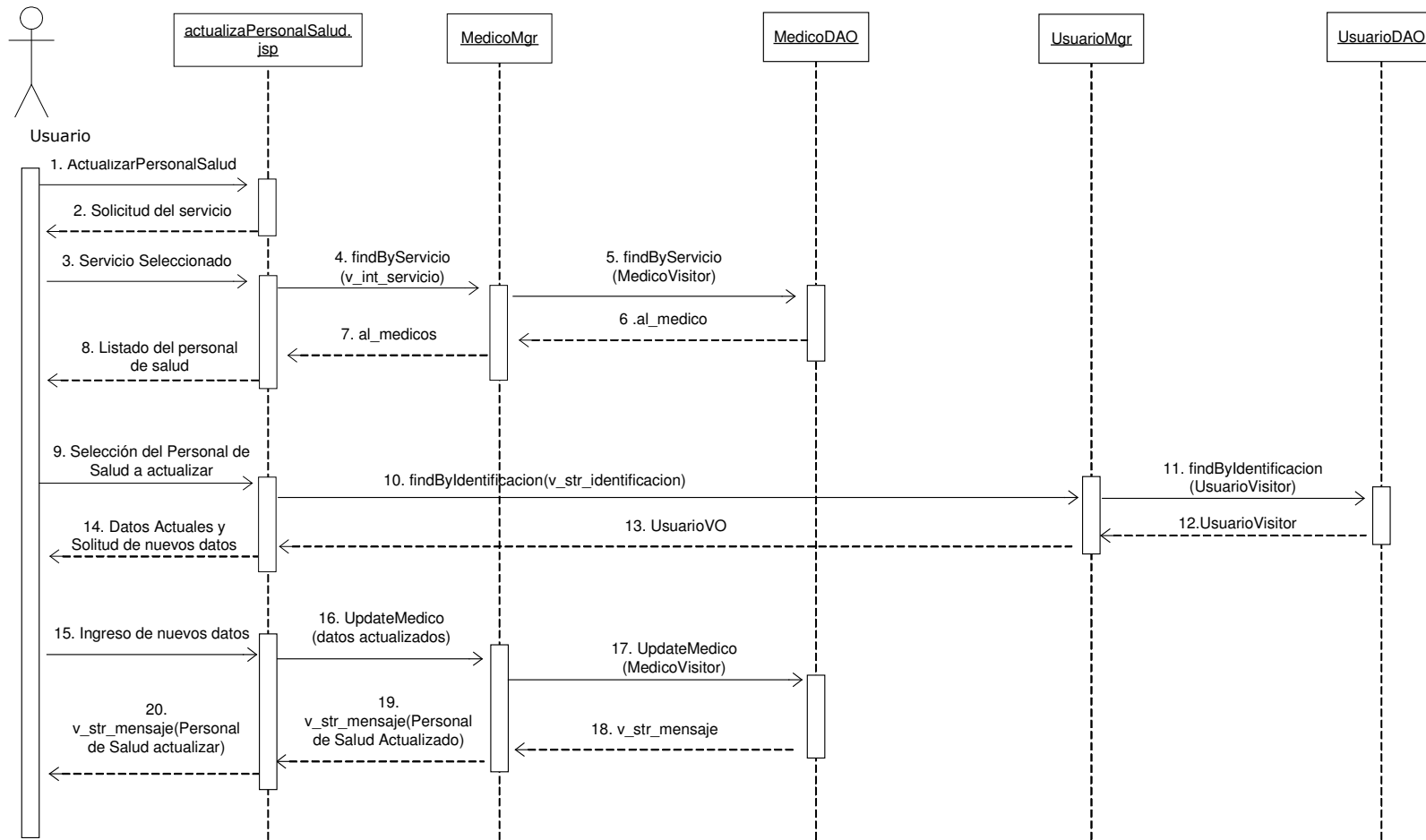
### 11.2.10.2 Registrar Personal de Salud



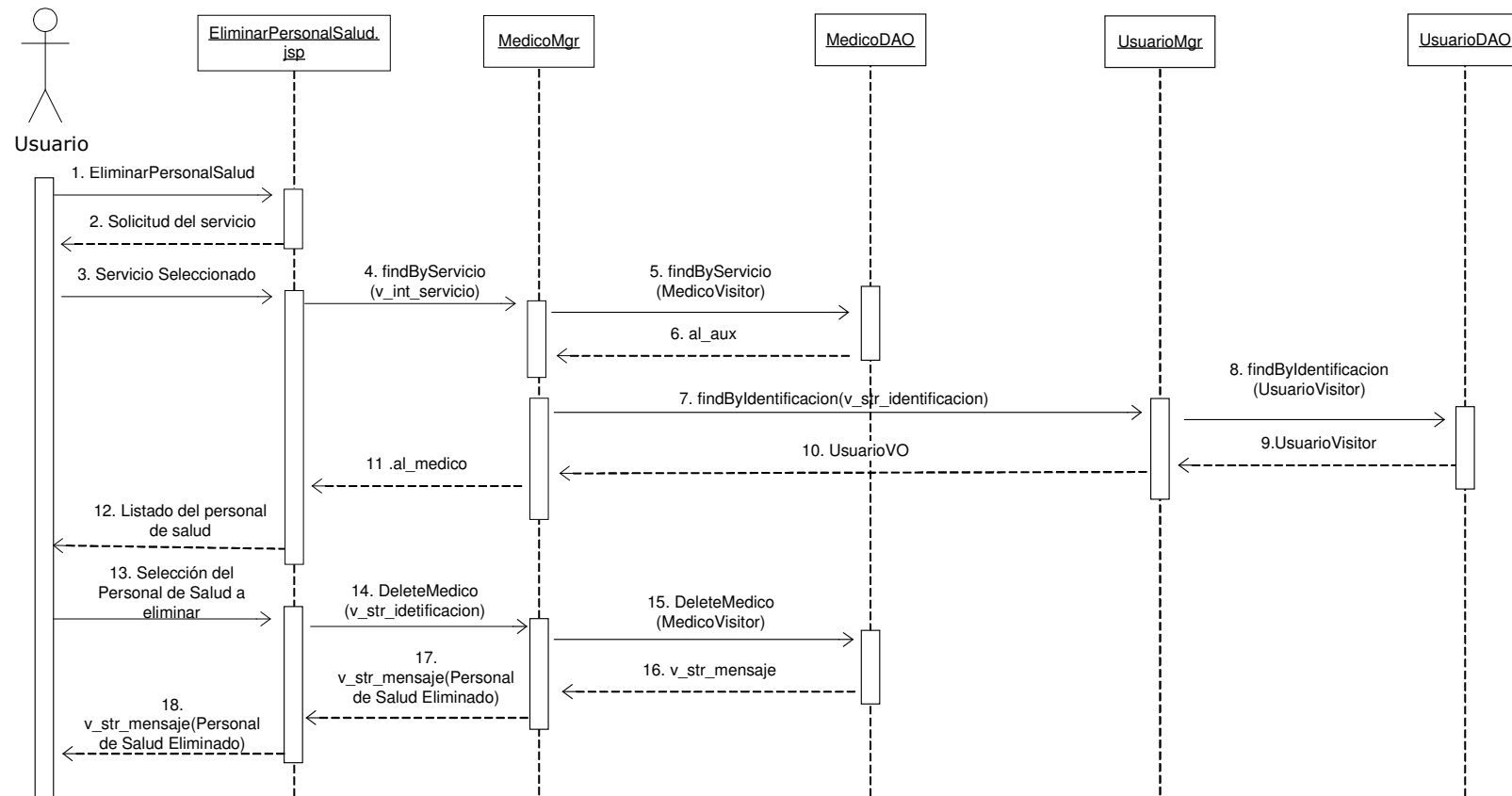
### 11.2.10.2 Consultar Personal de Salud



### 11.2.10.3 Actualizar Personal de Salud

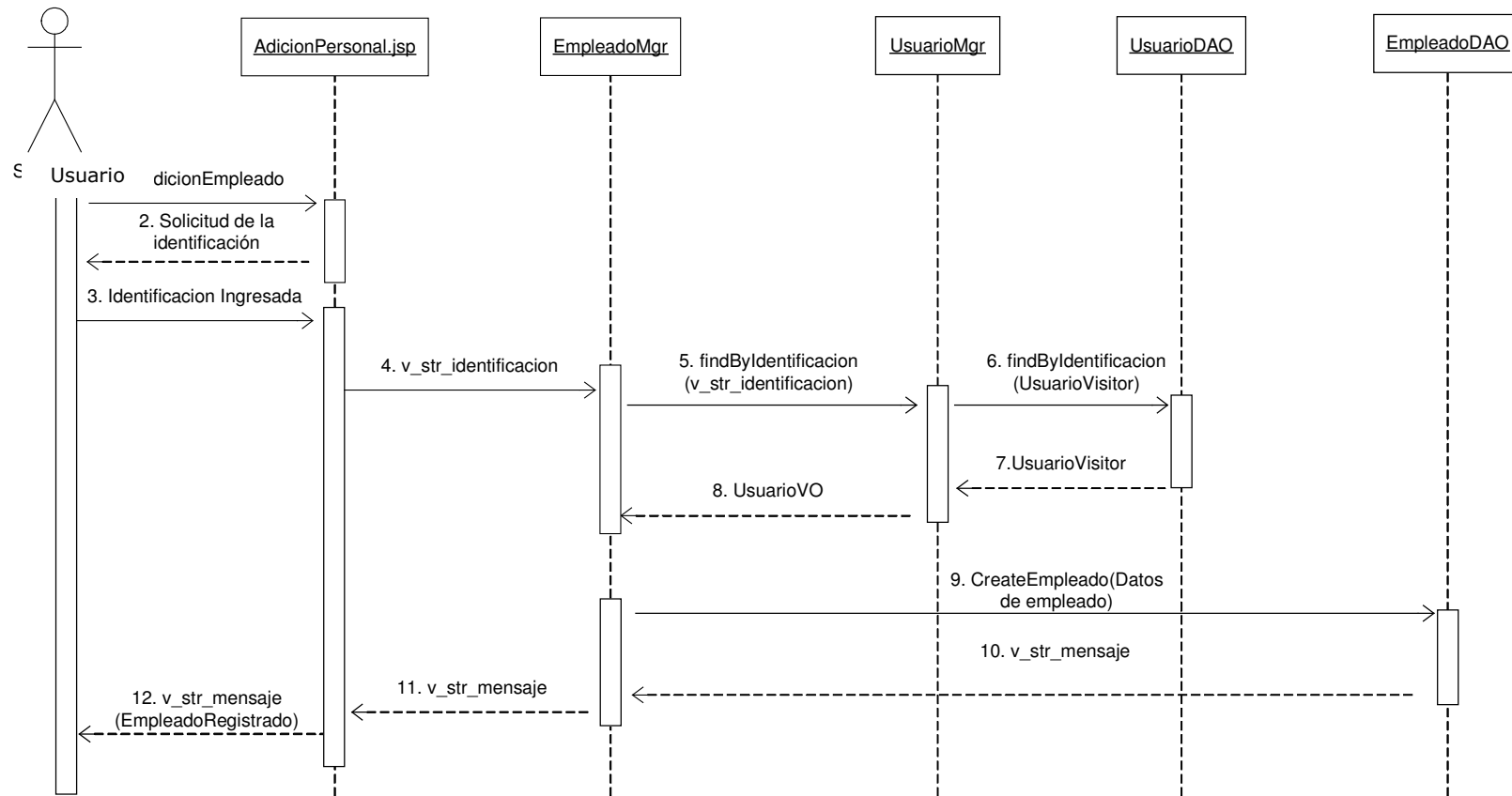


### 11.2.10.4 Eliminar Personal de Salud

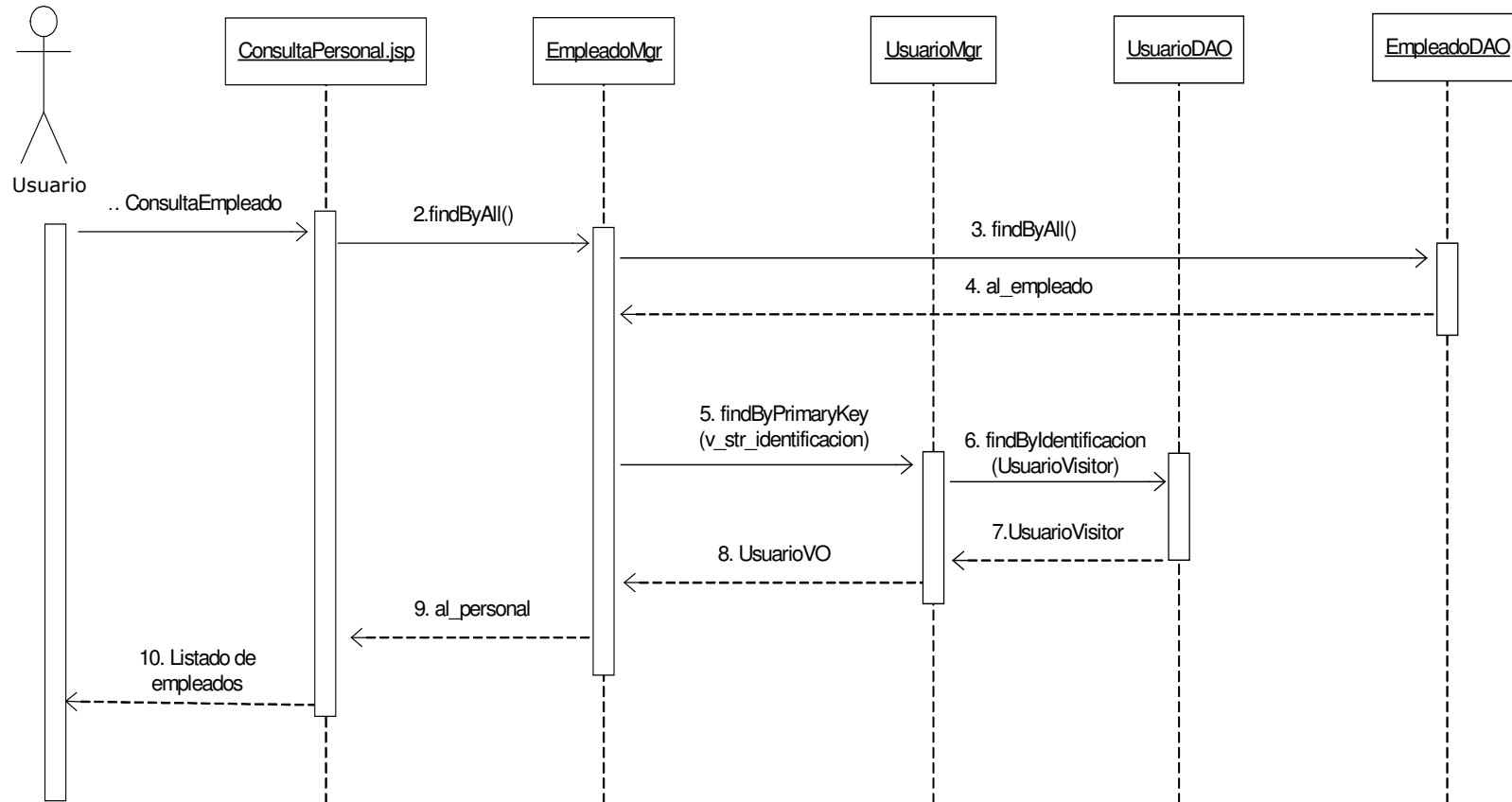


## 11.2.11 Administración del Personal Administrativo y Operativo

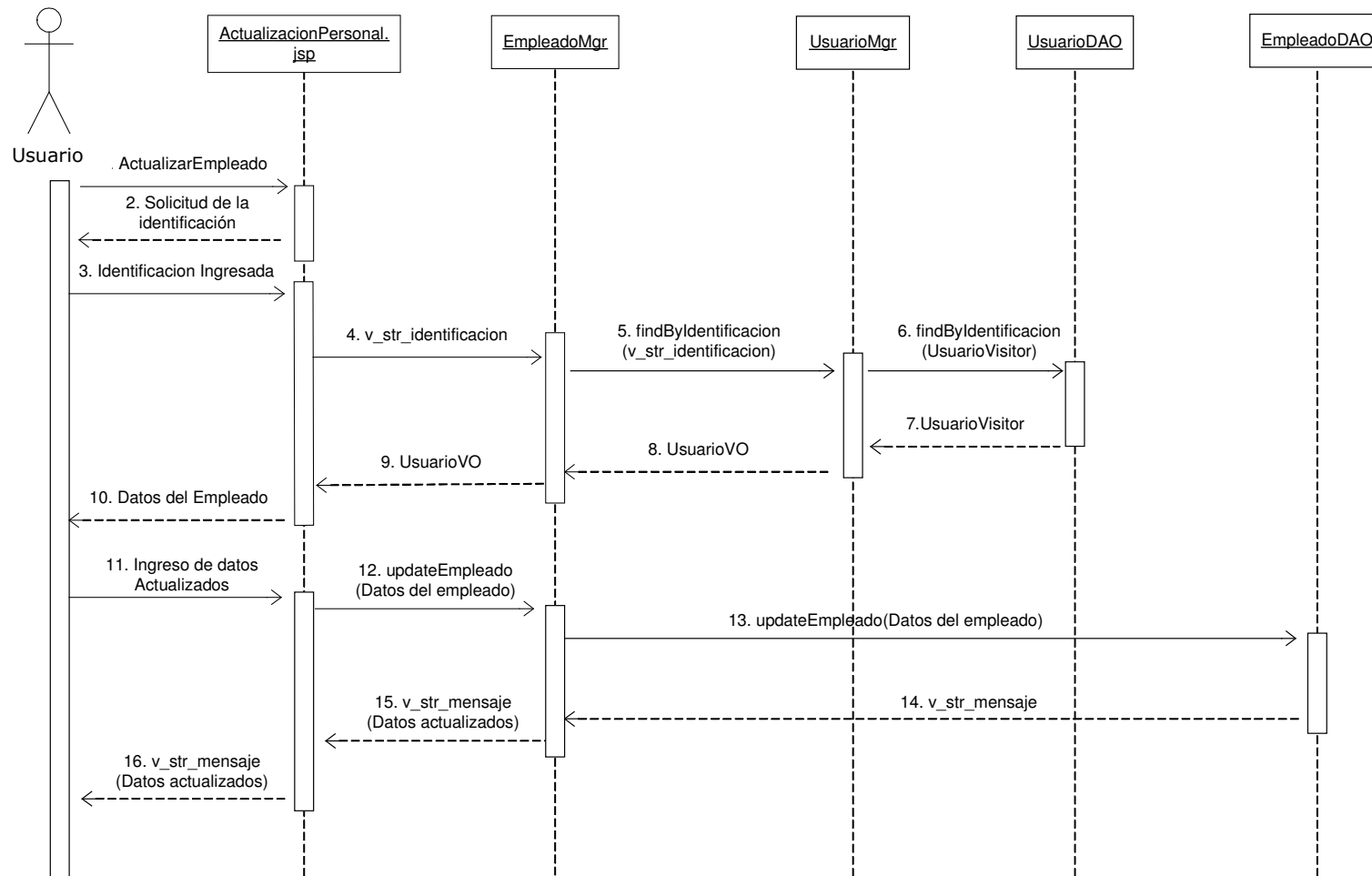
### 11.2.11.1 Registrar Personal Administrativo y Operativo



### 11.2.11.2 Consultar Personal Administrativo y Operativo

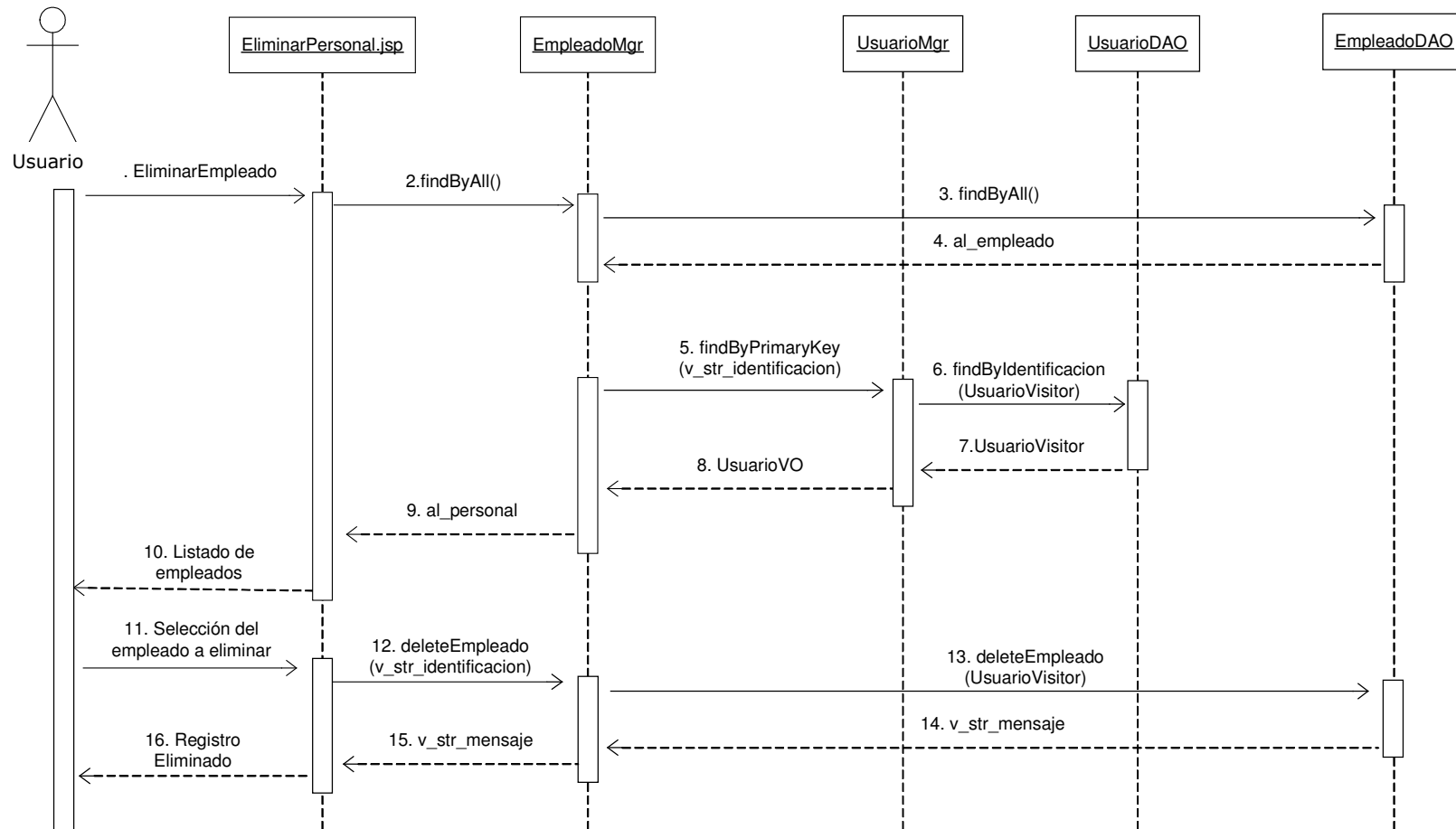


### 11.2.11.3 Actualizar Personal Administrativo y Operativo



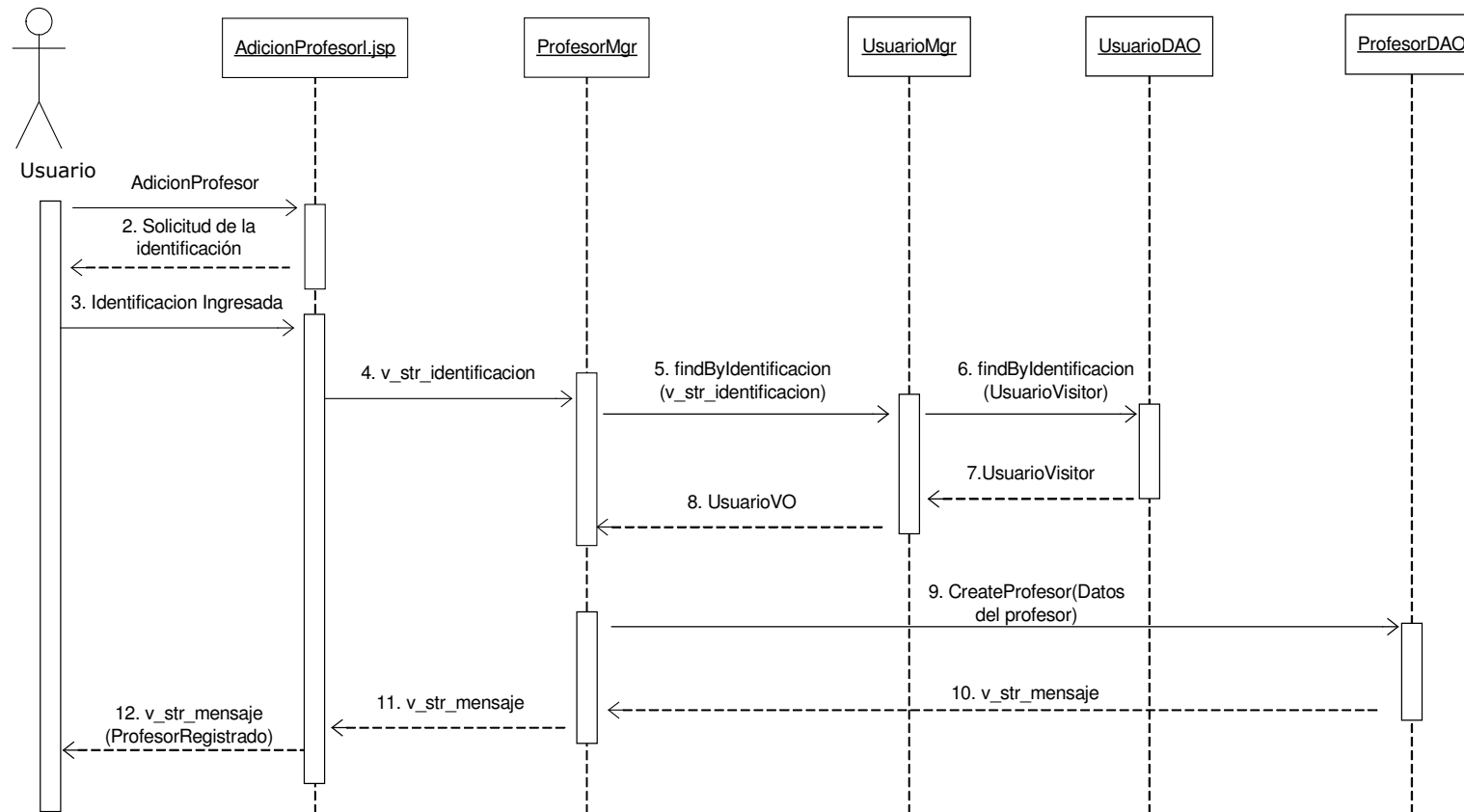


### 11.2.11.4 Eliminar Personal Administrativo y Operativo

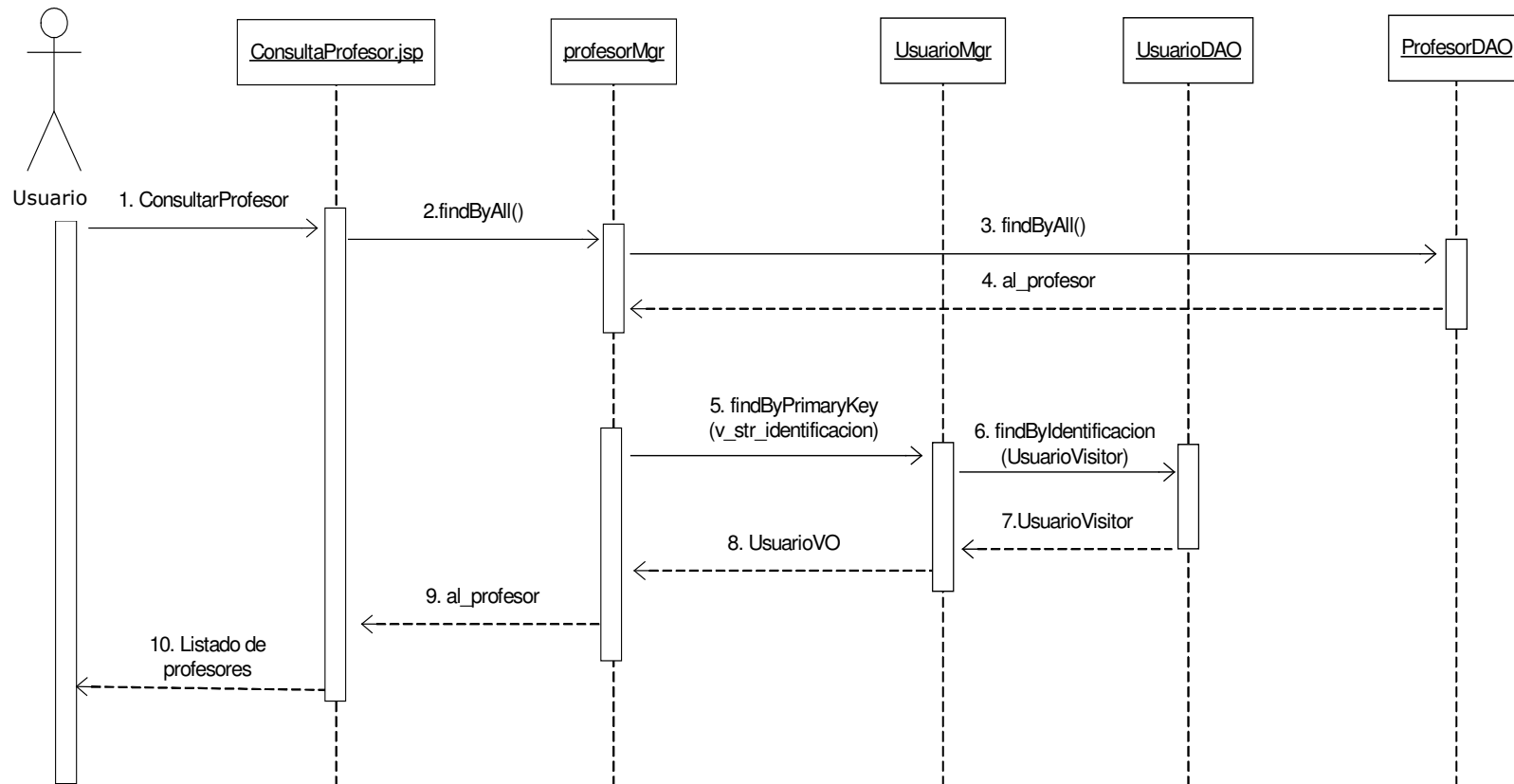


## 11.2.12 Administración de Profesores

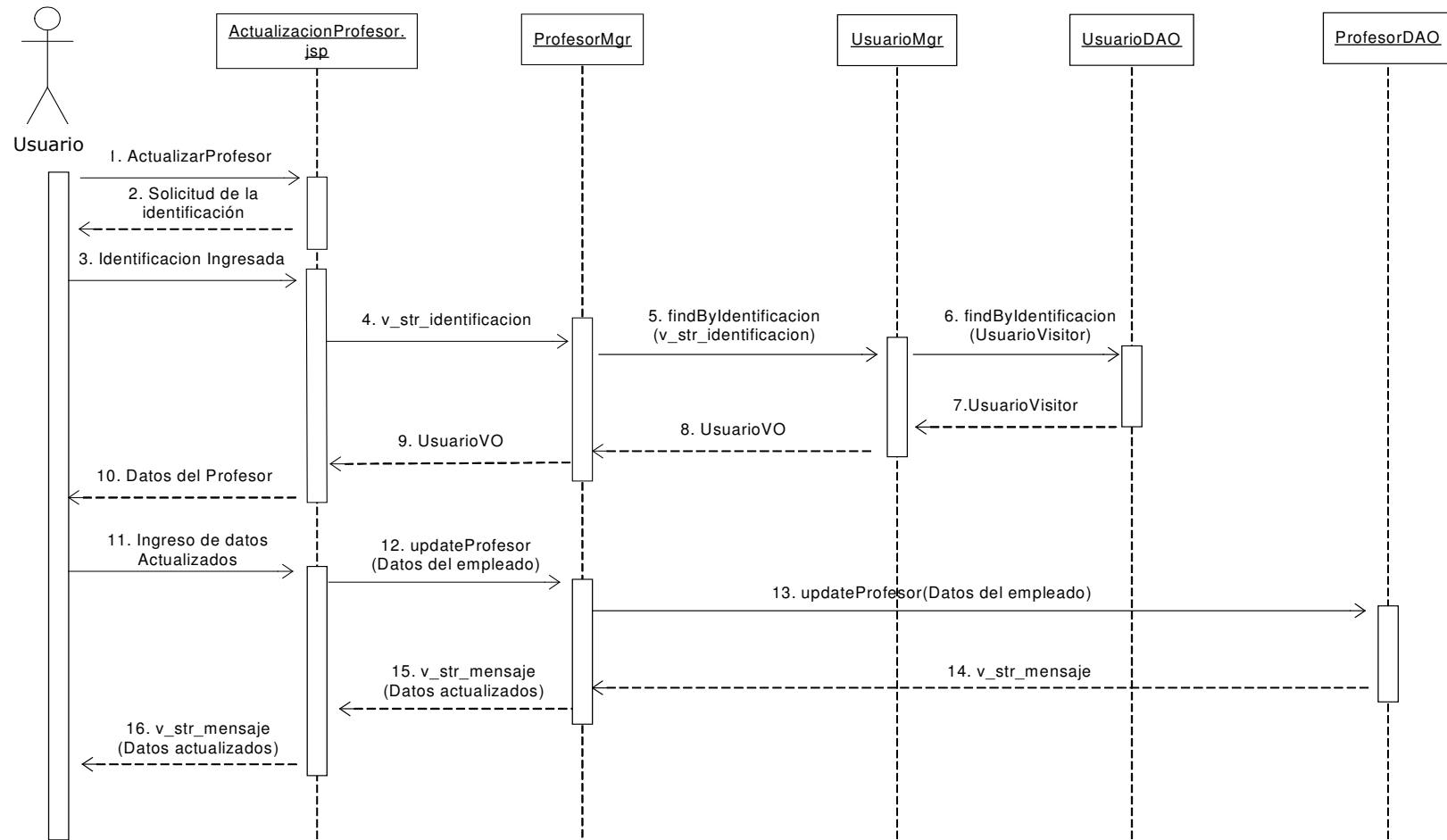
### 11.2.12.1 Registrar Profesor



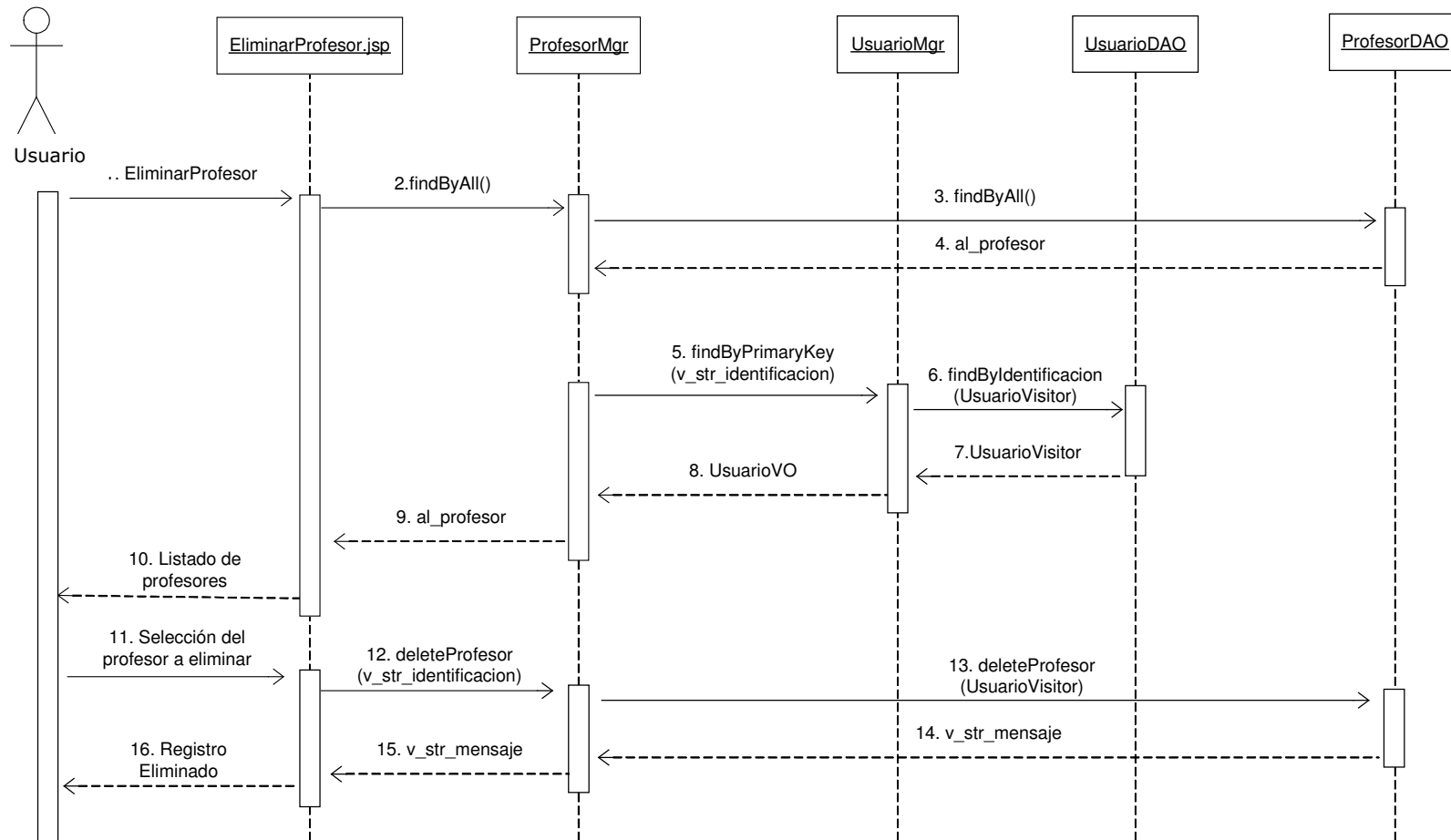
### 11.2.12.2 Consultar Profesor



### 11.2.12.3 Actualizar Profesor

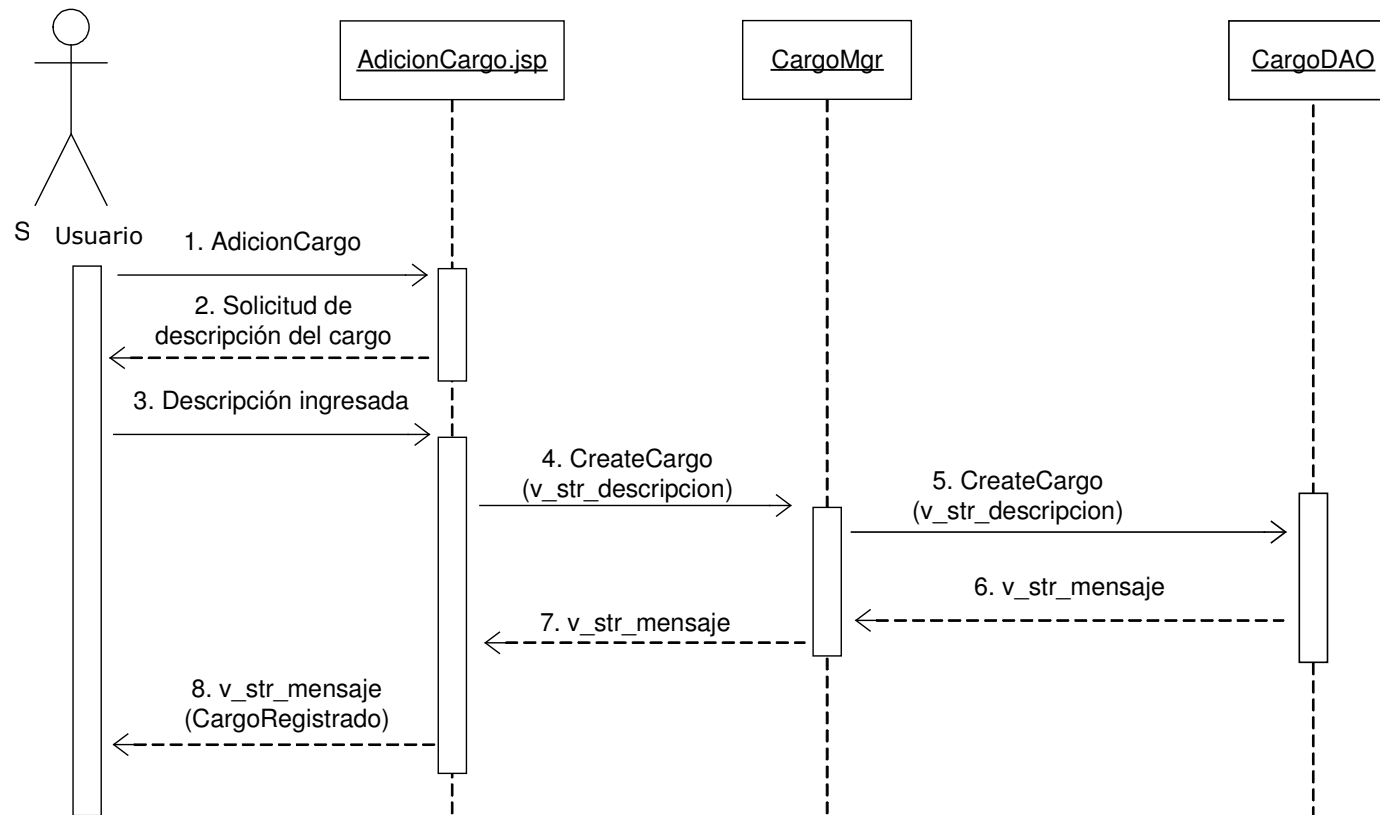


### 11.2.12.4 Eliminar Profesor

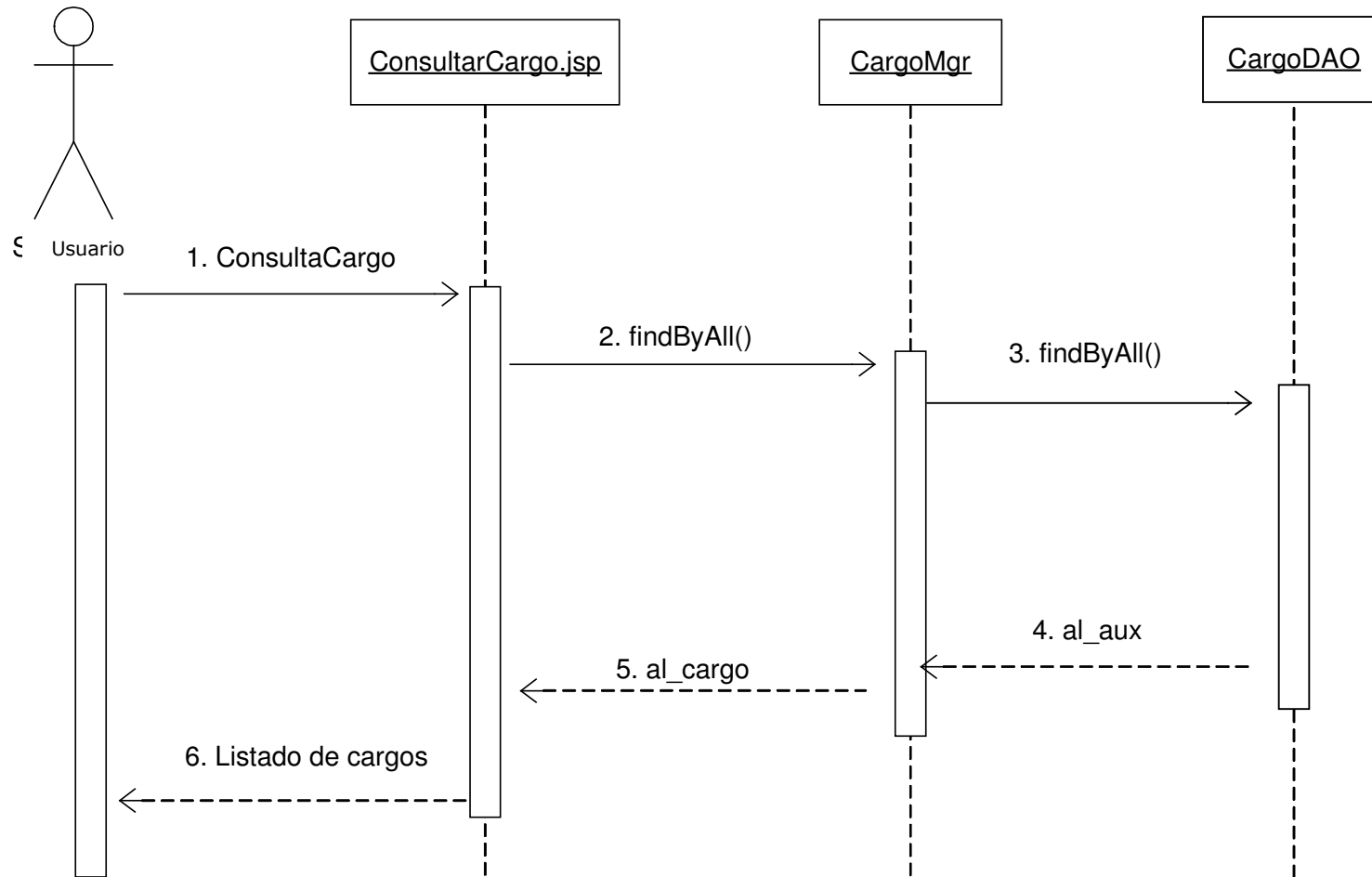


## 11.2.13 Administración de Cargos

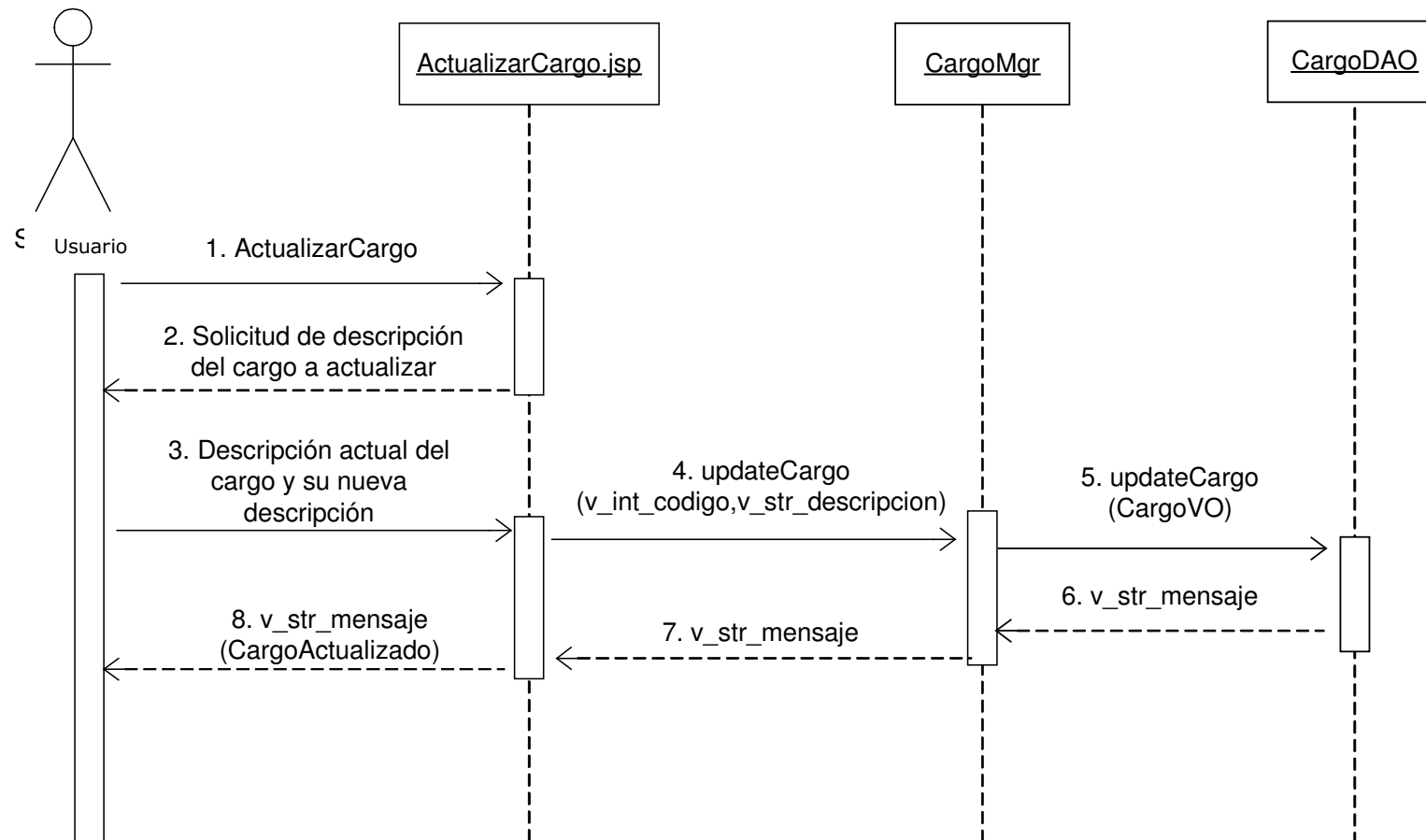
### 11.2.13.1 Registrar Cargos



### 11.2.13.2 Consultar Cargo

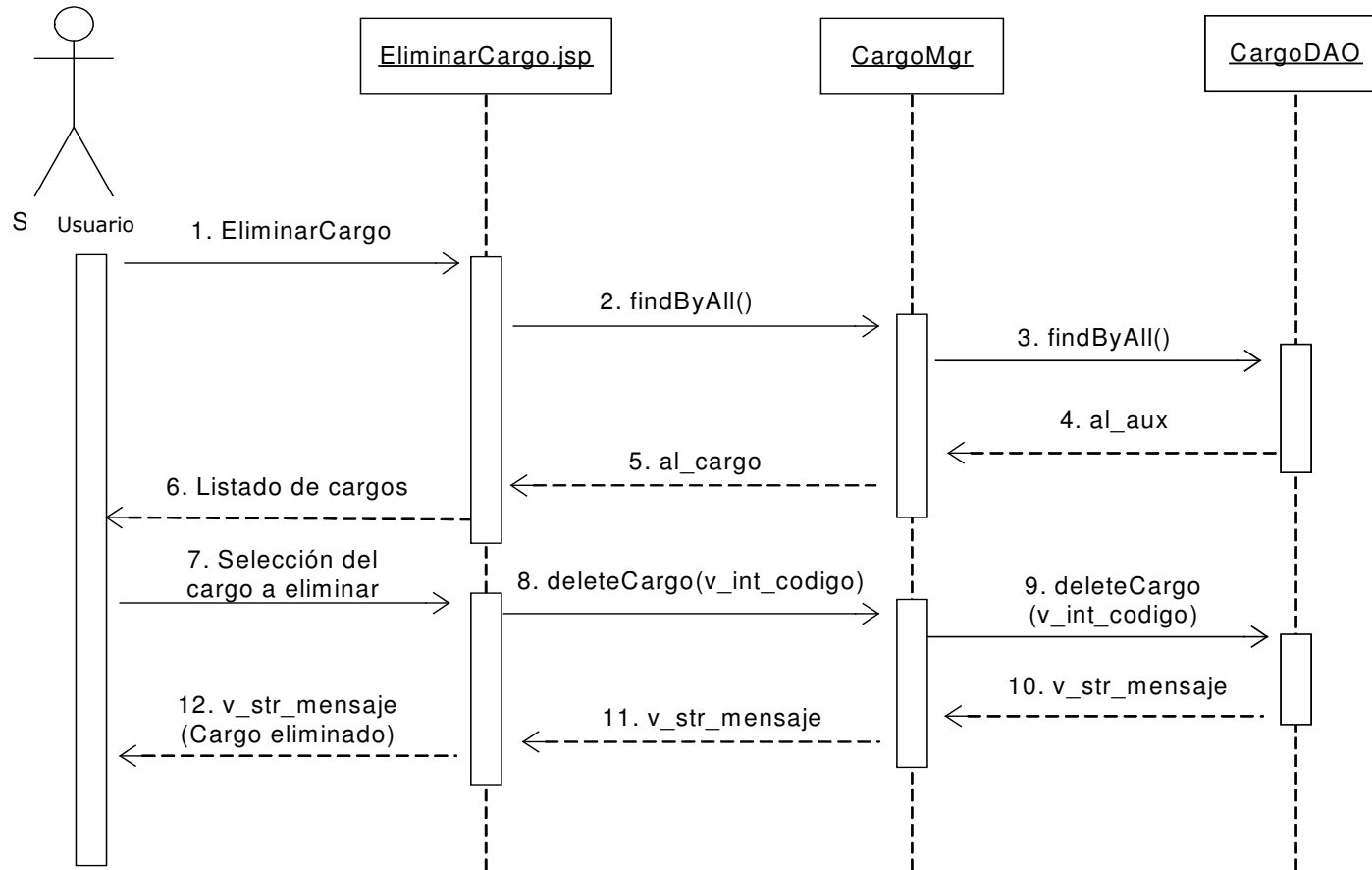


### 11.2.13.3 Actualizar Cargo



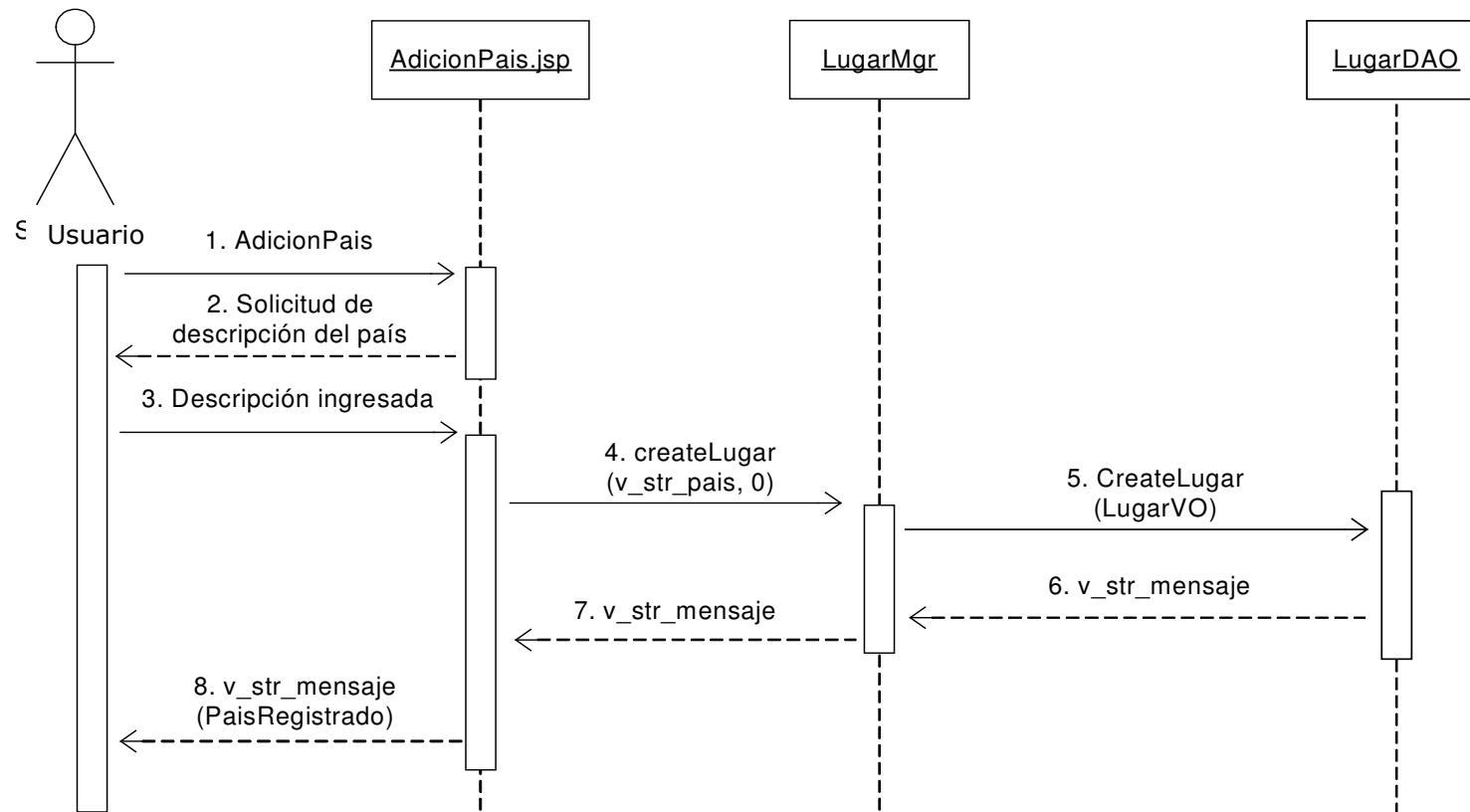


### 11.2.13.4 Eliminar Cargo

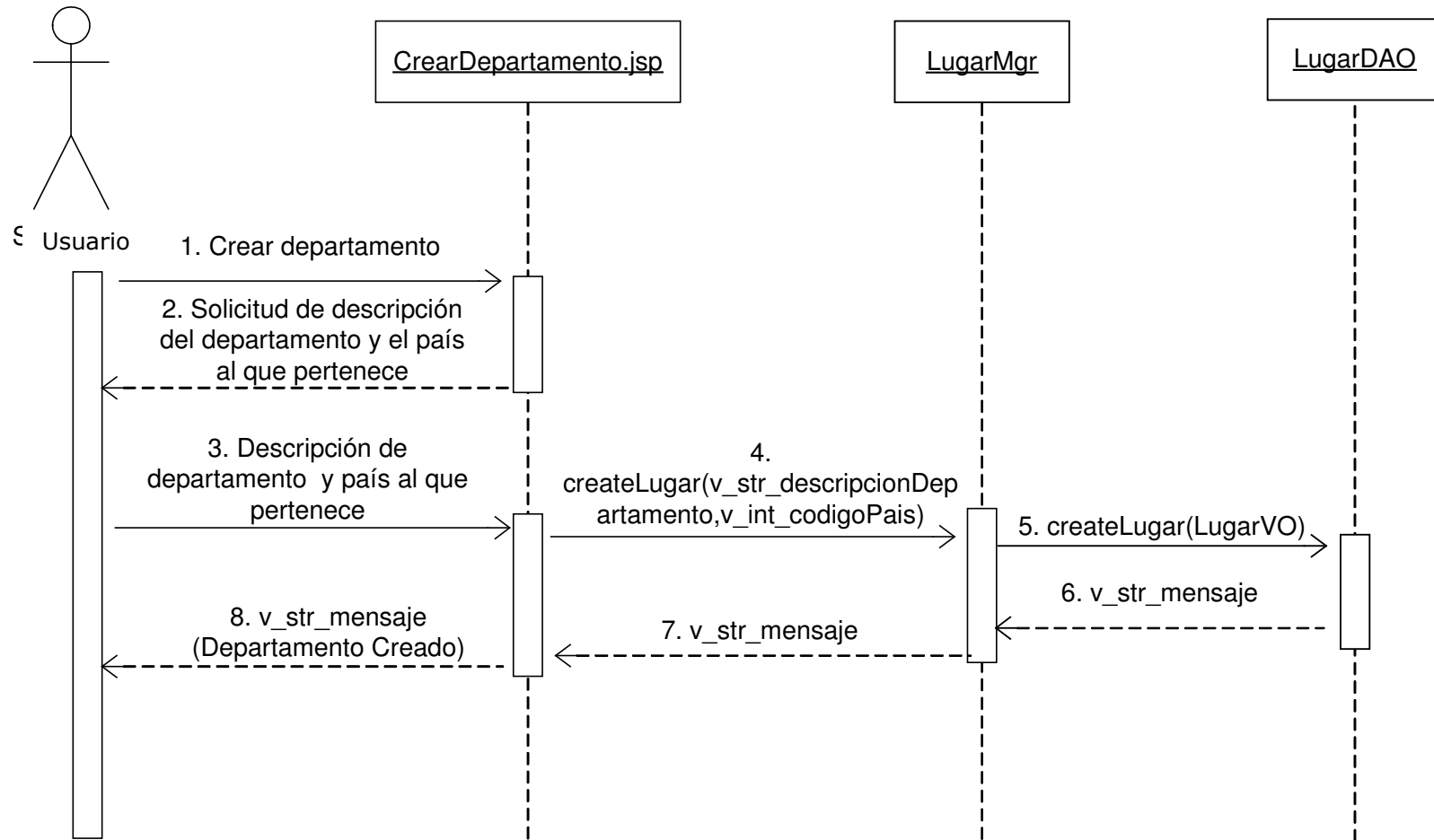


## 11.2.14 Administración de Lugares

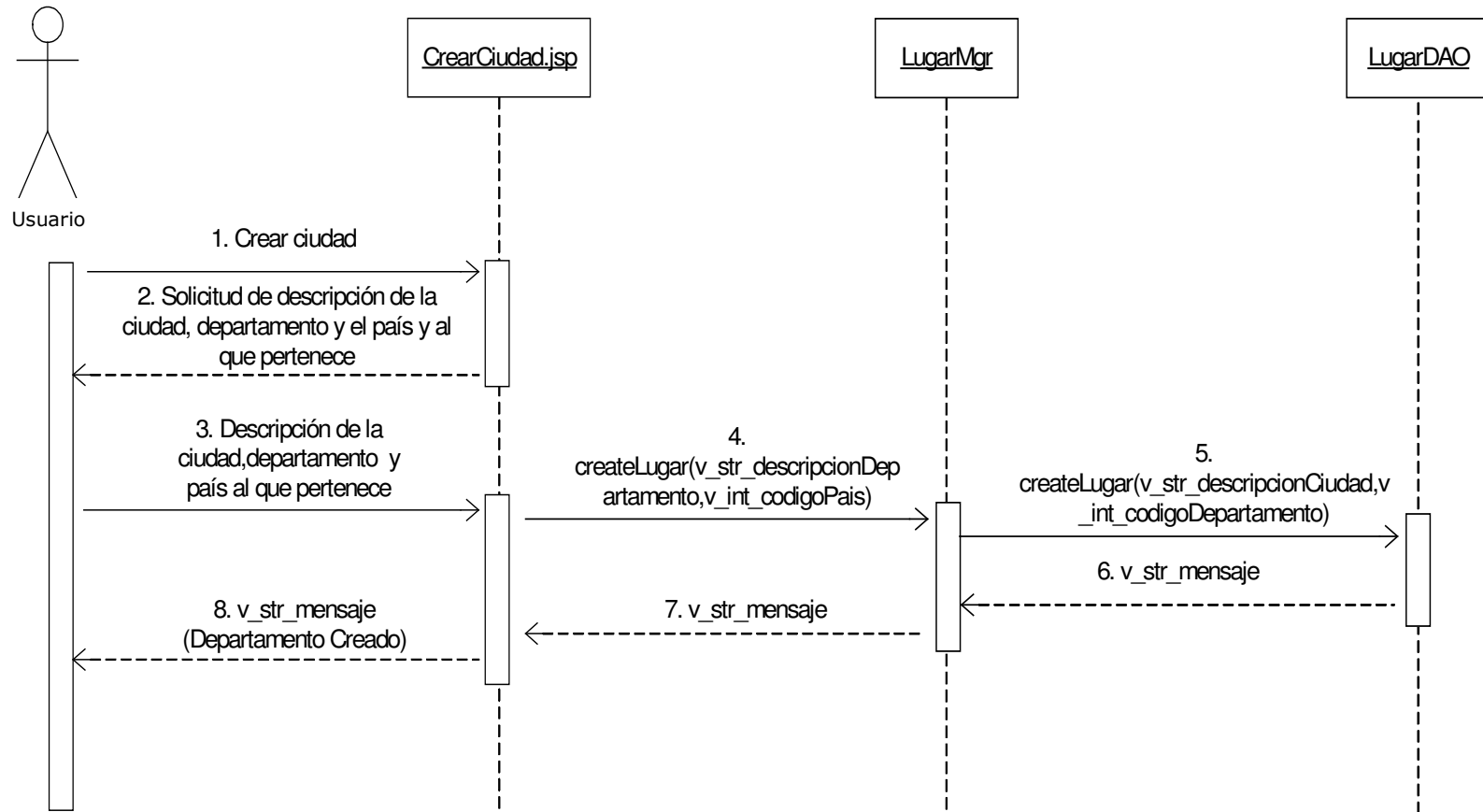
### 11.2.14.1 Registrar País



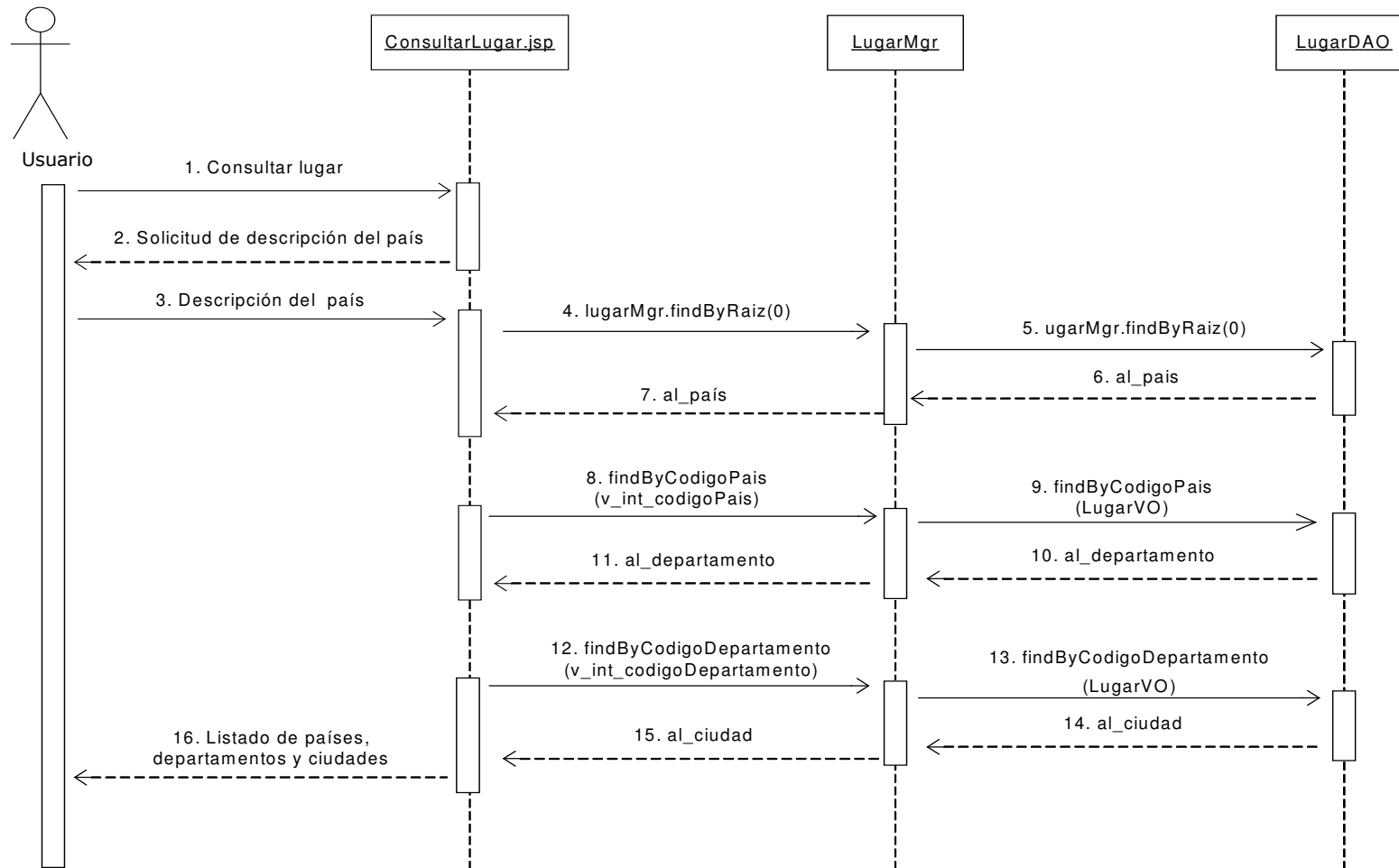
### 11.2.14.2 Registrar Departamento



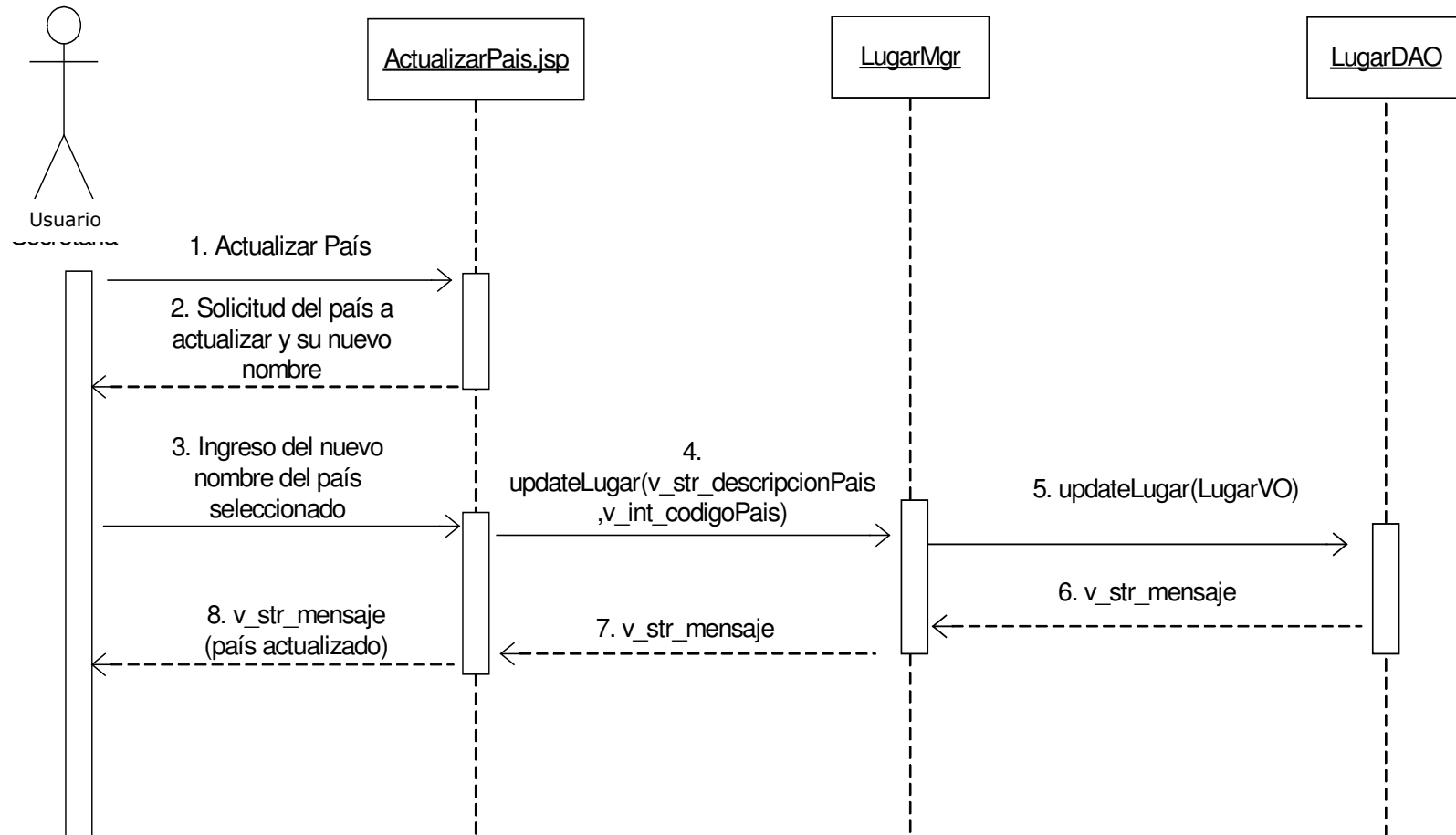
### 11.2.14.3 Registrar Ciudad



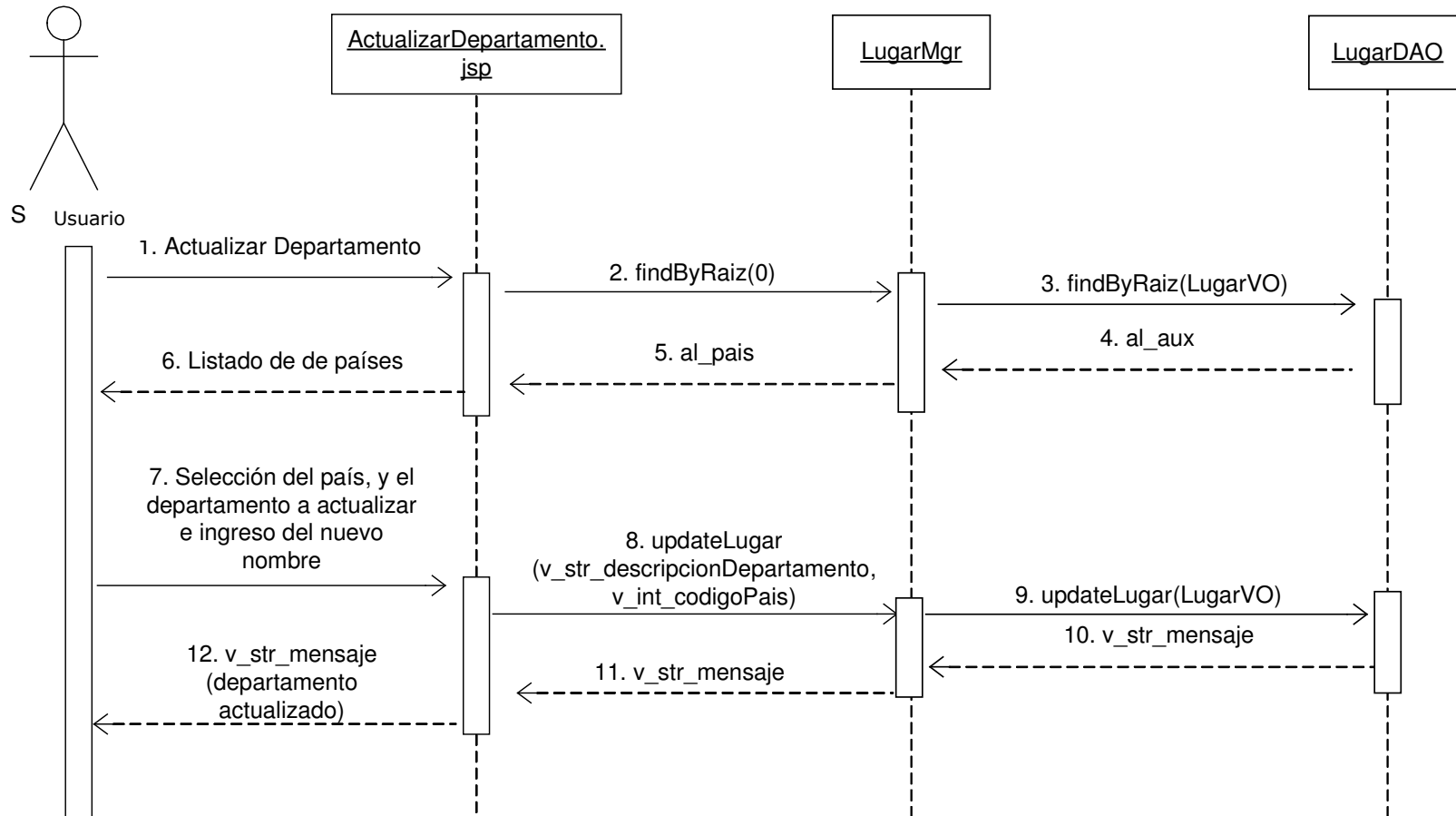
### 11.2.14.4 Consultar Lugares



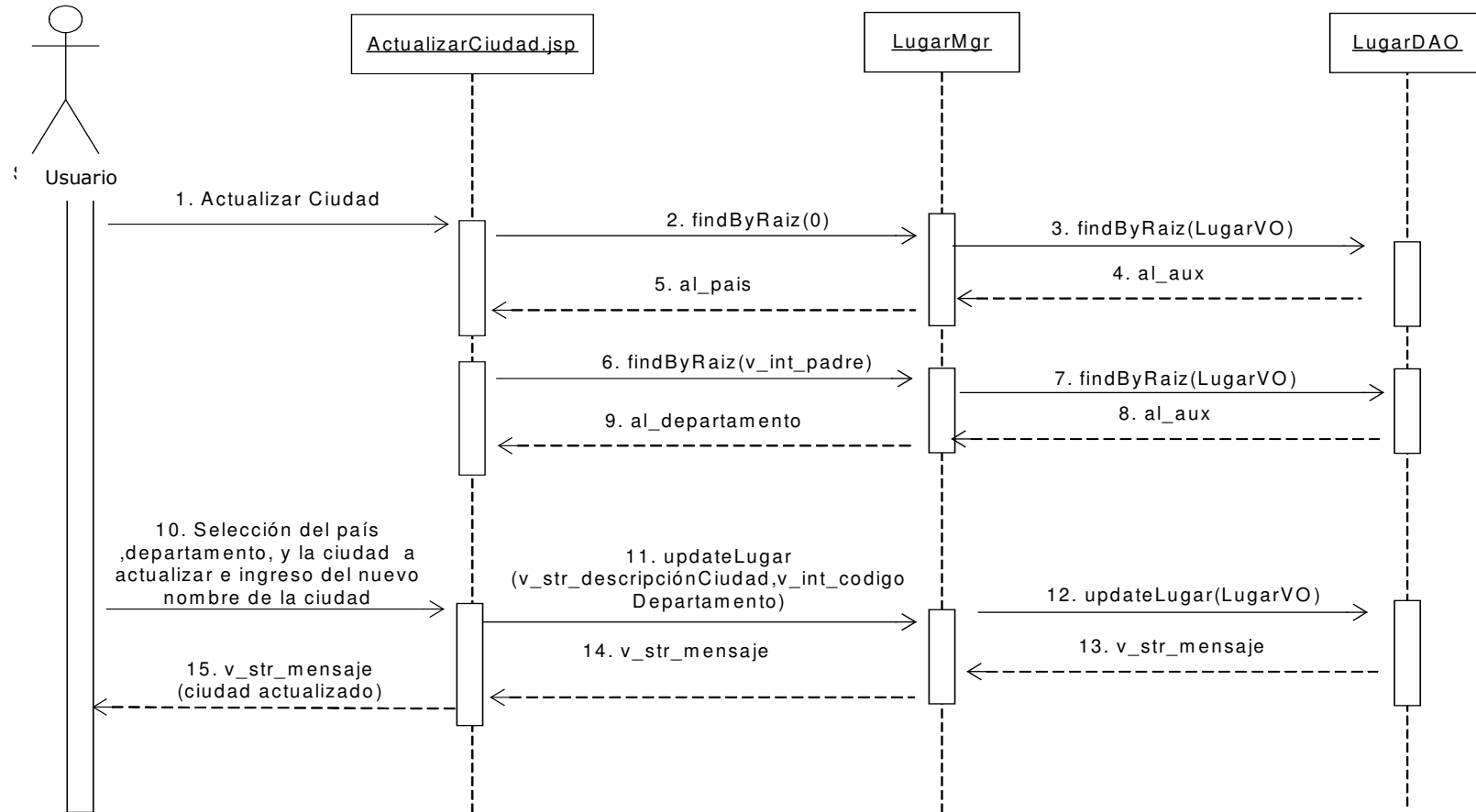
### 11.2.14.5 Actualizar País



### 11.2.14.6 Actualizar Departamento

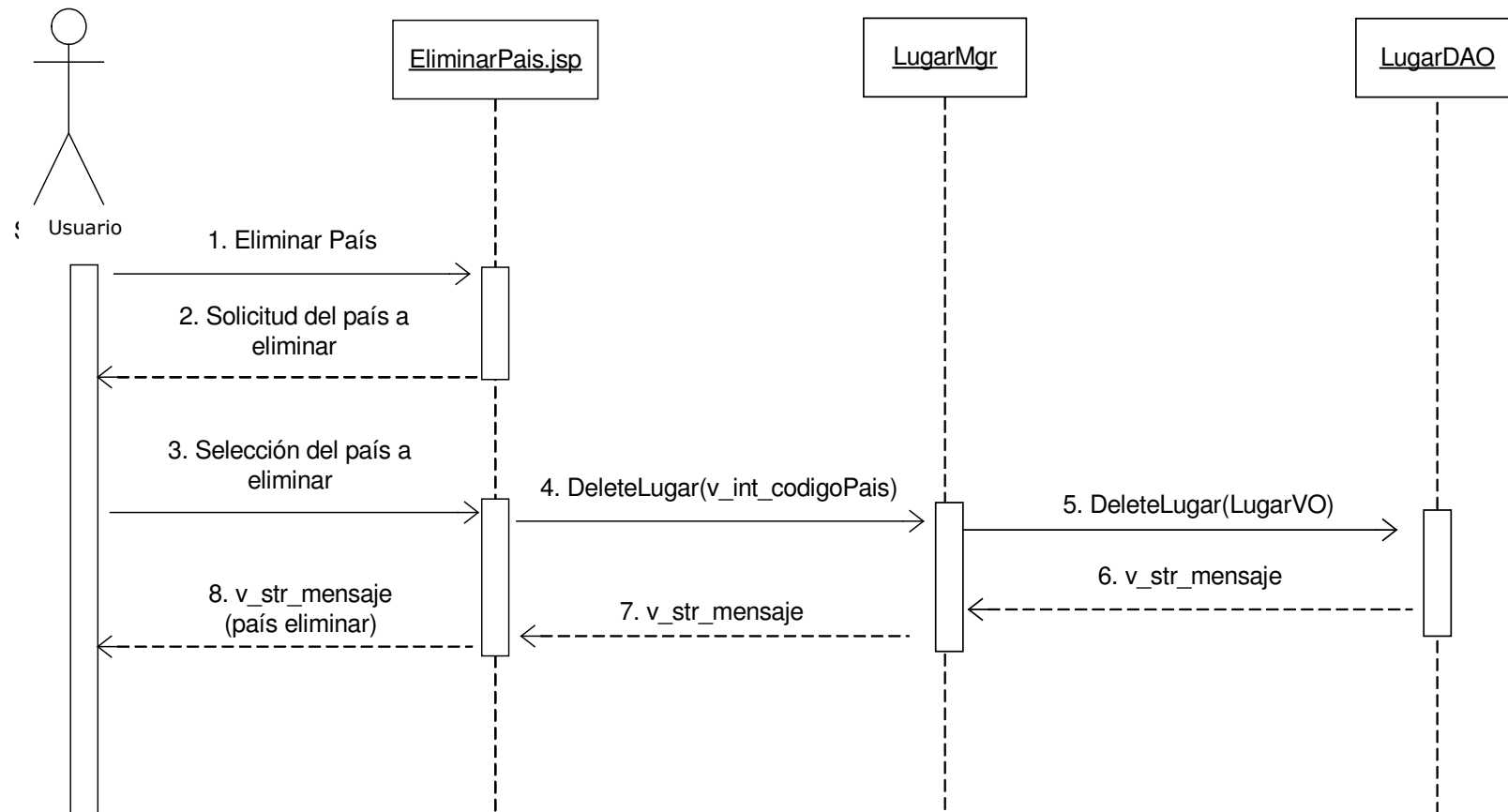


### 11.2.14.7 Actualizar Ciudad

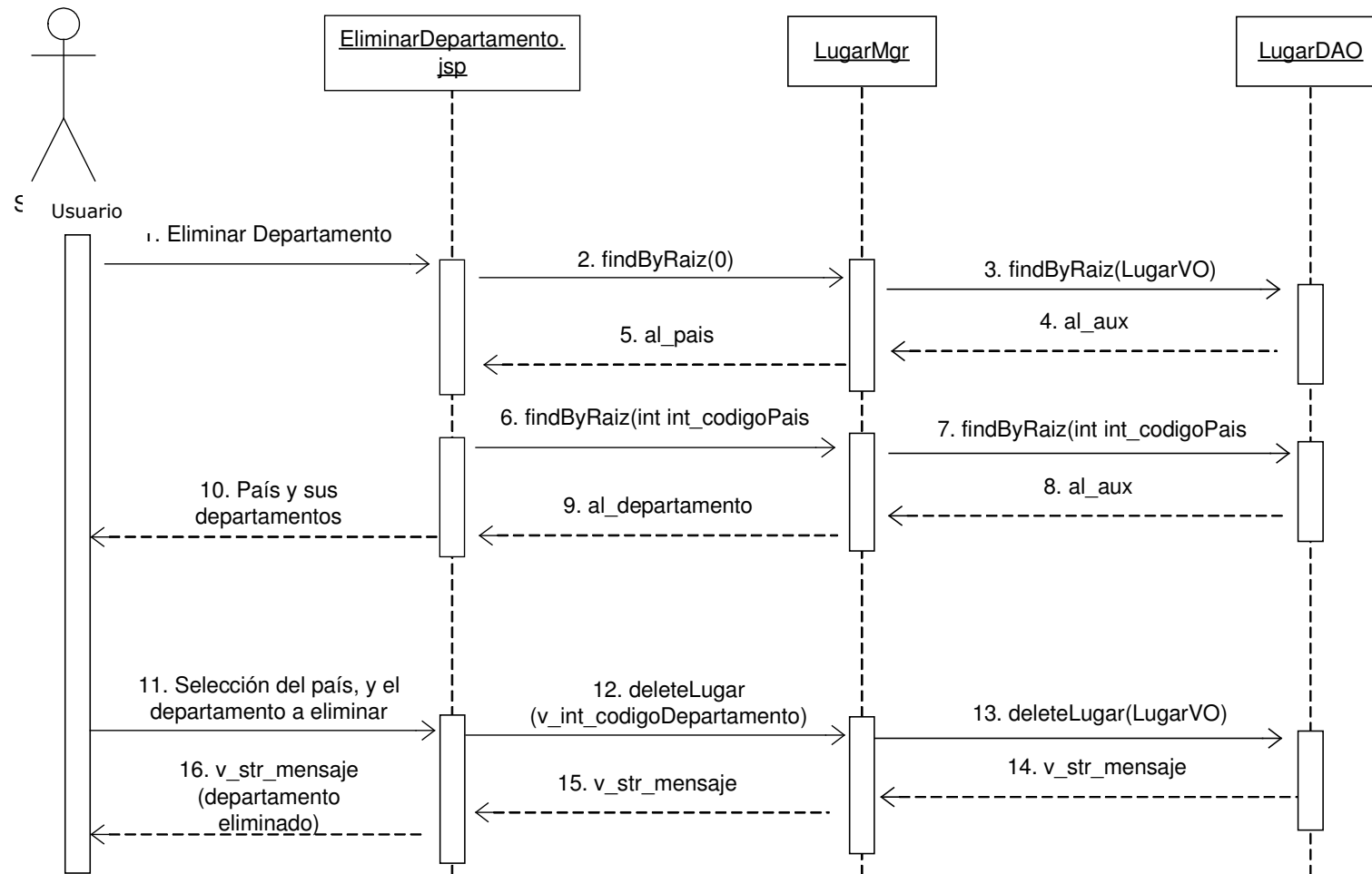




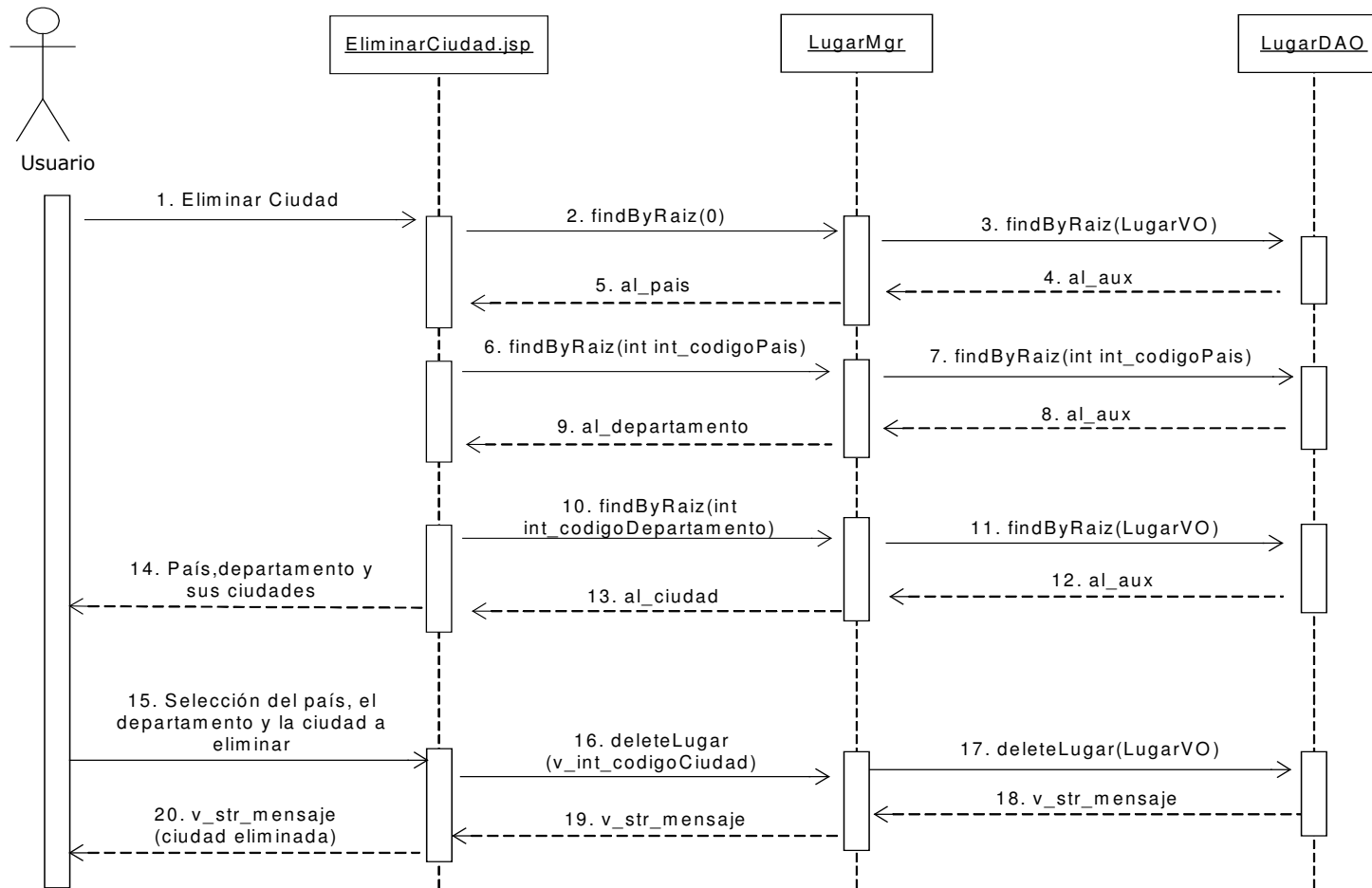
### 11.2.14.8 Eliminar País



### 11.2.14.9 Eliminar Departamento

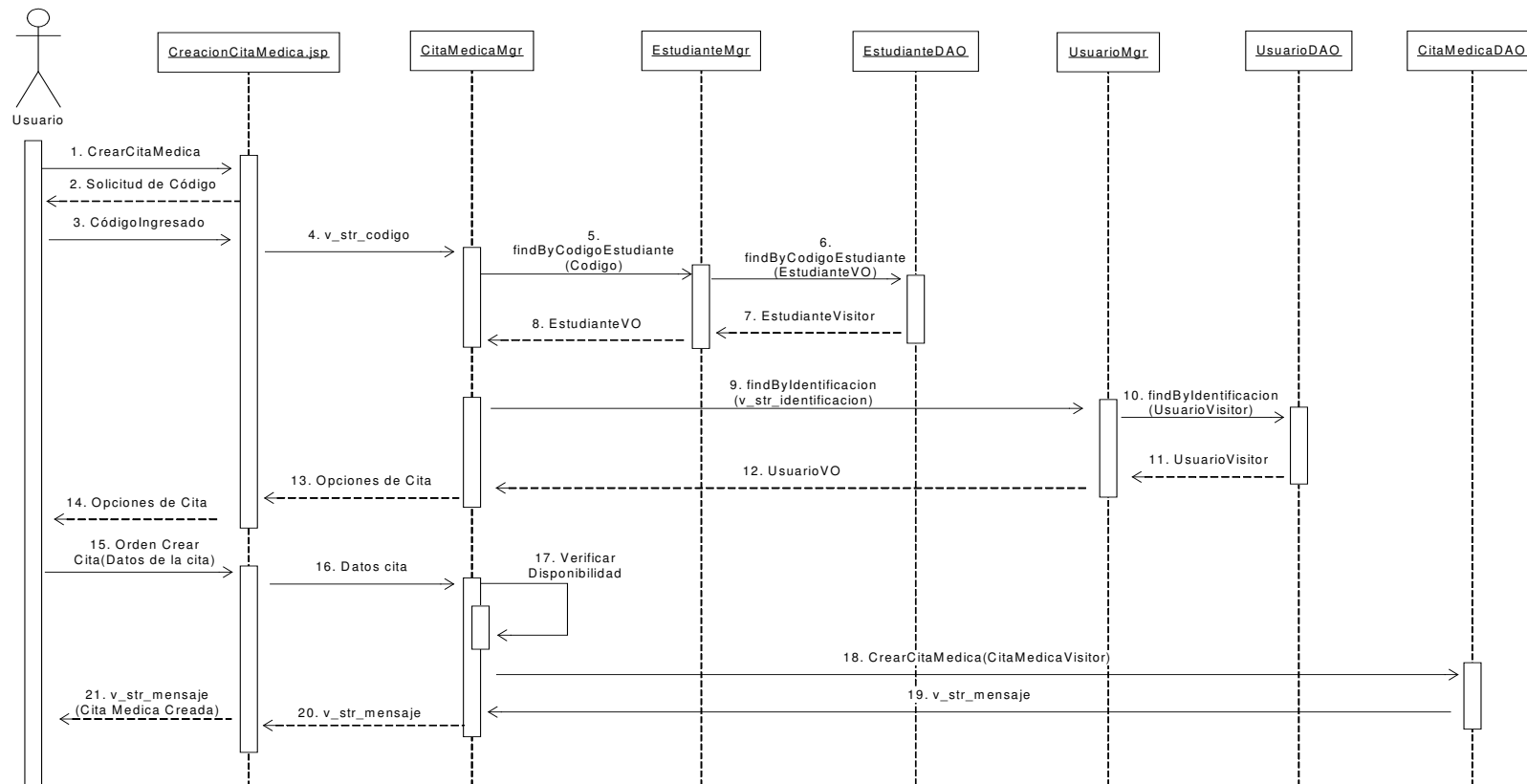


### 11.2.14.10 Eliminar Ciudad

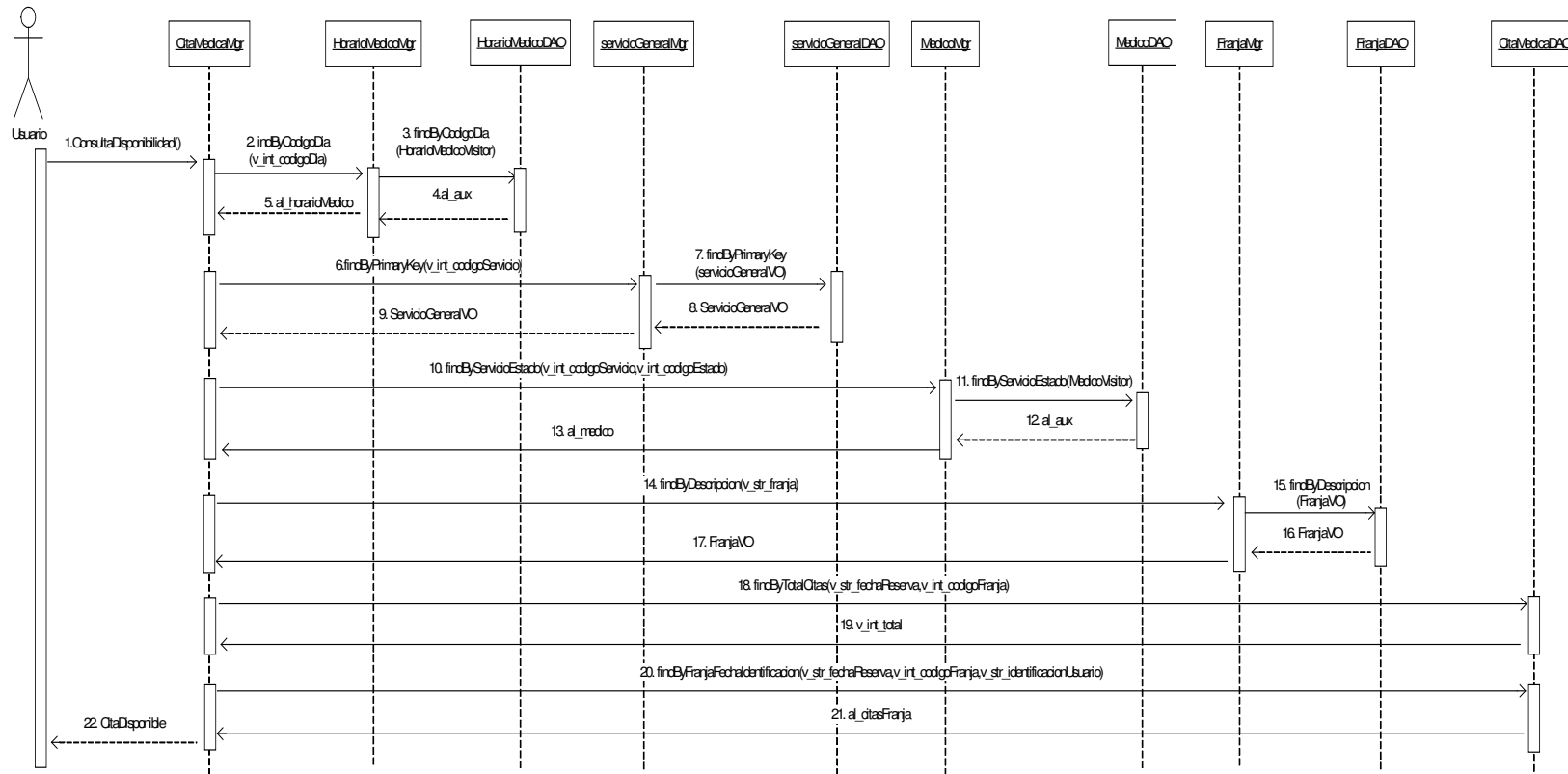


## 11.2.15 Administración de Citas Médicas

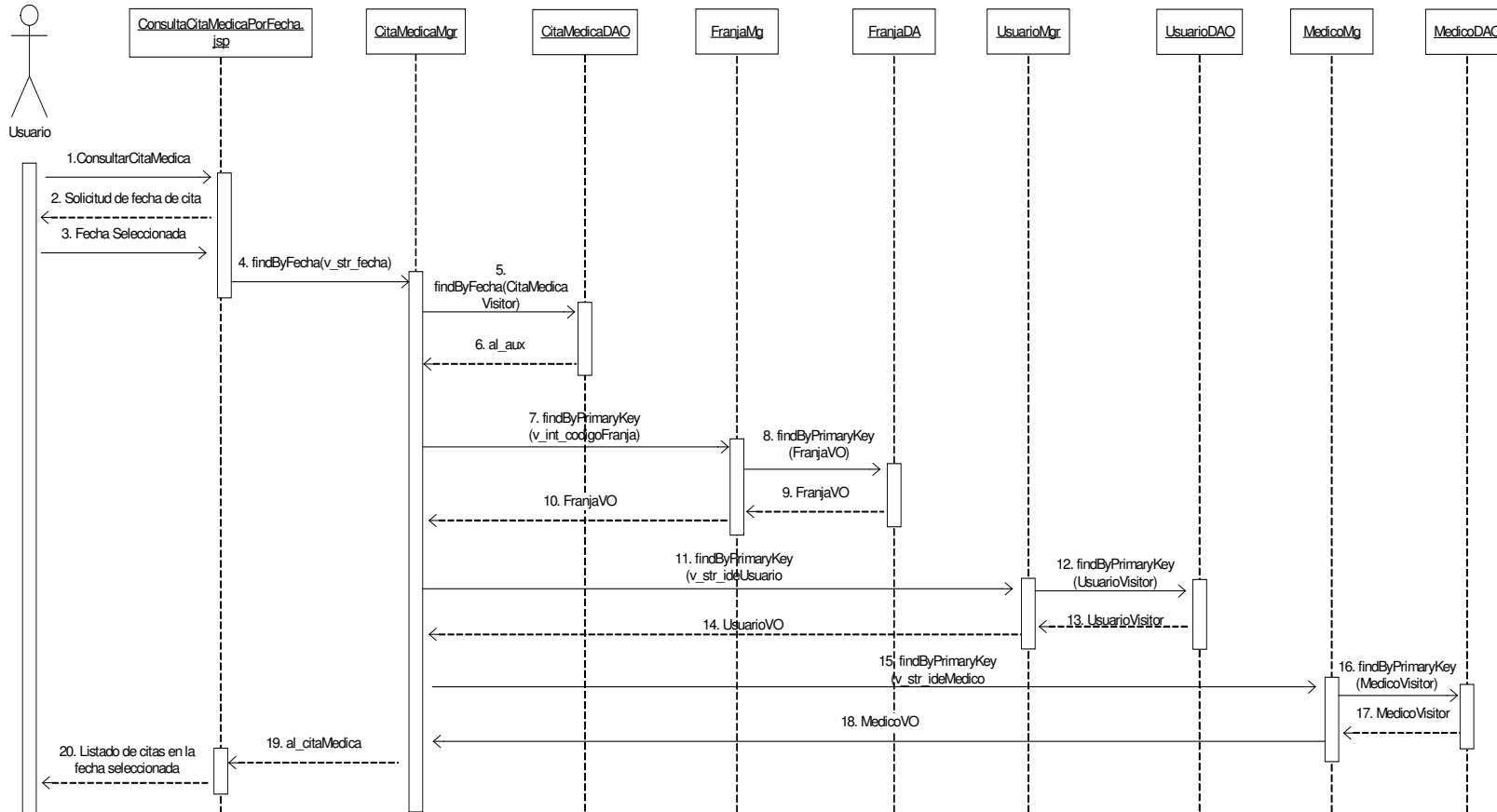
### 11.2.15.1 Crear Cita Médica



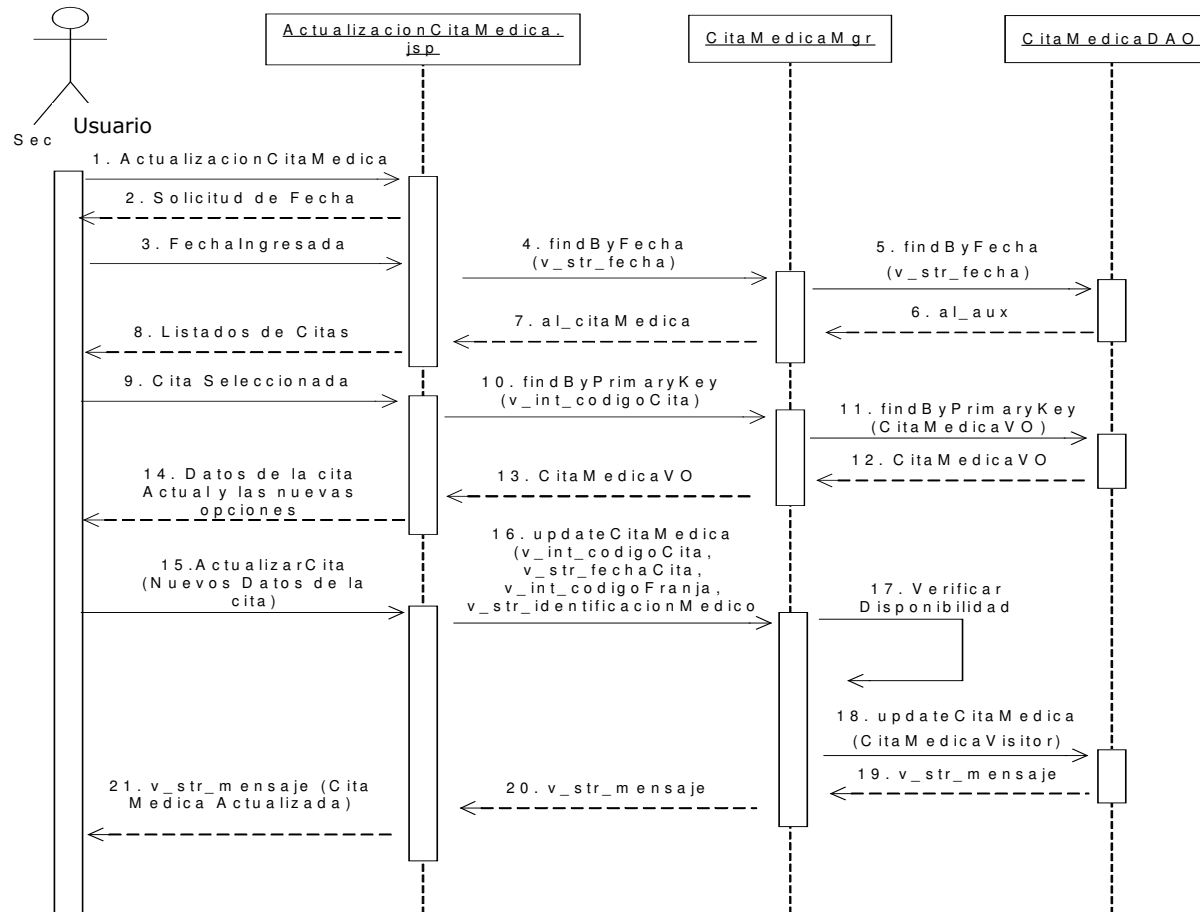
### 11.2.15.2 Consultar Disponibilidad de Citas Médicas



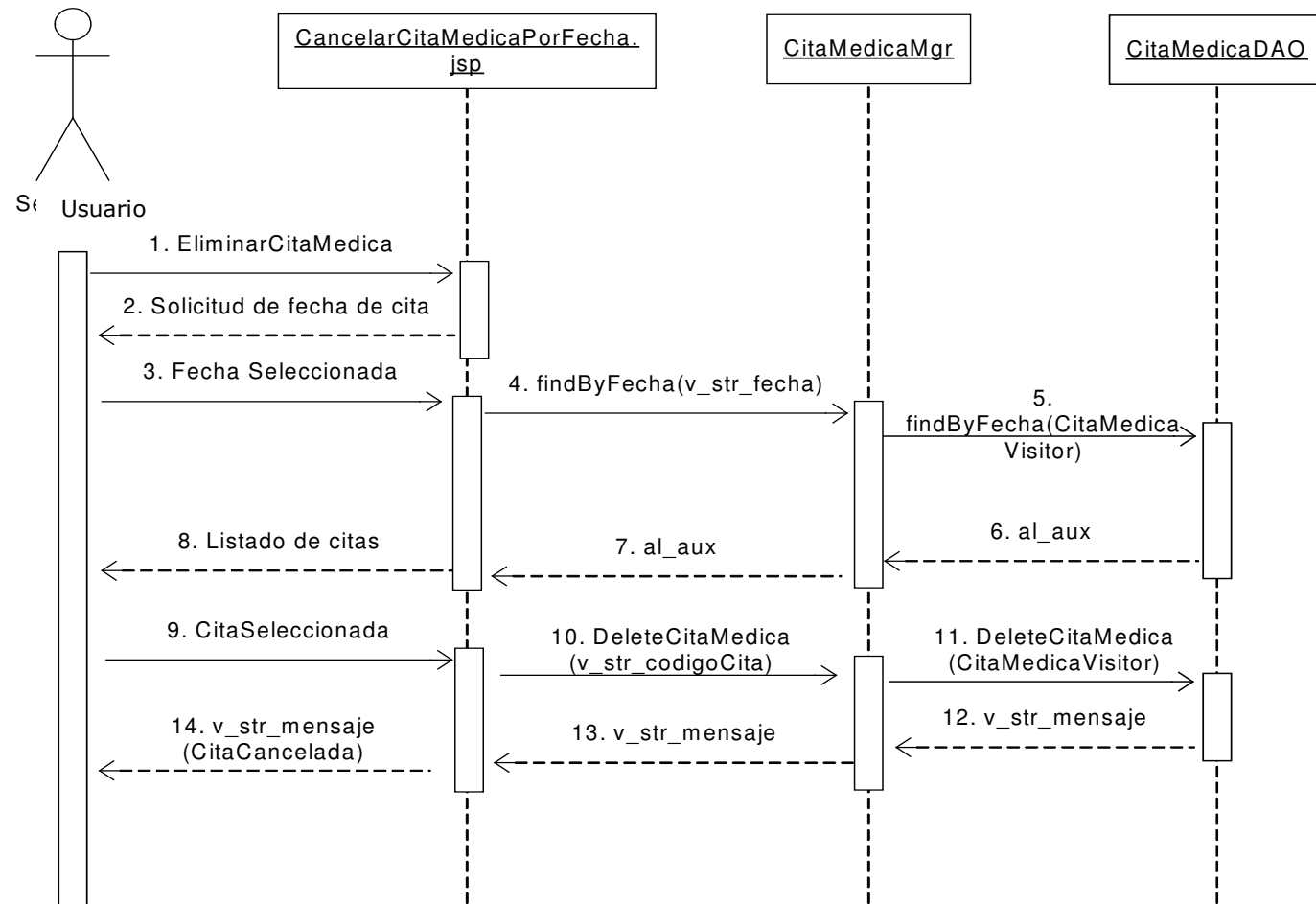
### 11.2.15.3 Consultar Cita Médica



### 11.2.15.4 Actualizar Cita Médica



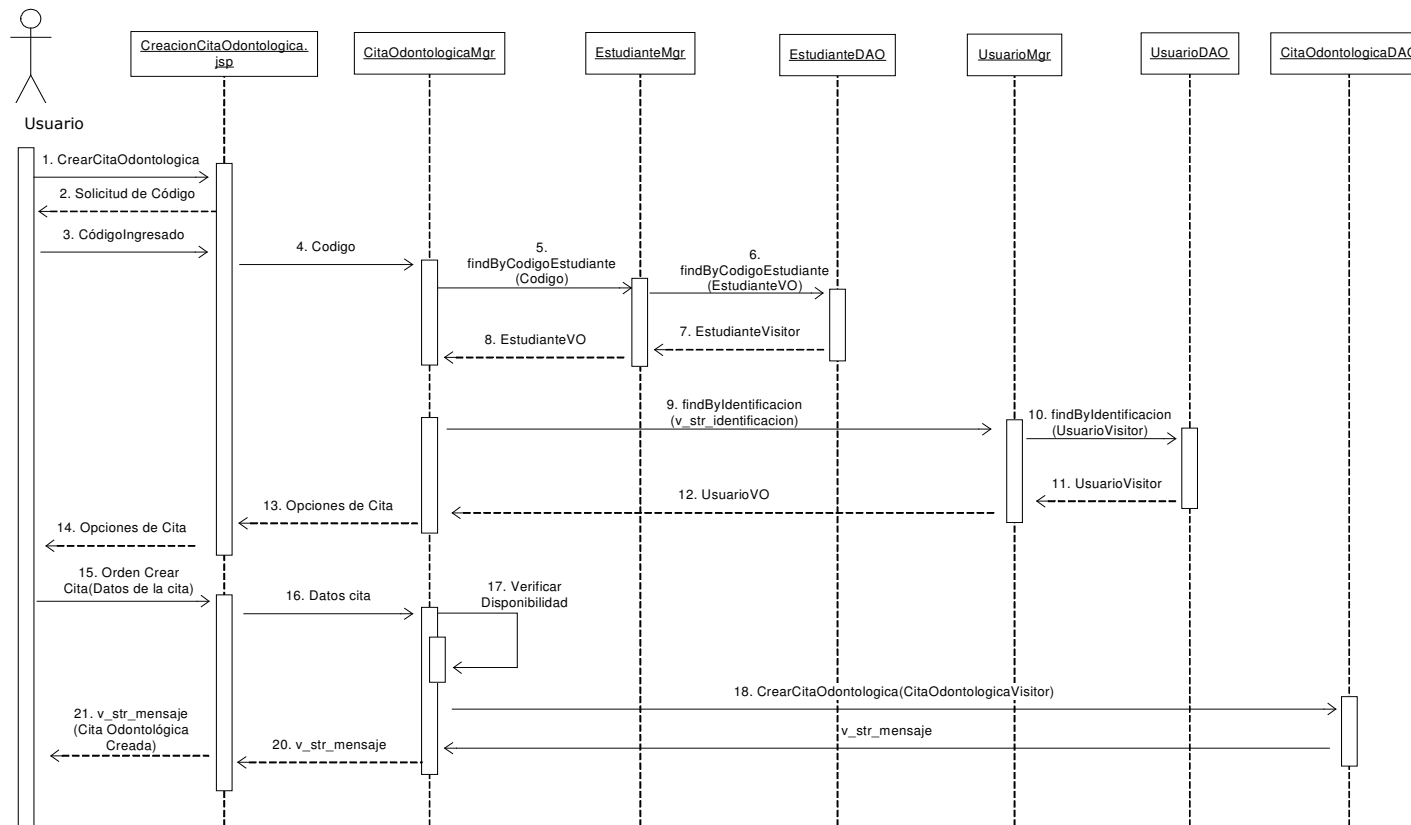
### 11.2.15.5 Cancelar Cita Médica



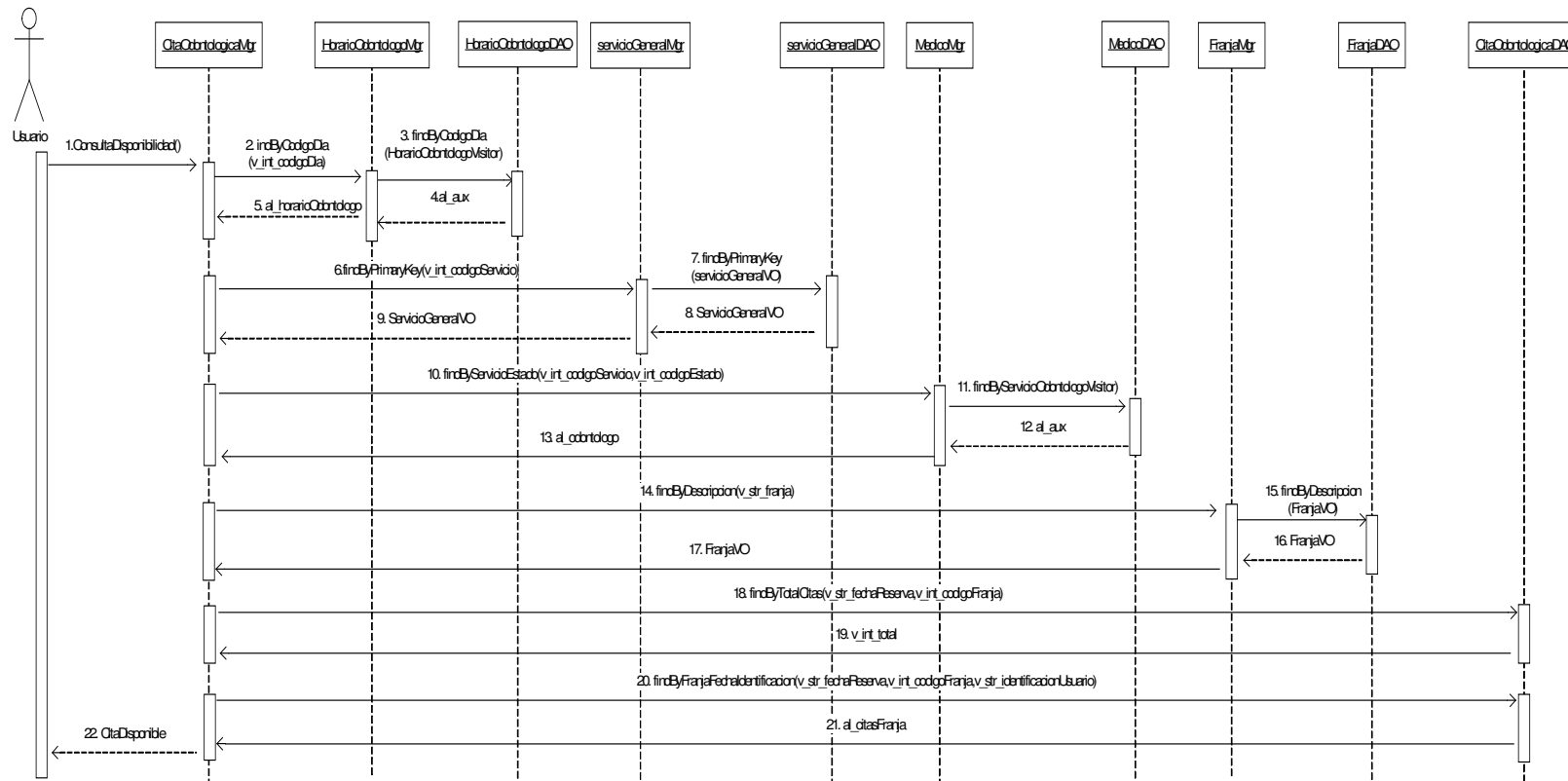


## 11.2.16 Administración de Citas Odontológicas

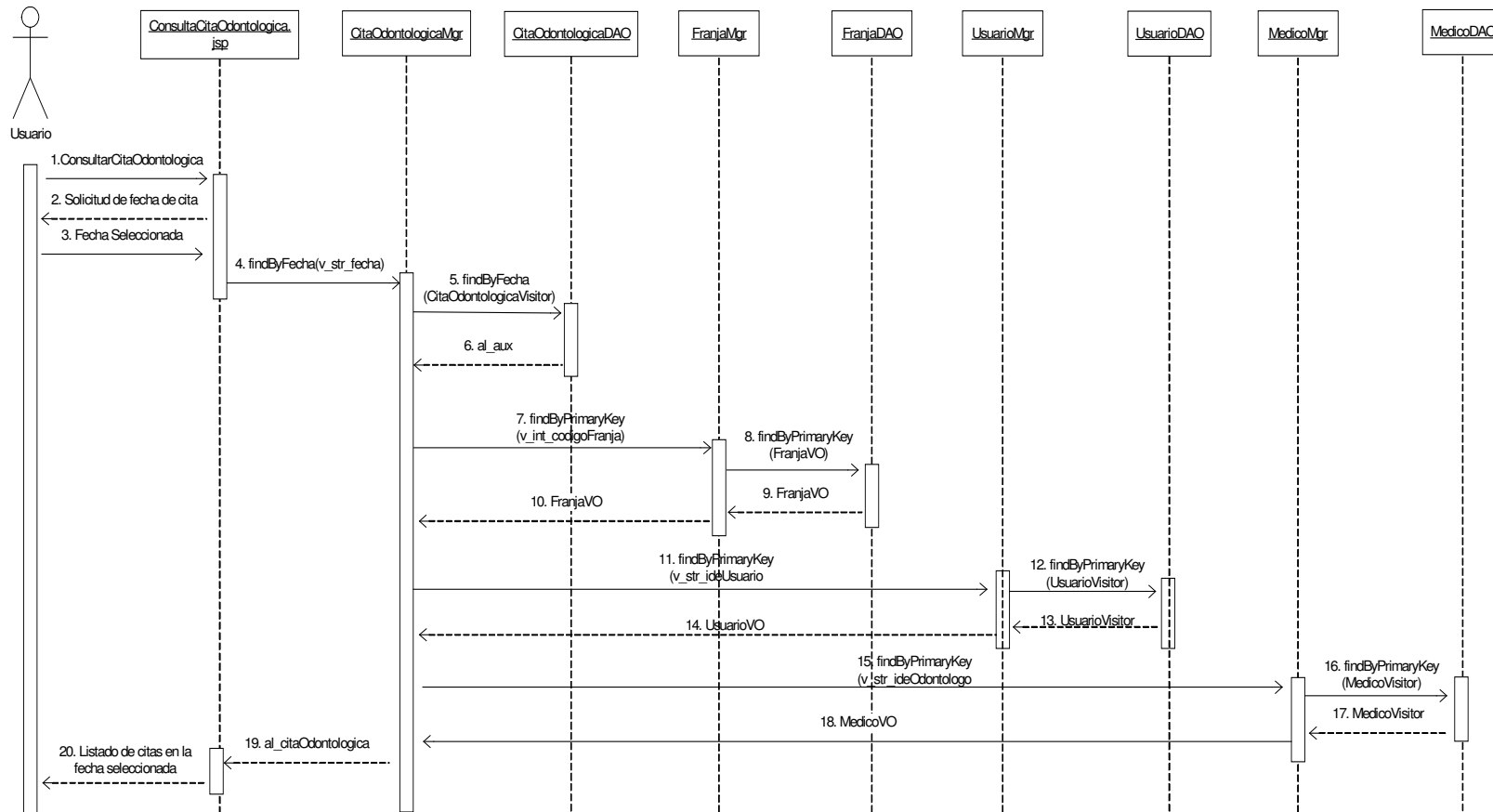
### 11.2.16.1 Crear Cita Odontológica



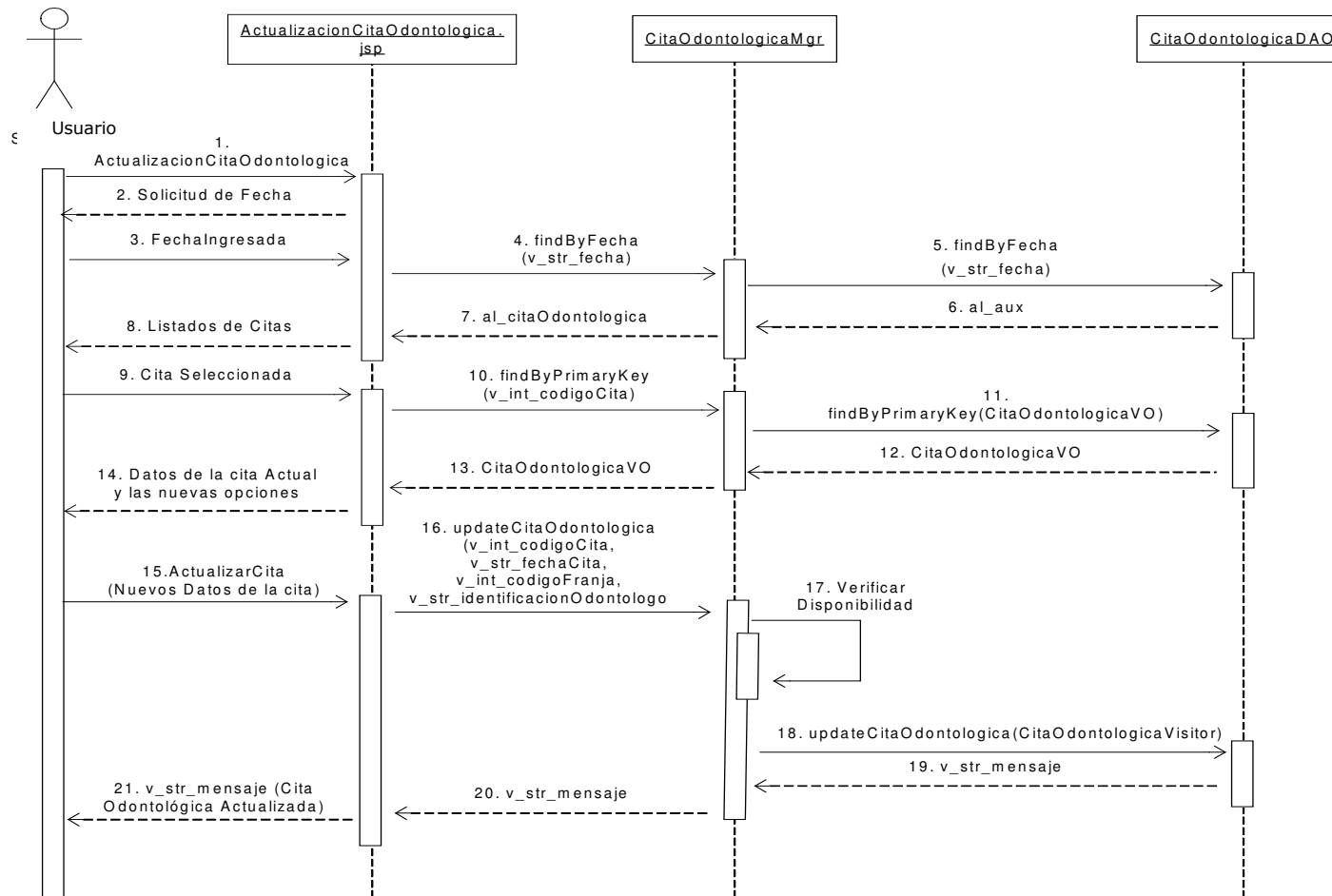
## 11.2.16.2 Consultar Disponibilidad de Citas Odontológicas



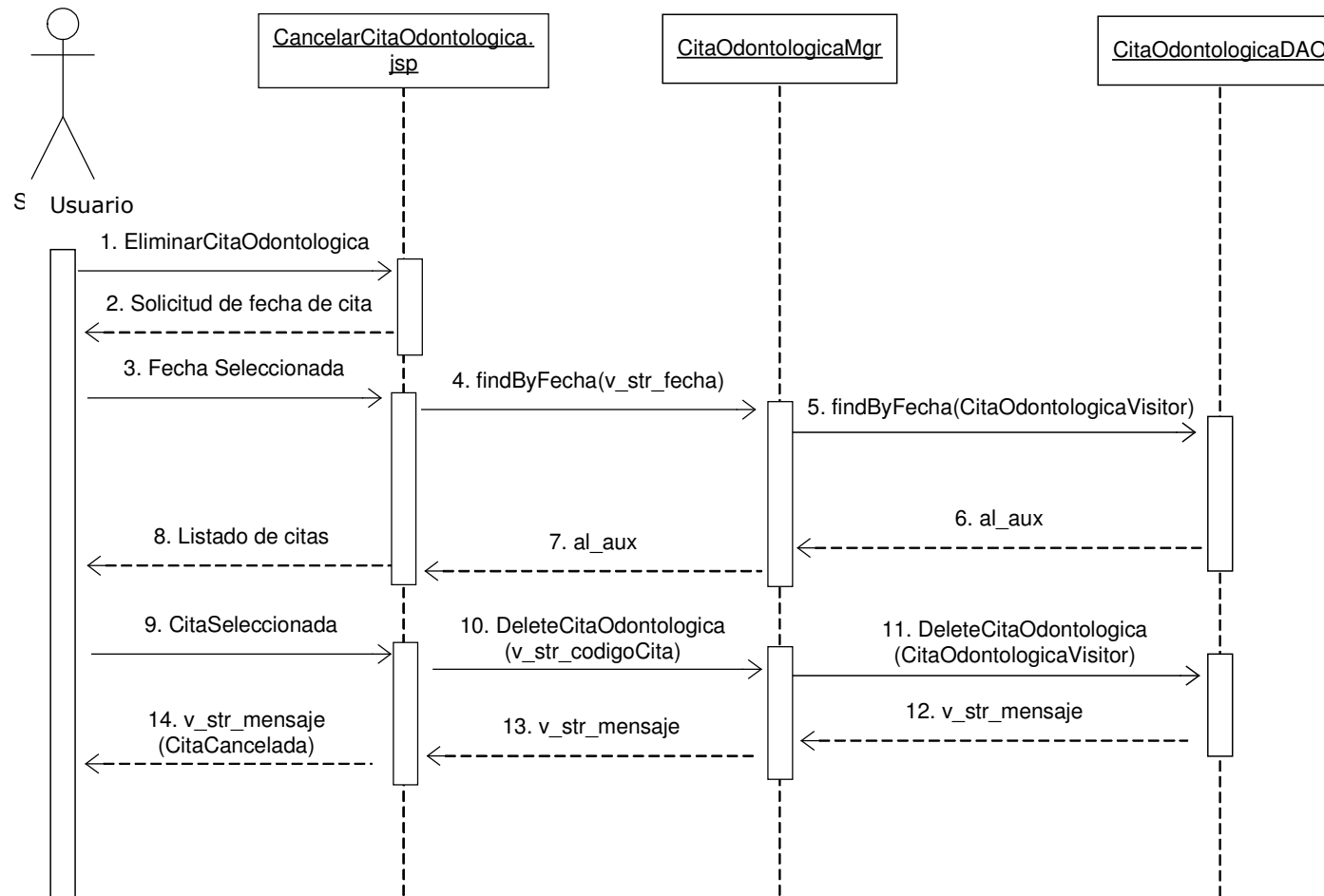
### 11.2.16.3 Consultar Cita Odontológica



### 11.2.16.4 Actualizar Cita Odontológica

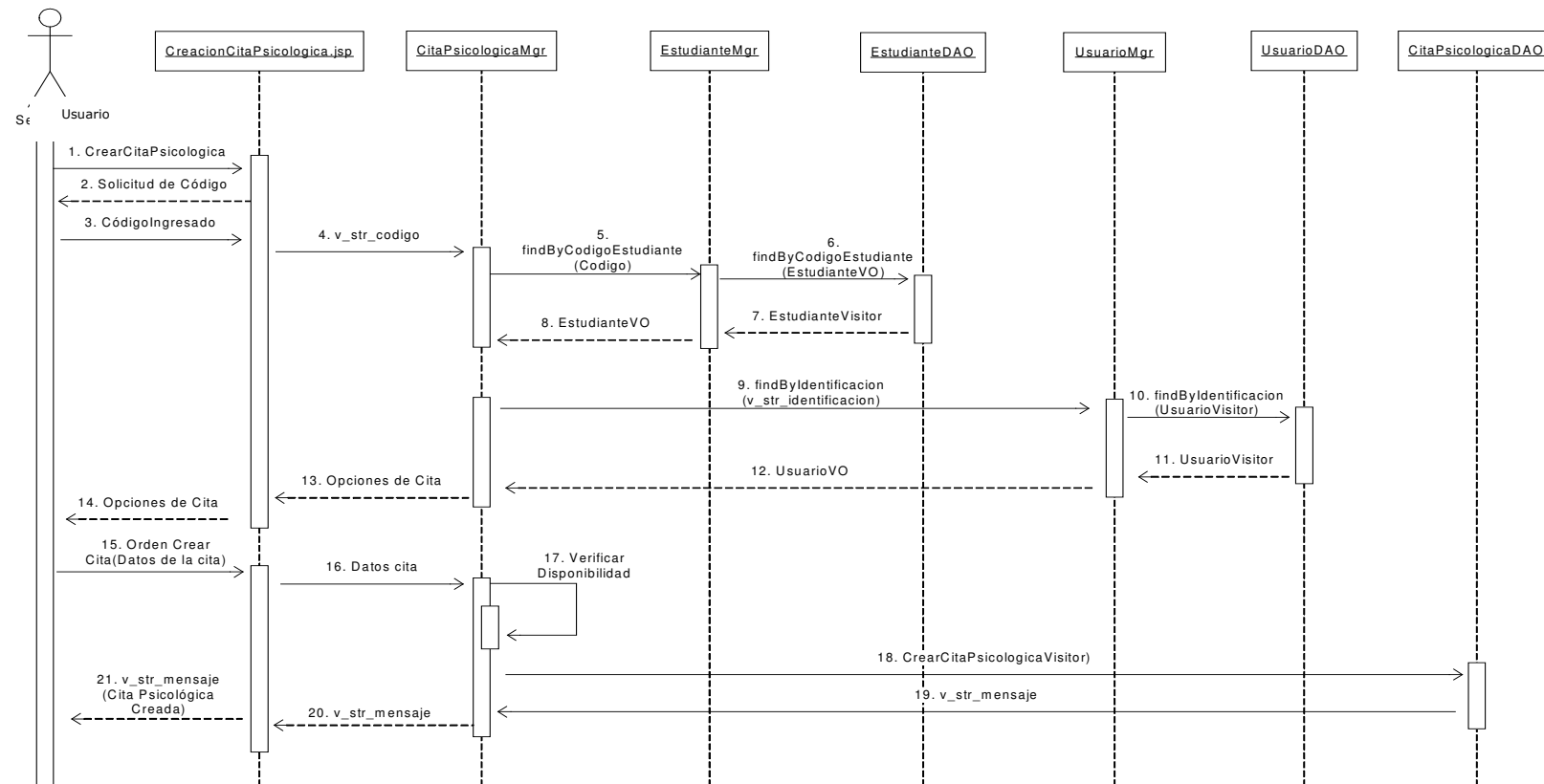


### 11.2.16.5 Cancelar Cita Odontológica

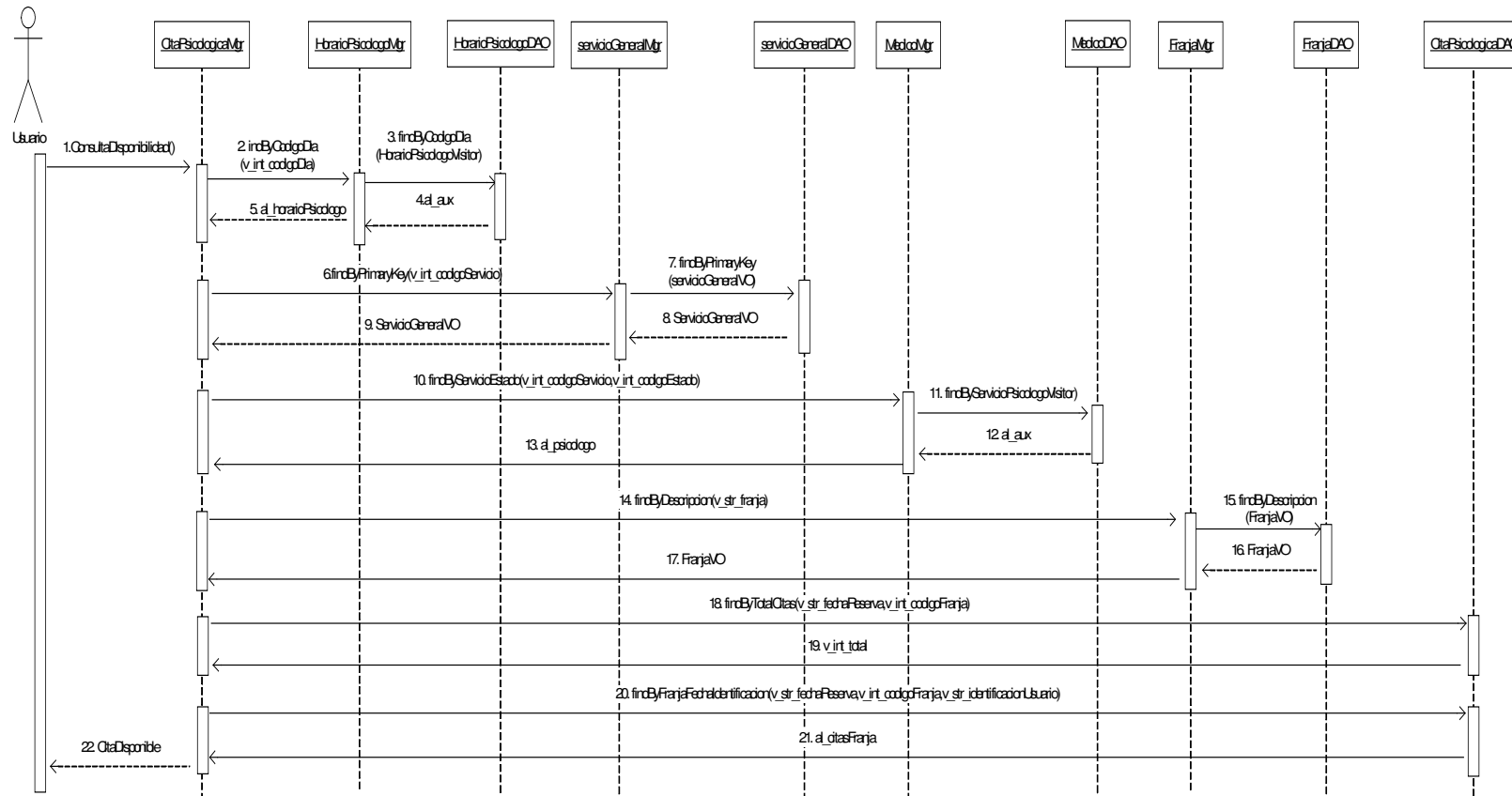


## 11.2.17 Administración De Citas Psicológicas

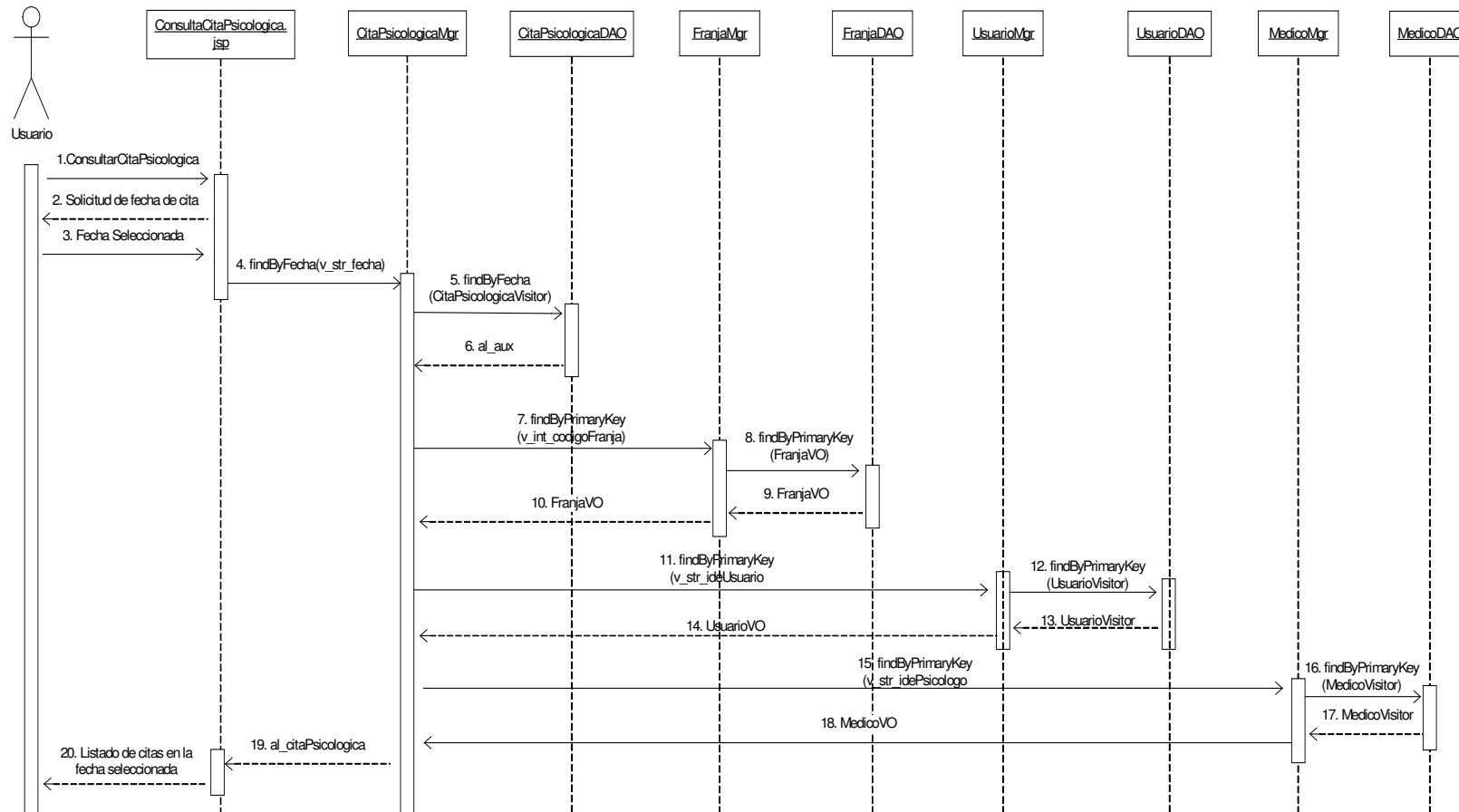
### 11.2.17.1 Crear Cita Psicológica



### 11.2.17.2 Verificar Disponibilidad Citas Psicológicas

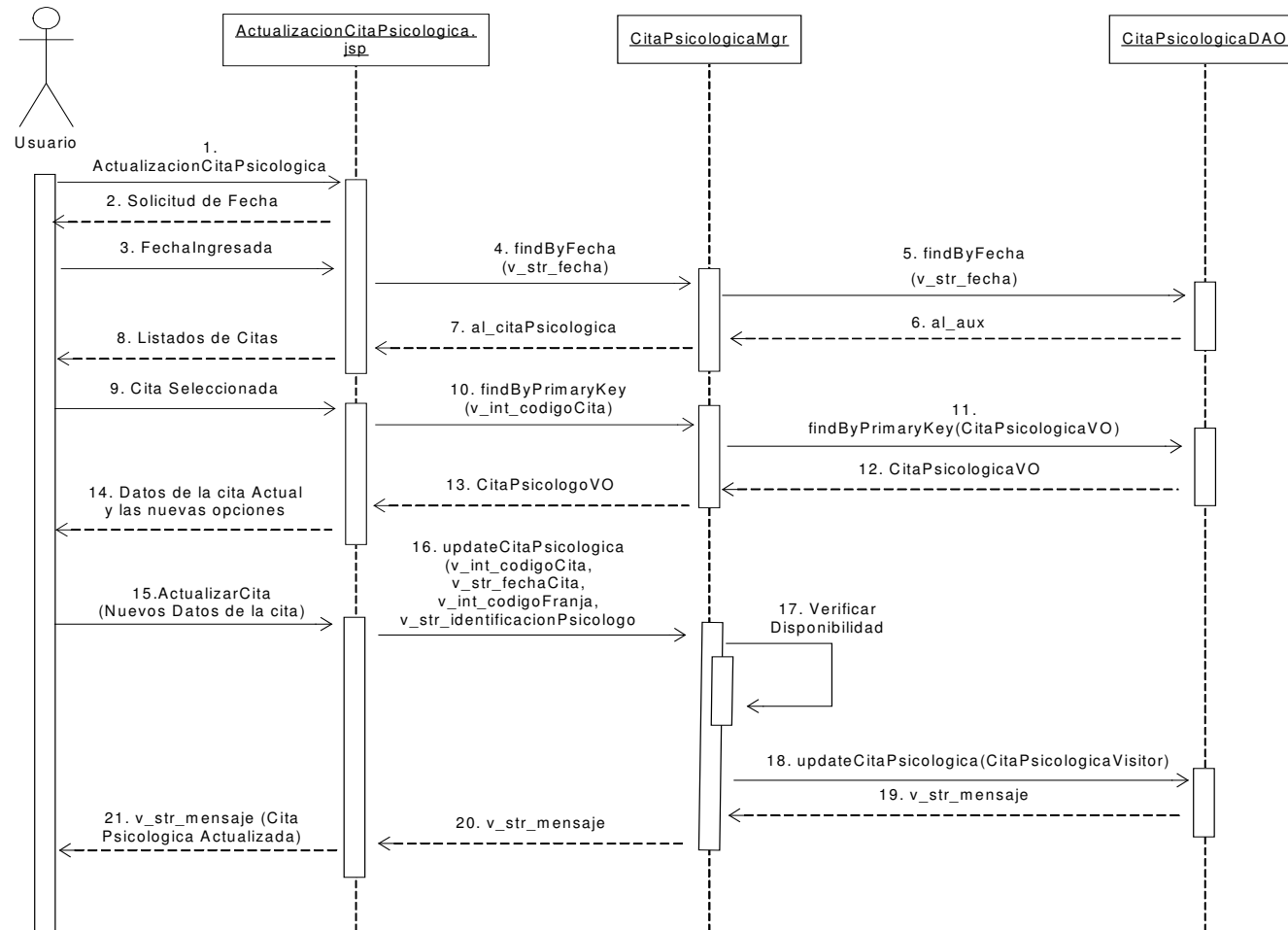


### 11.2.17.3 Consultar Cita Psicológica



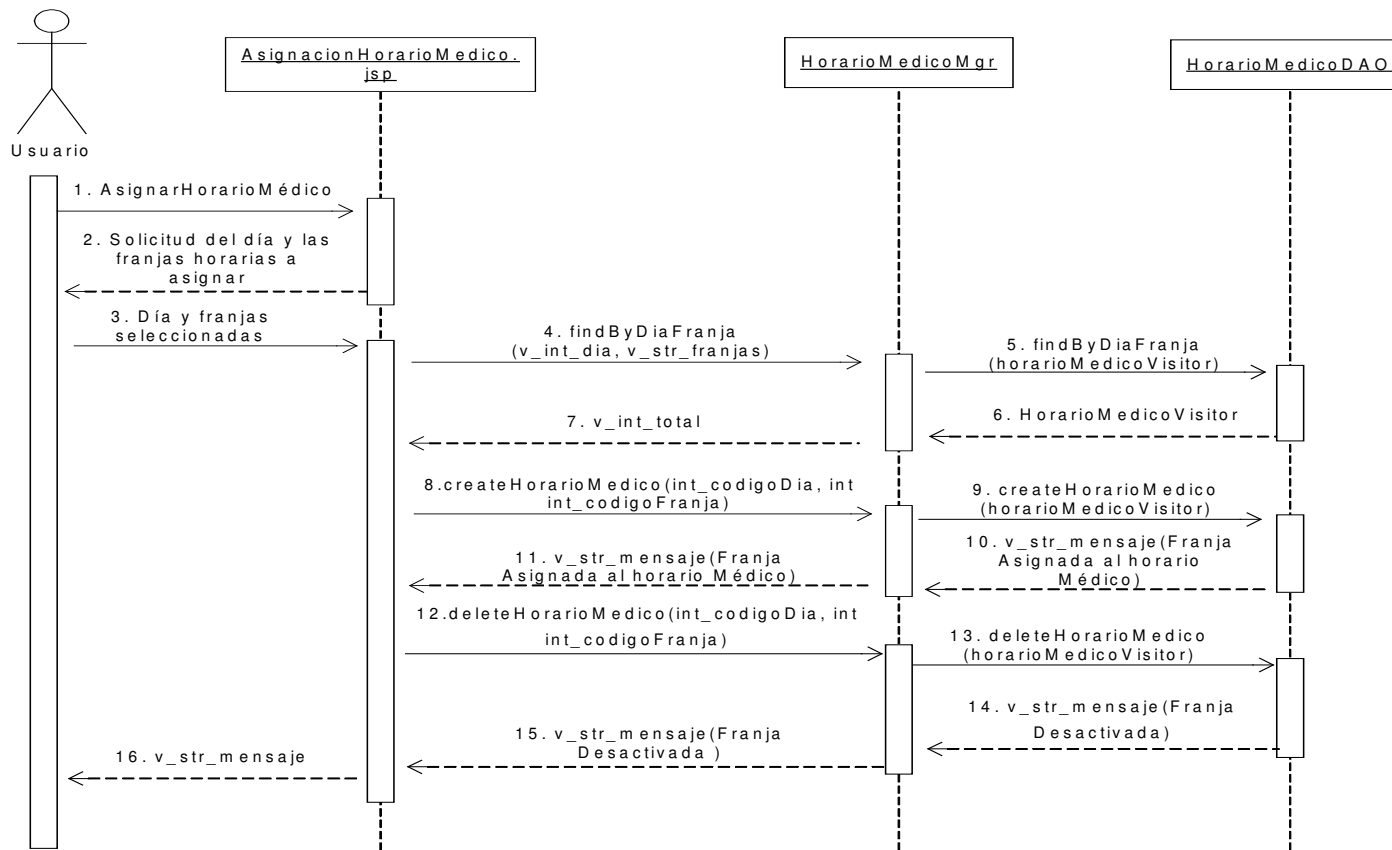


### 11.2.17.4 Actualizar Cita Psicológica

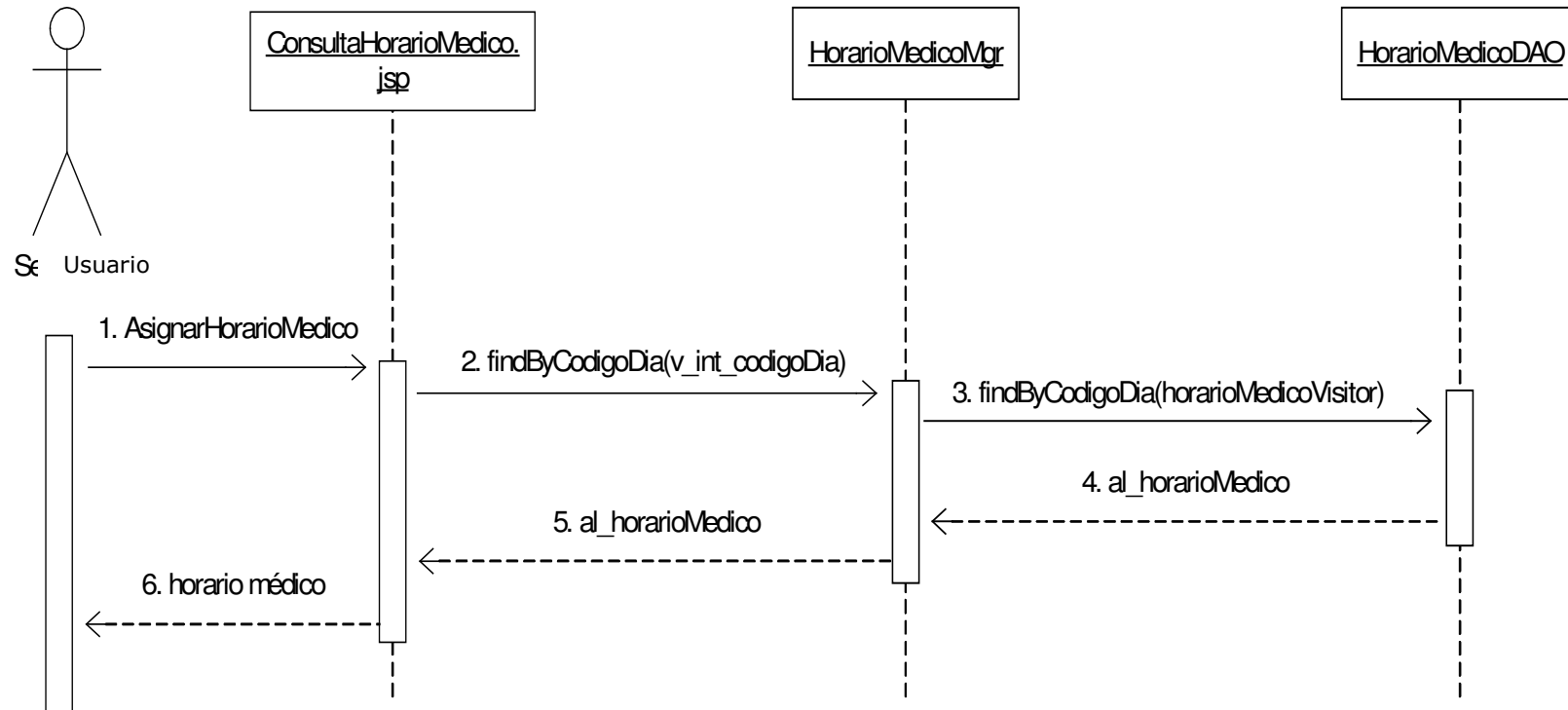


## 11.2.18 Administración de Horarios Médicos

### 11.2.18.1 Asignar Horario Médico

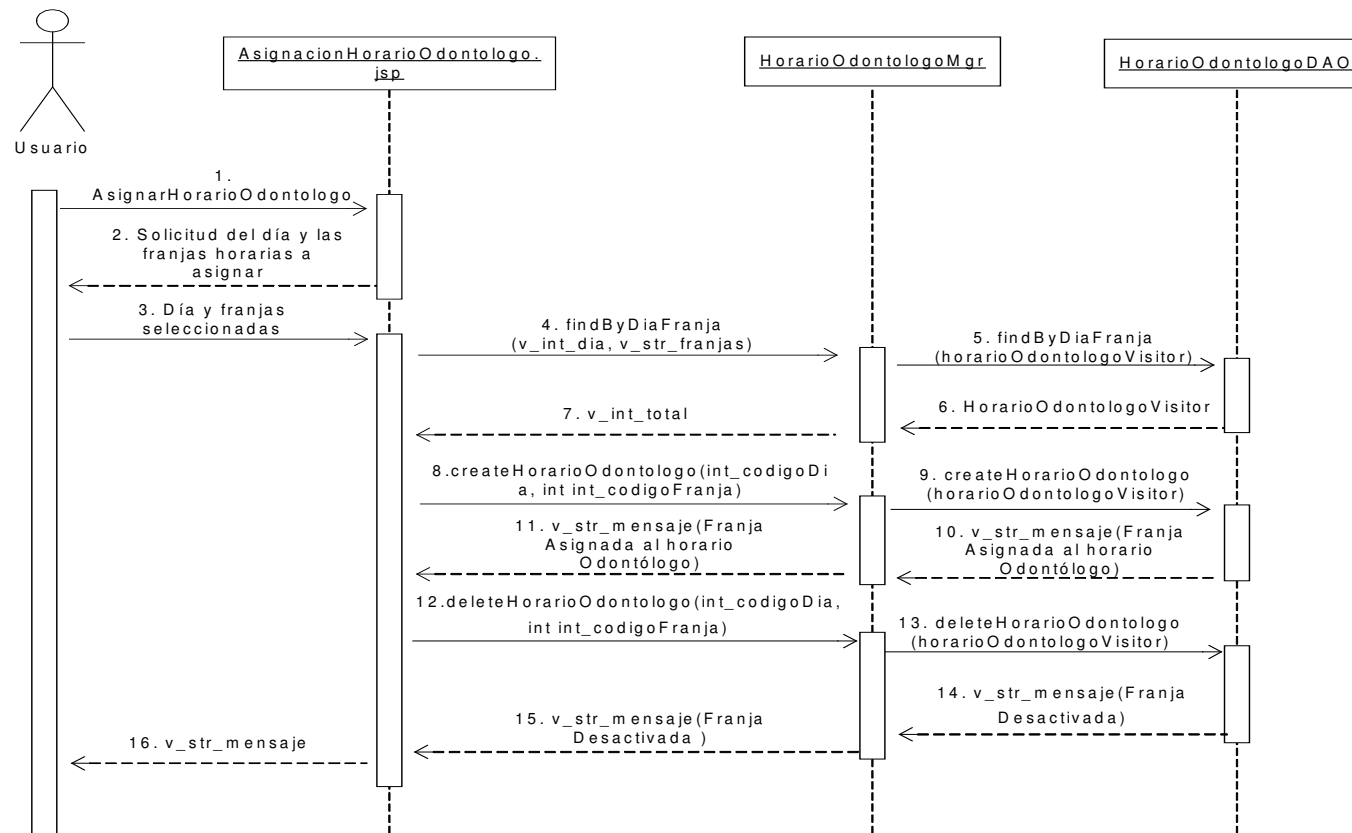


### 11.2.18.2 Consultar Horario Médico

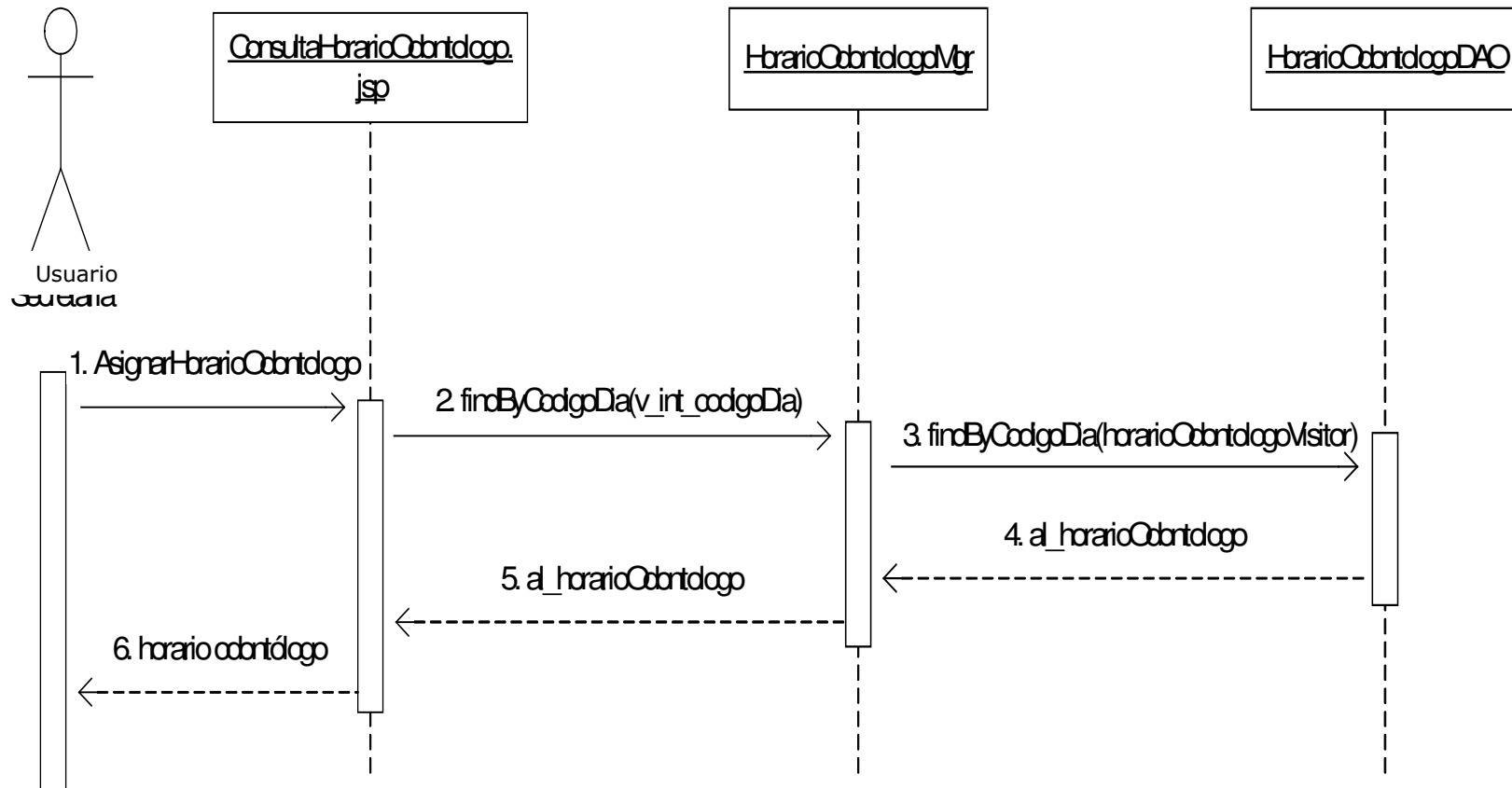


## 11.2.19 Administración de Horarios Odontológicos

### 11.2.19.1 Asignar Horario Odontológico

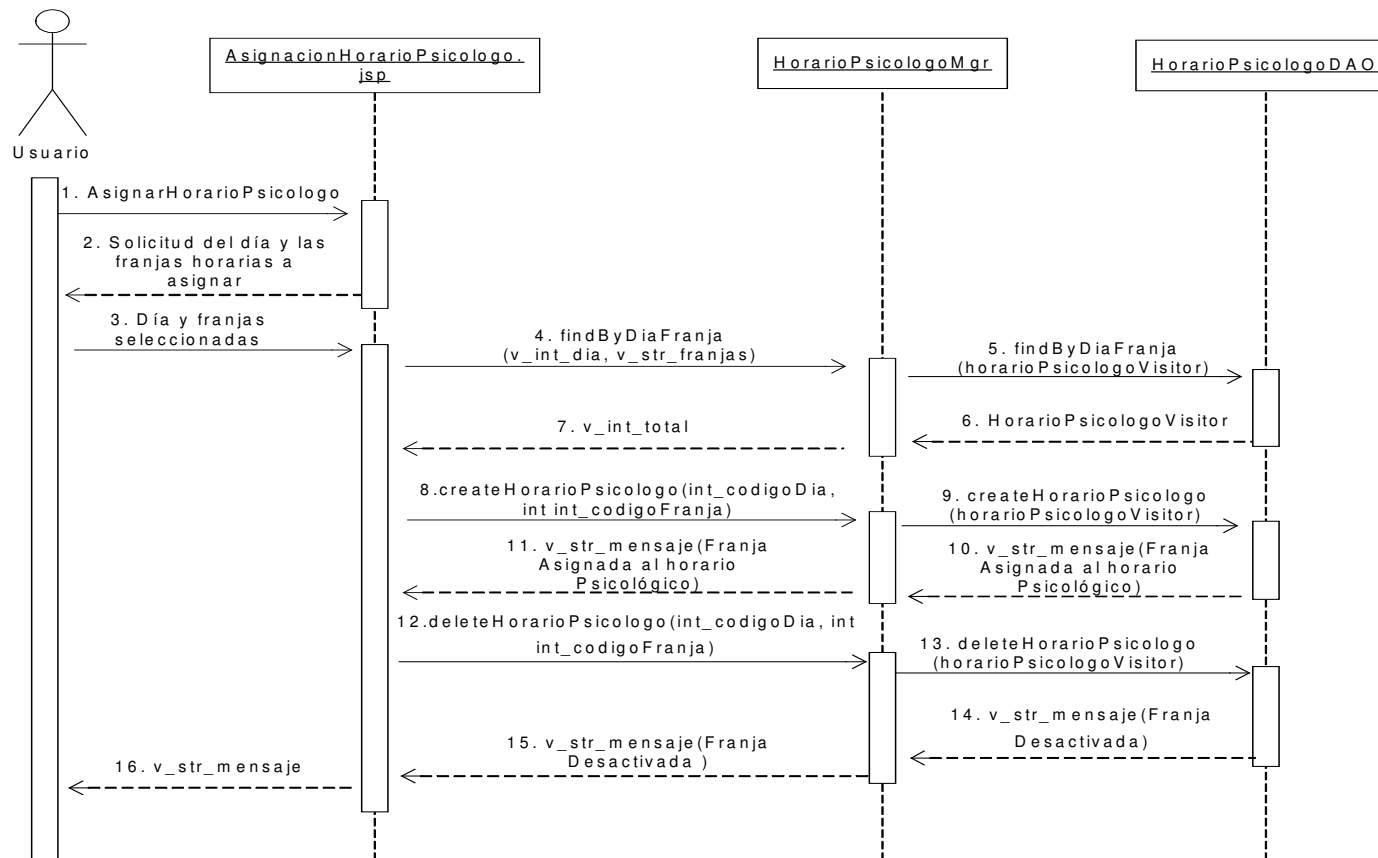


### 11.2.19.2 Consultar Horario Odontológico

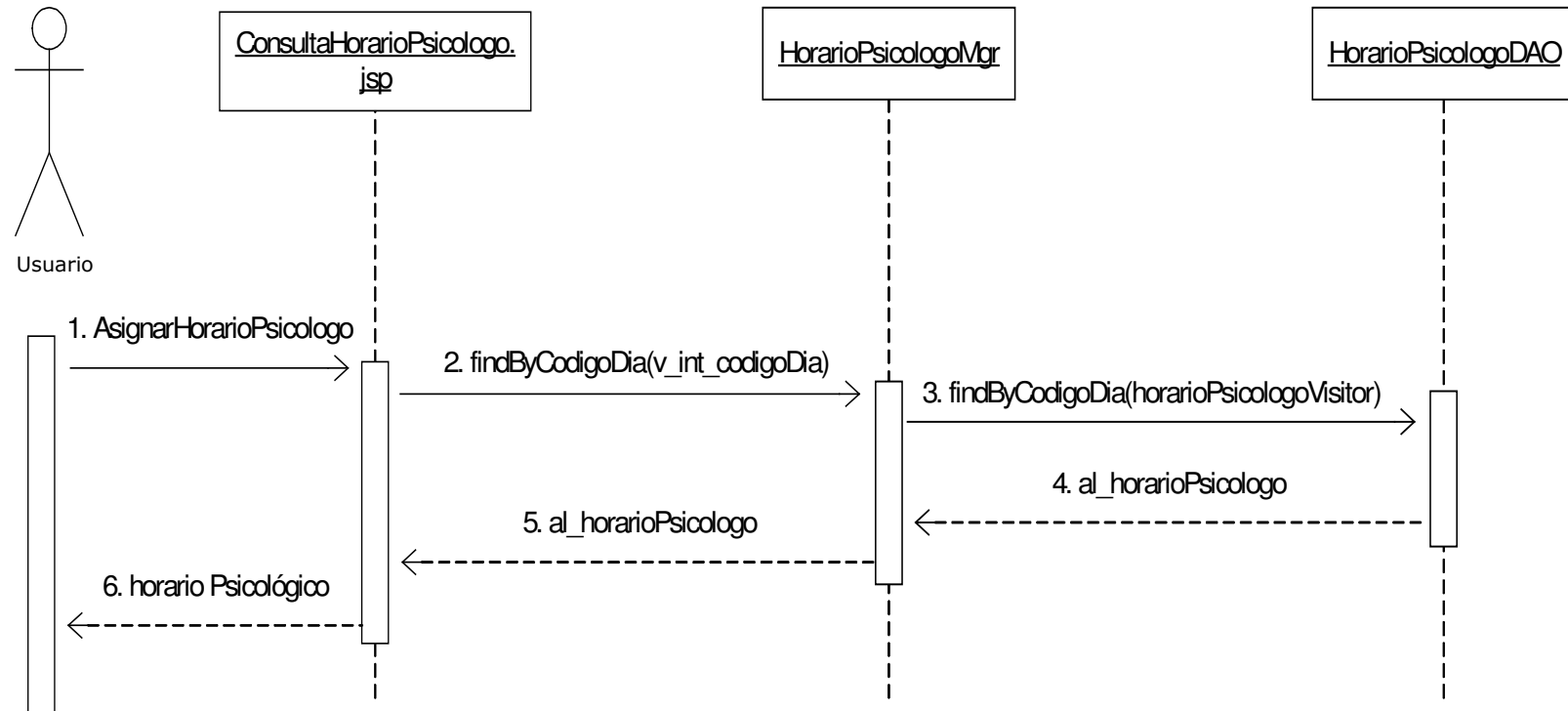


## 11.2.20 Administración de Horarios Psicológicos

### 11.2.20.1 Asignar Horario Psicológico

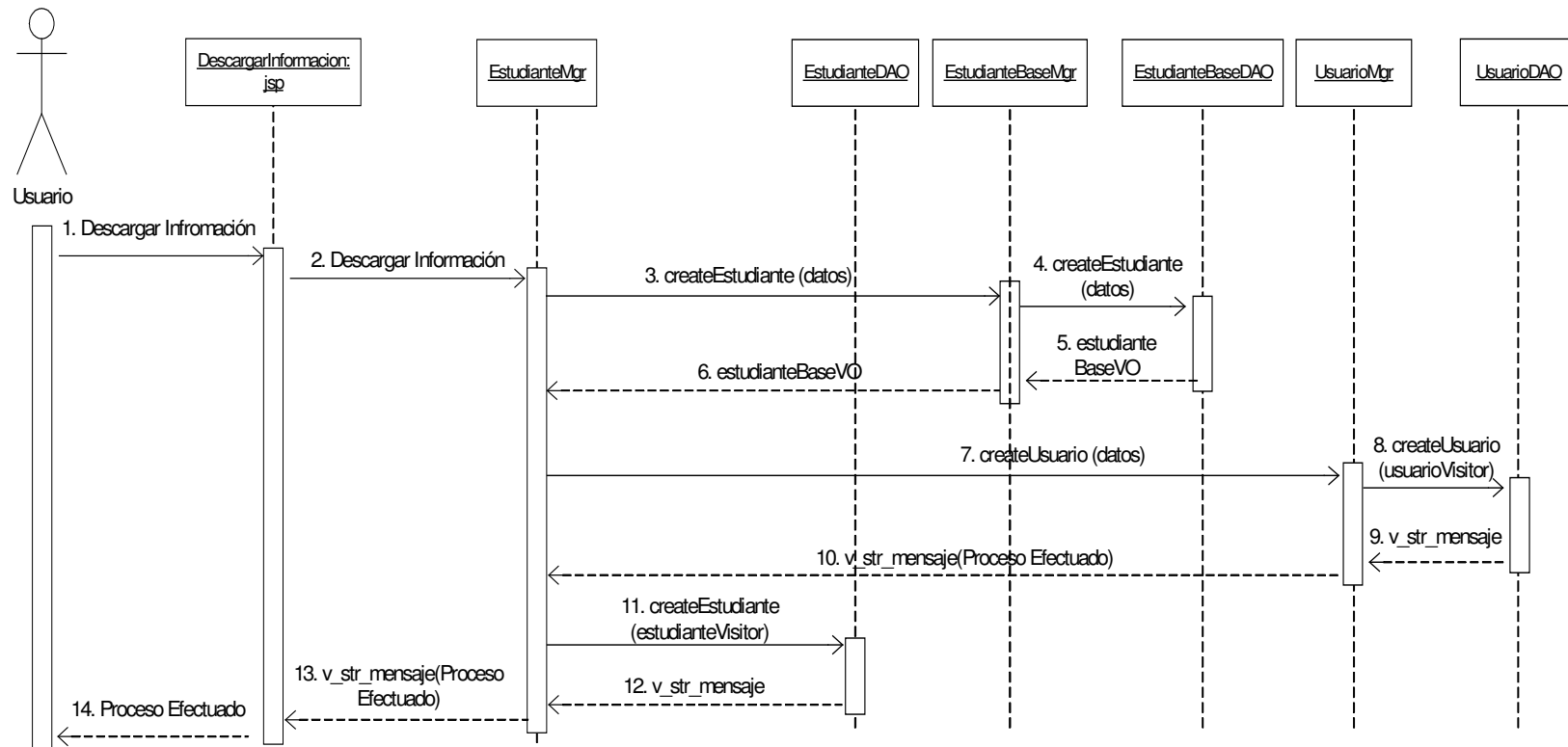


### 11.2.20.2 Consultar Horario Psicológico



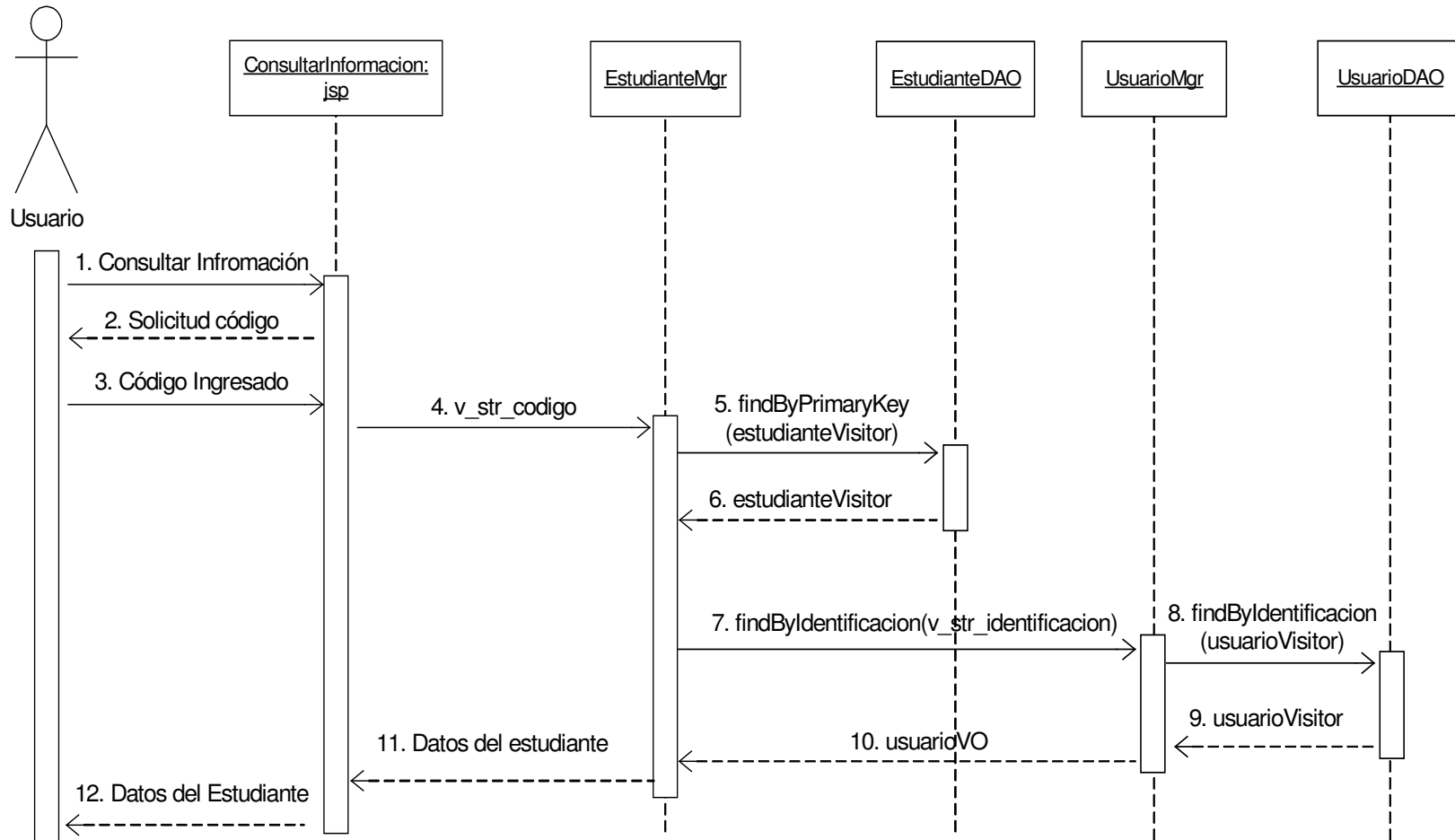
## 11.2.21 Administración de Estudiantes

### 11.2.21.1 Descargar Información Estudiantes de la Extensión



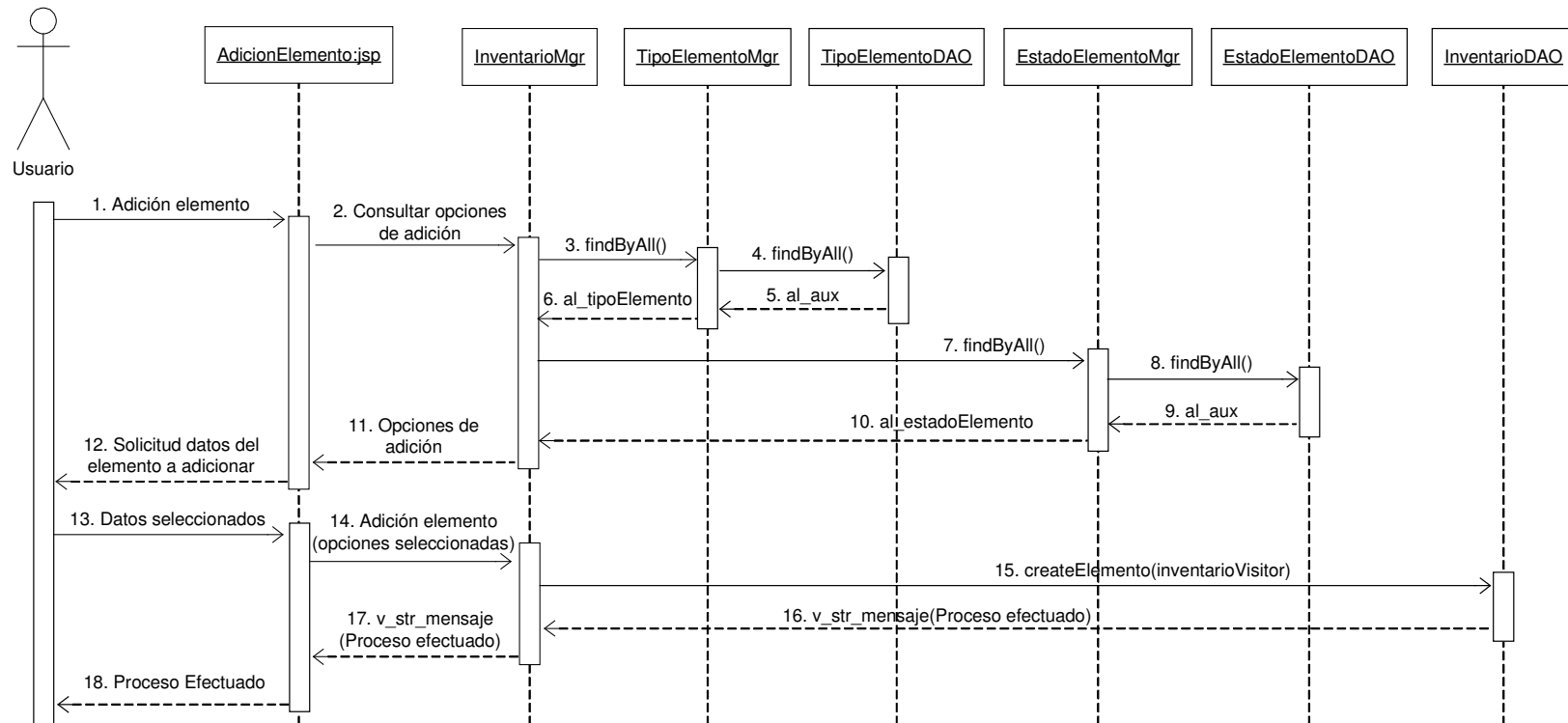


### 11.2.21.2 Consultar Datos de Estudiantes

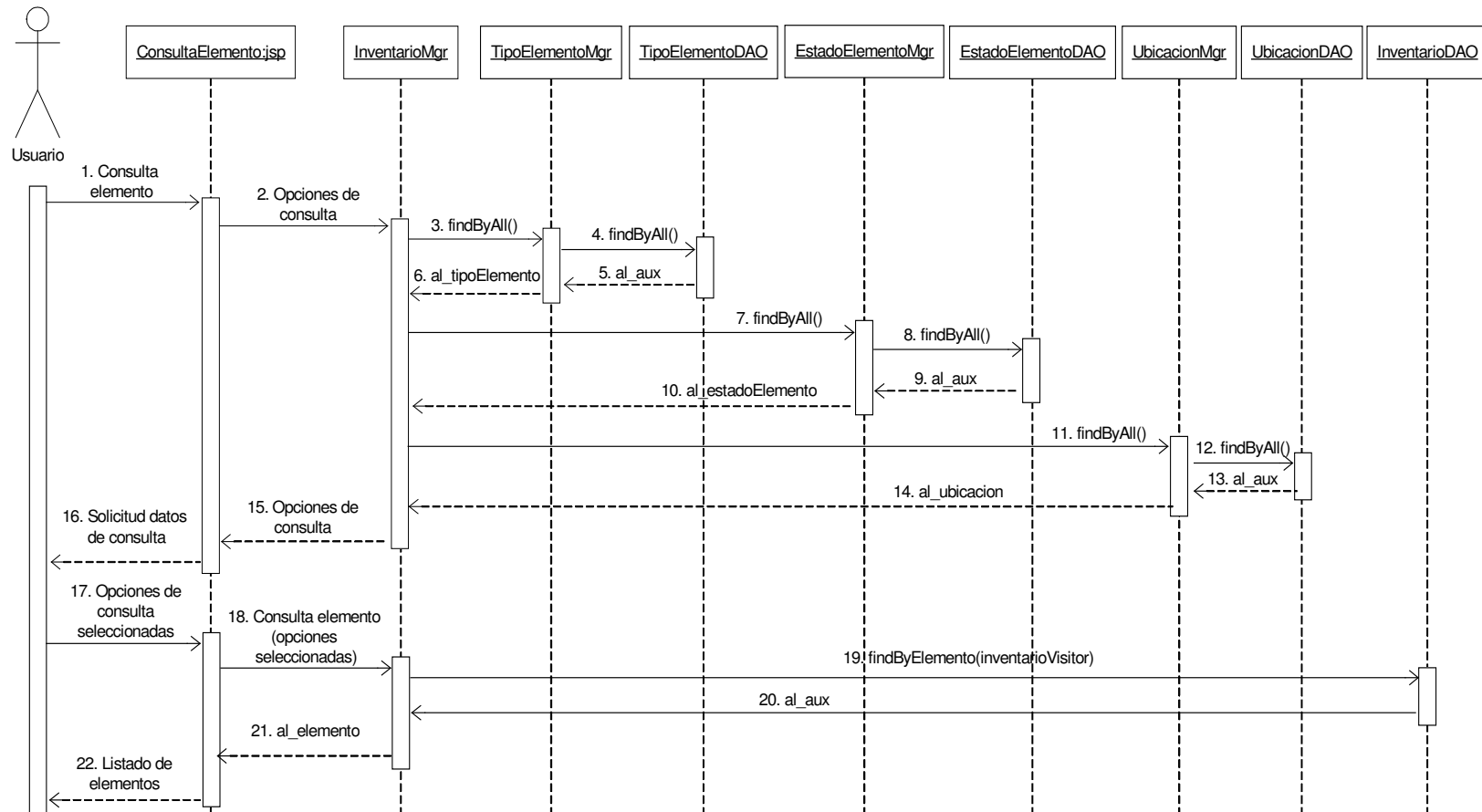


## 11.2.22 Administración de Elementos de Inventario

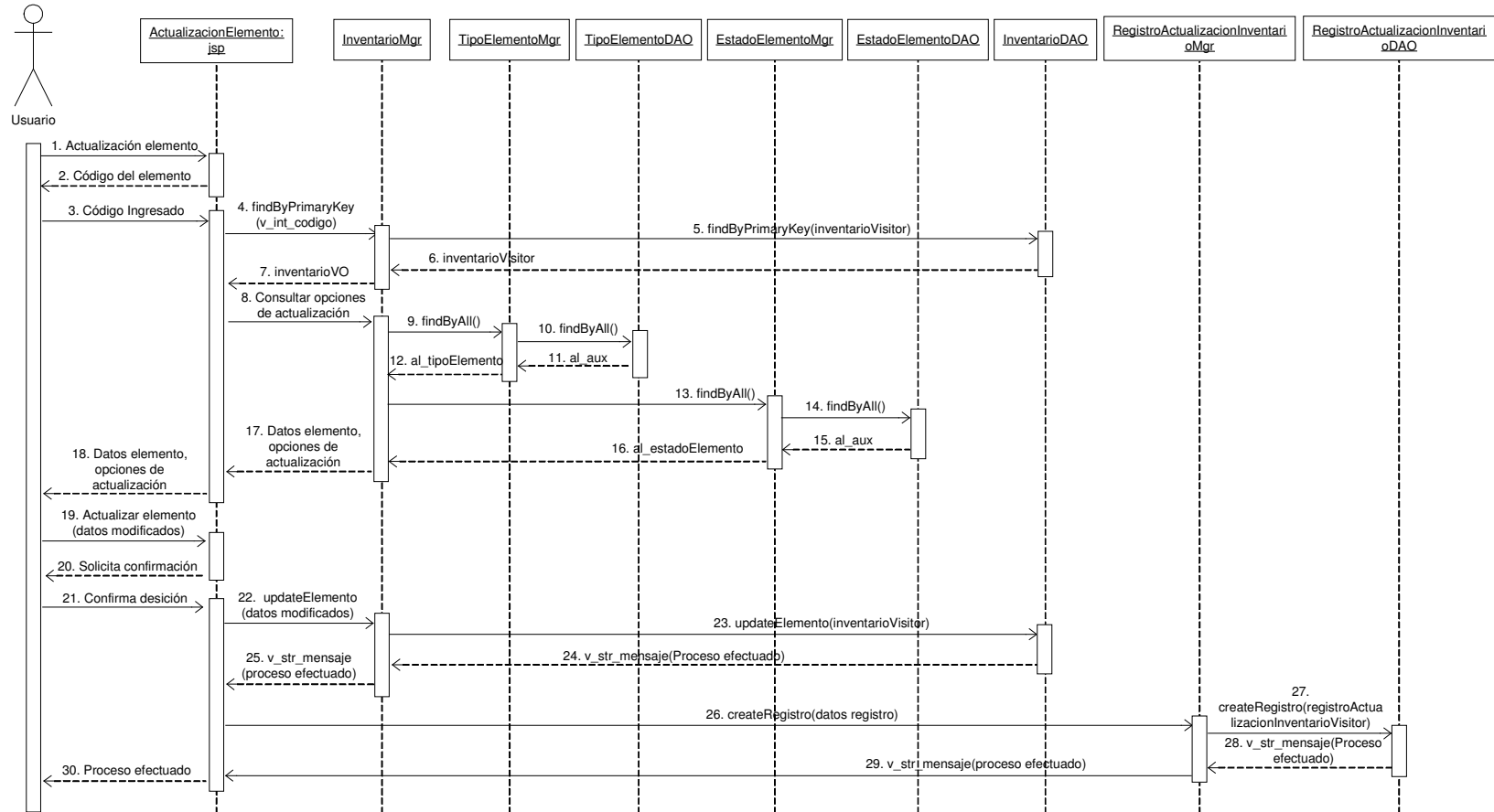
### 11.2.22.1 Adicionar Elementos de Inventario



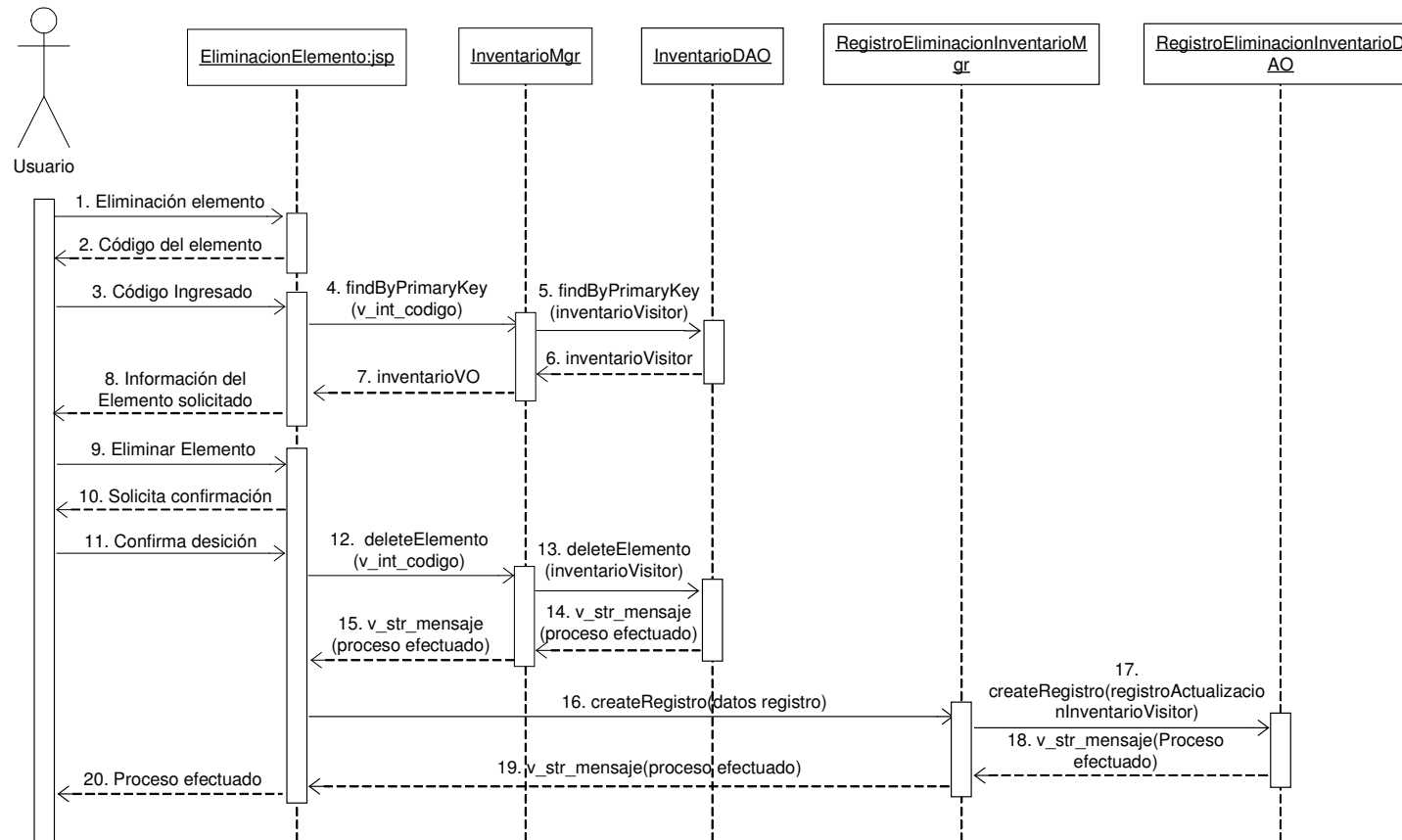
## 11.2.22.2 Consultar Elemento de Inventario



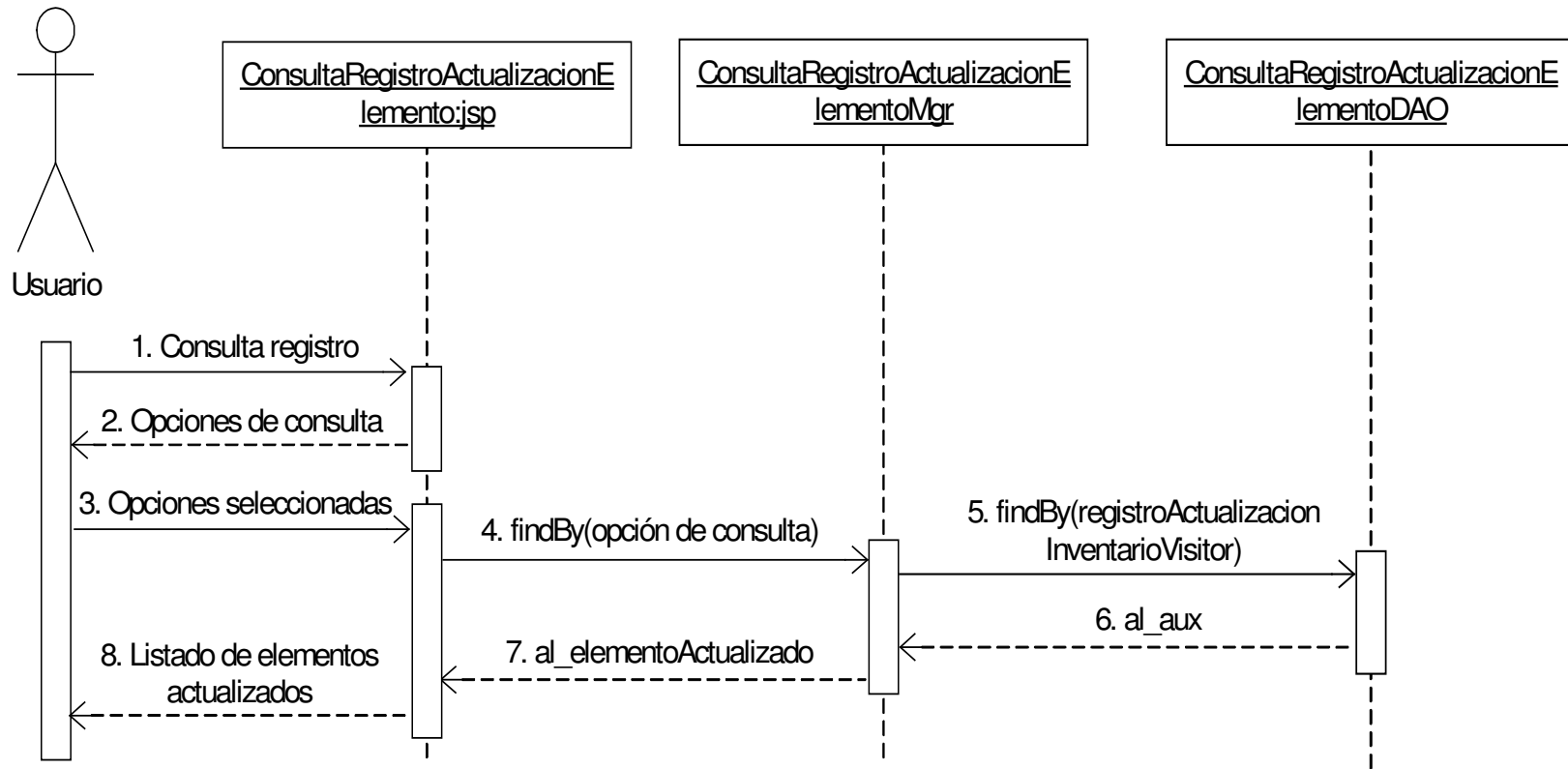
### 11.2.22.3 Actualizar Elemento de Inventario



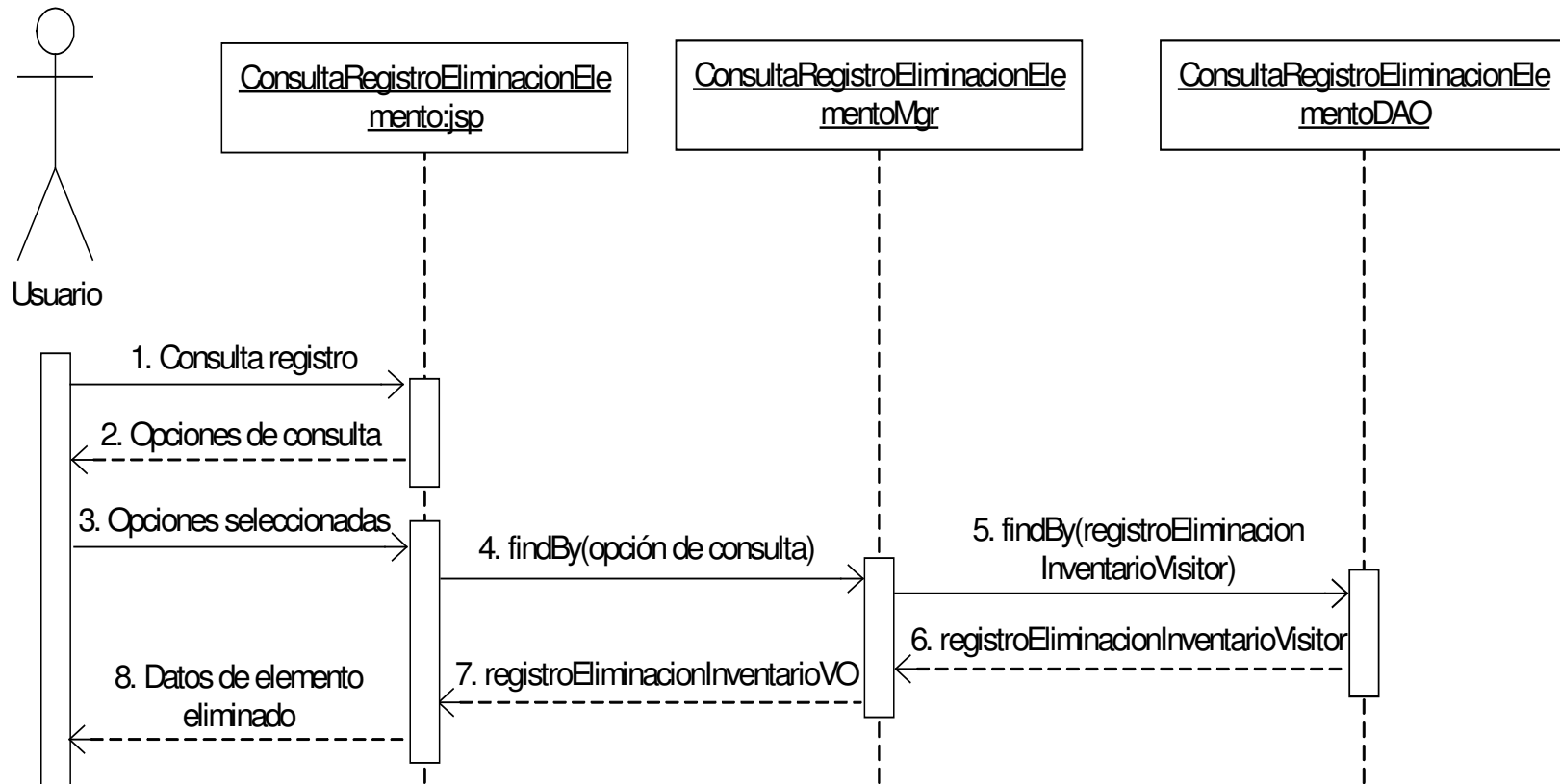
### 11.2.22.4 Eliminar Elemento de Inventario



### 11.2.22.5 Consultar Elemento Actualizados

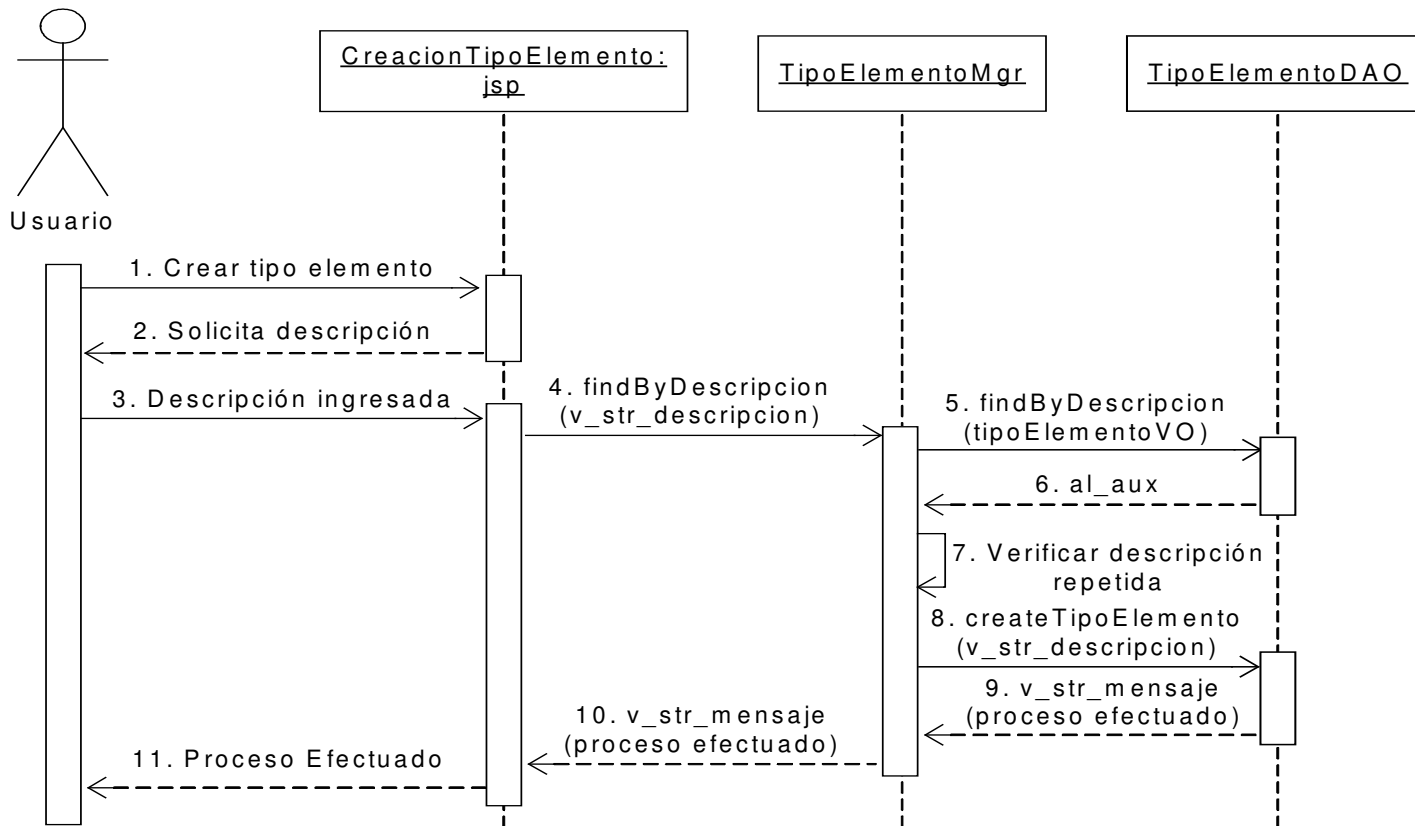


### 11.2.22.6 Consultar Elemento Eliminados



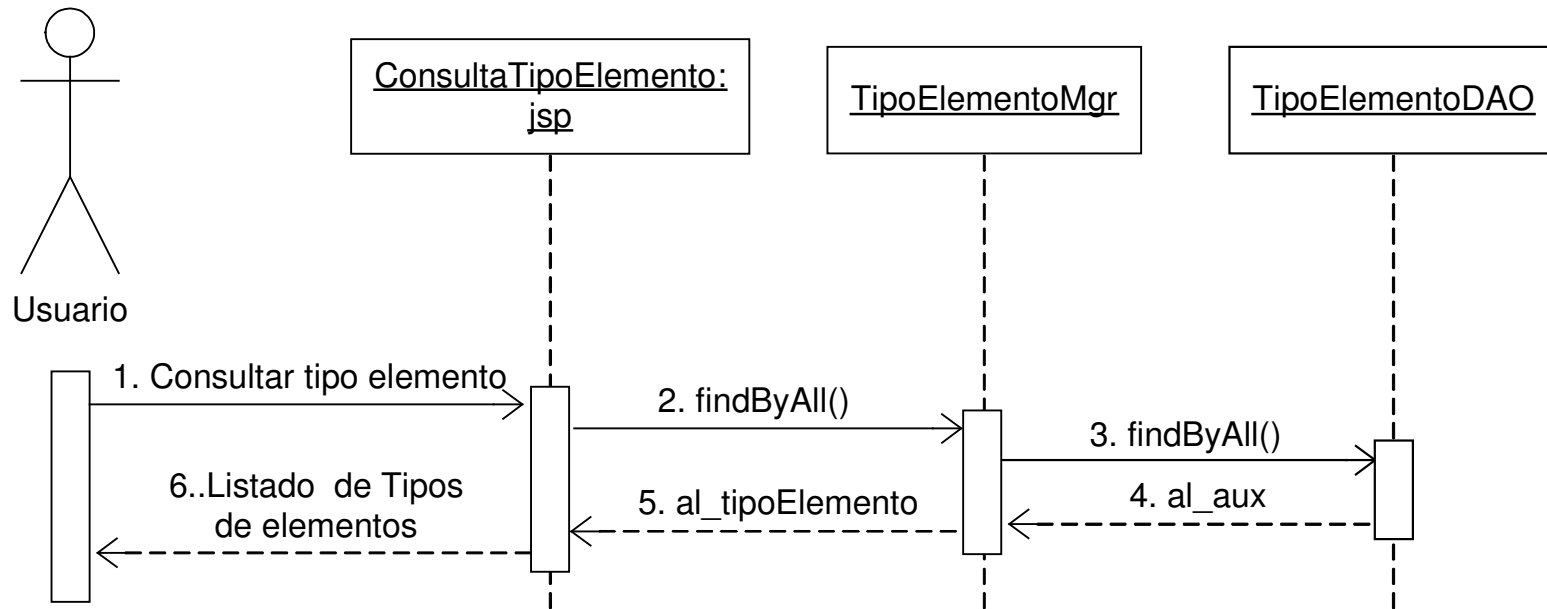
## 11.2.23 Administración Tipos de Elementos

### 11.2.23.1 Crear Tipo Elemento

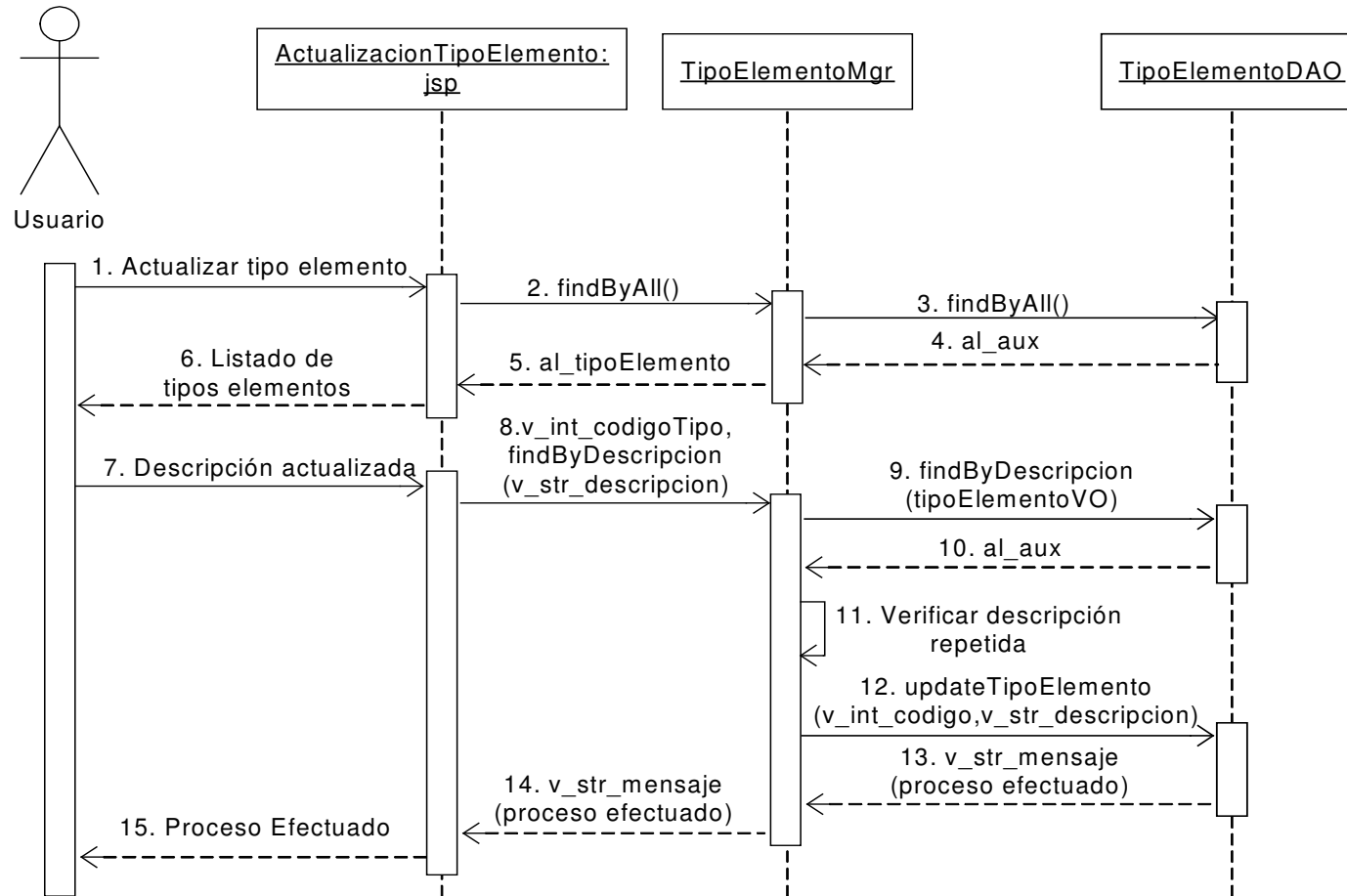




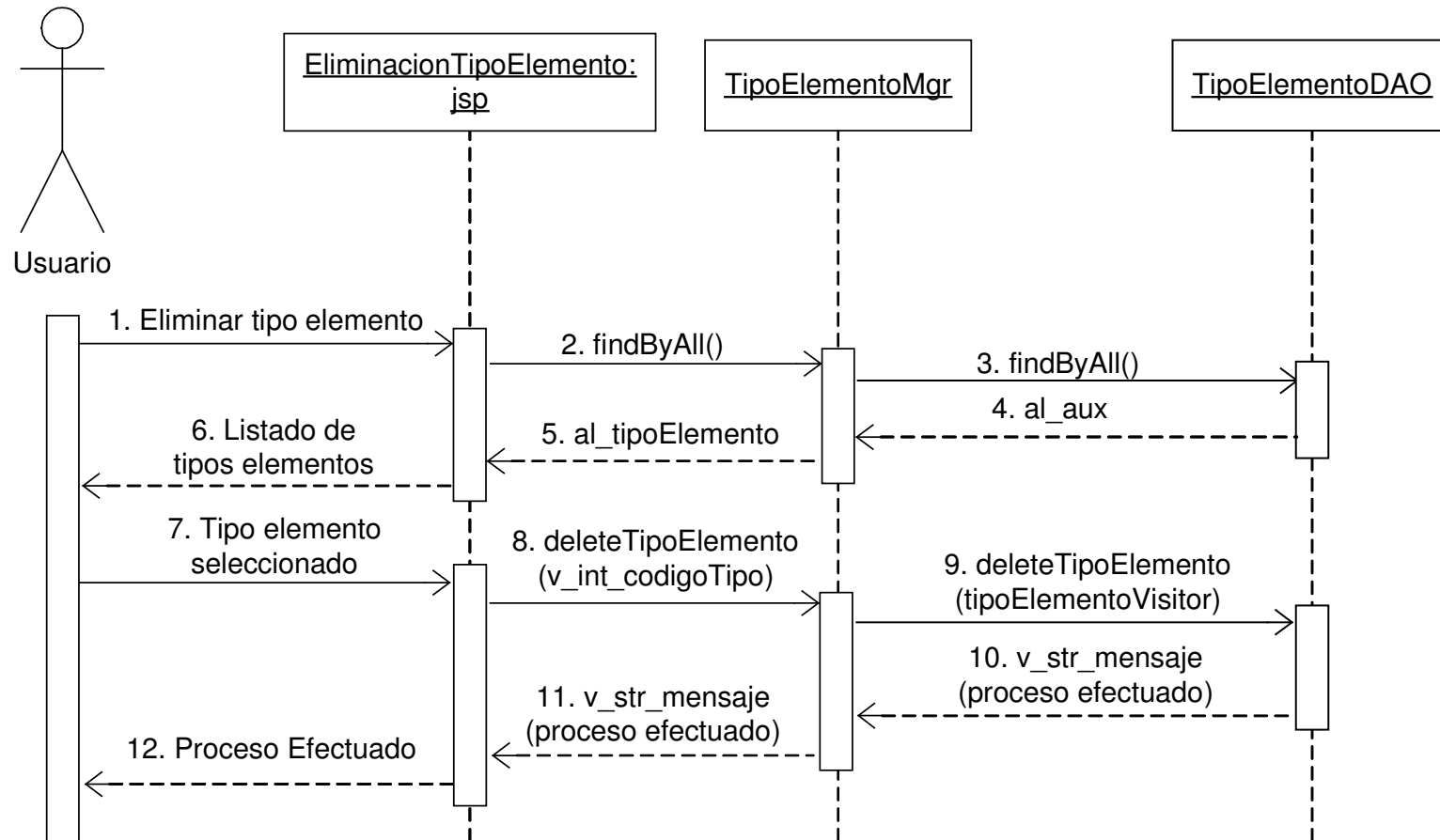
### 11.2.23.2 Consultar Tipo Elemento



### 11.2.23.3 Actualizar Tipo Elemento

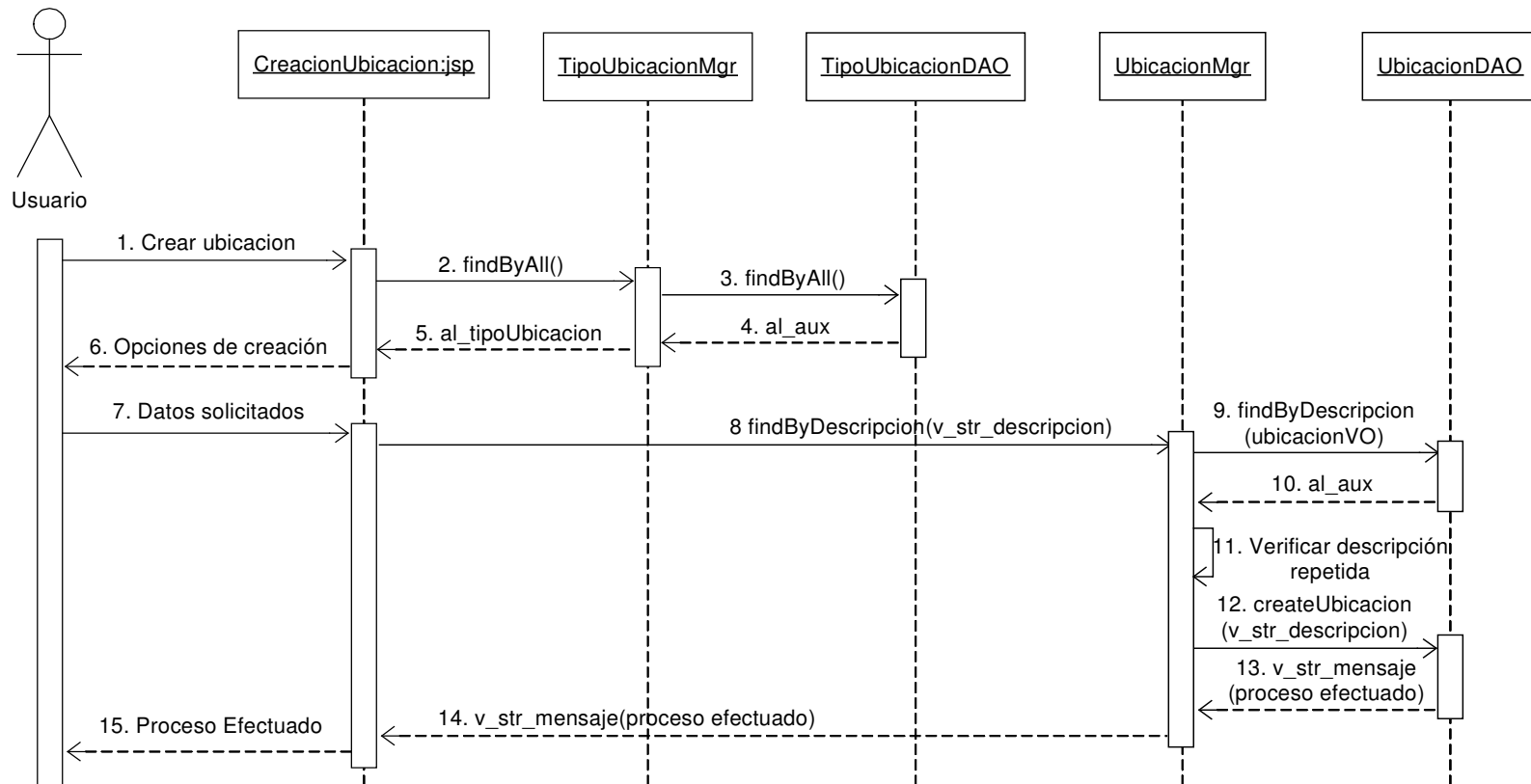


### 11.2.23.4 Eliminar Tipo Elemento

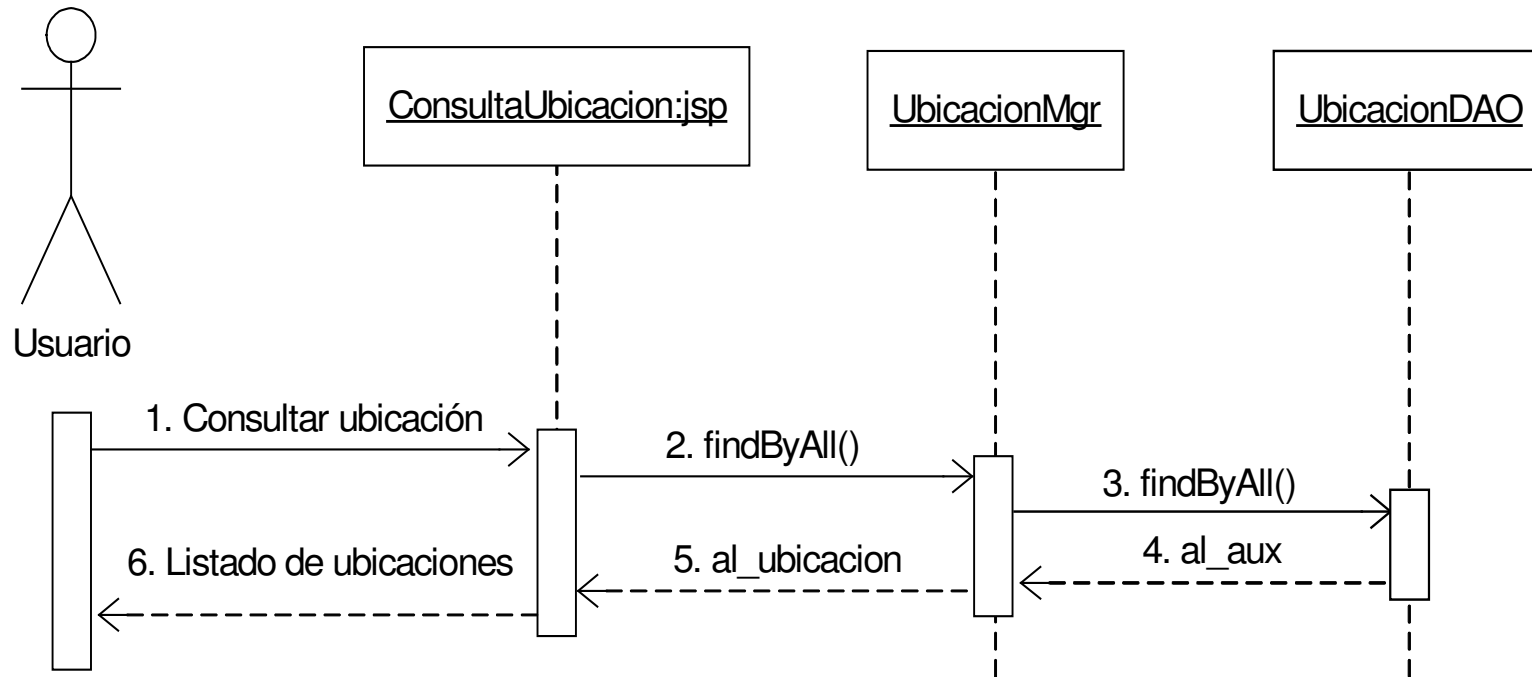


## 11.2.24 Administración Ubicaciones

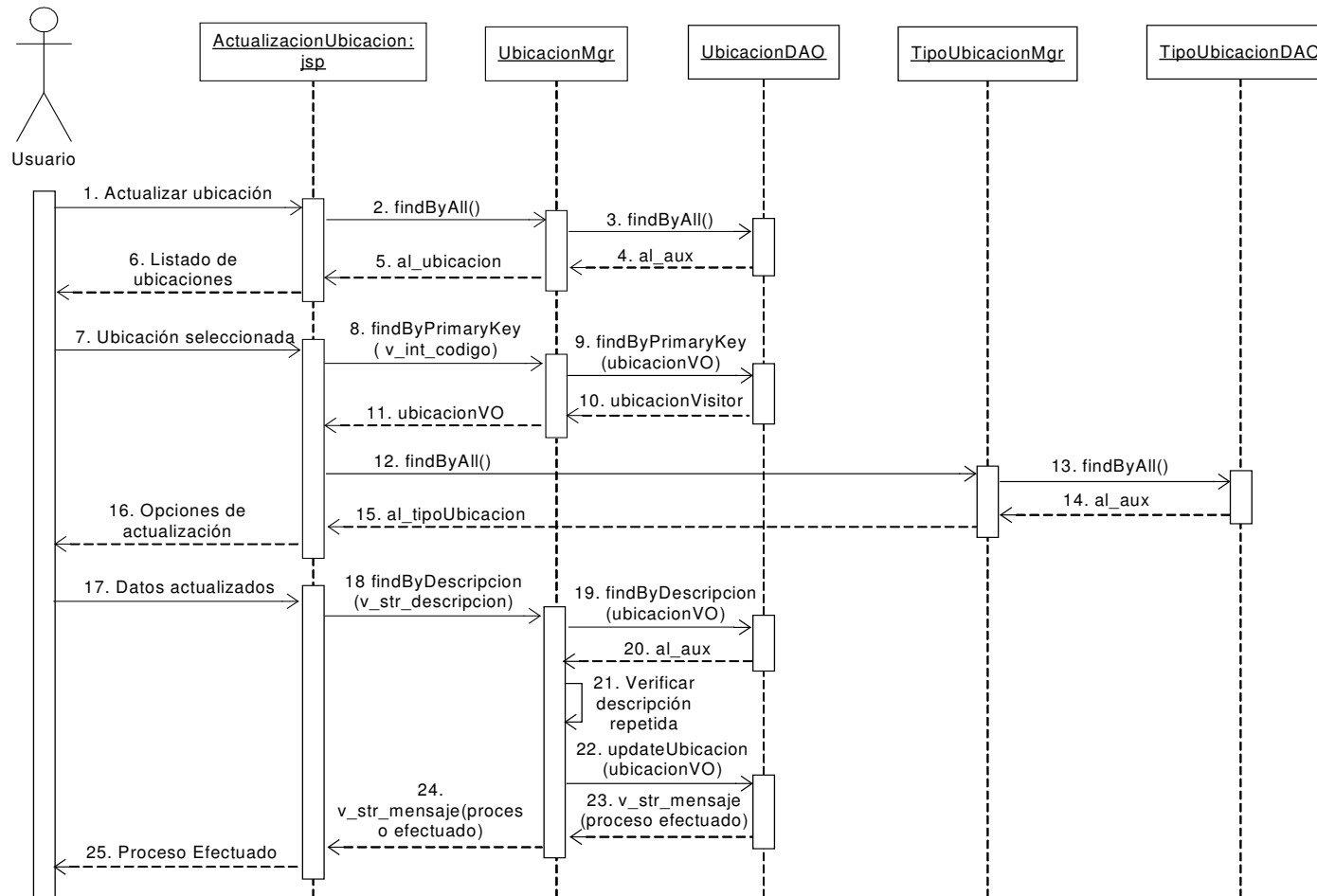
### 11.2.24.1 Crear Ubicación



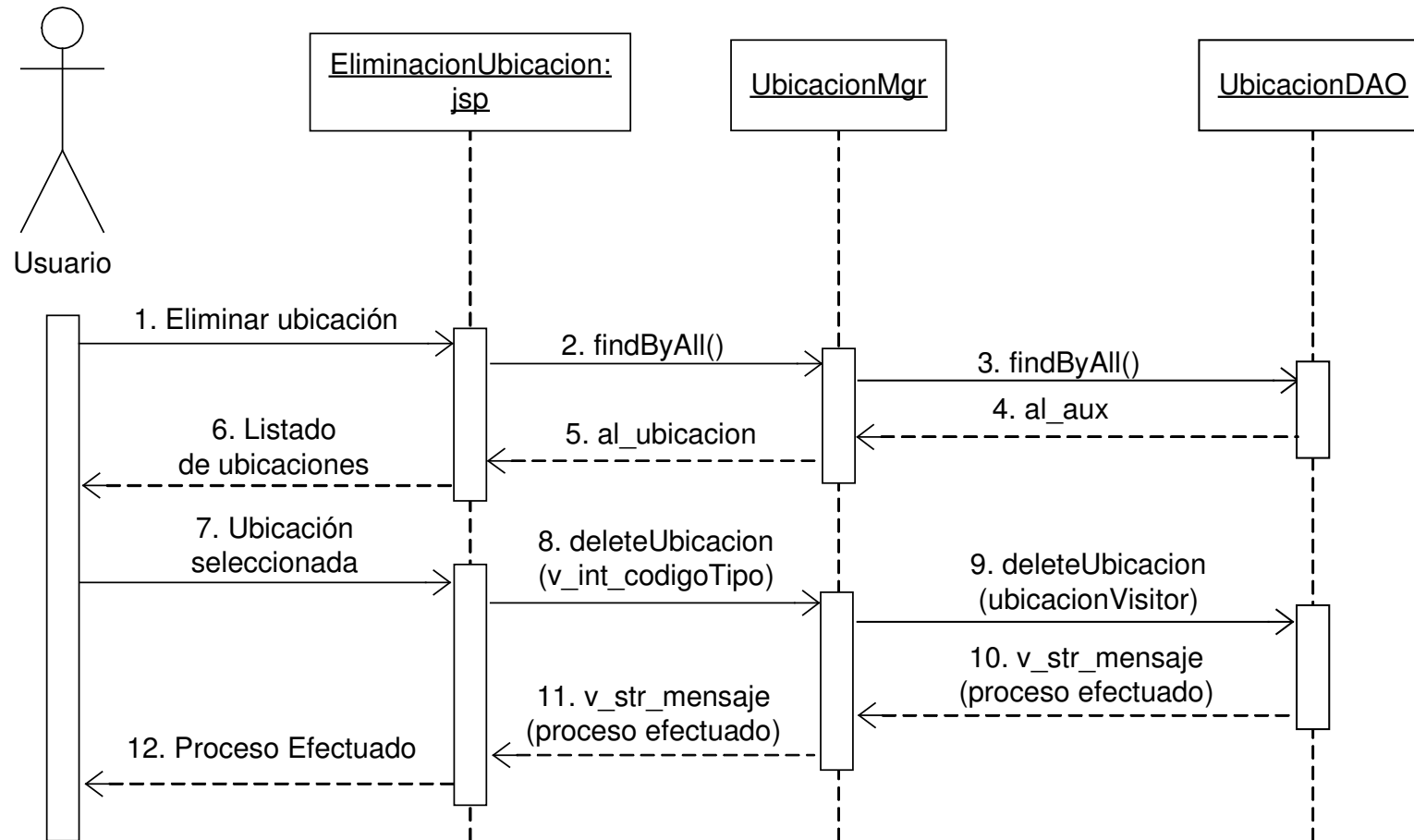
### 11.2.24.2 Consultar Ubicación



### 11.2.24.3 Actualizar Ubicación

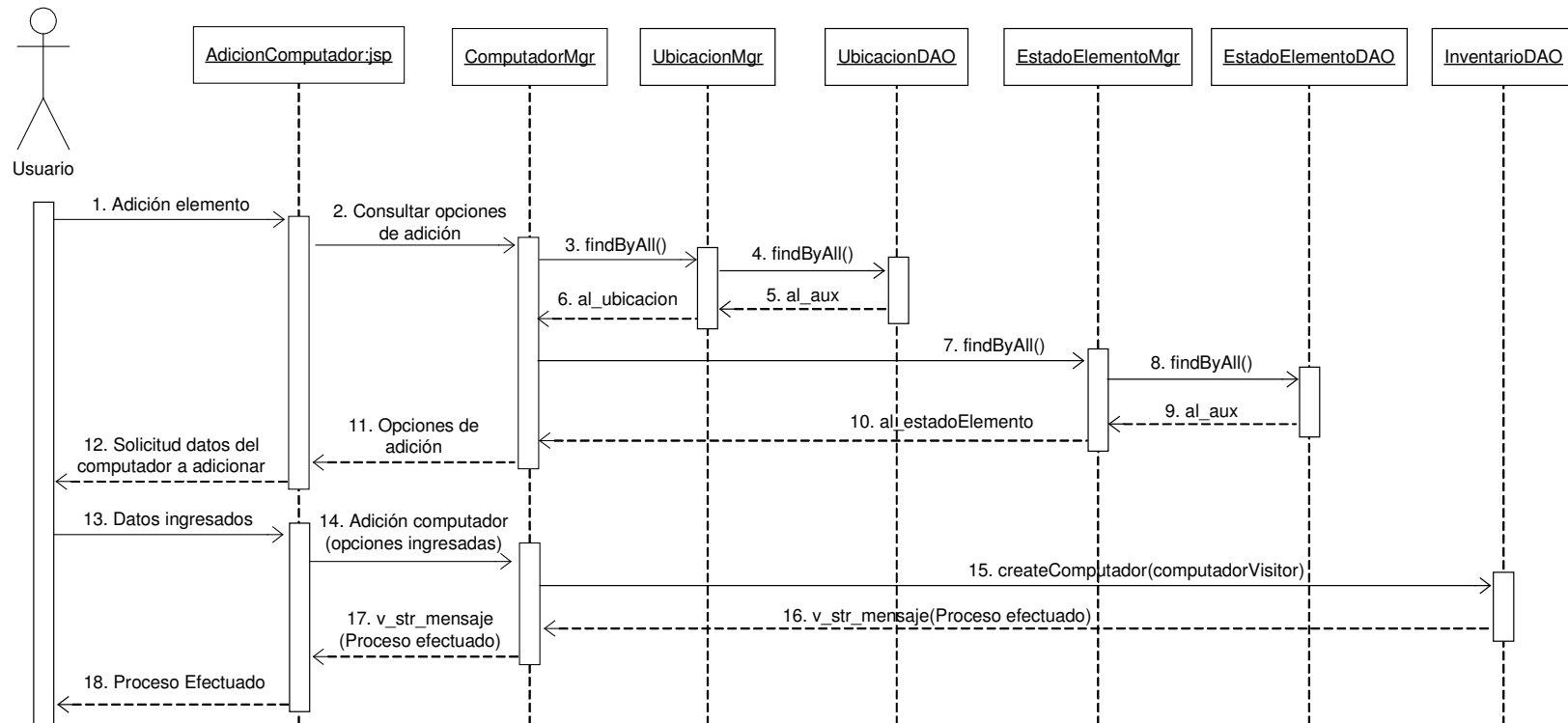


#### 11.2.24.4 Eliminar Ubicación



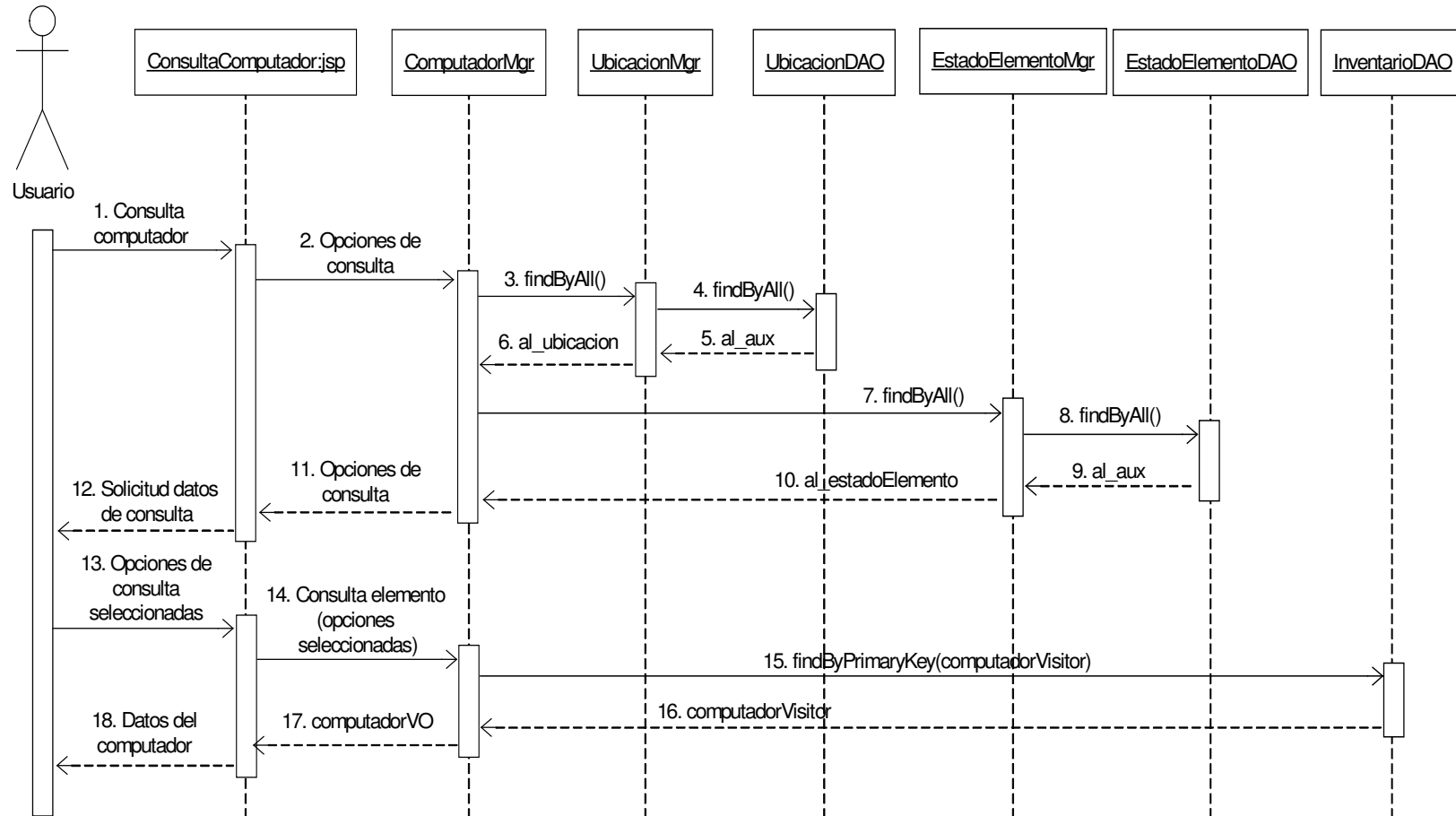
## 11.2.25 Administración de Computadores

### 11.2.25.1 Adicionar Computador

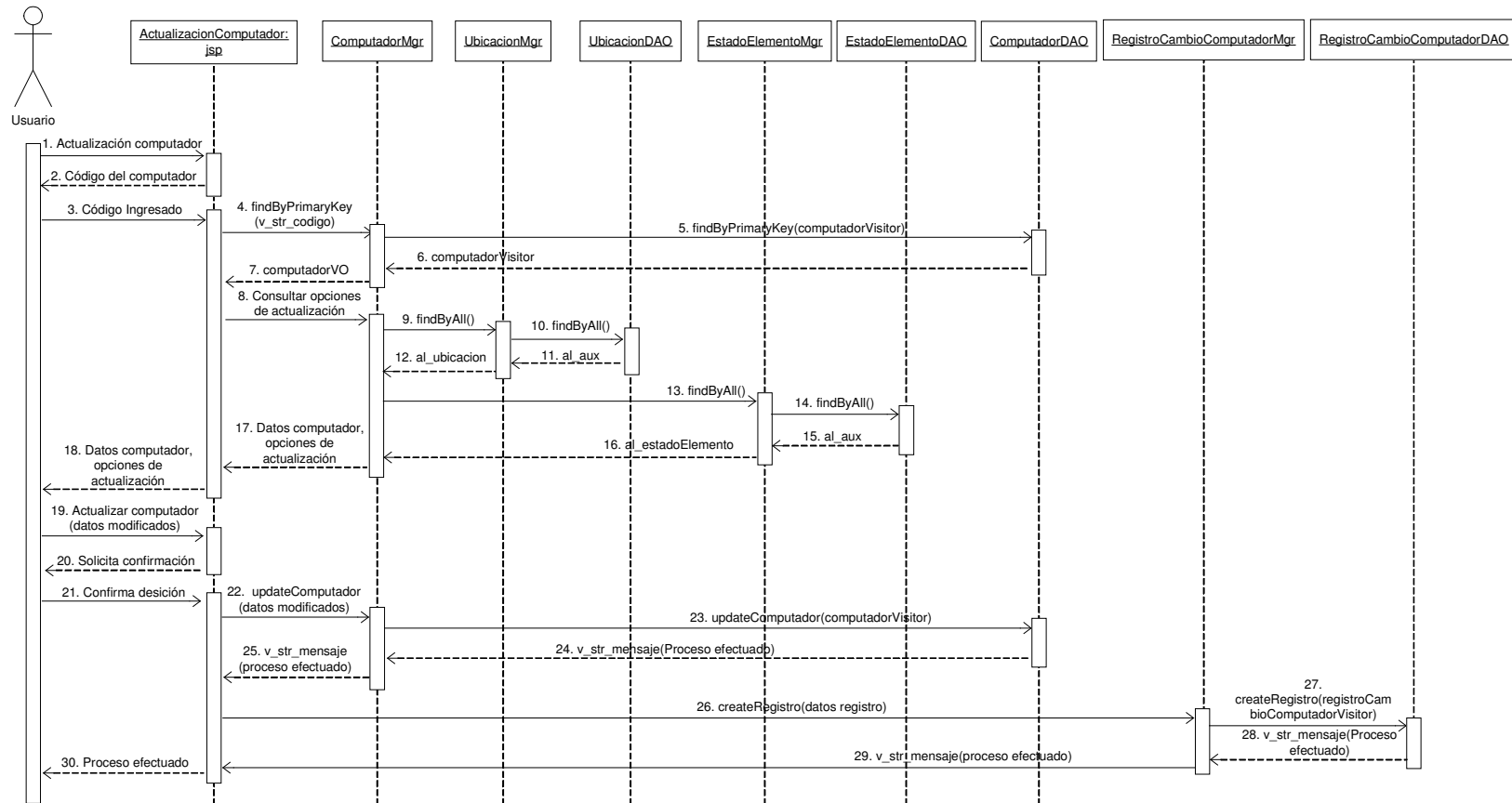




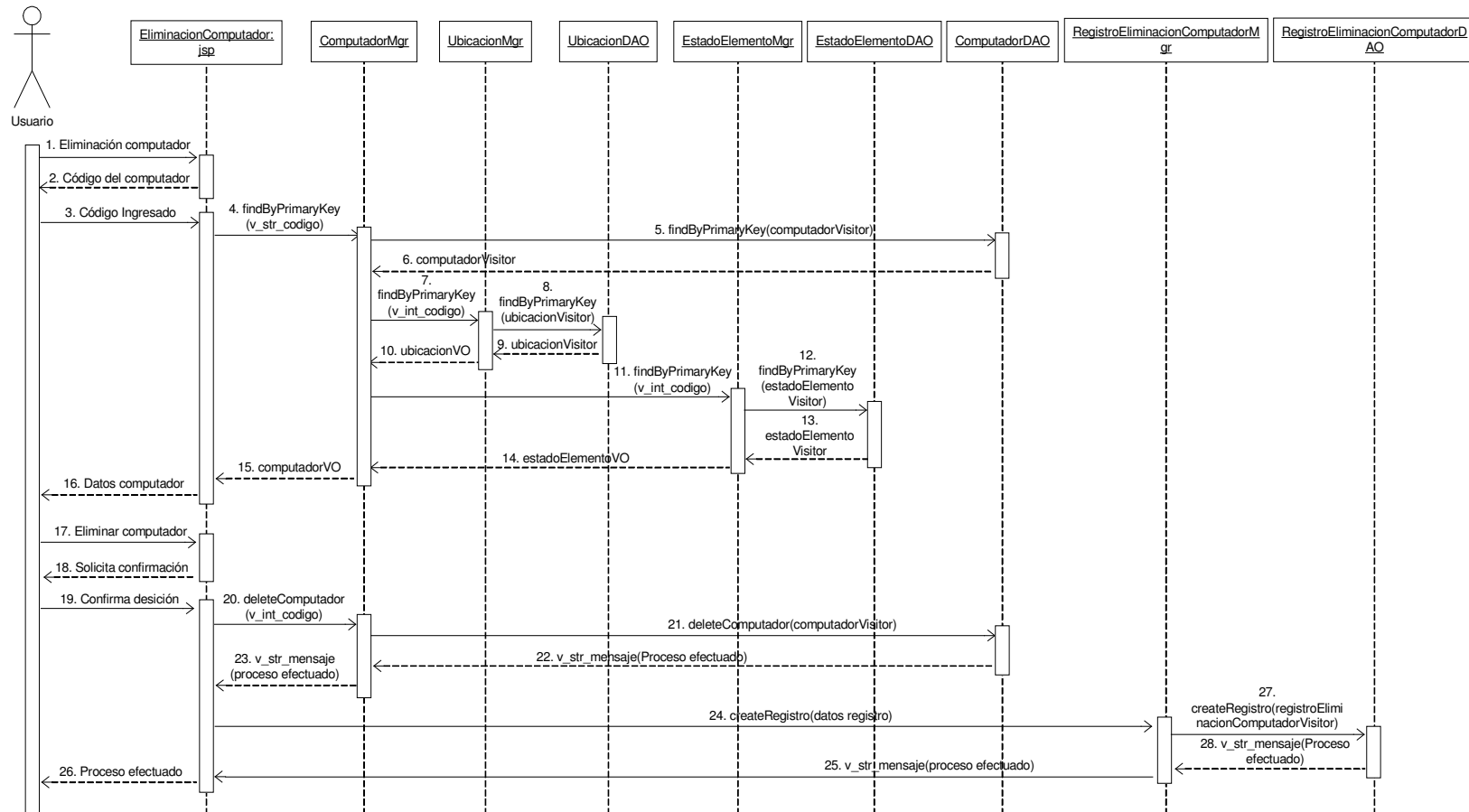
### 11.2.25.2 Consultar Computador



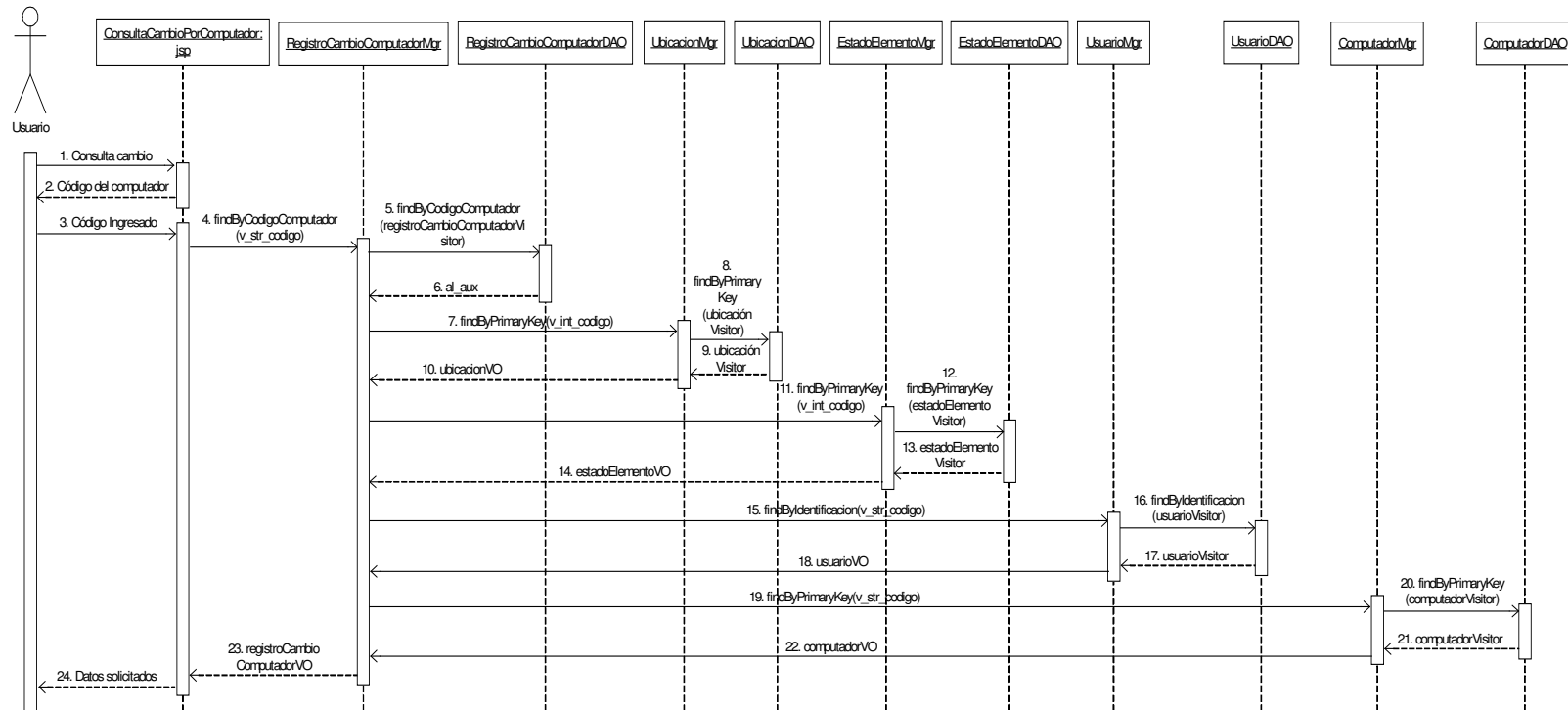
### 11.2.25.3 Actualizar Computador



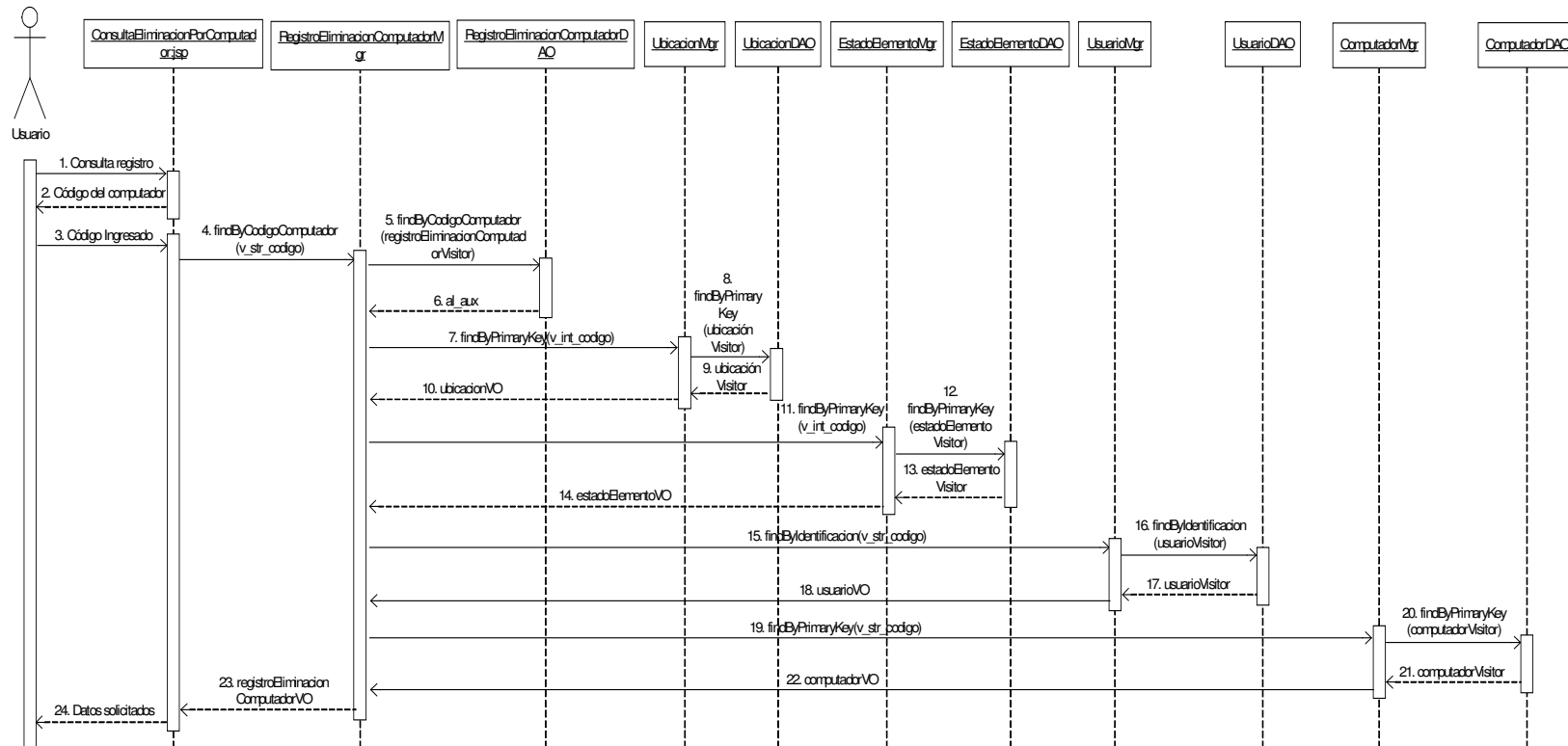
### 11.2.25.4 Eliminar Computador



### 11.2.25.5 Consultar Cambios Computador

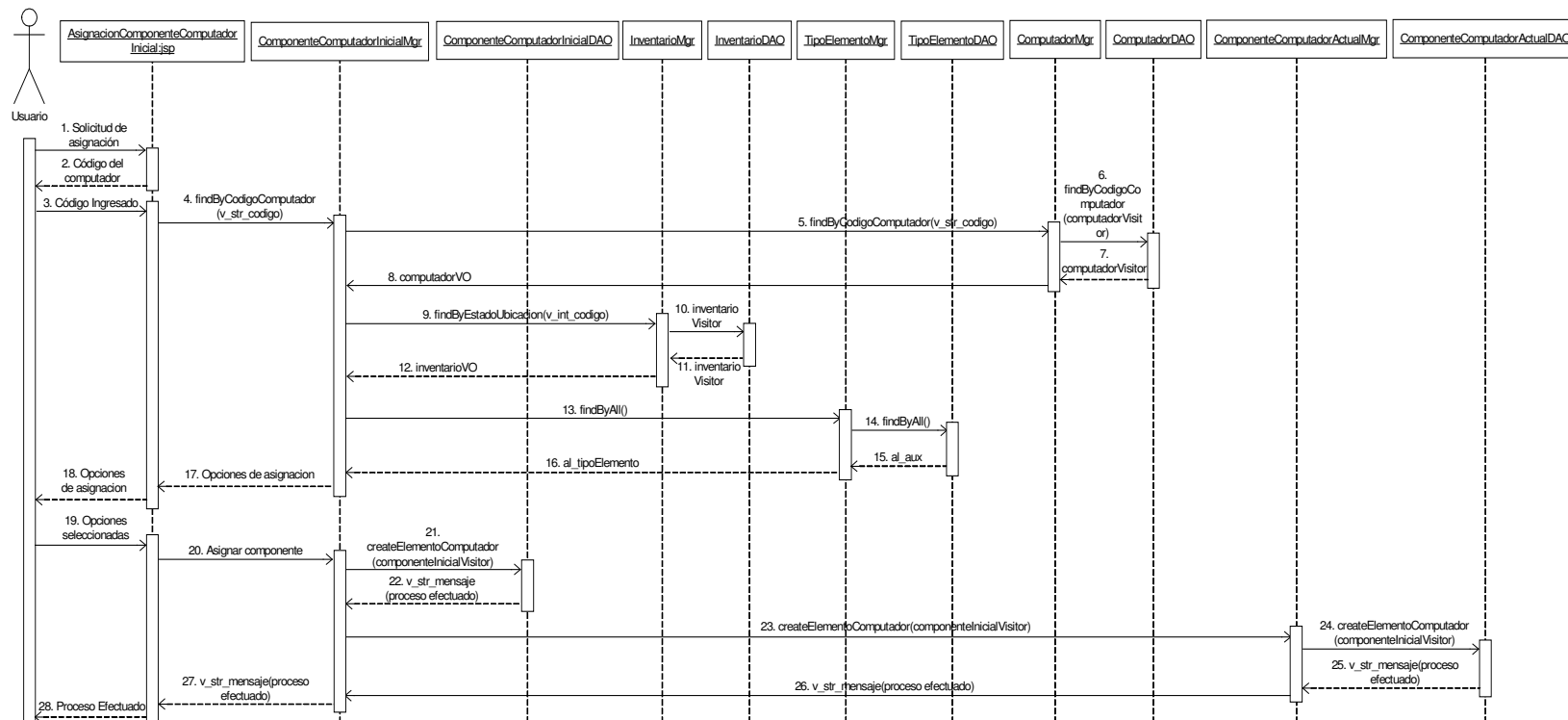


### 11.2.25.6 Consultar Eliminación Computador

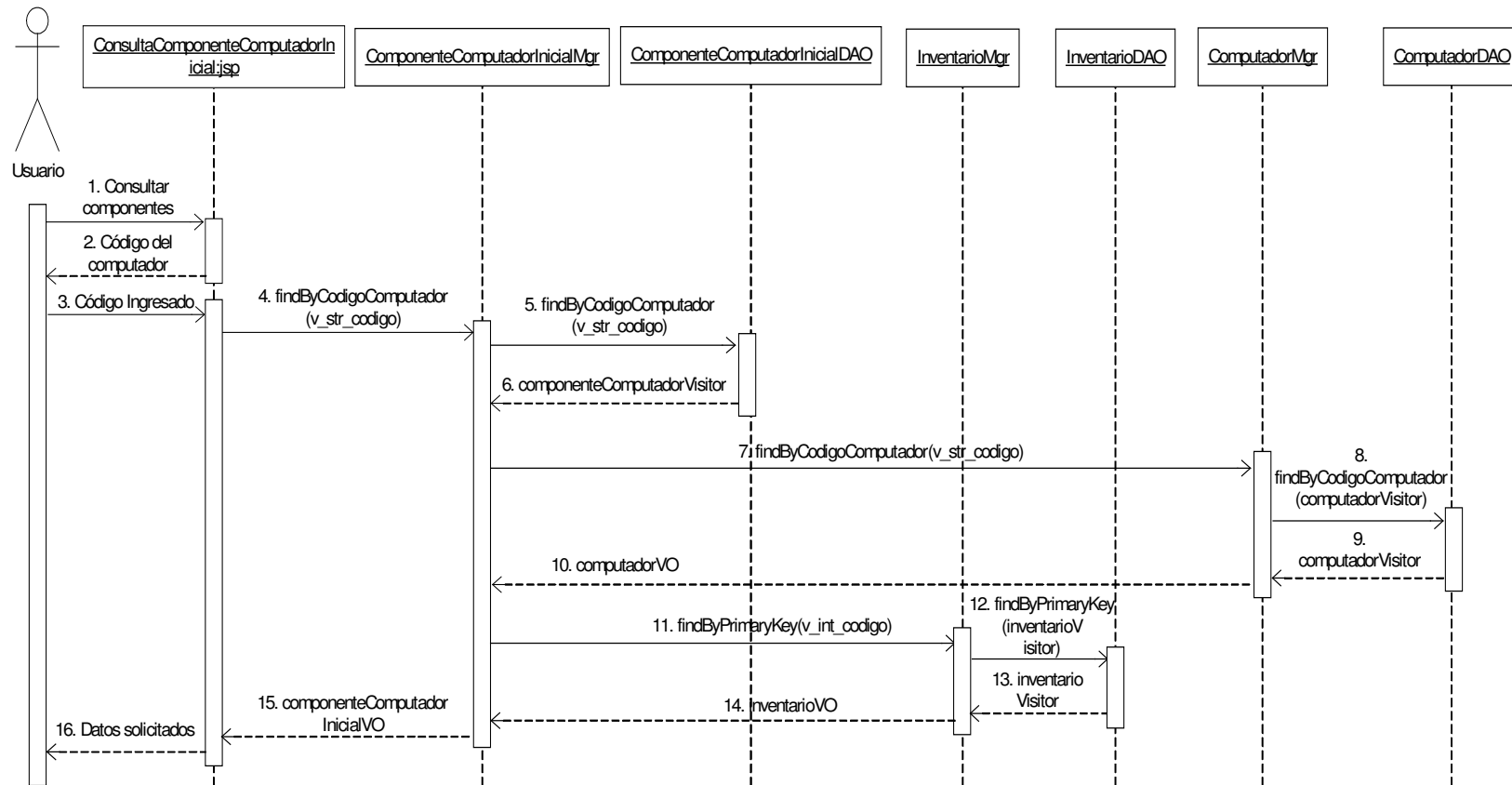


## 11.2.26 Administración de Componentes Computador Inicial

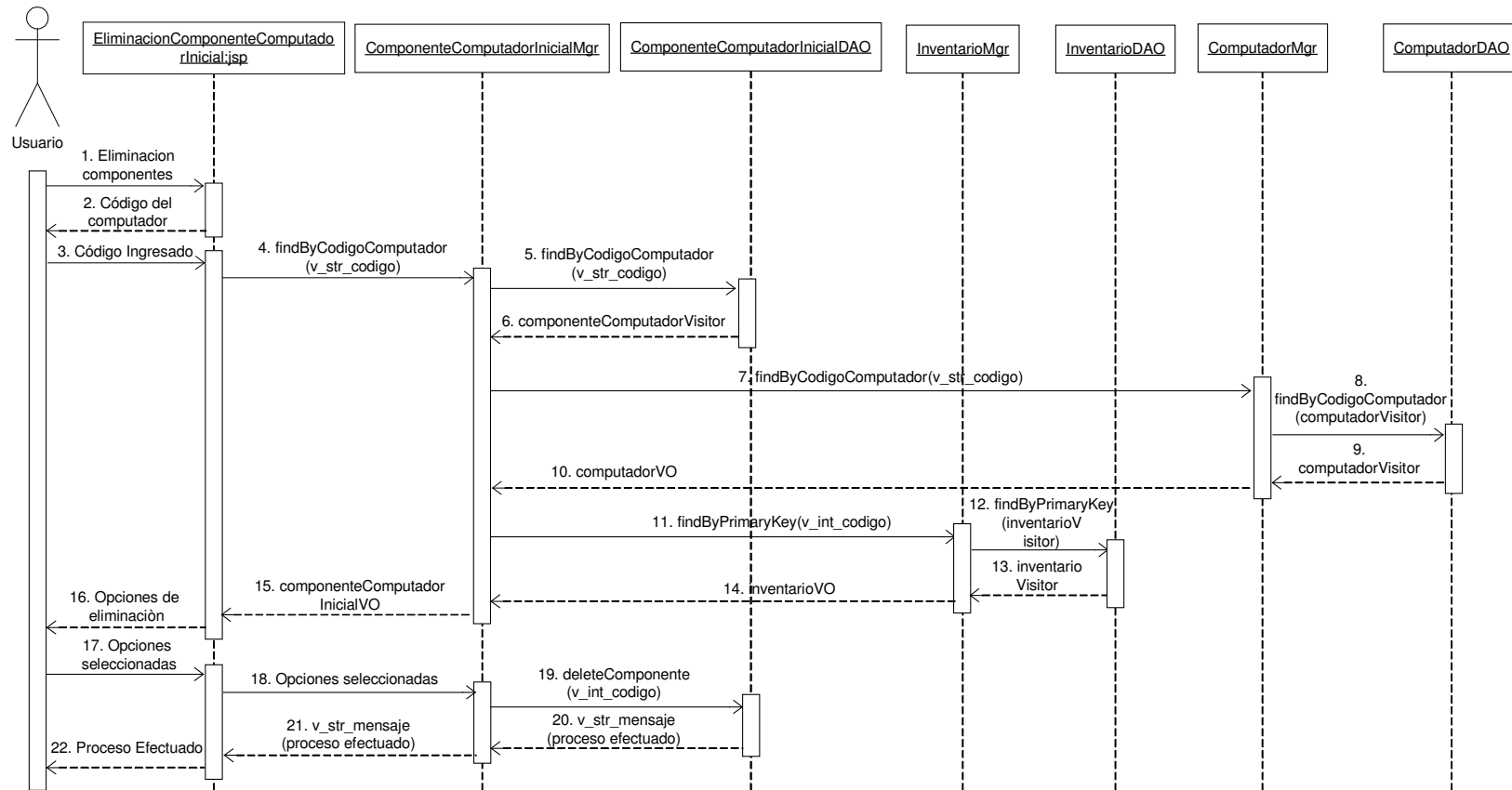
### 11.2.26.1 Asignar Componente Iniciales



### 11.2.26.2 Consultar Componente por Computador



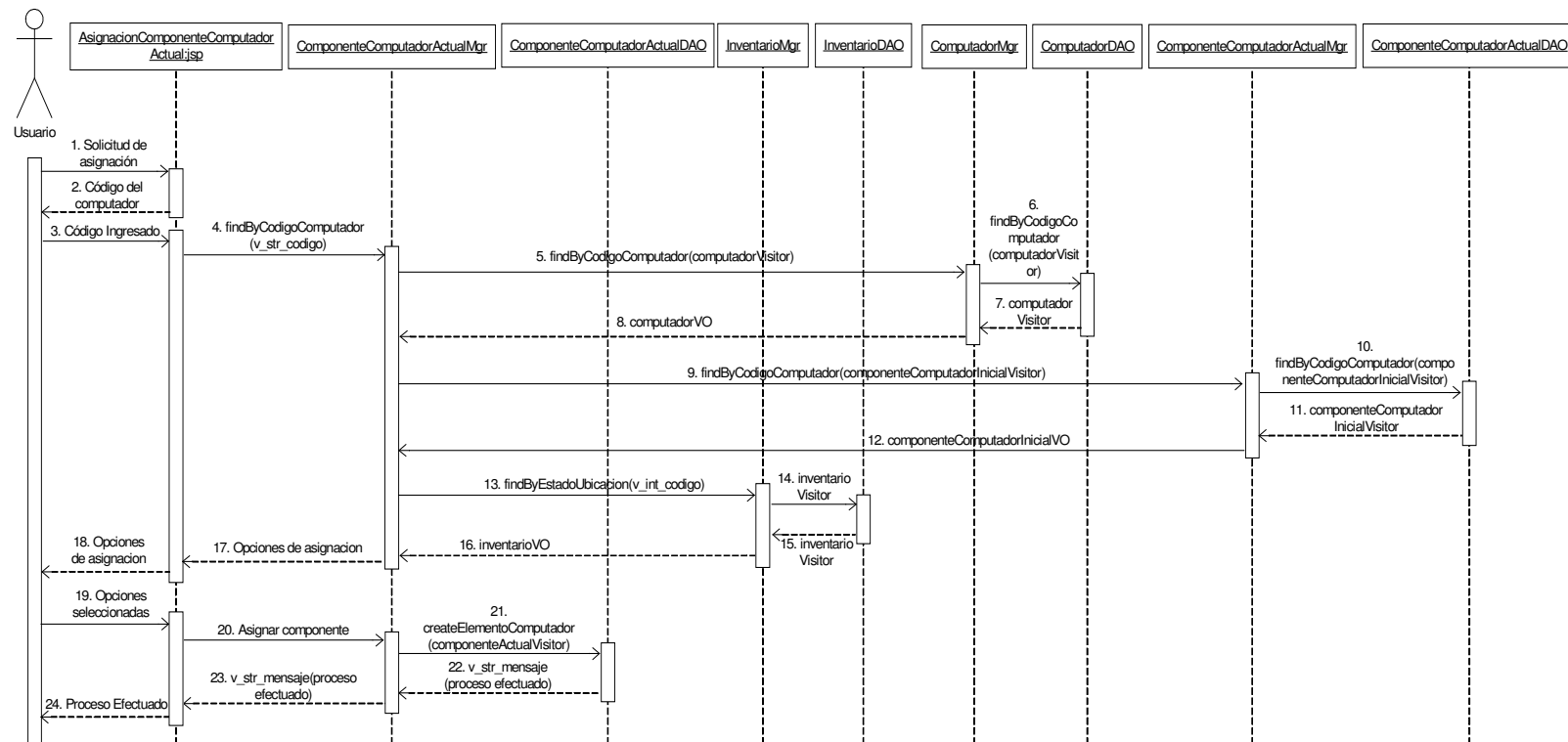
### 11.2.26.3 Quitar Componente Inicial



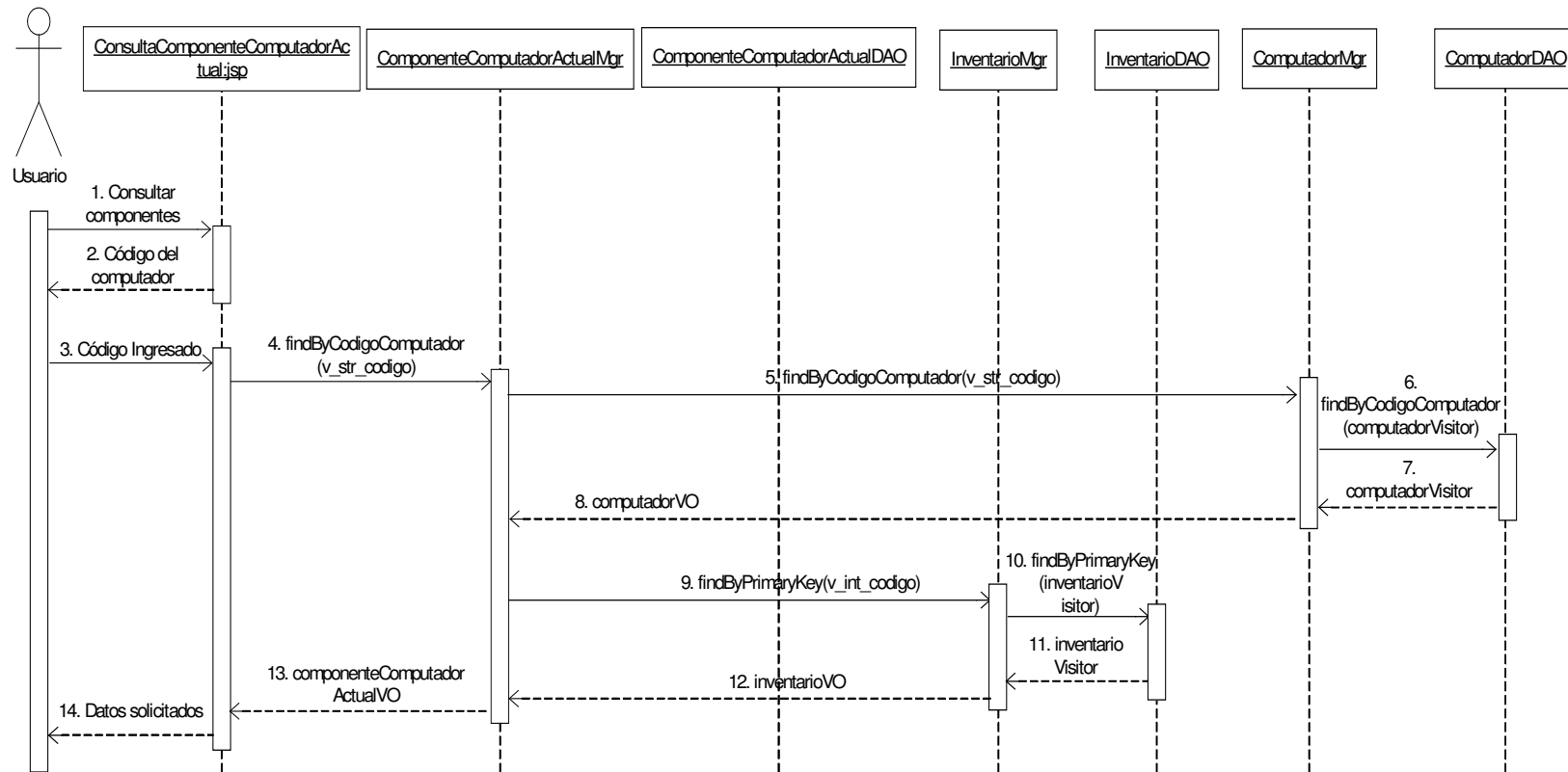


## 11.2.27 Administración de Componentes Computador Actual

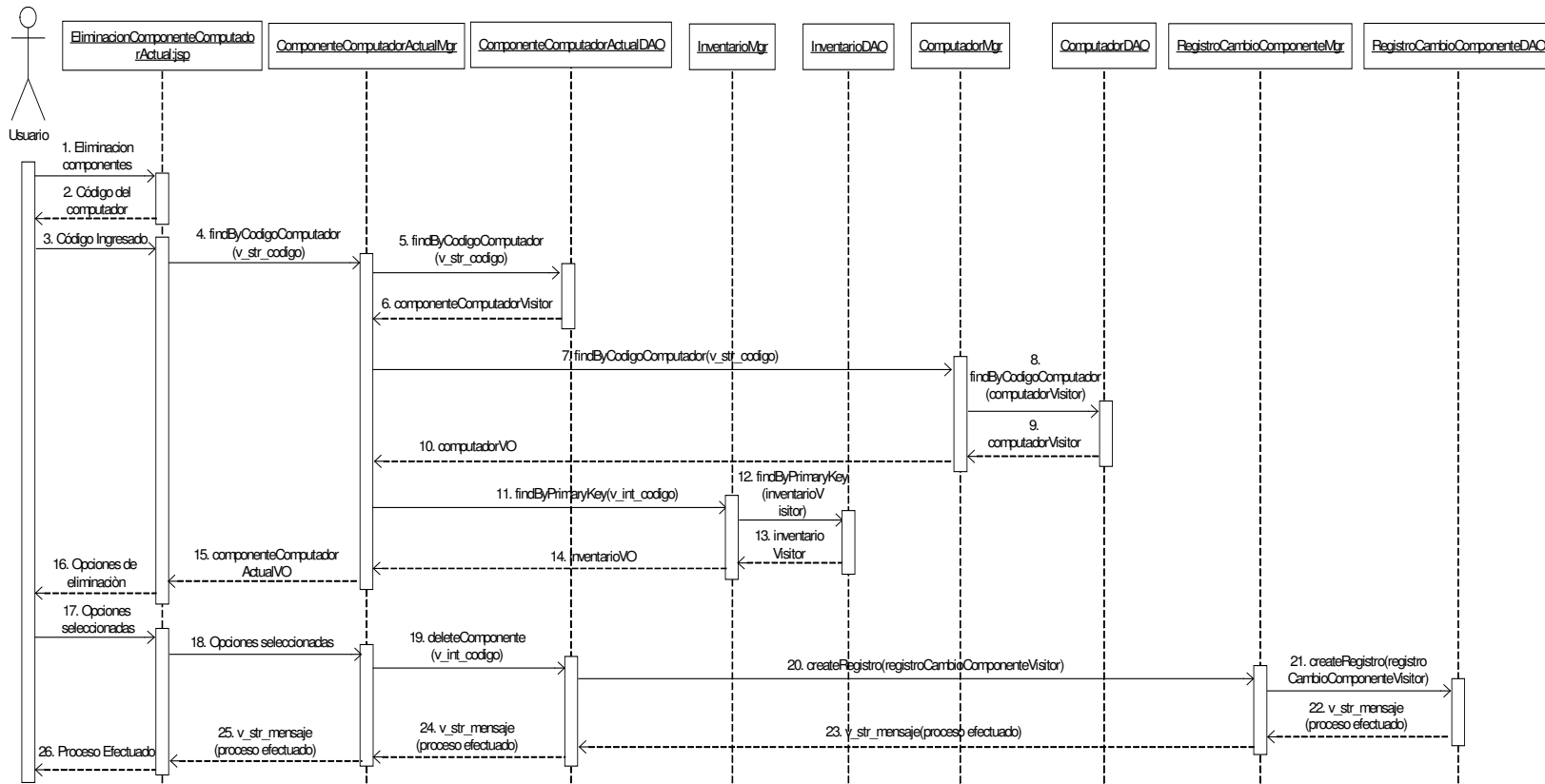
### 11.2.27.1 Asignar Componente Computador Actual



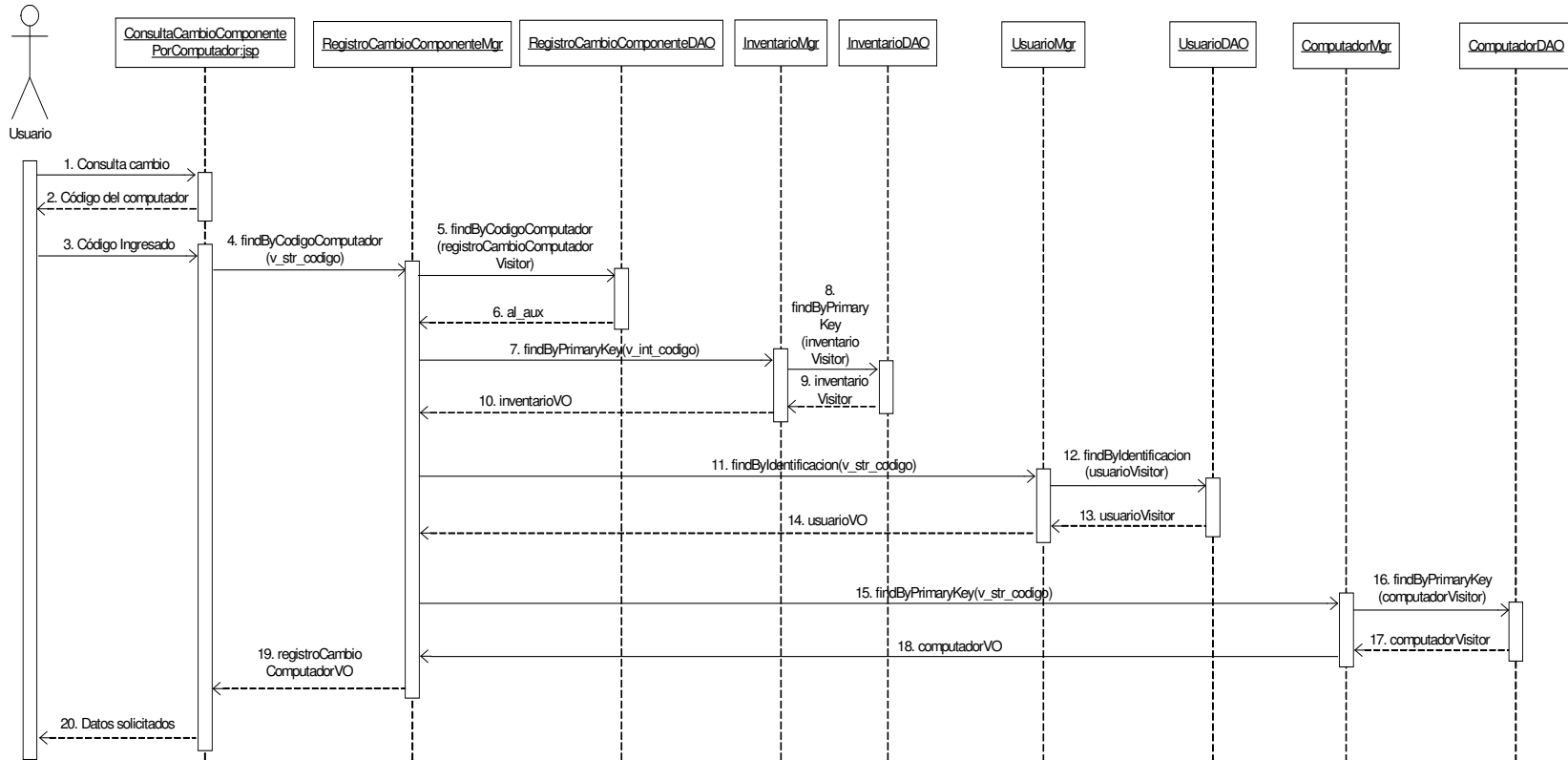
### 11.2.27.2 Consultar Componente por Computador



### 11.2.27.3 Quitar Componente Actual

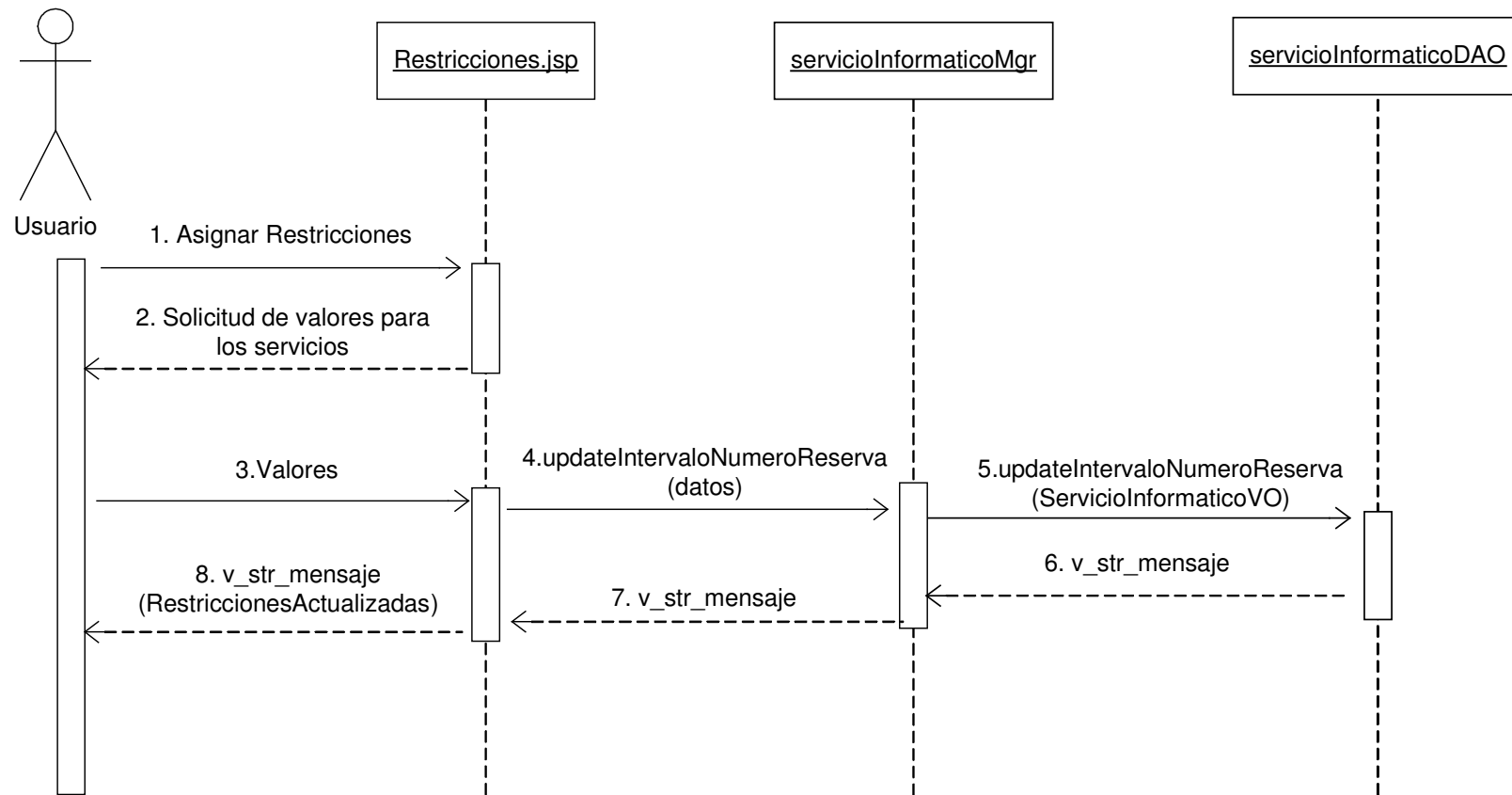


### 11.2.27.4 Consulta Registro Cambio Componente

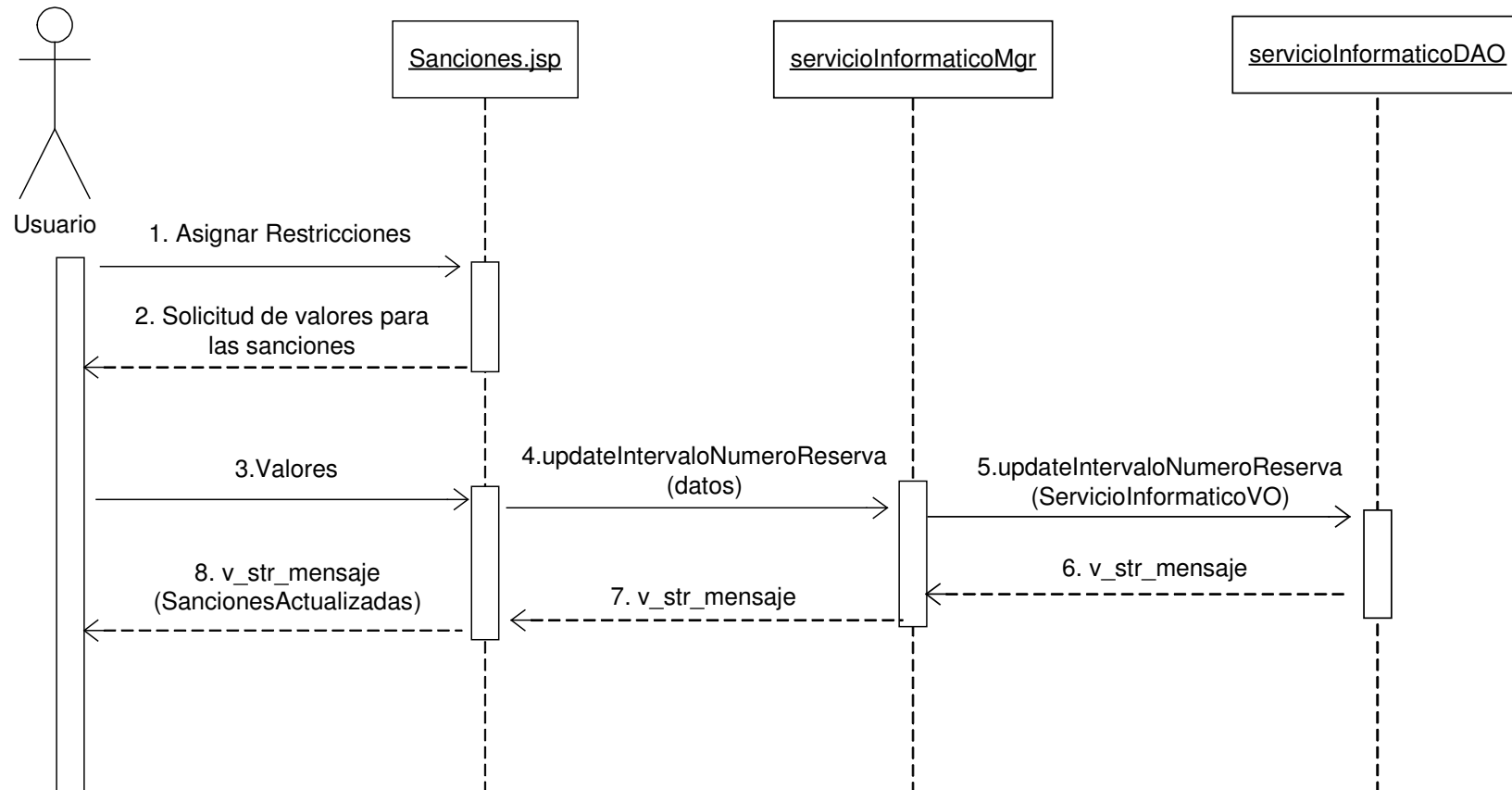


## 11.2.28 Administración de Servicios Informáticos

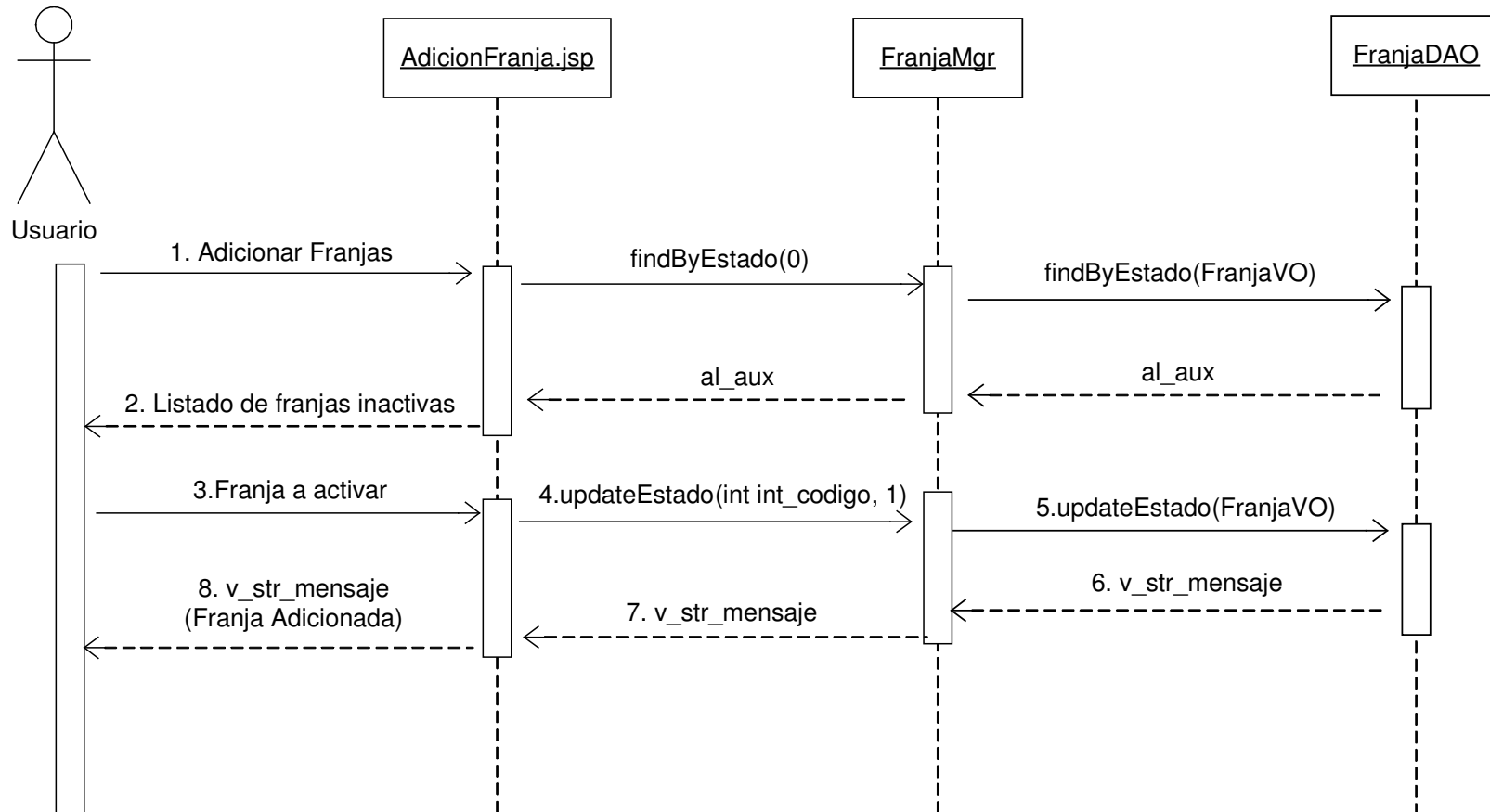
### 11.2.28.1 Actualizar Restricciones



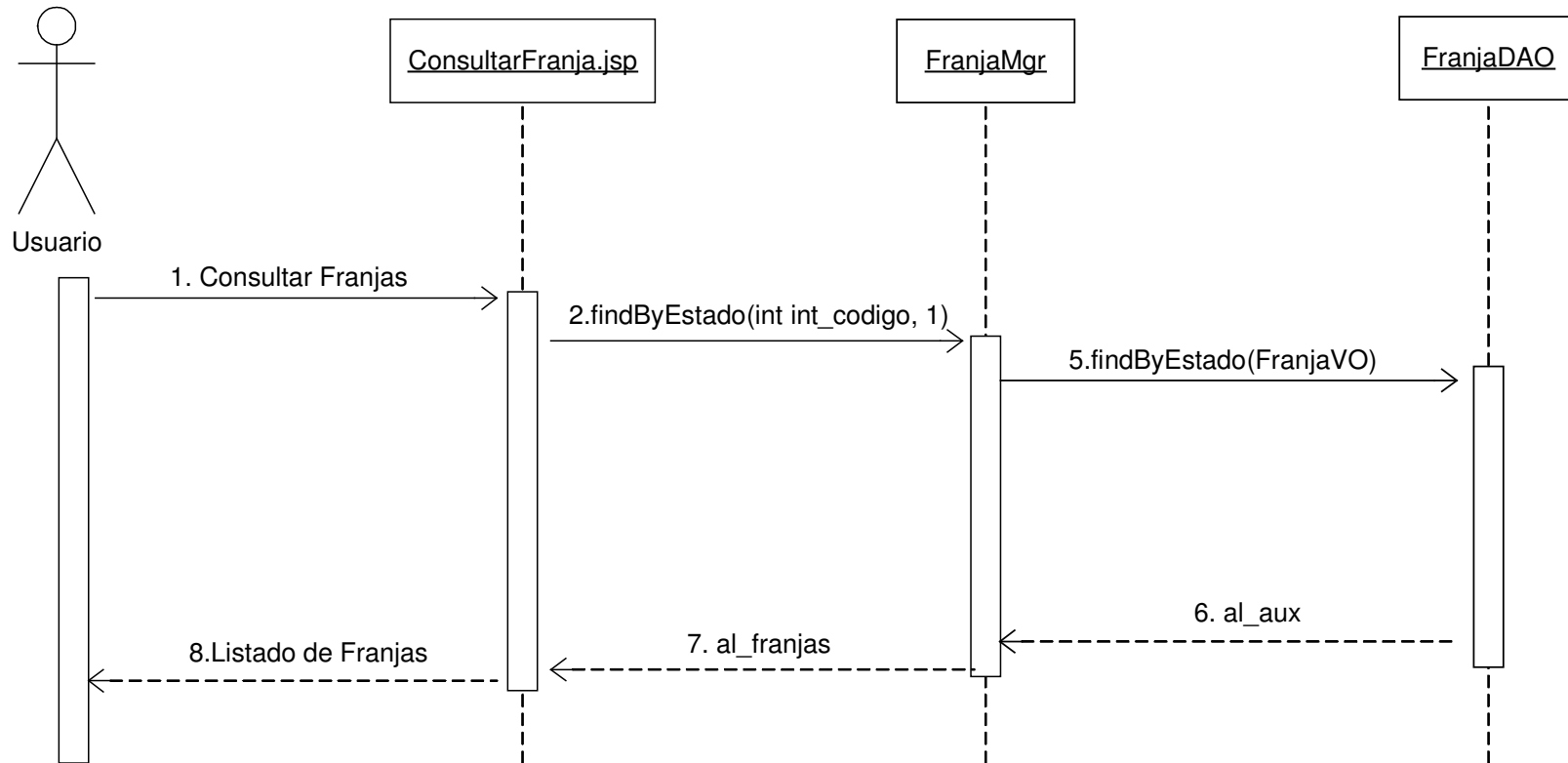
### 11.2.28.2 Actualizar Sanciones



### 11.2.28.3 Adicionar Franja Horaria

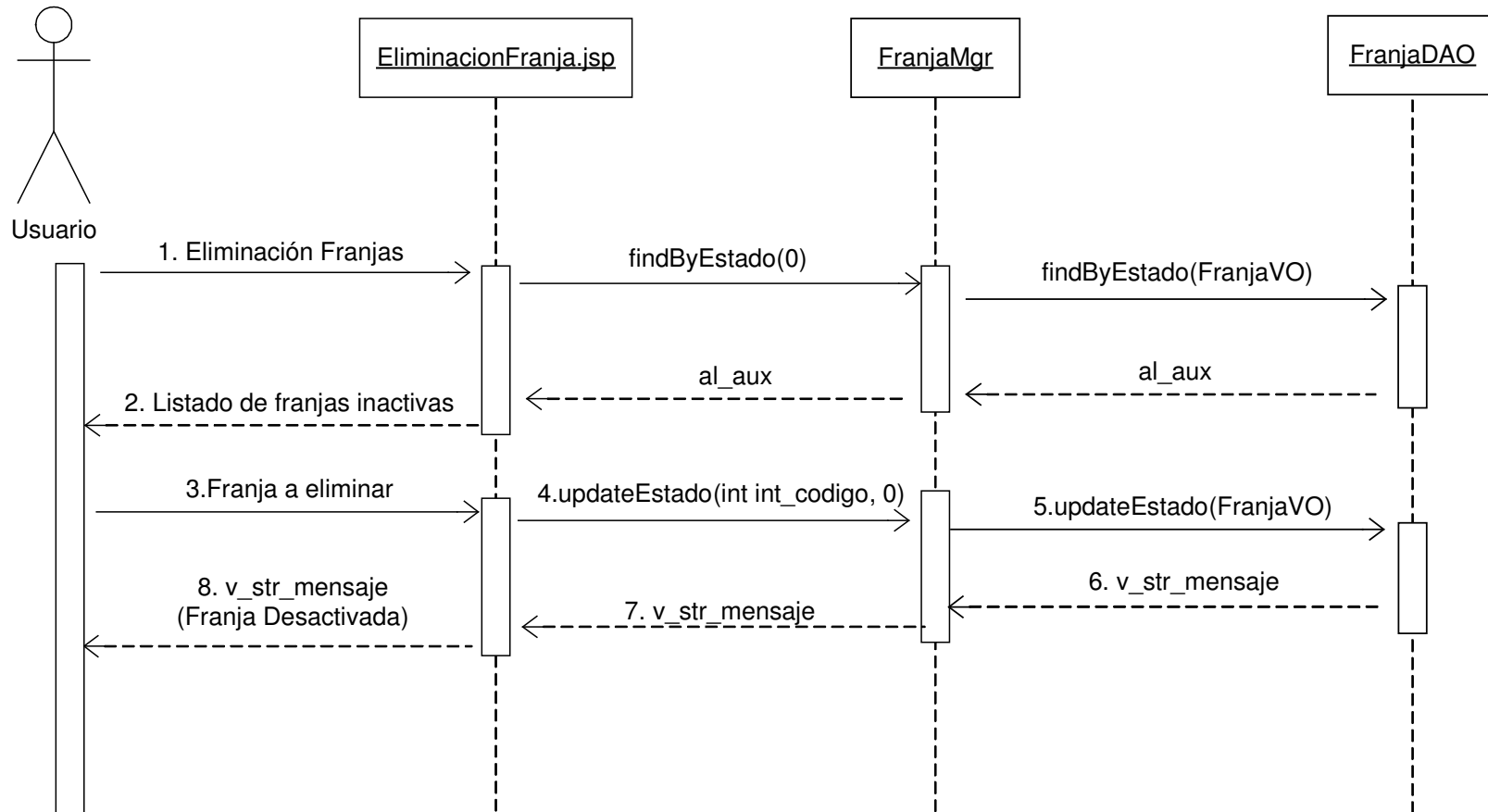


#### 11.2.28.4 Consultar Franja Horaria

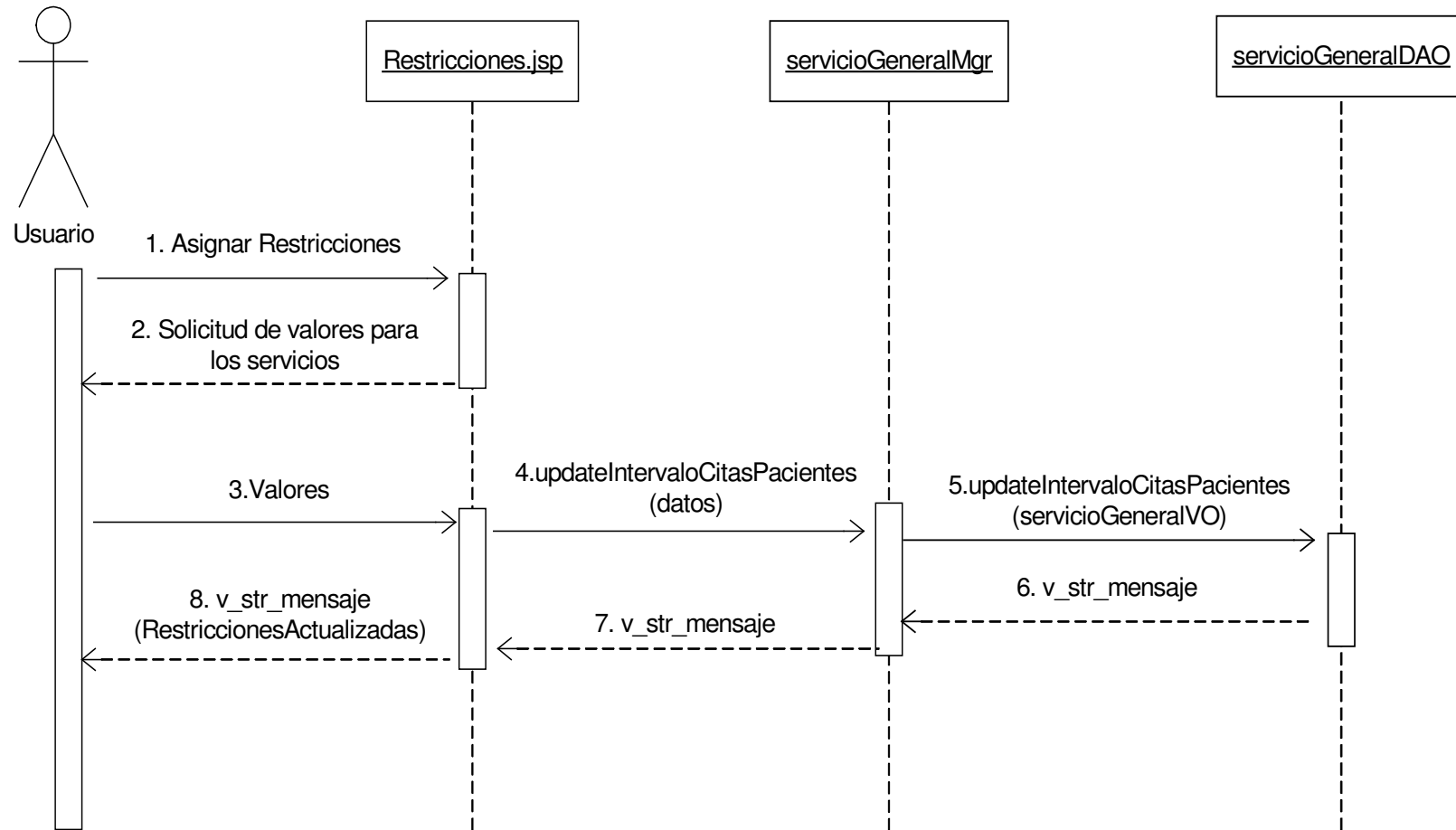




### 11.2.28.5 Eliminar Franja Horaria

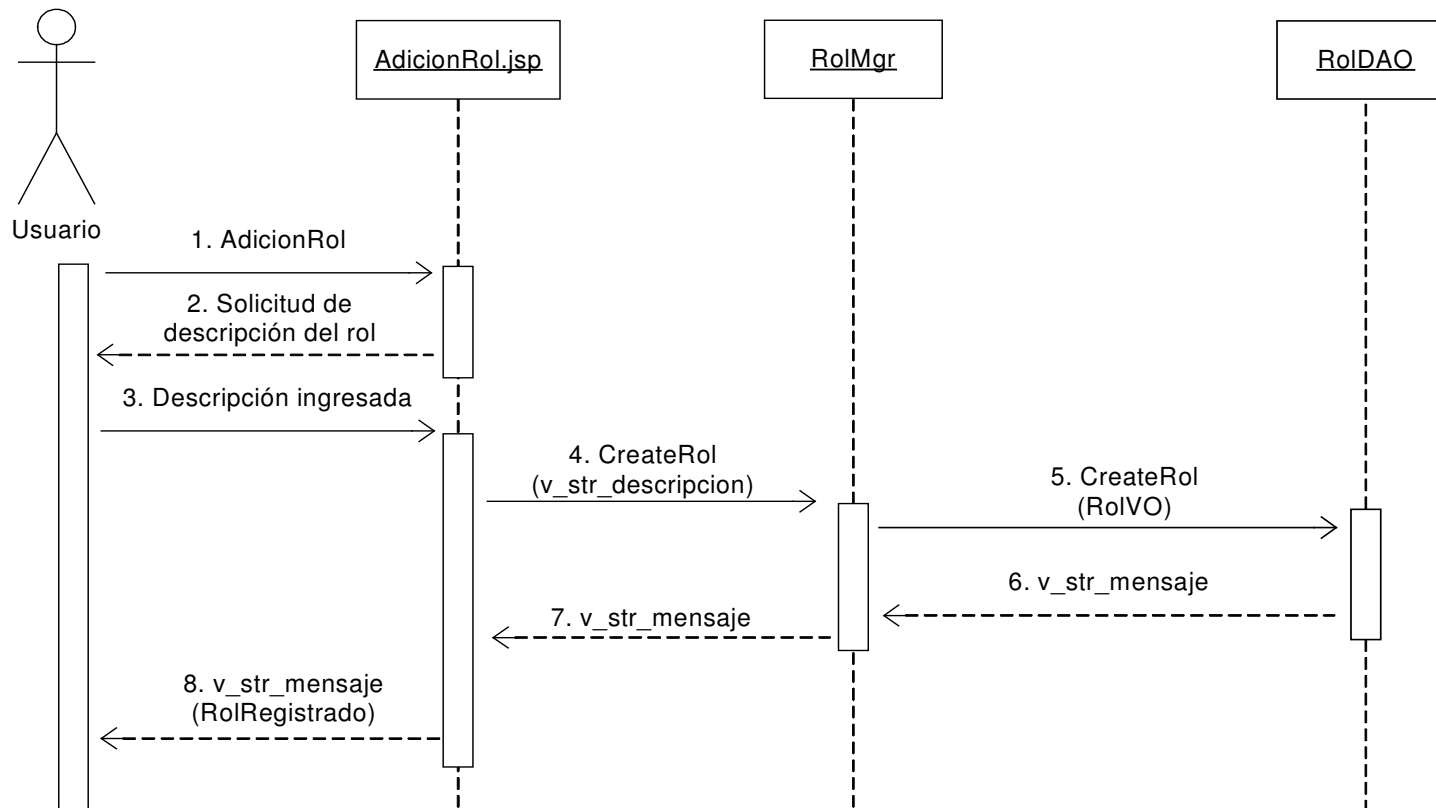


### 11.2.29 Actualizar Restricciones Médicas

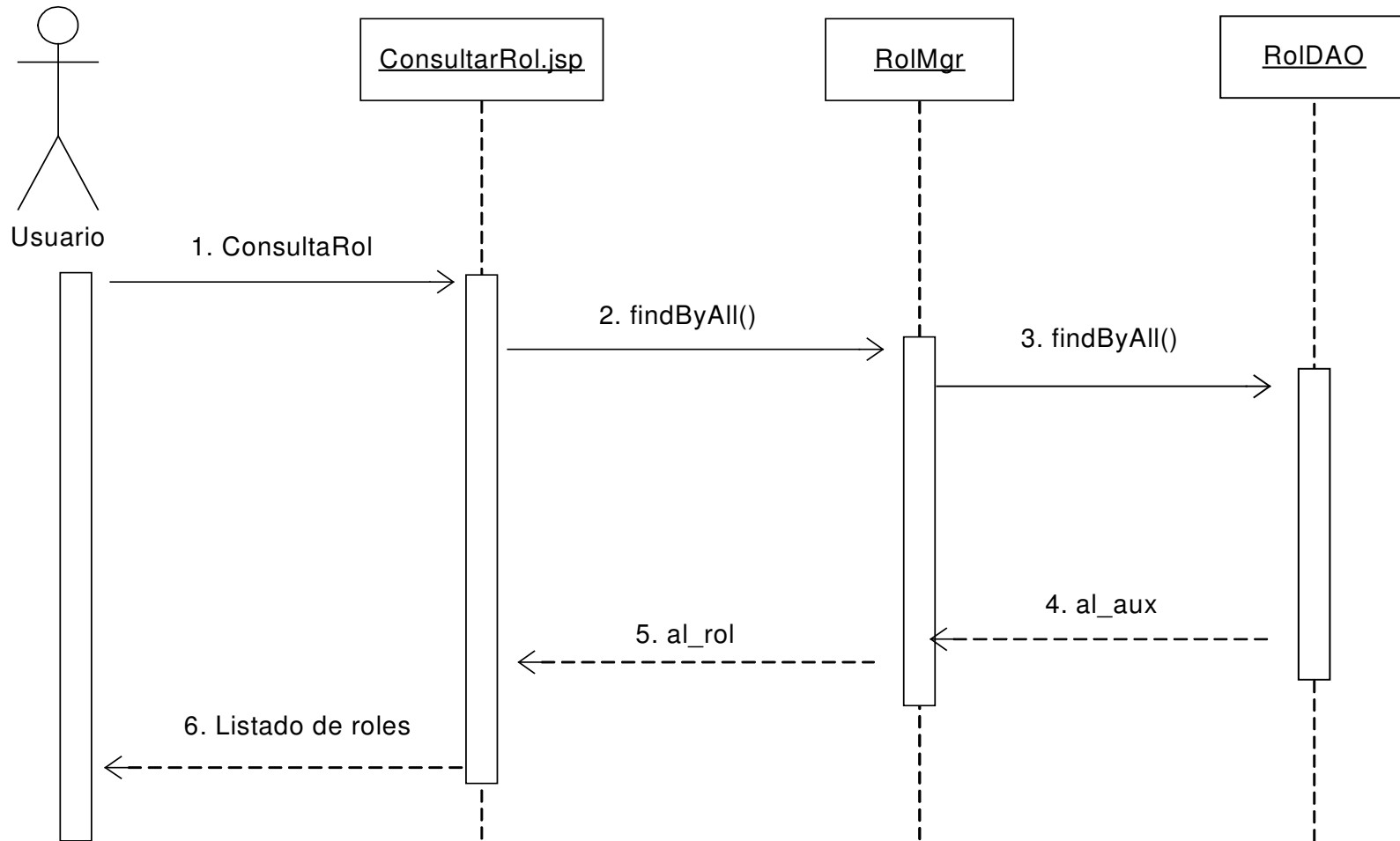


## 11.2.30 Administración de Roles

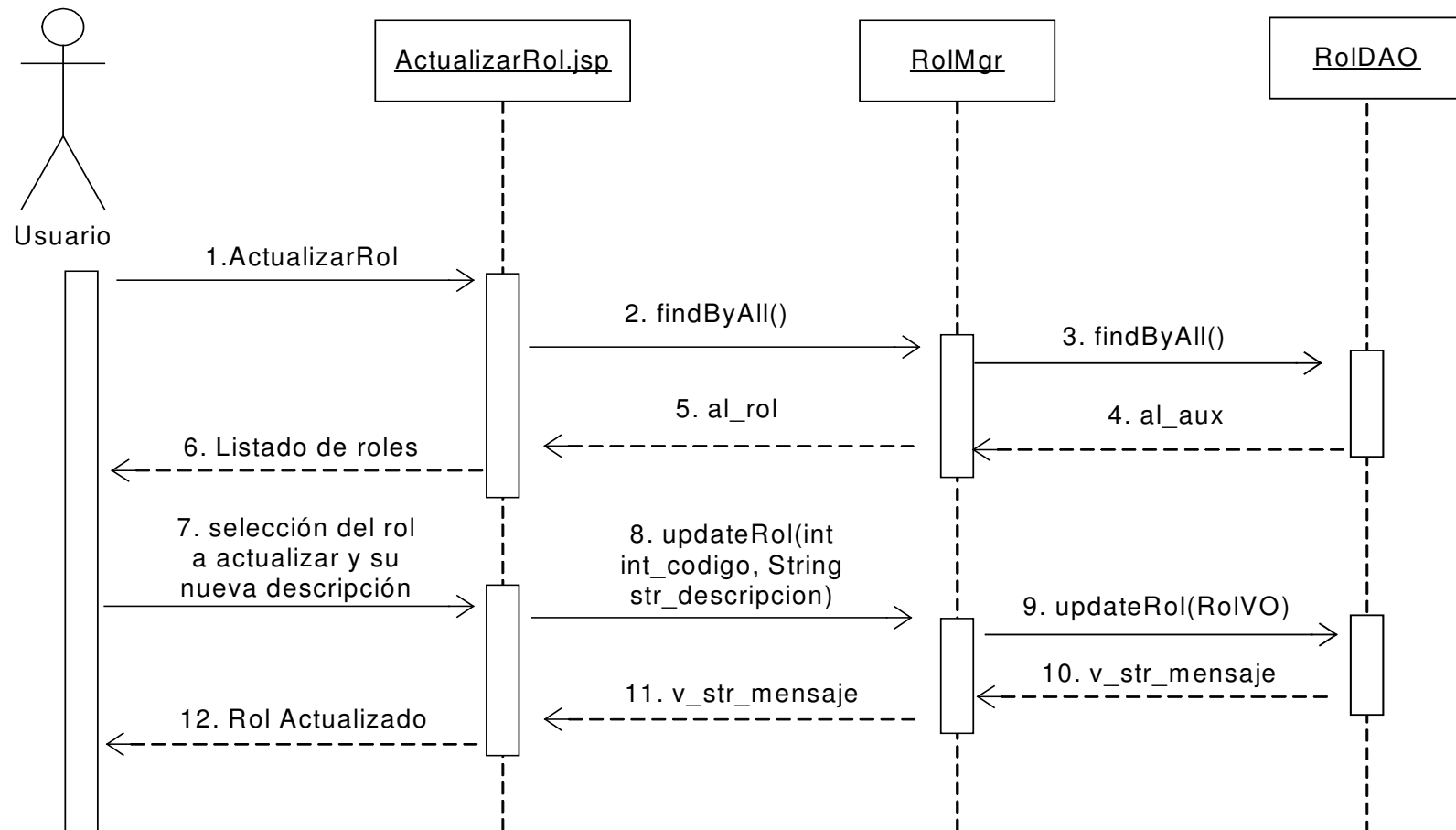
### 11.2.30.1 Crear Rol



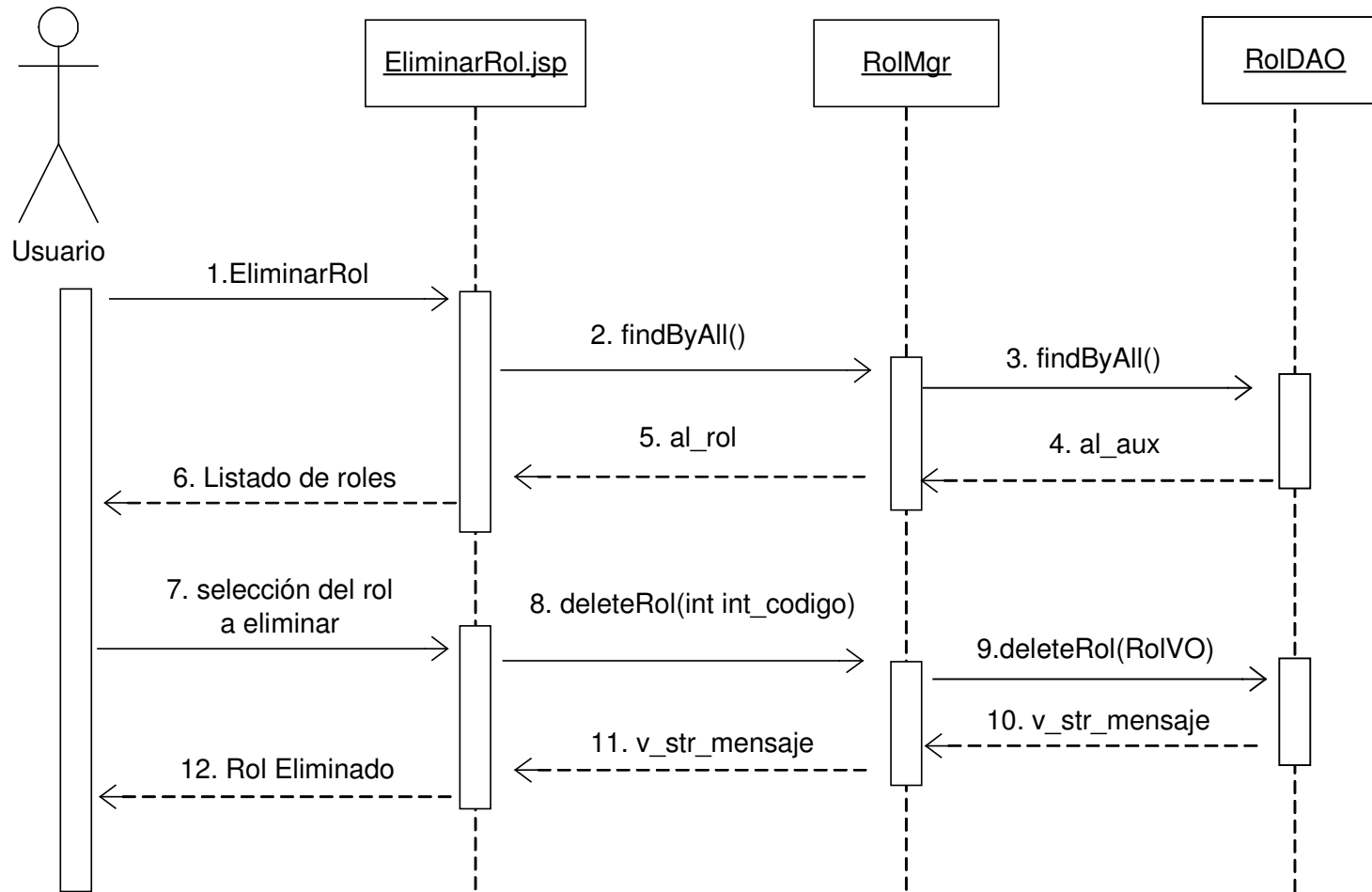
### 11.2.30.2 Consultar Rol



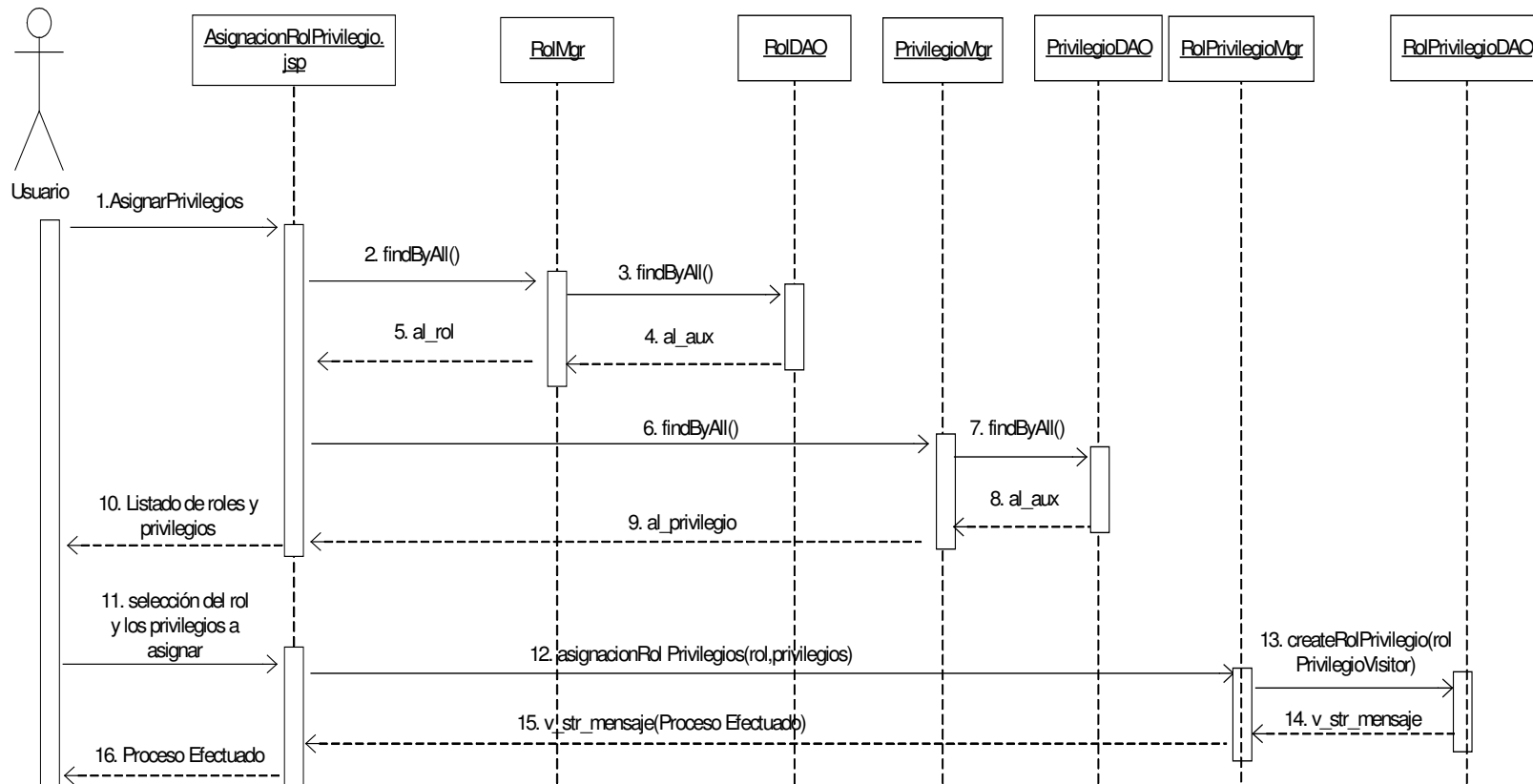
### 11.2.30.3 Actualizar Rol



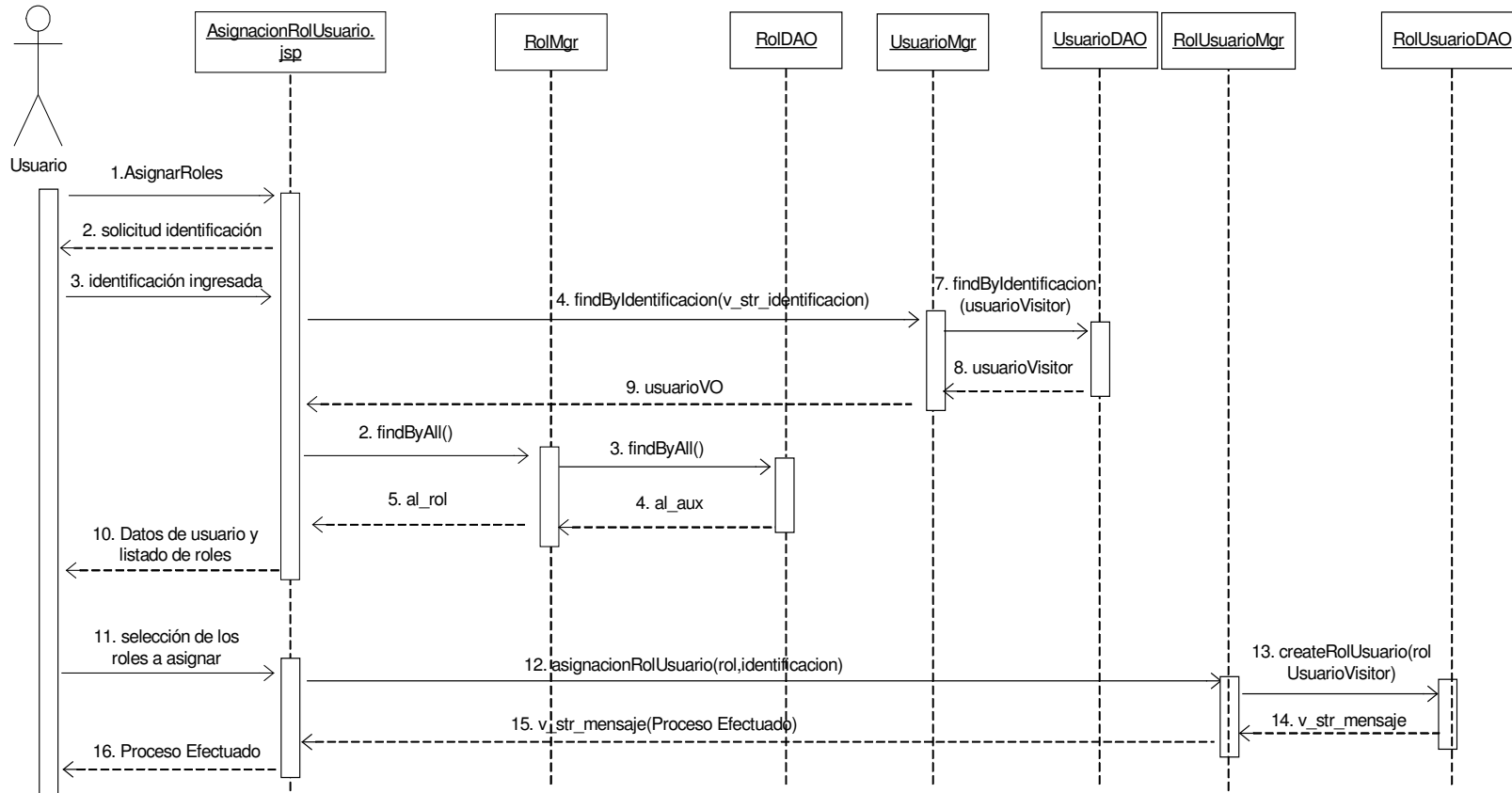
### 11.2.30.4 Eliminar Rol



### 11.2.30.5 Asignar Privilegios Rol



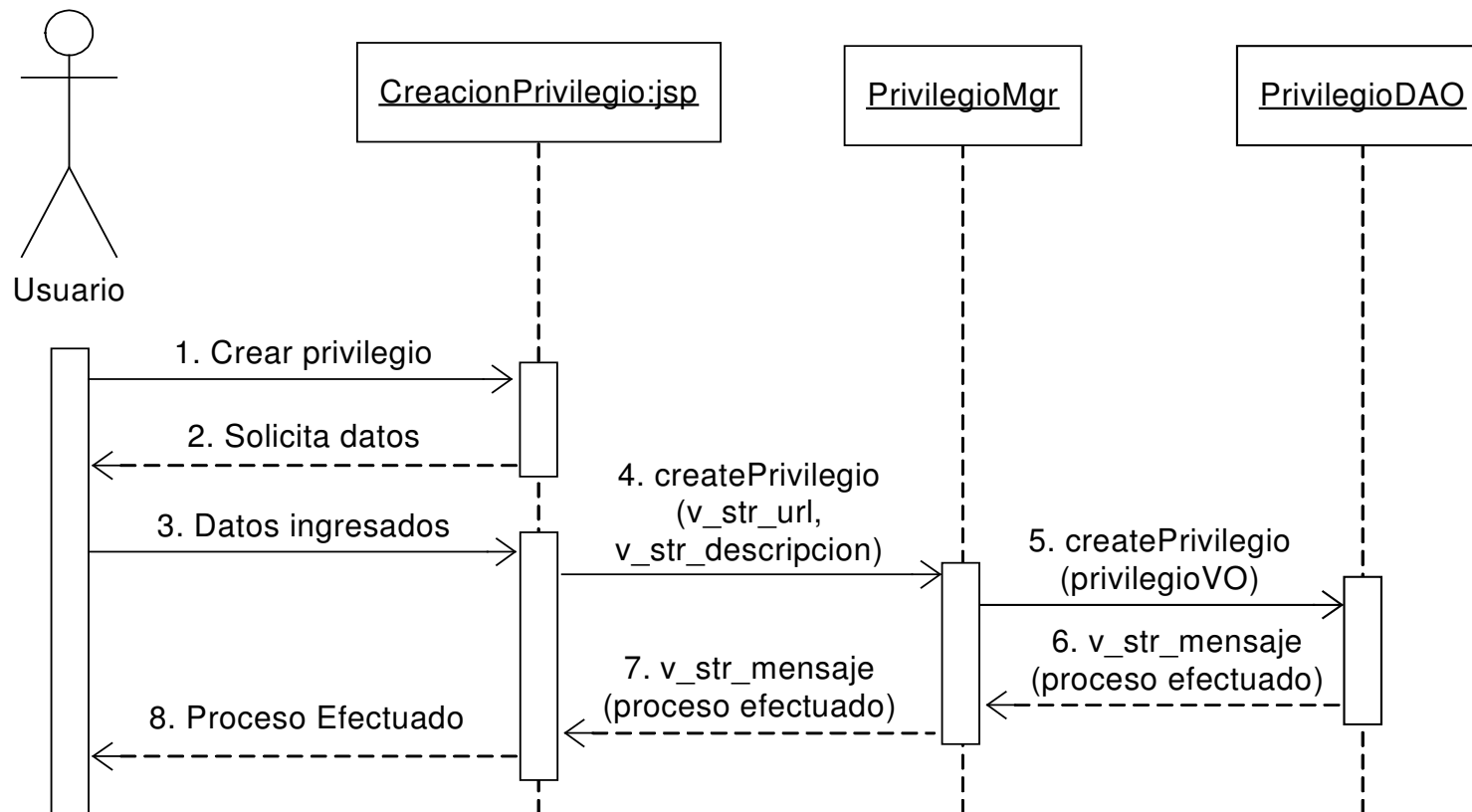
### 11.2.30.6 Asignar Roles Usuario



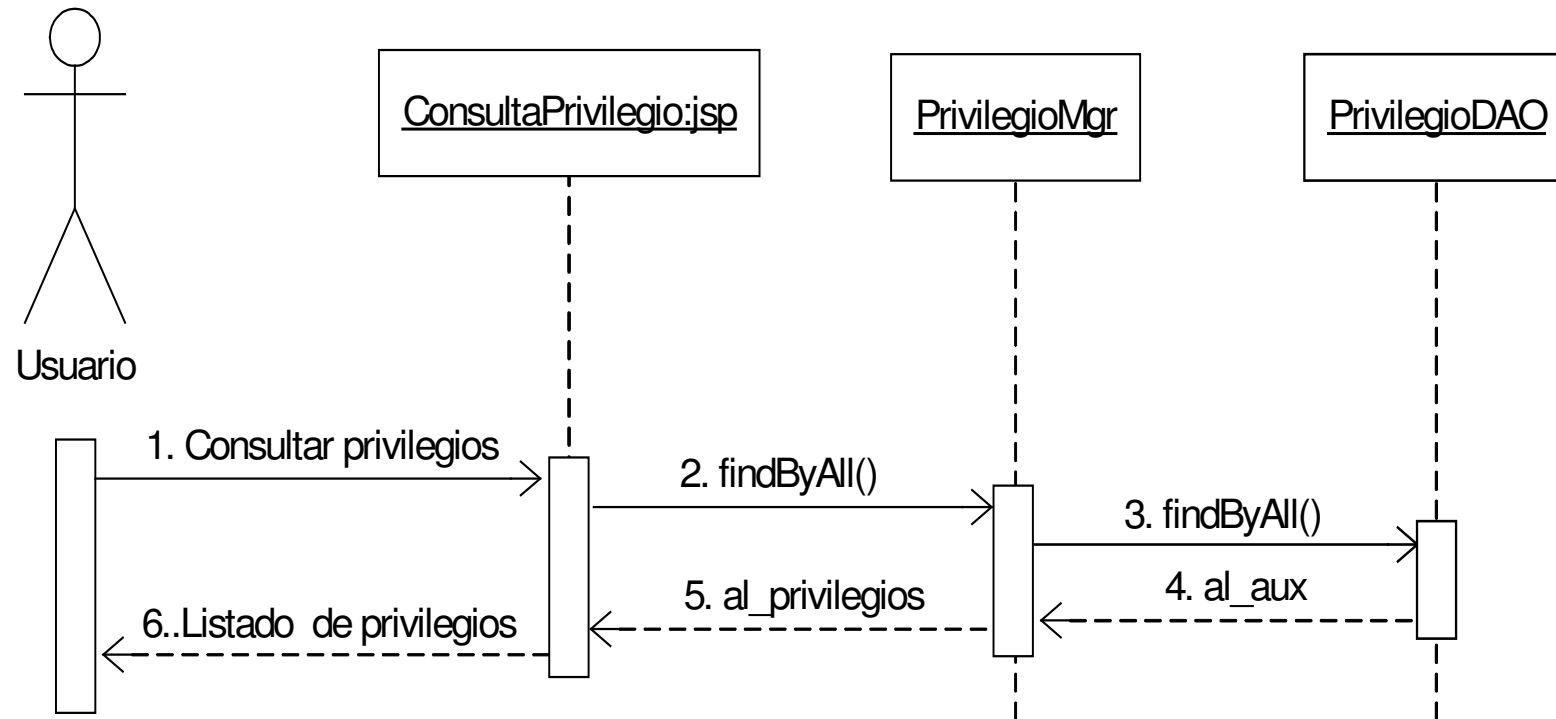


## 11.2.31 Administración Privilegios

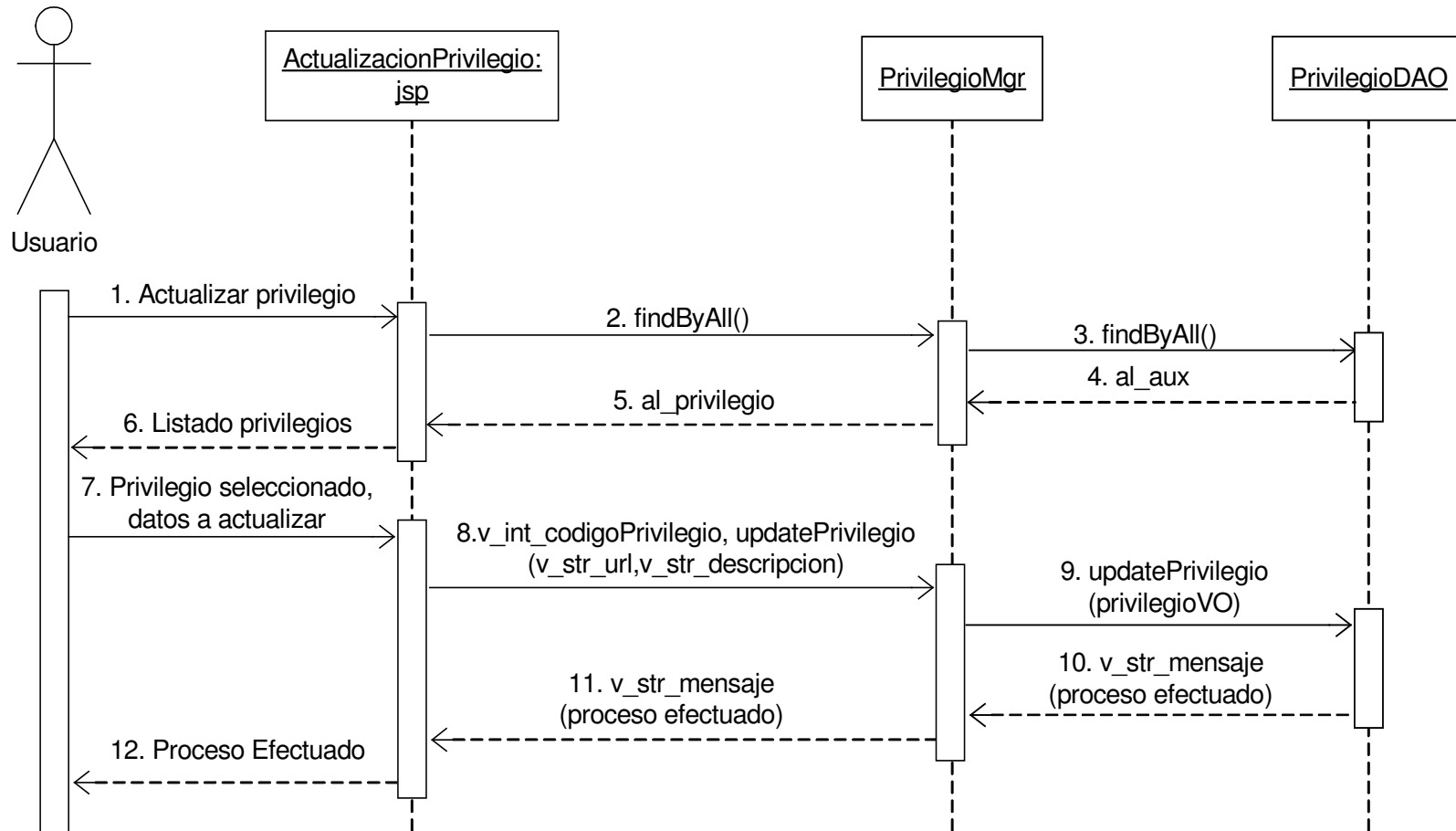
### 11.2.31.1 Crear Privilegio



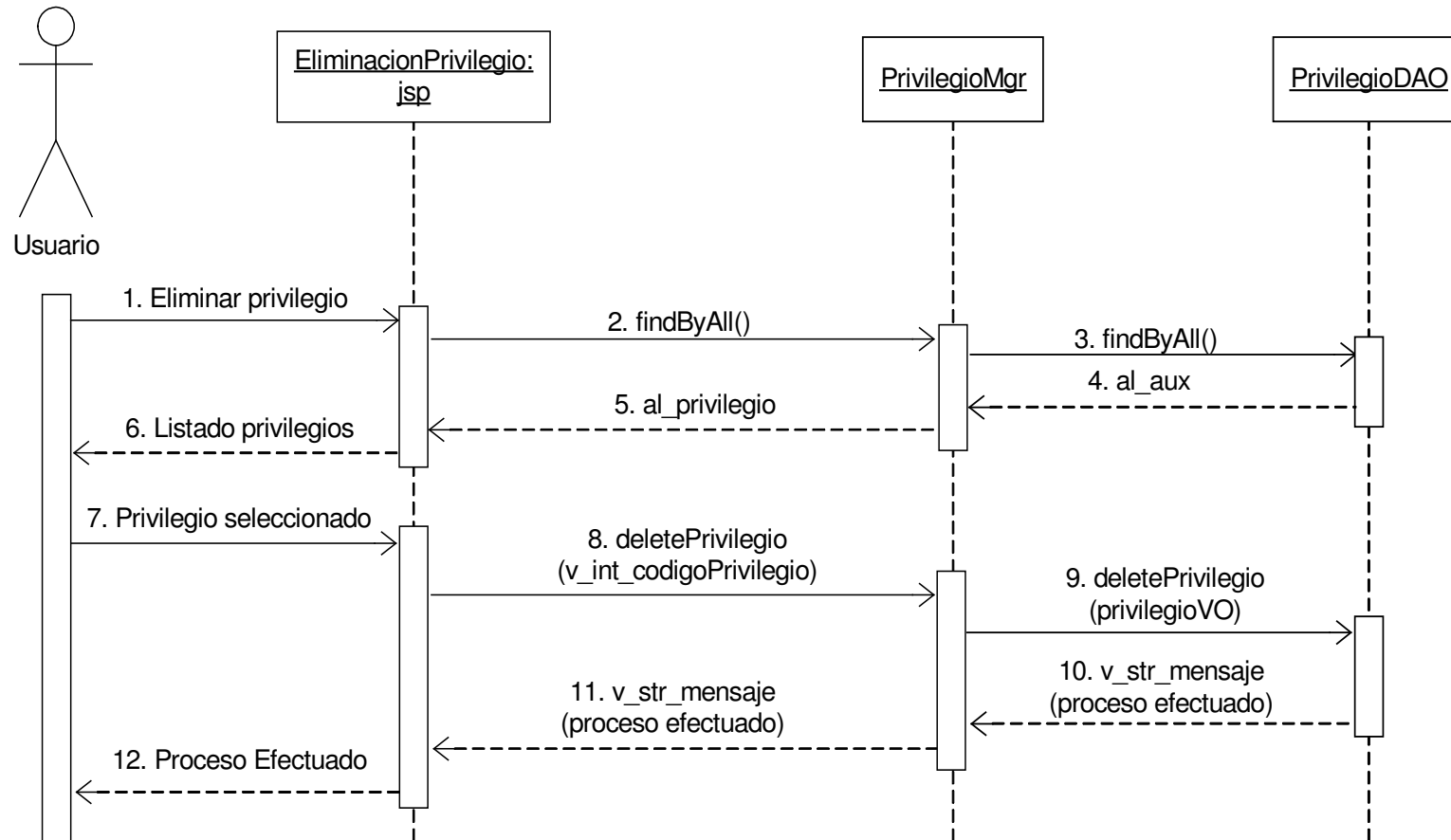
### 11.2.31.2 Consultar Privilegio



### 11.2.31.3 Actualizar Privilegio

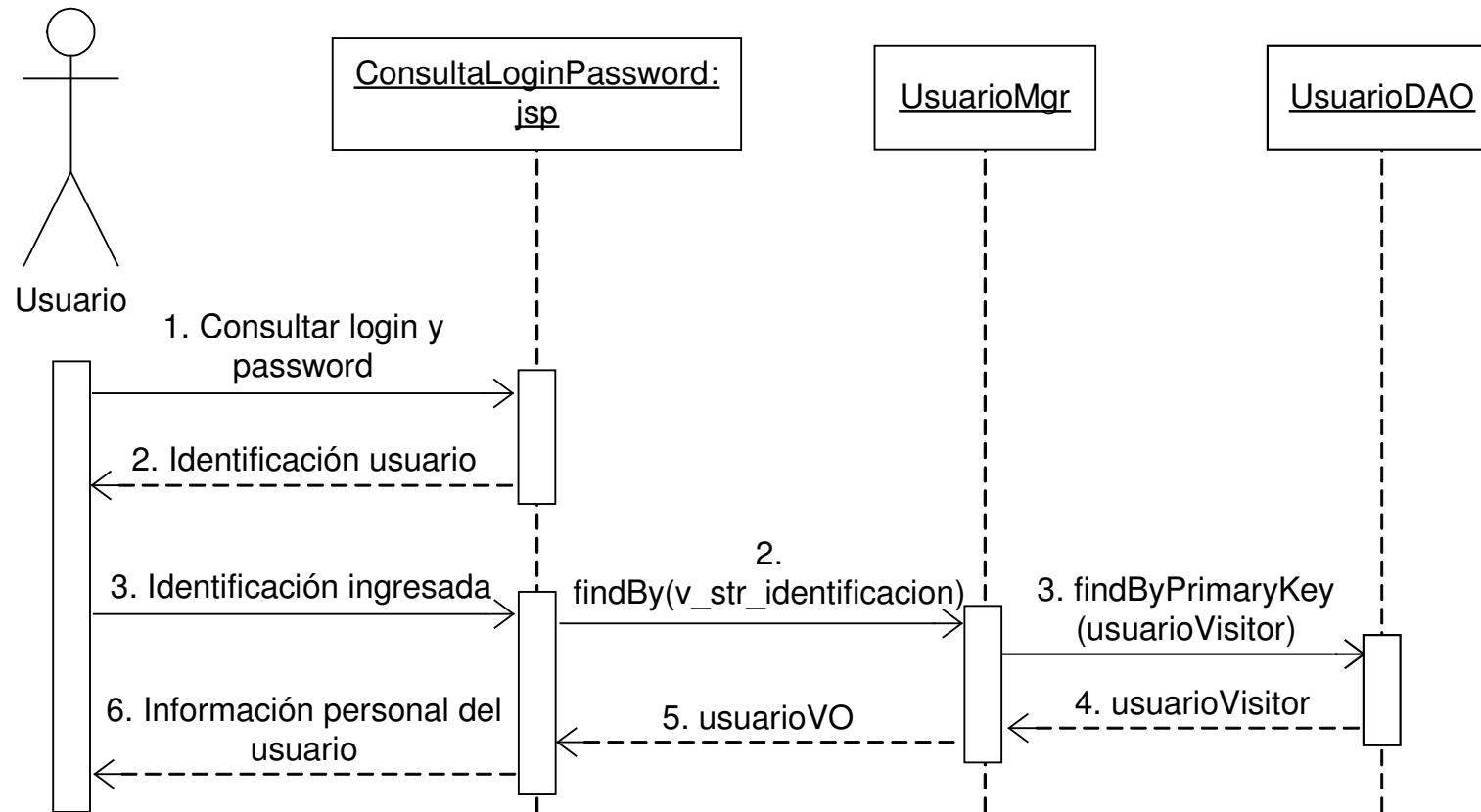


### 11.2.31.4 Eliminar Privilegio

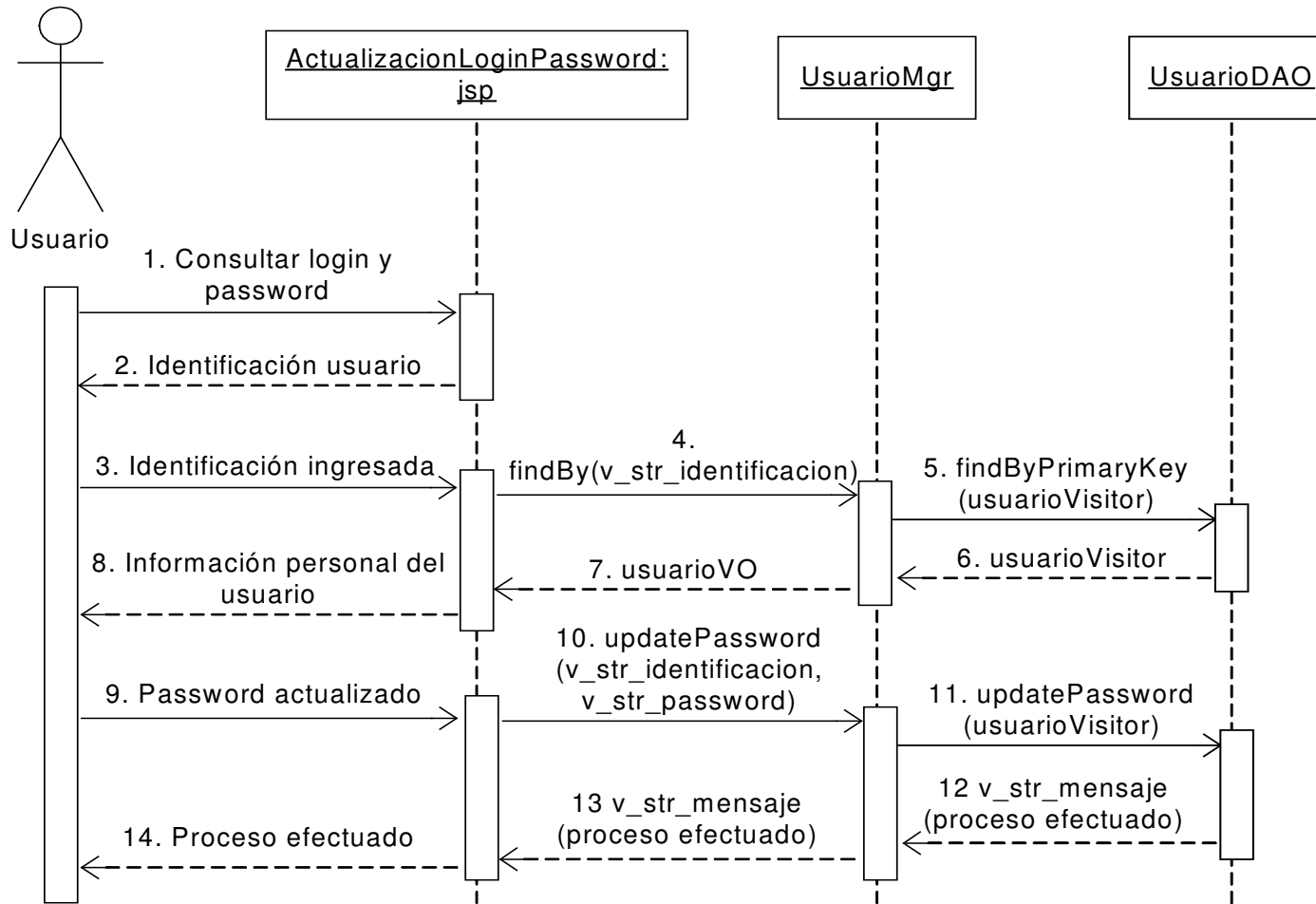


## 11.2.32 Administración Login y Password

### 11.2.32.1 Consultar Login y Password

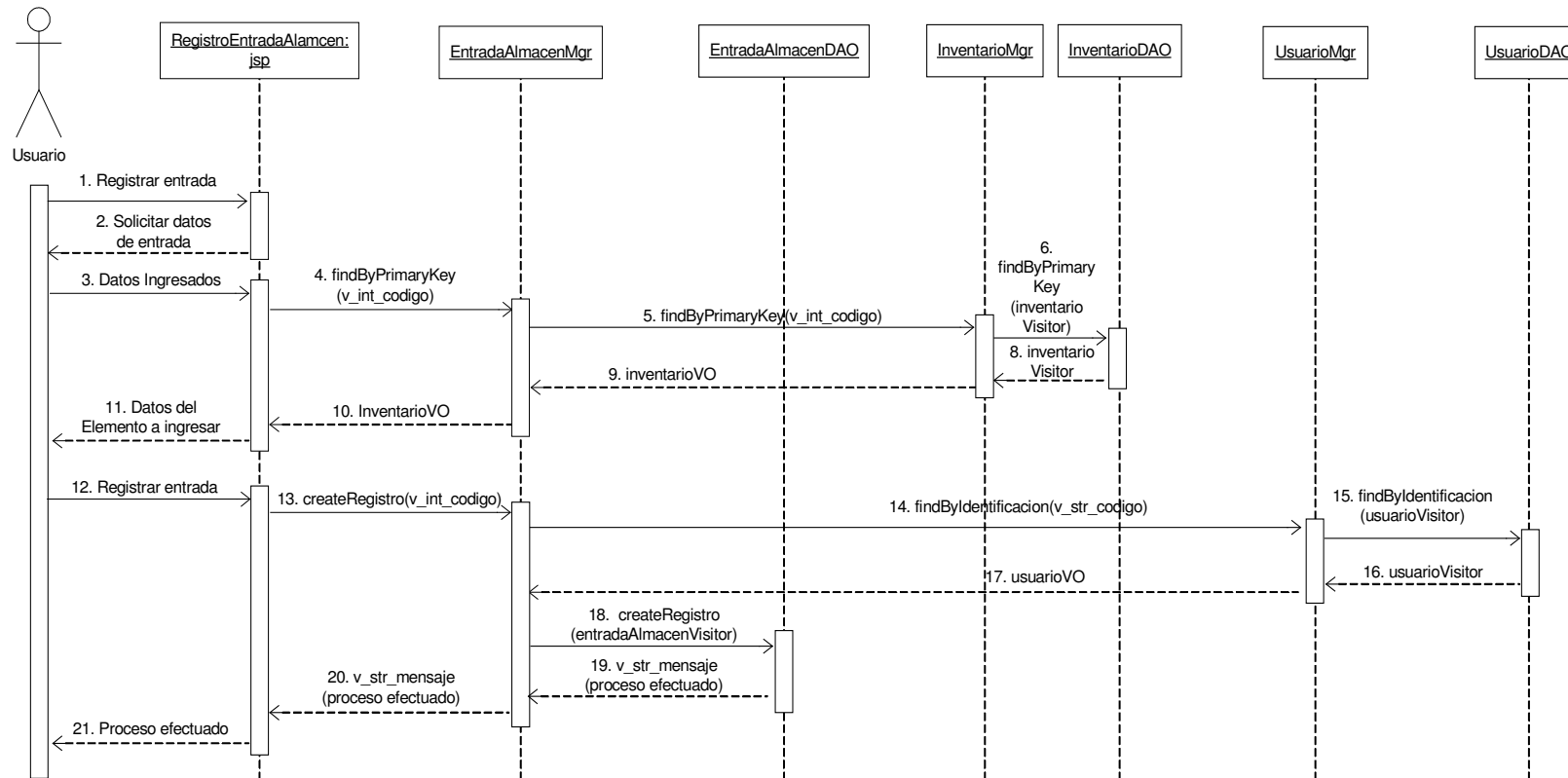


### 11.2.32.2 Actualizar Password

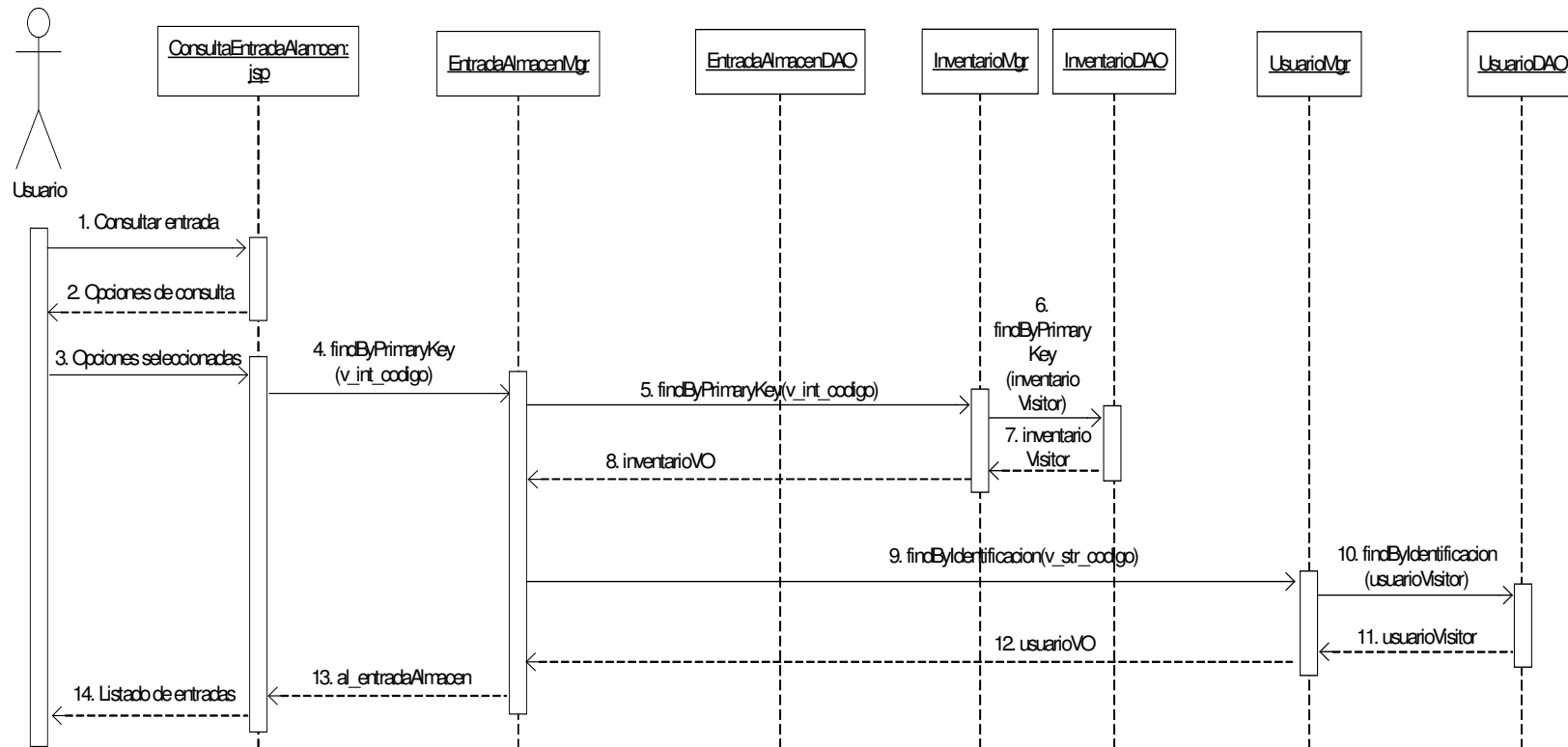


## 11.2.33 Administración de Entradas de Almacén

### 11.2.33.1 Registrar Entrada Almacén



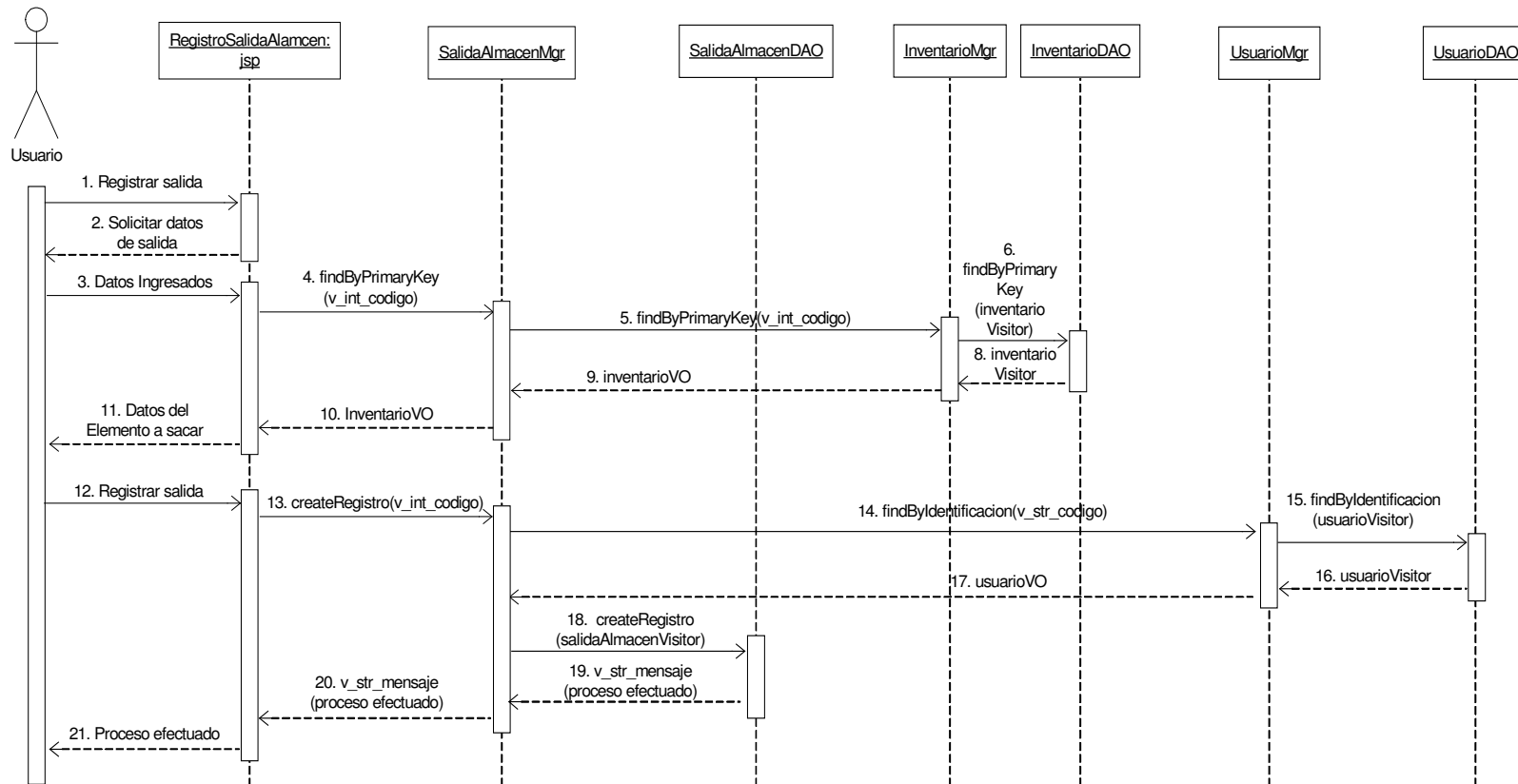
### 11.2.33.2 Consultar Entrada Almacén



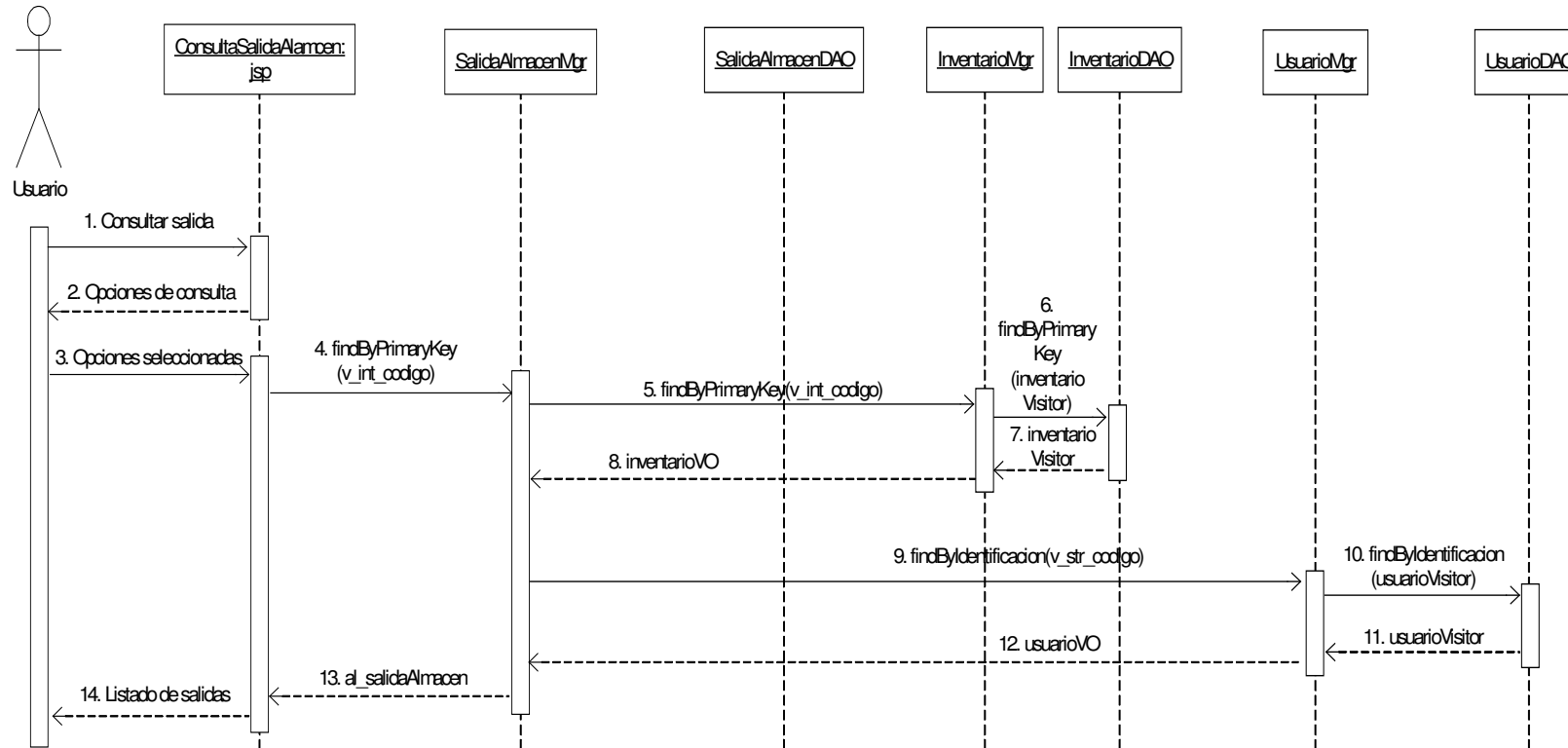


## 11.2.34 Administración de Salidas de Almacén

### 11.2.34.1 Registrar Salida Almacén

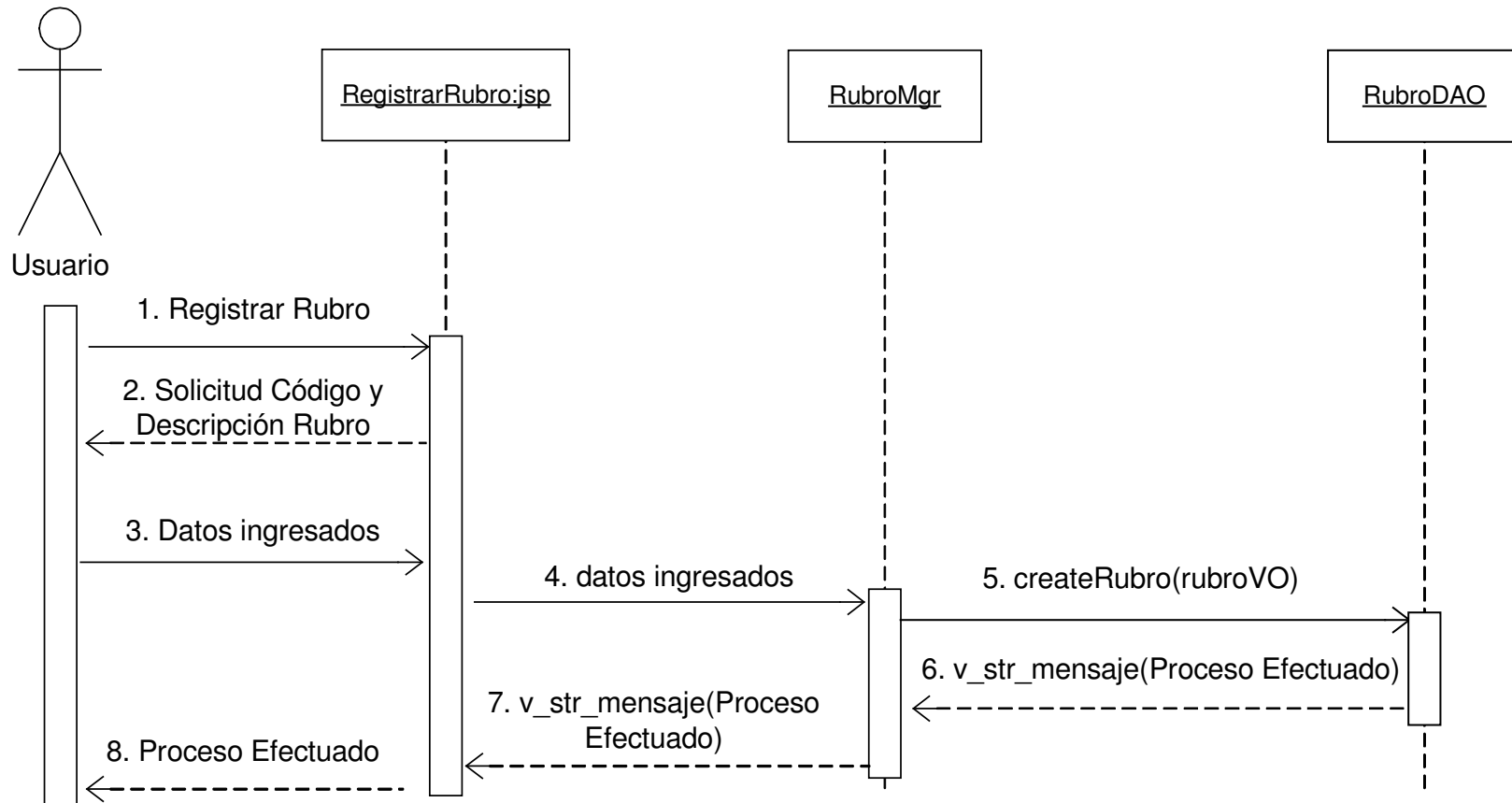


### 11.2.34.2 Consultar Salida Almacén

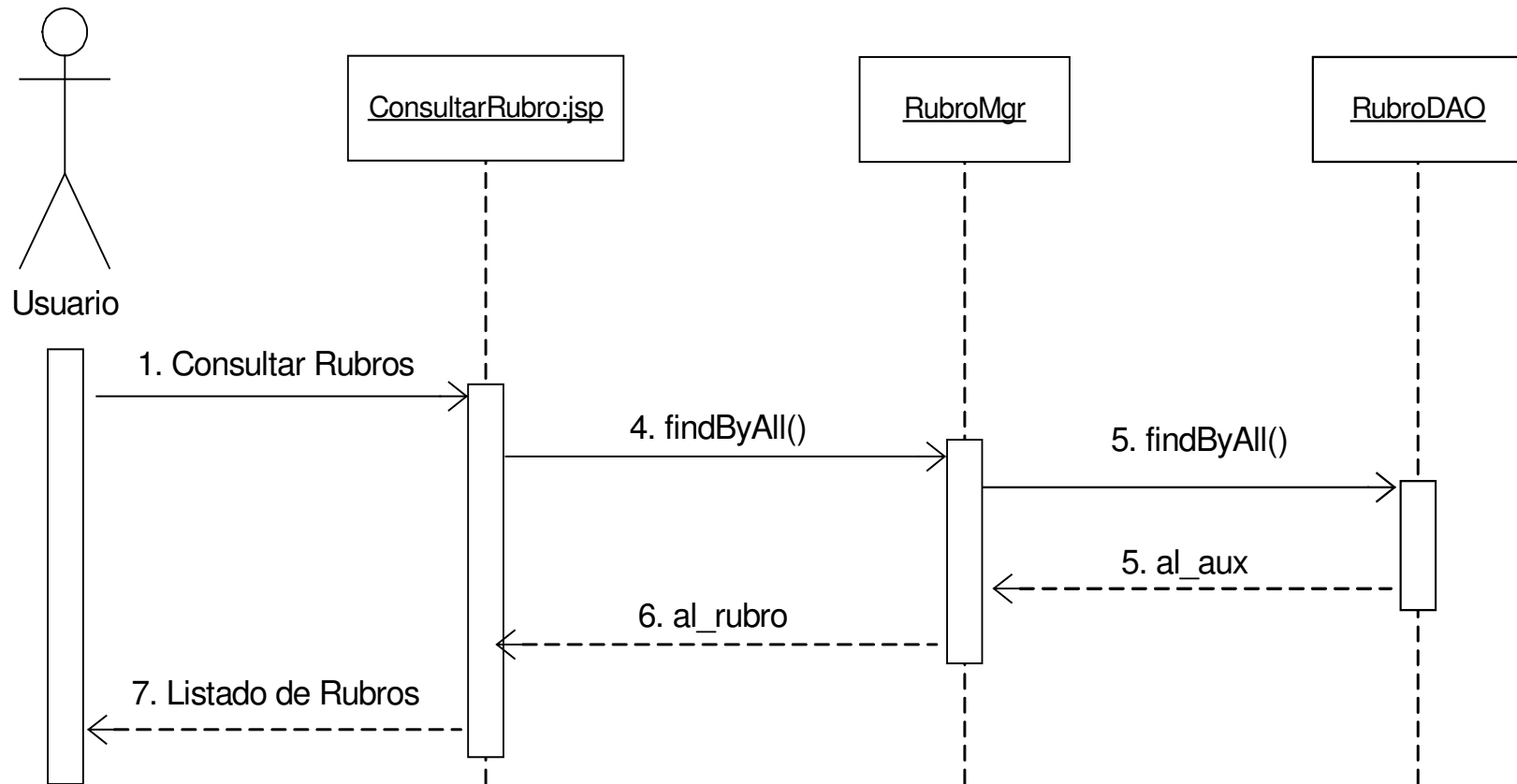


## 11.2.35 Administración de Rubros

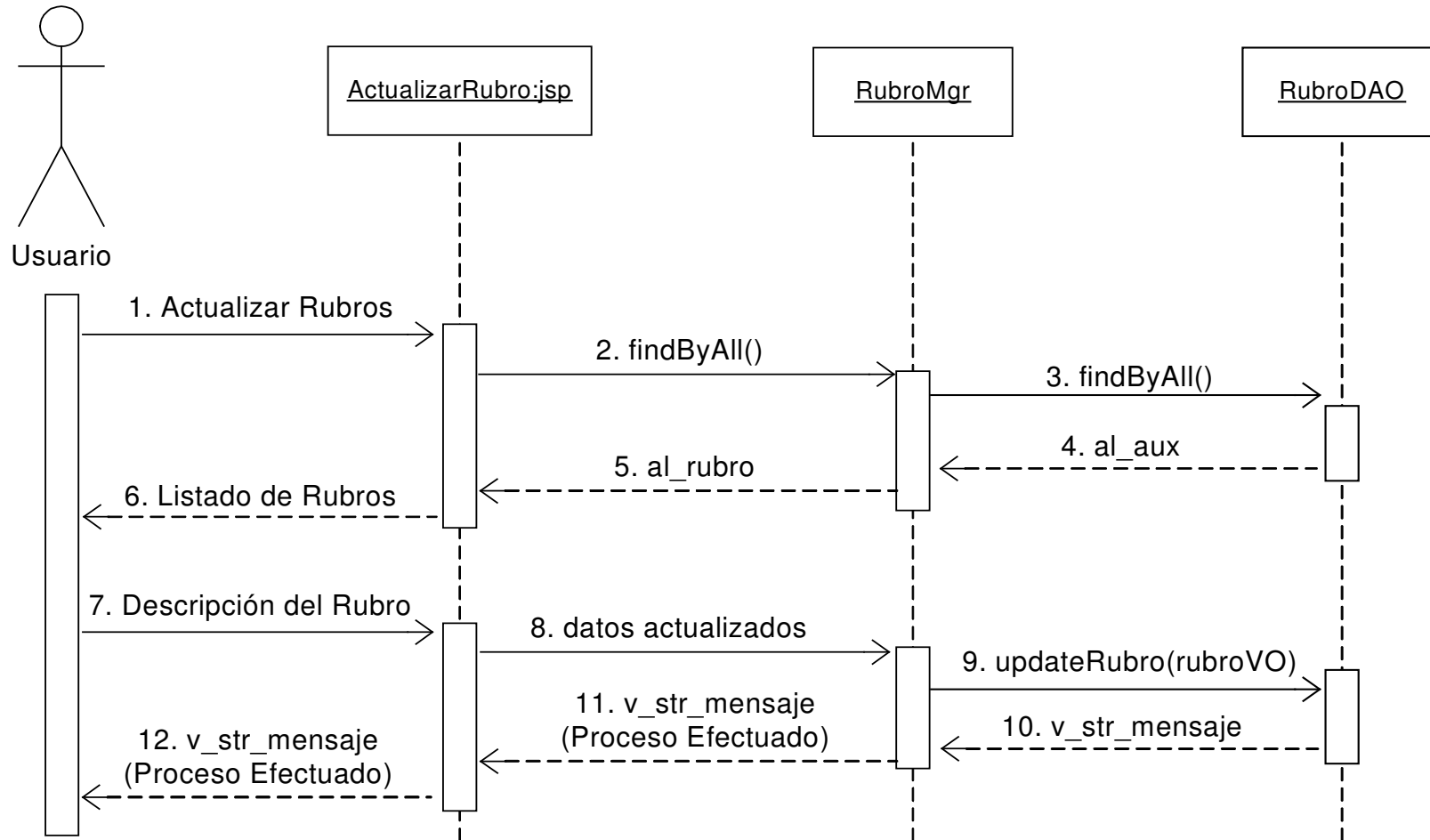
### 11.2.35.1 Registrar Rubro



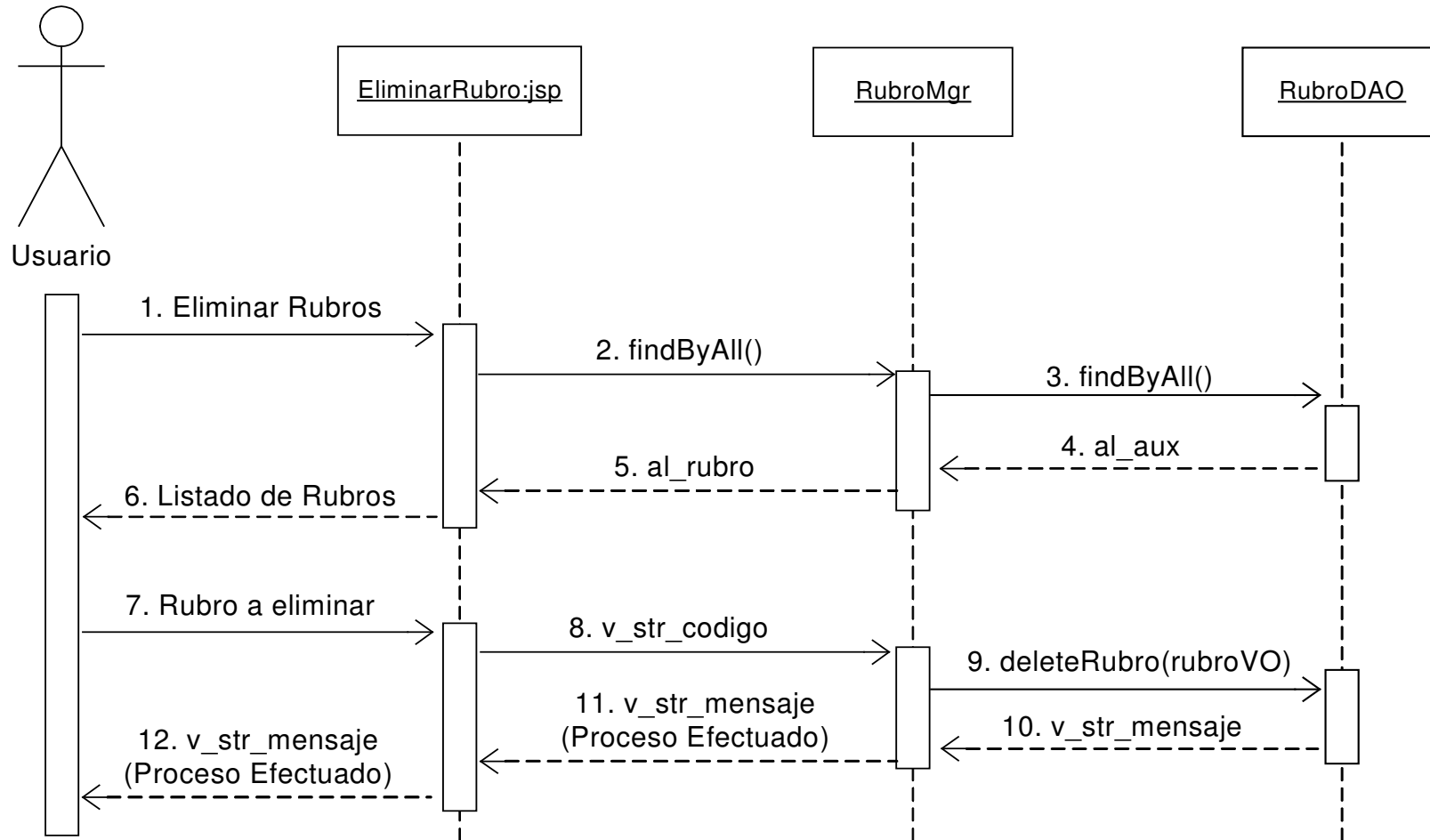
### 11.2.35.2 Consultar Rubro



### 11.2.35.3 Actualizar Rubro

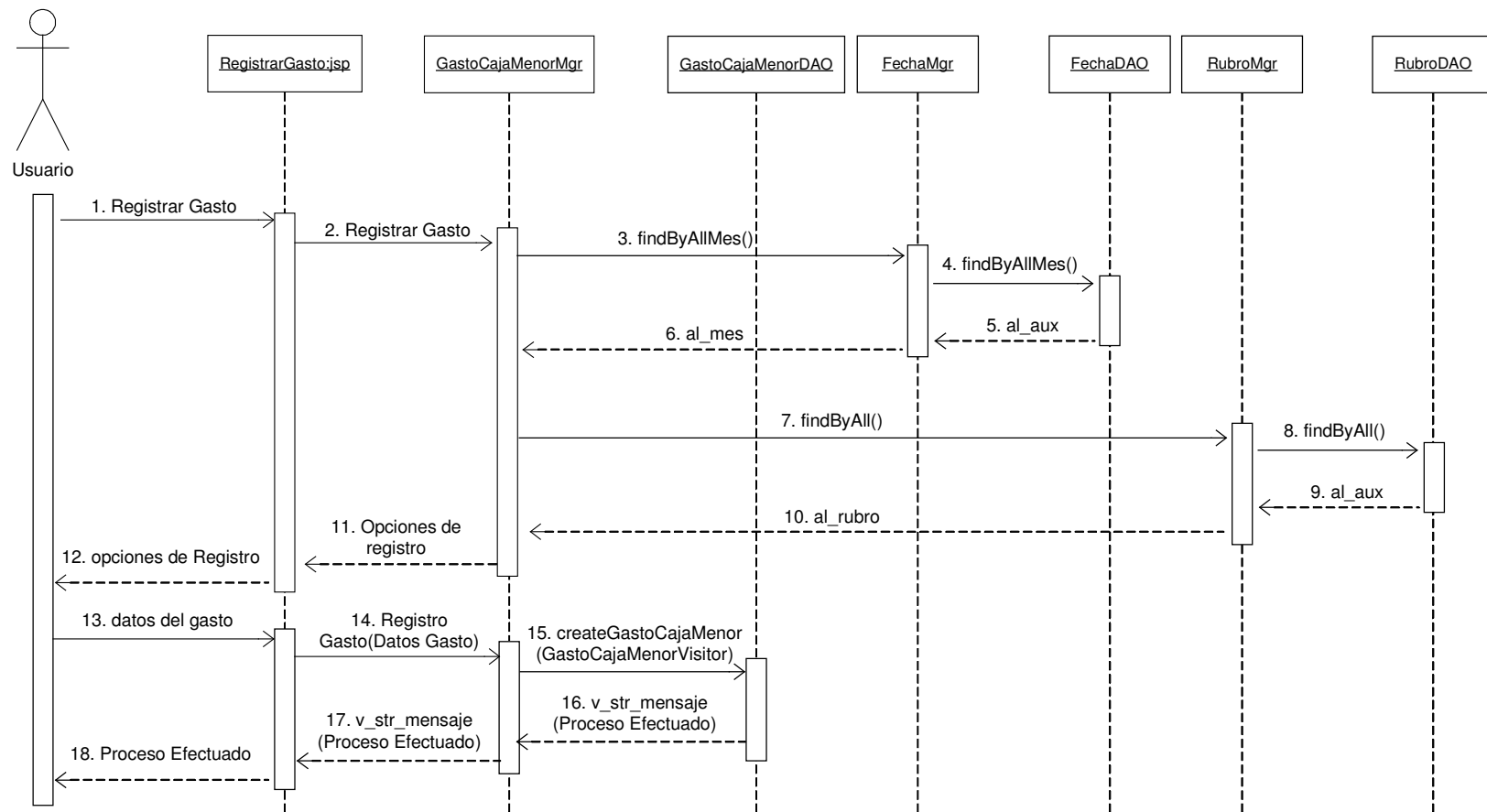


### 11.2.35.4 Eliminar Rubro

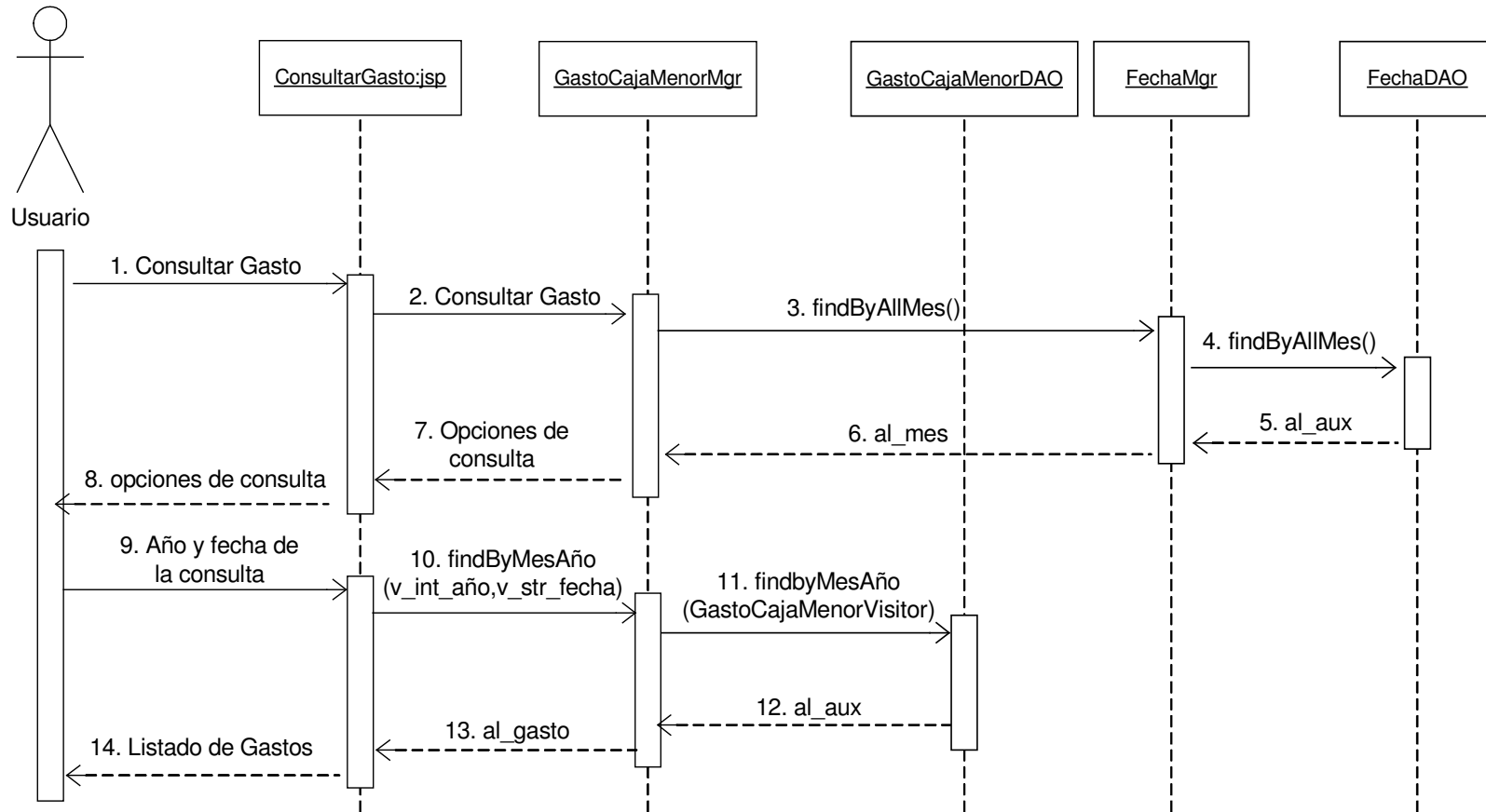


## 11.2.36 Administración de Gastos de Caja Menor

### 11.2.36.1 Registro Gasto Caja Menor

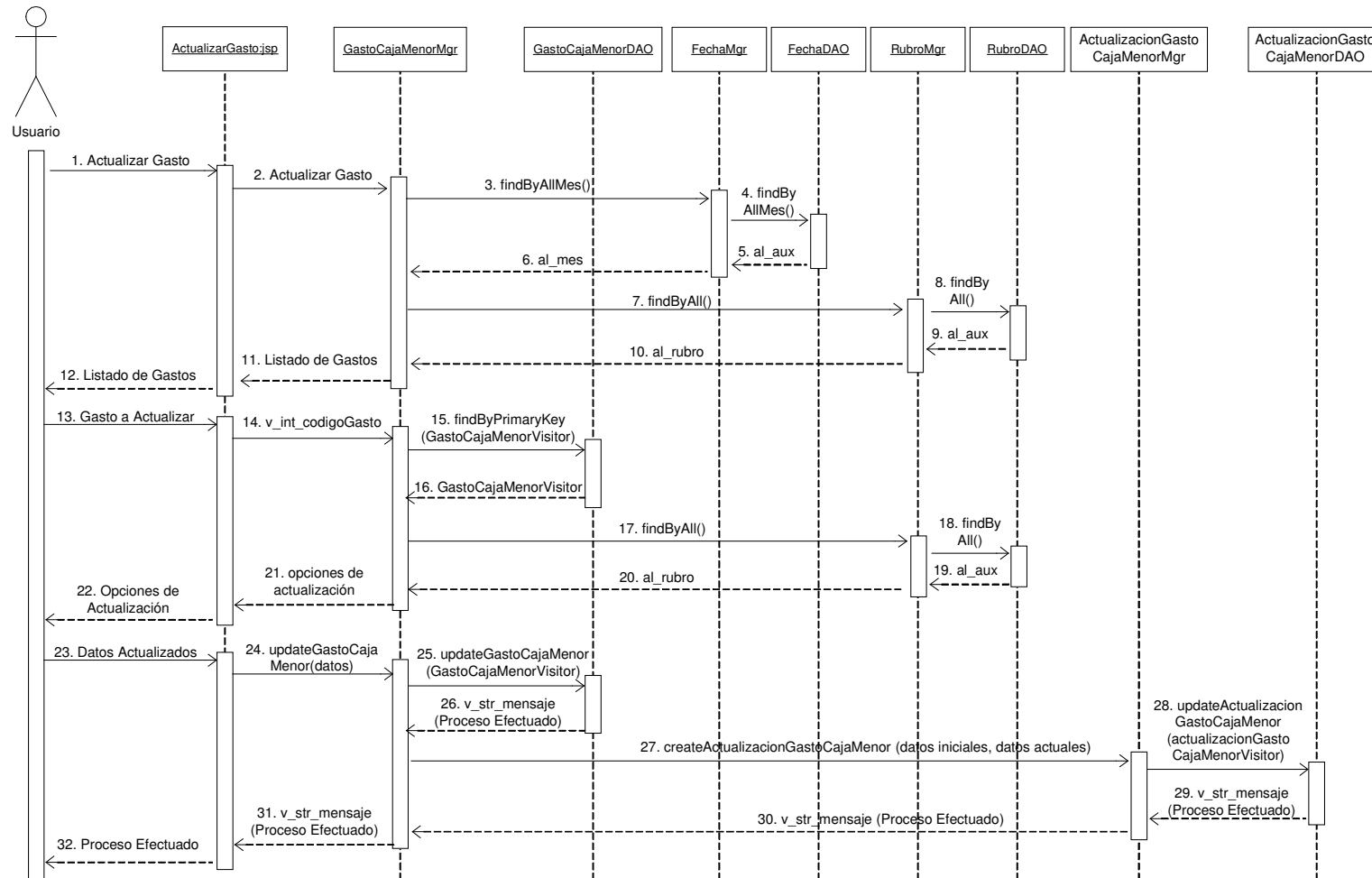


### 11.2.36.2 Consulta Gasto Caja Menor

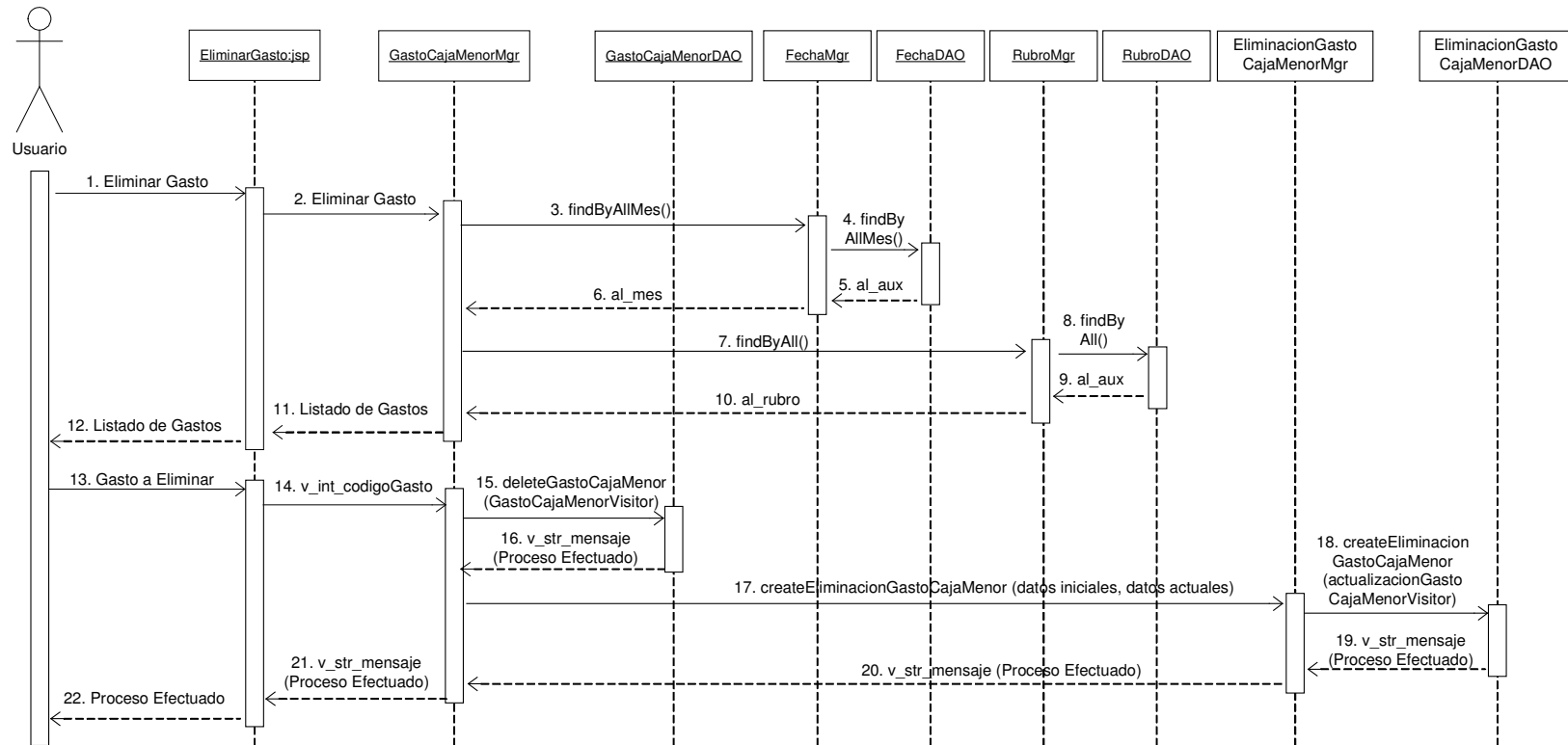




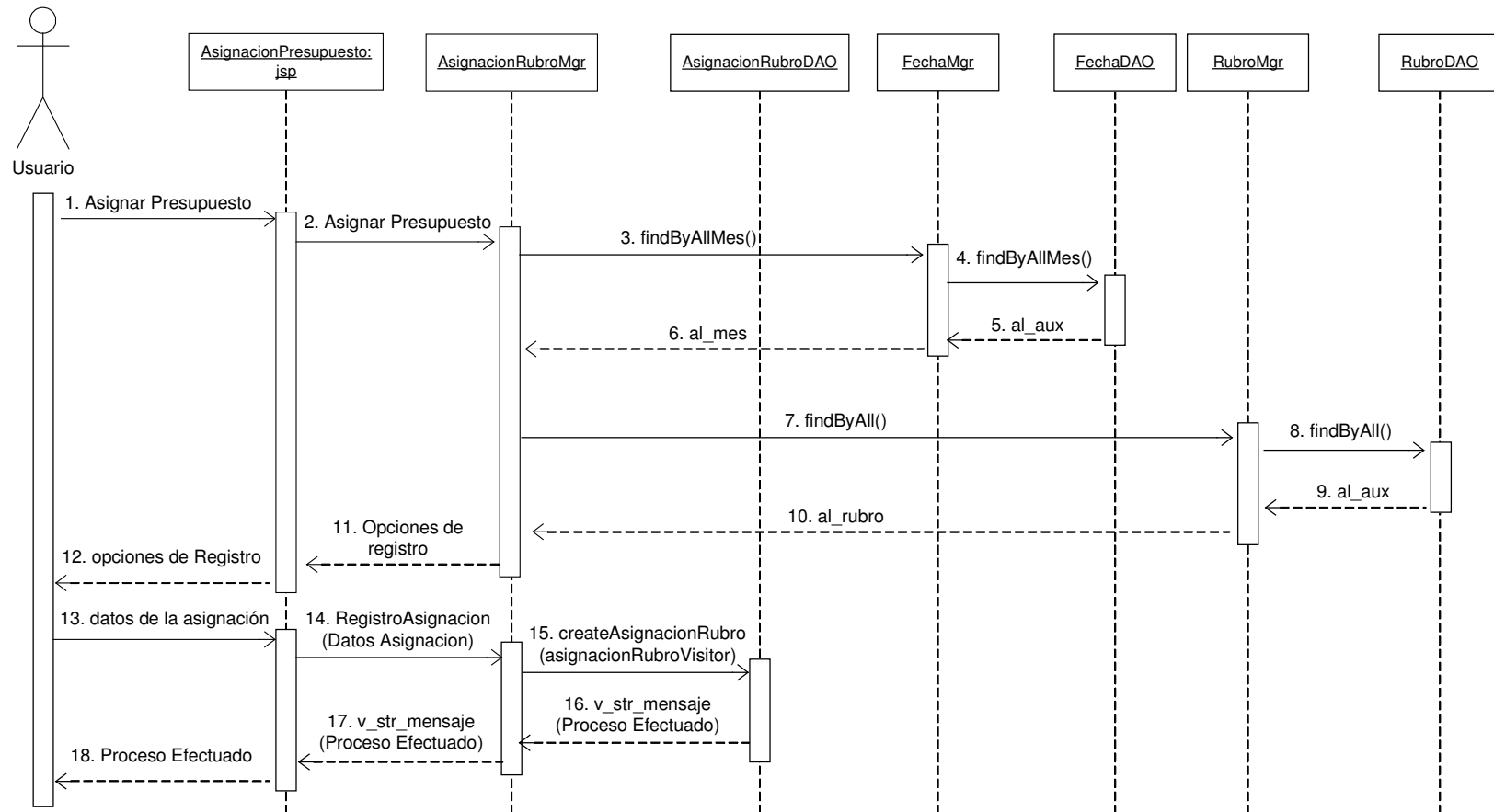
### 11.2.36.3 actualización Gasto Caja Menor



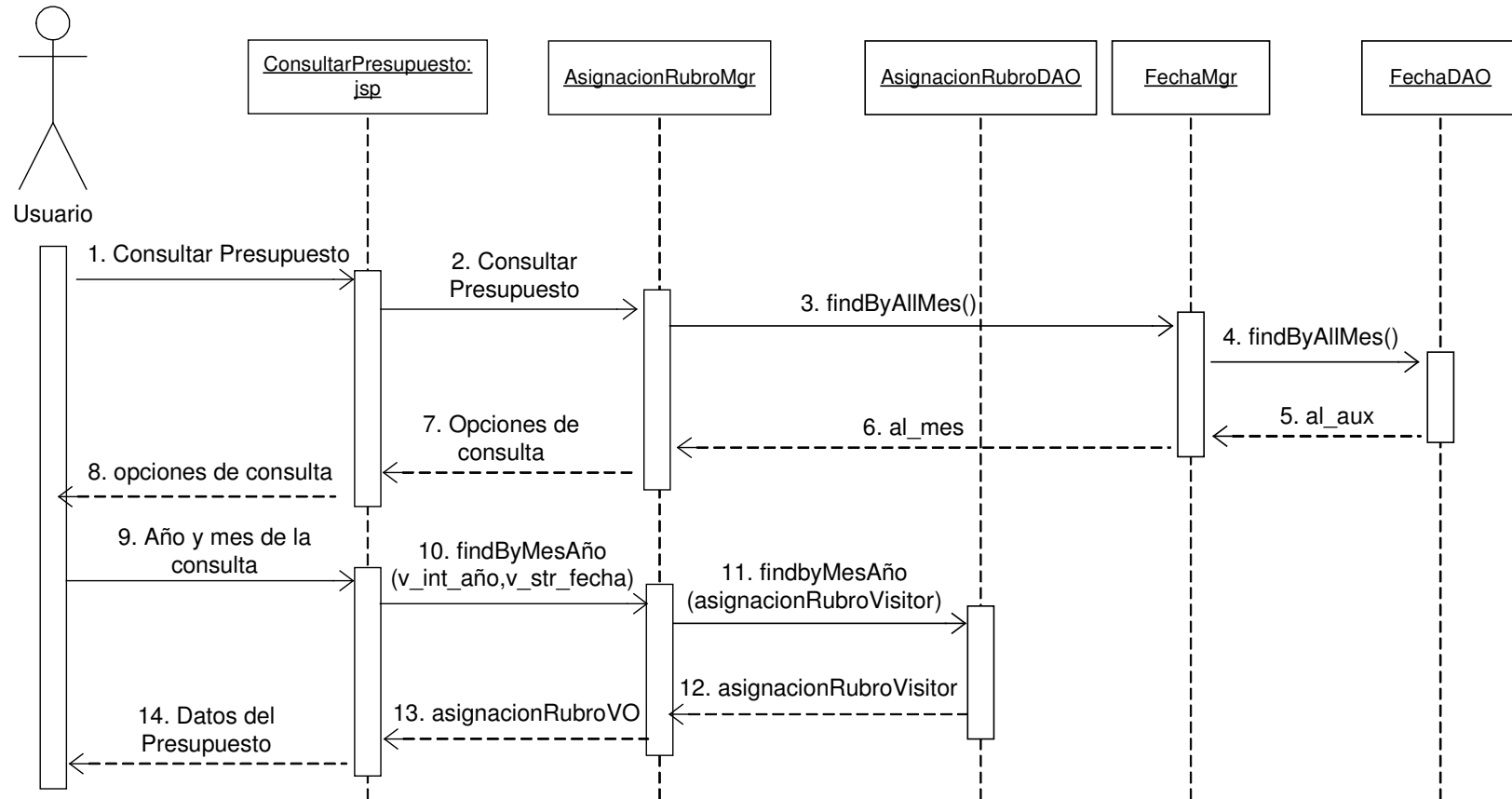
### 11.2.36.4 Eliminación Gasto Caja Menor



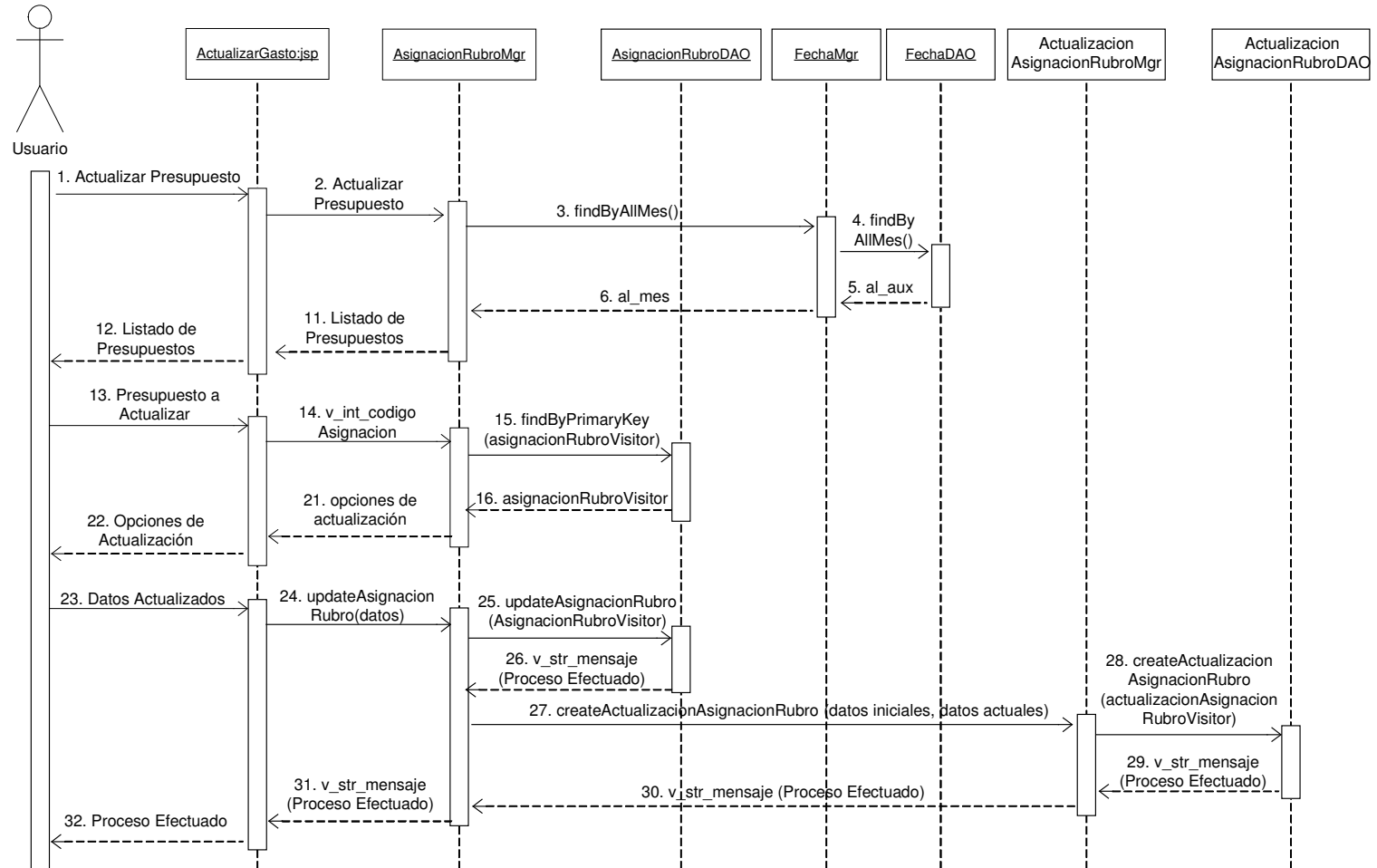
### 11.2.36.5 Asignación Presupuesto Rubro



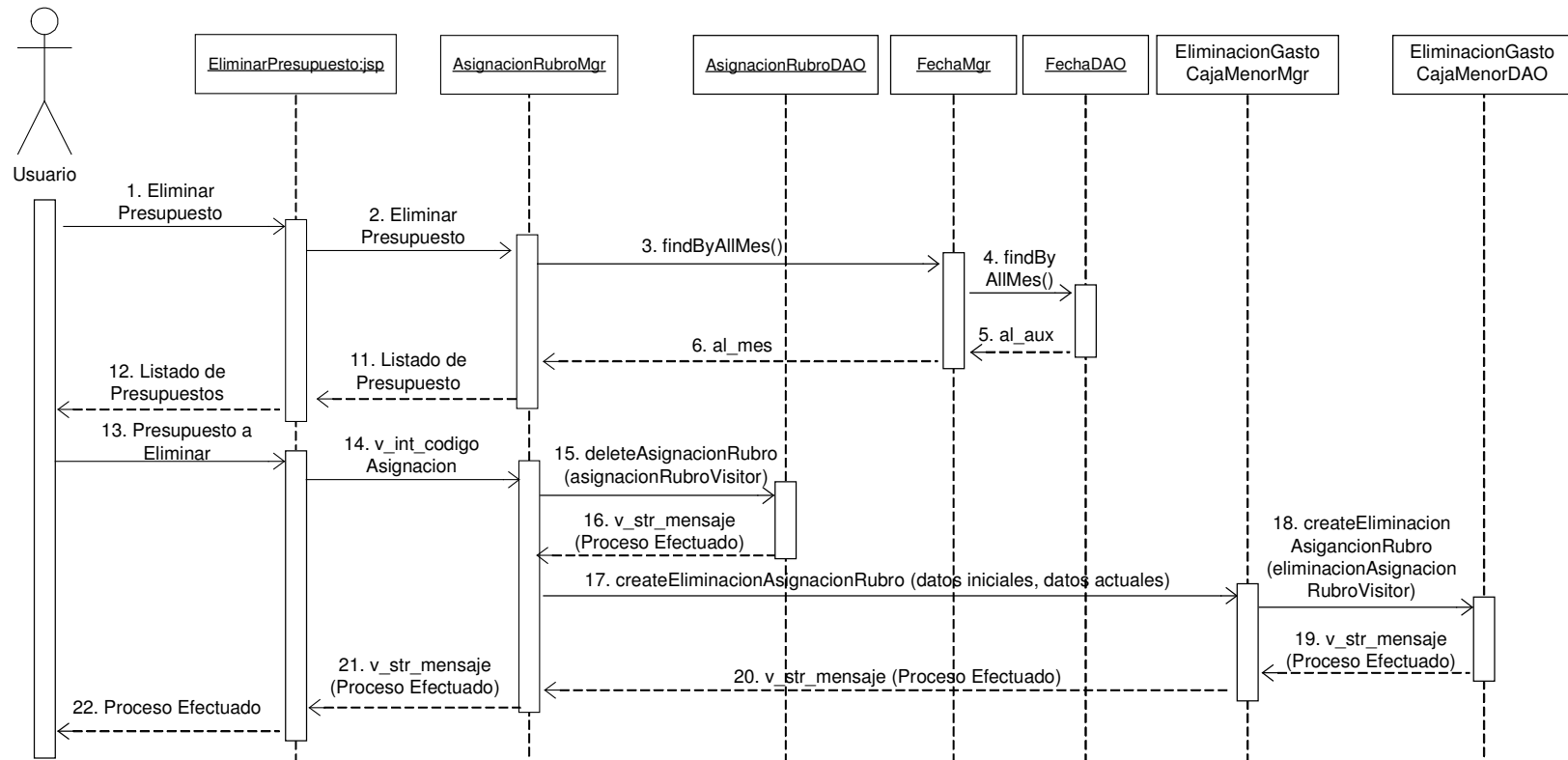
### 11.2.36.6 Consulta Asignación Presupuesto



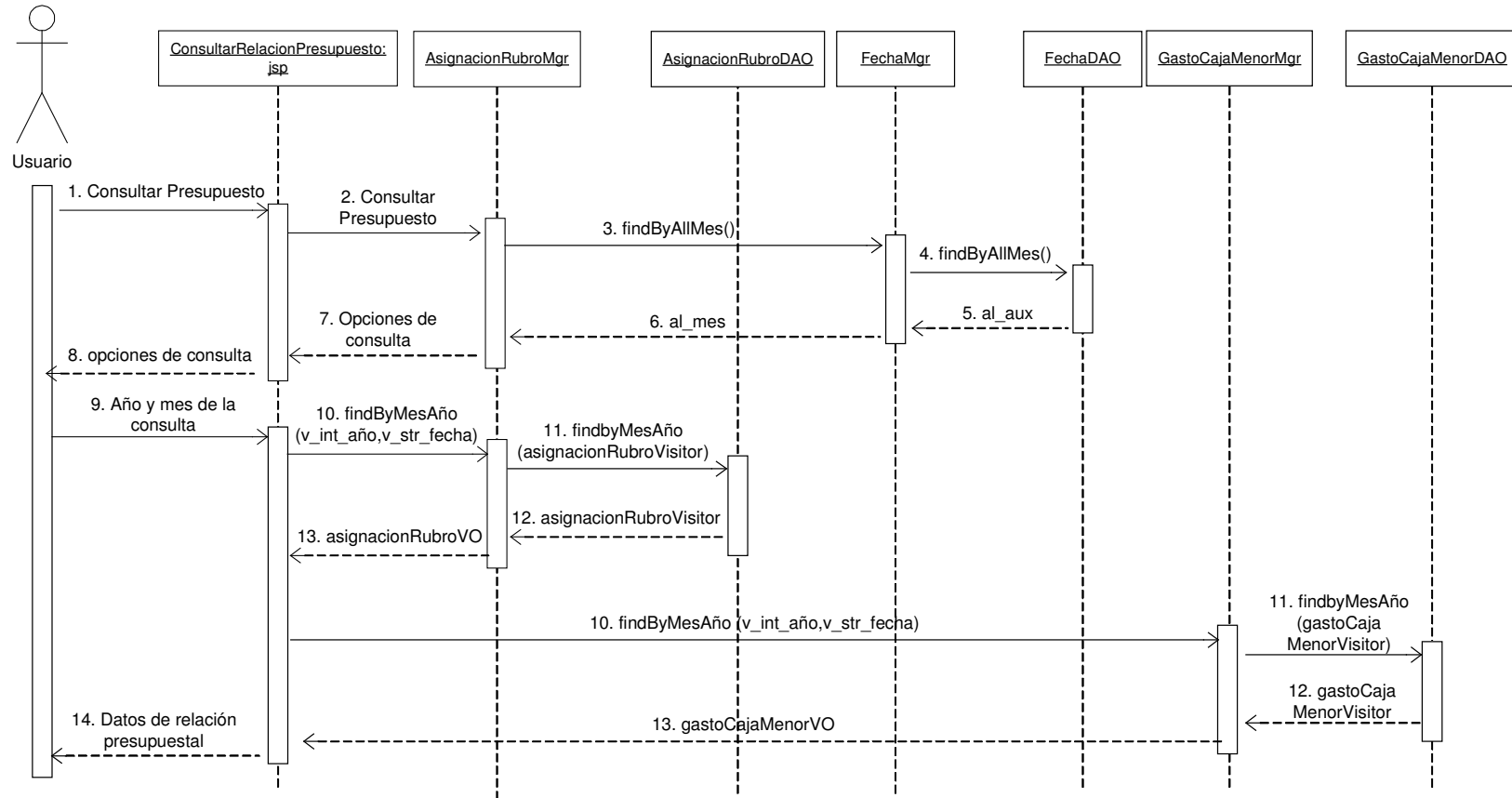
### 11.2.36.7 Actualización Asignación Presupuesto



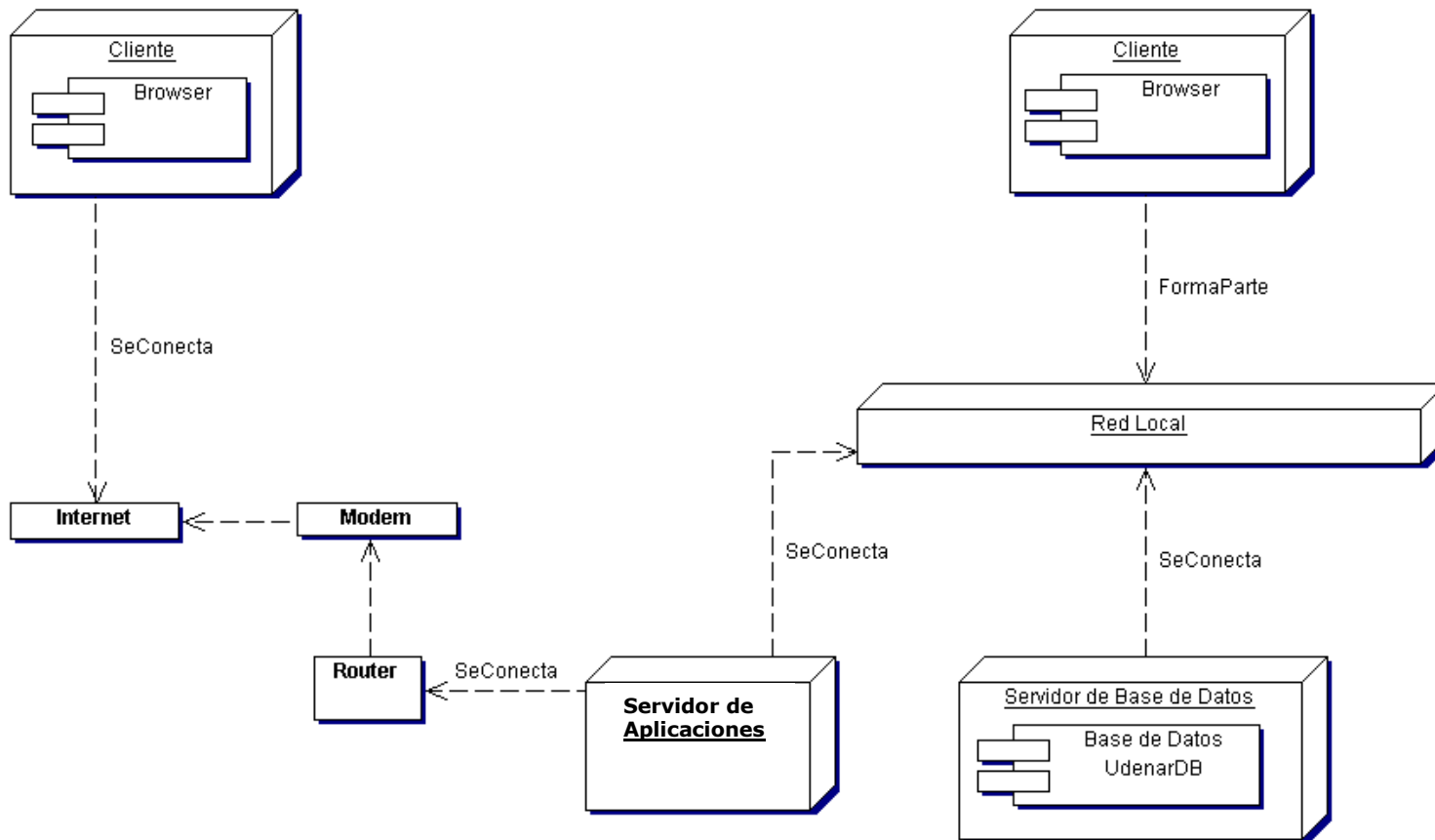
### 11.2.36.8 Eliminación Asignación Presupuesto



### 11.2.36.9 Consulta Relación Presupuestal



## 11.2 DIAGRAMA DE DESPLIEGUE





## 12. CONCLUSIONES

El proyecto de grado SISGAMISI se enfoca en el desarrollo de una aplicación que permita optimizar el manejo de la información de las dependencias de Monitoría Informática, Secretaría y Administración de Inventario de la Universidad de Nariño Extensión Ipiales.

El análisis de los requerimientos del sistema se llevó a cabo mediante la realización de entrevistas a los funcionarios directamente encargados de las dependencias. De igual manera mediante la participación directa en la ejecución de los procesos que se llevan a cabo en la dependencia de monitoría Informática y Administración de Inventario, se logró identificar las necesidades y dar solución a las debilidades que actualmente se presenta.

La Metodología Orientada a Objetos permite visualizar el mundo real a través de objetos que Pueden ser clasificados, descritos, organizados, combinados, manipulados y creados, conllevando a obtener una abstracción que modela el mundo de forma tal que ayuda a entenderlo y gobernarlo mejor.

El Lenguaje de Modelado Unificado (UML) permite visualizar, especificar, construir y documentar los componentes de un sistema. Este es un lenguaje muy expresivo, que abarca todos los panoramas necesarios para desarrollar y estructurar los artefactos necesarios del sistema. Mediante el uso de UML se logró implementar un proceso bien definido que contribuyó a la identificación de que componentes producir, que actividades y que herramientas usar para crearlos y manejarlos, y como usar esos componentes para validar y controlar los procesos finales.

J2EE (Java2 Enterprise Edition) ofrece un conjunto de especificaciones y técnicas que proporcionan soluciones completas, seguras, estables y escalables para el desarrollo, despliegue y gestión de aplicaciones en múltiples niveles de funcionalidad basadas en servidores. Se aprovechó las características propias de esta arquitectura para la construcción de software por componentes capaces de atender las solicitudes de los usuarios e interactuar con otros componentes.

### **13. RECOMENDACIONES**

Aprovechar las características y funcionalidad de la aplicación implementándola a nivel de internet.

Motivar y capacitar a los diferentes usuarios del sistema para lograr un mejor aprovechamiento de las ventajas y servicios que ofrece el software.

Desarrollar nuevos proyectos basados en la metodología utilizada que se puedan integrar a la aplicación para suplir los nuevos requerimientos o necesidades que puedan surgir en la Universidad de Nariño Extensión Ipiales.

## BIBLIOGRAFIA

BODOFF Stephanie, GREEN Dale, HAASE Kim, JENDROCK Eric, PAWLAN Monica, STEARNS Beth. The J2EE™ Tutorial Copyright © 2002 SUN Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303 USA.

BOULEVARD El dorado. Developing J2EE Compliant Applications. SUN Microsystems, Inc. Broomfield, Colorado 80021 USA. Addison Wesley 2002.

MORERO Francisco, Introducción a la Programación Orientada a Objetos. 1999-2000 © Grupo EIDOS.

RUMBAUGH James, JACOBSON Ivar, BOOCH Grady. The Unified Modeling Language Reference Manual. Addison Wesley. December 1999.

[www.java.com](http://www.java.com)

[www.java.sun.com](http://www.java.sun.com)

[www.mysql.com](http://www.mysql.com)

[www.javascript.com](http://www.javascript.com)

[www.uml.org](http://www.uml.org)

[jakarta.apache.org/site/downloads/](http://jakarta.apache.org/site/downloads/)

[java.sun.com/j2ee/1.4/docs/tutorial/doc/](http://java.sun.com/j2ee/1.4/docs/tutorial/doc/)

[www.programacion.com/java/tutoriales/j2ee](http://www.programacion.com/java/tutoriales/j2ee)

[www.monografias.com/trabajos/objetos/objetos.shtml](http://www.monografias.com/trabajos/objetos/objetos.shtml)

[java.sun.com/products/jsp/docs.html](http://java.sun.com/products/jsp/docs.html)

[www.programacion.com/java/tutorial/servlets\\_jsp/](http://www.programacion.com/java/tutorial/servlets_jsp/)