

**SISTEMA DE INFORMACIÓN AUTOMATIZADO PARA EL CONTROL DE
INGRESO Y SALIDA DE TRABAJADORES DE LA EMPRESA DE OBRAS
SANITARIAS EMPOPASTO S.A. E.S.P.**

CARLOS ENRIQUE RICAURTE PARDO

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2008**

**SISTEMA DE INFORMACIÓN AUTOMATIZADO PARA EL CONTROL DE
INGRESO Y SALIDA DE TRABAJADORES DE LA EMPRESA DE OBRAS
SANITARIAS EMPOPASTO S.A. E.S.P.**

CARLOS ENRIQUE RICAURTE PARDO

**Trabajo de grado presentado como requisito parcial para optar por el título
de Ingeniero de Sistemas**

**Directora:
Ing. PAOLA ARTURO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2008**

NOTA DE ACEPTACION

Jurado

Jurado

Director

San Juan de Pasto, Mayo de 2008

“Las ideas y conclusiones aportadas en el Trabajo de Grado son responsabilidad exclusiva del autor.”

Artículo 1º del Acuerdo N.º. 324 de octubre 11 de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño

AGRADECIMIENTOS

En primer lugar, a Dios por ofrecerme el don de la vida y la felicidad de disfrutar toda su creación.

A mi familia por haberme entendido en cada una de mis decisiones acertadas y no. Sobre todo a mi abuelo y mi madre quienes me dieron su apoyo incondicional.

A la Universidad de Nariño, por acogerme en su gran familia, alma mater, donde me formaron académica y personalmente; nunca voy a olvidar los partidos de fútbol.

A mis amigos que estuvieron en las buenas y en las malas, sobre todo en las malas, en el aula de clases, en la cancha de fútbol, en la plaza fuchi, en la calle, en la casa, en las fiestas, en los toques, en el estadio, en todas partes siempre estuvieron.

A mis profesores quienes me compartieron sus conocimientos, ejemplo y sabiduría, especialmente a Vicente Chamorro, Paola Arturo y Ricardo Timarán.

A la familia EMPOPASTO quienes me dieron la gran oportunidad de aprender de forma práctica la verdadera aplicación de la ingeniería de sistemas, además de haber conocido gente maravillosa que me acogió hasta convertirnos en grandes amigos.

Al Dr, Pedro Vicente Obando, quien me ayudo a continuar en la Universidad en un momento muy difícil de mi vida.

DEDICATORIA

A toda mi familia a quienes les debo gran parte de lo que soy y quiero ser, en especial a mi madre Virginia, a mi abuelito Carlos y mis sobrinas Sarita y Angelita, quienes me dan la fuerza de seguir adelante en este mundo lleno de contradicciones, anhelos y esperanzas.

A Dios, fuente de paz y apoyo, que me brinda su mano en aquellos momentos de debilidad.

A mis amigos quienes me rodearon en los momentos difíciles de la vida, me dieron la mano en la alegría y desánimo.

CONTENIDO

INTRODUCCION	17
1. TEMA	19
1.1 TÍTULO	19
1.2 MODALIDAD.....	19
1.3 LÍNEA DE INVESTIGACIÓN.....	19
1.4 ALCANCE Y DELIMITACIONES	19
2. DESCRIPCIÓN DEL PROBLEMA	20
2.1 PLANTEAMIENTO DEL PROBLEMA.....	20
2.2 FORMULACIÓN DEL PROBLEMA.....	21
2.3 SISTEMATIZACIÓN DEL PROBLEMA.....	21
3. OBJETIVOS	22
3.1 OBJETIVO GENERAL	22
3.2 OBJETIVOS ESPECIFICOS.....	22
4. JUSTIFICACION	23
5. ANTECEDENTES	25
6. MARCO TEORICO	26
6.1. ANALISIS Y DISEÑO ORIENTADO A OBJETOS.....	26
6.2. HERRAMIENTA UML	26
6.2.1 Cómo nació UML	27
6.3. JAVA.....	28
6.3.2 Pasado y presente de Java.....	28
6.4. PHP	29
6.5. PostgreSQL	29
6.6. XAJAX	30
6.6. Aplicación Cliente Servidor	31
7. METODOLOGIA	32
7.1 VENTAJAS CICLO DE VIDA EN ESPIRAL	32

7.2. INCONVENIENTES CICLO DE VIDA EN ESPIRAL.....	32
8.1 MODELO DE CASOS DE USO	34
8.1.1 Listado de características.	34
8.1.2 Listado de actores.....	36
8.1.4 Diagrama de casos de uso.	60
8.1.5 Diagramas de Secuencia del Sistema.	64
8.2 MODELO DE ANÁLISIS	71
8.2.1 Listado de conceptos.	71
8.2.2 Diagrama de clases del análisis.....	72
8.3 MODELO DE DISEÑO.....	73
8.3.1 Diagramas de secuencia.....	73
8.3.2 Diagrama de clases del diseño.....	78
8.3.3 Diseño de datos.	80
8.4 MODELO DE IMPLEMENTACION	83
8.4.1 Diagrama de paquetes.....	83
8.5 PRUEBAS DEL SISTEMA	84
CONCLUSIONES	88
RECOMENDACIONES.....	89
BIBLIOGRAFIA	90

LISTA DE TABLAS

Tabla 1. Cronograma de actividades del proyecto.....	33
Tabla 2. Listado de conceptos	71
Tabla 3. Pruebas del sistema.....	84

LISTA DE FIGURAS

figura 1. Mapa de procesos de EMPOPASTO S.A. E.S.P.	24
figura 2. Caso de uso iniciar sesión	37
figura 3. Caso de uso administrar empleado	38
figura 4. Caso de uso eliminar empleado	39
figura 5. Caso de uso adicionar empleado.....	40
figura 6. Caso de uso modificar empleado.....	41
figura 7. Caso de uso estado empleado	42
figura 8. Caso de uso administrar sede	43
figura 9. Caso de uso eliminar sede.....	43
figura 10. Caso de uso adicionar sede.....	44
figura 11. Caso de uso modificar sede.....	44
figura 12. Caso de uso administrar dependencia.....	45
figura 13. Caso de uso eliminar dependencia	46
figura 14. Caso de uso agregar dependencia	47
figura 15. Caso de uso modificar dependencia	47
figura 16. Caso de uso administrar cargo	48
figura 17. Caso de uso eliminar cargo	49
figura 18. Caso de uso adicionar cargo	50
figura 19. Caso de uso modificar cargo	50
figura 20. Caso de uso administrar horario	51
figura 21. Caso de uso eliminar horario	52
figura 22. Caso de uso adicionar horario	53
figura 23. Caso de uso modificar horario	53
figura 24. Caso de uso administrar alertas	54
figura 25. Caso de uso administrar contraseña.....	55
figura 26. Caso de uso administrar reportes	57
figura 27. Caso cerrar sesión	58
figura 28. Caso de uso registrar fecha y hora de entrada y salida.....	59

LISTADO DE DIAGRAMAS

diagrama 1. Ciclo de vida espiral	32
diagrama 2. Casos de uso modulo administrativo	60
diagrama 3. Casos de uso cargos y sedes	61
diagrama 4. Casos de uso dependencias y horario	62
diagrama 5. Casos de uso alertas, contraseña, cerrar sesión, reporte	63
diagrama 6. Casos de uso registrar fecha y hora	63
diagrama 7. Sescuencia iniciar sesión	64
diagrama 8. Secuencia adicionar empleado	64
diagrama 9. Secuencia eliminar empleado	65
diagrama 10. Secuencia modificar empleado	65
diagrama 11. Secuencia estado empleado	66
diagrama 12. Secuencia adicionar horario	66
diagrama 13. Secuencia modificar horario	67
diagrama 14. Secuencia eliminar horario	67
diagrama 15. Secuencia adicionar cargo	68
diagrama 16. Secuencia eliminar cargo	68
diagrama 17. Secuencia modificar cargo	69
diagrama 18. Secuencia generar reporte	69
diagrama 19. Secuencia registrar fecha y hora de entrada y salida	70
diagrama 20. Analisis del sistema	72
diagrama 21. Secuencia agregar sede	73
diagrama 22. Secuencia modificar sede	73
diagrama 23. Secuencia eliminar sede	74
diagrama 24. Secuencia registrar empleado	74
diagrama 25. Secuencia modificar empleado	75
diagrama 26. Secuencia eliminar empleado	75
diagrama 27. Secuencia estado empleado	76
diagrama 28. Secuencia registrar ingreso empleado	76
diagrama 29. Secuencia generar reportes	77
diagrama 30. Clases del diseño	78
diagrama 31. Clases de diseño 2	79
diagrama 32. Diseño de datos 1	80
diagrama 33. Diseño de datos 2	81
diagrama 34. Diseño de datos 3	82
diagrama 35. Diseño de datos 4	82
diagrama 36. Diagrama de paquetes	83
diagrama 38. Paquetes de clase	83

LISTA DE ANEXOS

ANEXO A. MANUAL DEL SISTEMA	93
ANEXO B. MANUAL DE USUARIO	101

GLOSARIO

ALERTA: Parámetro que permite visualizar de forma gráfica que tan tarde llegó o que tan temprano salió un empleado de la Empresa

CARGO: Nombre de Obligación de hacer o cumplir algo.

CICLO DE VIDA EN ESPIRAL: El modelo espiral de los procesos software es un modelo del ciclo de meta-vida. En este modelo, el esfuerzo de desarrollo es iterativo. Tan pronto como uno completa un esfuerzo de desarrollo, otro comienza.

CLIENTE – SERVIDOR: Un servidor es una aplicación que ofrece un servicio a usuarios de Internet; un cliente es el que pide ese servicio. Una aplicación consta de una parte de servidor y una de cliente, que se pueden ejecutar en el mismo o en diferentes sistemas.

CONTRASEÑA o PASSWORD: Palabra o signo secreto que permite el acceso al Sistema, a alguien o a un grupo de personas.

DEPENDENCIA: Sección o colectividad subordinada a la EMPOPASTO SA ESP

E.S.P.: Empresa de Servicios Públicos

EMPLEADO: Persona que desempeña un empleo de EMPOPASTO SA ESP. Pertenece a una Dependencia, tiene un Cargo, trabaja en una Sede y cumple un Horario.

EMPOPASTO: Empresa de Obras Sanitarias de Pasto

HORARIO: Cuadro indicador de las horas en que deben ejecutarse determinadas actividades.

JAVA: Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

PHP: (Hypertext Pre-processor) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

POSTGRESQL: Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD.

S.A.: Sociedad Anónima

SEDE: Lugar donde tiene su domicilio la empresa EMPOPASTO SA ESP (MIJITAYO, CASONA, CENTENARIO, SAN FELIPE)

UML: Lenguaje de Modelado Unificado

USUARIO: Dicho de una persona: Que tiene derecho de usar del sistema con cierta limitación.

XAJAX: Ajax es una tecnología que utiliza a su vez otra combinación de tecnologías, como XML y Javascript, para realizar peticiones de contenido o computación de servidor sin tener que recargar la página en la que está el usuario. Es una tecnología que permite una nueva gama de aplicaciones interactivas en la web, mucho más ricas y rápidas, dado que no se precisa recargar todo el contenido de una página para realizar peticiones al servidor.

RESUMEN

El factor más importante de toda empresa es el recurso humano, y por otra parte el control es una etapa primordial en la administración, pues aunque una empresa cuente con magníficos planes, una estructura organizacional adecuada y una dirección eficiente, el ejecutivo no puede verificar cuál es la situación real de la organización si no existe un mecanismo que se cerciore e informe si los hechos van de acuerdo con los objetivos.

Para esto se ha desarrollado una herramienta sistematizada permite solucionar el problema registrar, administrar y generar reportes con la información de los ingresos y salidas de los empleados con el fin de obtener un control eficiente y eficaz de todo el personal. Esta información sirve de apoyo para la toma de decisiones con respecto al área de talento humano de esta entidad.

El sistema registra la hora de ingreso y salida de los empleados de la Empresa de Obras Sanitarias de Pasto EMPOPASTO SA ESP pasando el carné por el lector de código de barras identificando el nombre del usuario, fecha, hora y en qué sede marcó su registro. Por otra parte la aplicación permite administrar empleados, sedes, dependencias, cargos, alertas, contraseñas, horarios y lo más importante genera reportes de manera personalizada y parametrizable que ayuda a controlar el recurso humano de manera gráfica, segura, eficiente y eficaz.

El proyecto fue realizado con herramientas que facilitan el desarrollo del proyecto de manera sistemática y minimizando el porcentaje de error. Tales herramientas son: UML la cual ayuda a esquematizar toda la aplicación y logra identificar muchos errores que de no haberse encontrado en la etapa de análisis hubieran sido cruciales en la etapa de desarrollo, atrasando el cronograma de actividades. Otra de las herramientas es el ciclo de vida en espiral, el cual guía con su metodología un mapa de las actividades a ejecutar para desarrollar el proyecto asegurando la satisfacción del cliente debido a que el está en constante interacción en todas las etapas del proyecto.

ABSTRACT

The most important factor of every company is the human resource, and moreover the control is a fundamental stage in the administration, so although a company has a magnificent plans, an appropriate organizational structure and an efficient leadership, the worker can't verify what is the real situation of the organization if doesn't exist a mechanism to ascertain and report if the facts and the objectives are going in the same way.

To this has developed a systematic tool which make possible to resolve the problem register, manage and generate reports with the information of the incomes and expenses of employees in order to obtain an efficient and effective control of all staff. This information will serve an support for decision-making whit regard to the human talent division of this entity.

The system records the time of entry and way out of employees of the Company EMPOPASTO SA EMP whit only passing by the bar code reader the ID card to identify the usser name, date, time and in wath site was marked his register. Moreover, the implementation will manage employees, offices, dependences, positions, alerts, passwords, schedules and most importantly it allow generate personalized reports and parametrized that helps control the human resource in graphically way and resource, efficient and effective.

The proyect was developed with tools that facilitated the development of the proyect in a systematic way for minimizing the error porcentaje. Such tools are: UML which helped to do diagrams of entire application and helped to archieve to identify many errors that, if hadn't been found during the analysis, it had been crucial in the development stage, delaying the activities schedule. Another tool was the life cycle in spyral, which with it's methodology led to activities map to do and develop the proyect ensuring customer satisfaction due that this one is in constant interaction at all stage of proyect.

INTRODUCCION

Gracias a la ayuda de la tecnología informática y las metodologías para el desarrollo de Software de la Ingeniería de Sistemas se pueden crear herramientas que permiten mejorar los procesos en cualquier tipo de empresa. En el caso de la Empresa de Obras Sanitarias de Pasto EMPOPASTO S.A. E.S.P., el control del personal es una de las actividades más importantes para su buen funcionamiento y productividad, que necesita ser sistematizado.

El proyecto surge del problema: ¿Cómo optimizar el sistema de control de ingreso y salida del personal de la empresa de Obras Sanitarias EMPOPASTO S.A. E.S.P. de forma segura, eficiente y eficaz?

El problema lleva a generar un objetivo así: Generar un sistema de control con el fin de maximizar la productividad de todo el personal de la Empresa de Obras Sanitarias EMPOPASTO S.A. E.S.P. y así poder brindar un mejor servicio a la comunidad.

El sistema de información para el control de ingreso y salida de trabajadores de la empresa de obras sanitarias EMPOPASTO S.A. E.S.P., será una herramienta de apoyo para brindar un mejor servicio a la comunidad gracias a la utilización del cien por ciento de la productividad humana dentro de los horarios establecidos, en un contexto de cultura de autocontrol acorde a los procesos de una empresa que brinda servicios públicos.

El desarrollo del producto software se realizará utilizando la metodología del ciclo de vida en espiral el cual permite llevar un control detallado de las tareas del proyecto en cada una de sus etapas y además es retroalimentado por el usuario final quien esta presente en todas las etapas del proyecto, lo que permite asegurar un producto final que satisfaga las expectativas esperadas del cliente.

La línea de investigación del proyecto fue Software y manejo de información, y la modalidad del trabajo de grado fue la de pasantía.

Para el análisis y diseño del sistema fue utilizada la herramienta de Lenguaje Unificado de Modelado (UML) ya que ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como lógica del negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Para el desarrollo del proyecto se utilizó PHP, XAJAX, JAVA, POSTGRES, APACHE. Lo que permite realizar una aplicación CLIENTE – SERVIDOR.

Se presenta el siguiente documento que contiene el desarrollo del proyecto, con la descripción del problema, los objetivos, la metodología para el desarrollo de la solución y los resultados obtenidos.

1. TEMA

1.1 TÍTULO

SISTEMA DE INFORMACIÓN AUTOMATIZADO PARA EL CONTROL DE INGRESO Y SALIDA DE TRABAJADORES DE LA EMPRESA DE OBRAS SANITARIAS EMPOPASTO S.A. E.S.P.

1.2 MODALIDAD

El presente proyecto de trabajo de grado corresponde a la modalidad estipulada como: PASANTIA.

1.3 LÍNEA DE INVESTIGACIÓN

El proyecto pertenece a la línea de investigación: SOFTWARE Y MANEJO DE INFORMACIÓN.

1.4 ALCANCE Y DELIMITACIONES

El sistema de información se encargará de capturar, procesar y administrar la información del Ingreso - Salida de los ciento sesenta y cinco trabajadores de la empresa de obras sanitarias EMPOPASTO S.A. E.S.P. en sus diferentes sedes: Mijitayo, Centro, Centenario y San Felipe; el sistema se encargará de generar reportes personalizados y datos estadísticos que faciliten el control de manera segura y confiable, lo cual servirá para facilitar el seguimiento individual y en grupo de los funcionarios con respecto al cumplimiento del horario establecido en cada área de la empresa. La empresa tendrá con este aplicativo la información acertada para la toma de decisiones, estudio de tiempos y movimientos.

2. DESCRIPCIÓN DEL PROBLEMA

La empresa de obras sanitarias EMPOPASTO S.A. E.S.P. cuenta con ciento sesenta y cinco empleados en las diferentes sedes (Mijitayo, Centenario, Centro y San Felipe), que deben registrarse por lo menos cuatro veces al día en el control de ingresos y salidas.

Para el área de Talento Humano controlar y realizar el seguimiento personalizado de la hora de ingreso y salida de los empleados, dependiendo del horario de cada uno de ellos se convierte en un proceso muy dispendioso. De igual manera la verificación de los permisos y las justificaciones correspondientes no supe las necesidades de control de una empresa proyectada a mejorar día a día sus procesos.

2.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad la empresa EMPOPASTO S.A. E.S.P cuenta con un sistema que no supe las necesidades de control de ingreso y salida de empleados para la sección de Talento Humano, tampoco sirve como herramienta colaborativa para la liquidación de nómina.

- El sistema actual no brinda total credibilidad. Los empleados pueden marcar la hora de ingreso y salida con el carné del compañero sin control alguno.
- El sistema actual no maneja la información generada en las sedes (Mijitayo, Centenario, San Felipe y Centro) de manera centralizada, sino independientemente, lo que dificulta la recolección de información.
- El sistema actual no genera reportes ni algún otro dato estadístico, simplemente lista las horas de ingreso y salida de los empleados
- El sistema actual no tiene en cuenta que hay empleados con horarios diferentes; esto no permite hacer un seguimiento eficiente del personal.
- El sistema actual no es seguro, puesto que no maneja usuarios ni contraseñas para acceder a el y permite adicionar, modificar y eliminar registros sin ninguna restricción.
- El sistema actual no interactúa con el sistema de nómina.
- El sistema actual no tiene contemplado los periodos de tiempo adicional como horas extras diurnos y nocturnos, dominicales y festivos.
- El sistema actual funciona independientemente de los demás sistemas de la empresa aunque maneje los mismos datos, lo que dificulta su integración.

Esta situación dificulta la detección de retrasos sin causa justificada lo que va en contra del rendimiento de EMPOPASTO S.A. E.S.P. e infringe el reglamento interno, generando pérdidas económicas significativas para la empresa debido a su poca eficiencia y baja producción por ausencia y retraso del personal.

2.2 FORMULACIÓN DEL PROBLEMA

¿Cómo optimizar el proceso de control de ingreso y salida del personal de la empresa de Obras Sanitarias EMPOPASTO S.A. E.S.P de forma segura, eficiente y eficaz?

2.3 SISTEMATIZACIÓN DEL PROBLEMA

¿Cómo facilitar la tarea de encontrar faltas al reglamento interno de trabajadores y empleados públicos de EMPOPASTO S.A. E.S.P. con respecto al cumplimiento de los horarios establecidos?

¿Qué políticas deberán adoptarse para que los trabajadores de EMPOPASTO S.A. E.S.P. no se puedan suplantar a la hora de marcar el registro del sistema de control de ingresos y salidas de la empresa?

¿Cómo generar reportes claros de la hora de ingreso y salida de los trabajadores de manera inmediata y confiable que faciliten el seguimiento del personal, evaluación y la toma de decisiones al área de Talento Humano?

¿Cómo mantener segura la información del ingreso y salida de empleados a la empresa, para que solo los usuarios autorizados puedan tener acceso al sistema para realizar modificaciones y generar reportes?

¿Cómo tener un espacio donde el empleado pueda justificar su retraso o salida prematura de la empresa de obras sanitarias EMPOPASTO S.A. E.S.P?

¿Cómo optimizar el sistema actual de control de ingreso y salida de empleados de EMPOPASTO S.A. E.S.P.?

¿Cómo integrar el sistema de entrada y salida de empleados con los sistemas internos de la empresa?

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Generar un sistema de información para el control de ingreso y salida de trabajadores de EMPOPASTO S.A. E.S.P., con el fin de maximizar la productividad de todo el personal y así poder brindar un mejor servicio a la comunidad.

3.2 OBJETIVOS ESPECIFICOS

- Manejar el proceso de ingreso y salida de empleados que se desarrolla actualmente en la empresa de obras sanitarias EMPOPASTO S.A. E.S.P
- Generar reportes actualizados del ingreso y salida del personal de acuerdo a las necesidades del área de Talento Humano de EMPOPASTO S.A. E.S.P.
- Maximizar la productividad de la empresa para suministrar un mejor servicio a la comunidad, basado en el respeto y cumplimiento de las normas internas de EMPOPASTO S.A. E.S.P.
- Establecer políticas de seguridad para que el control del personal sea confiable y consistente.
- Forjar una cultura de cumplimiento de los horarios establecidos por la empresa de obras sanitarias EMPOPASTO S.A. E.S.P.

4. JUSTIFICACION

EMPOPASTO S.A. E.S.P. es una de las entidades más importantes en el departamento de Nariño que proporciona un servicio vital, básico para los hogares y el desarrollo de la región; por esta razón ha entrado al proceso de certificación de calidad, cuya política es “Entregar a los clientes agua potable, de manera continua y en cantidades suficientes y prestar el servicio de recolección de aguas residuales a precio equitativo, atendiendo en forma oportuna y efectiva los requerimientos de servicios asociados, para lo cual cuenta con un personal competente, **capacidad tecnológica y procesos normalizados con enfoque de mejoramiento continuo**”.

El sistema de información automatizado para el control de ingreso y salida de trabajadores de la empresa de obras sanitarias EMPOPASTO S.A. E.S.P. es una herramienta de ayuda para brindar un mejor servicio a la comunidad gracias a la utilización del cien por ciento de la productividad humana dentro de los horarios establecidos, en un contexto de cultura de autocontrol acorde a los procesos de una empresa que brinda servicios públicos.

Conjuntamente va a servir al área de Talento Humano para agilizar los procesos de detección de faltas al reglamento con respecto a los horarios del personal, factor esencial para obtener la certificación de calidad teniendo en cuenta que el recurso humano es el factor más importante en toda empresa y el desempeño de los trabajadores es directamente proporcional a la productividad.

Como parte del proceso de certificación se necesita de herramientas que ayuden al control de calidad, es por esta razón que esta herramienta software encaja dentro del proceso de sistemas de información. Además ayuda a realizar control del personal el cual es el recurso más importante con el que cuenta la empresa.

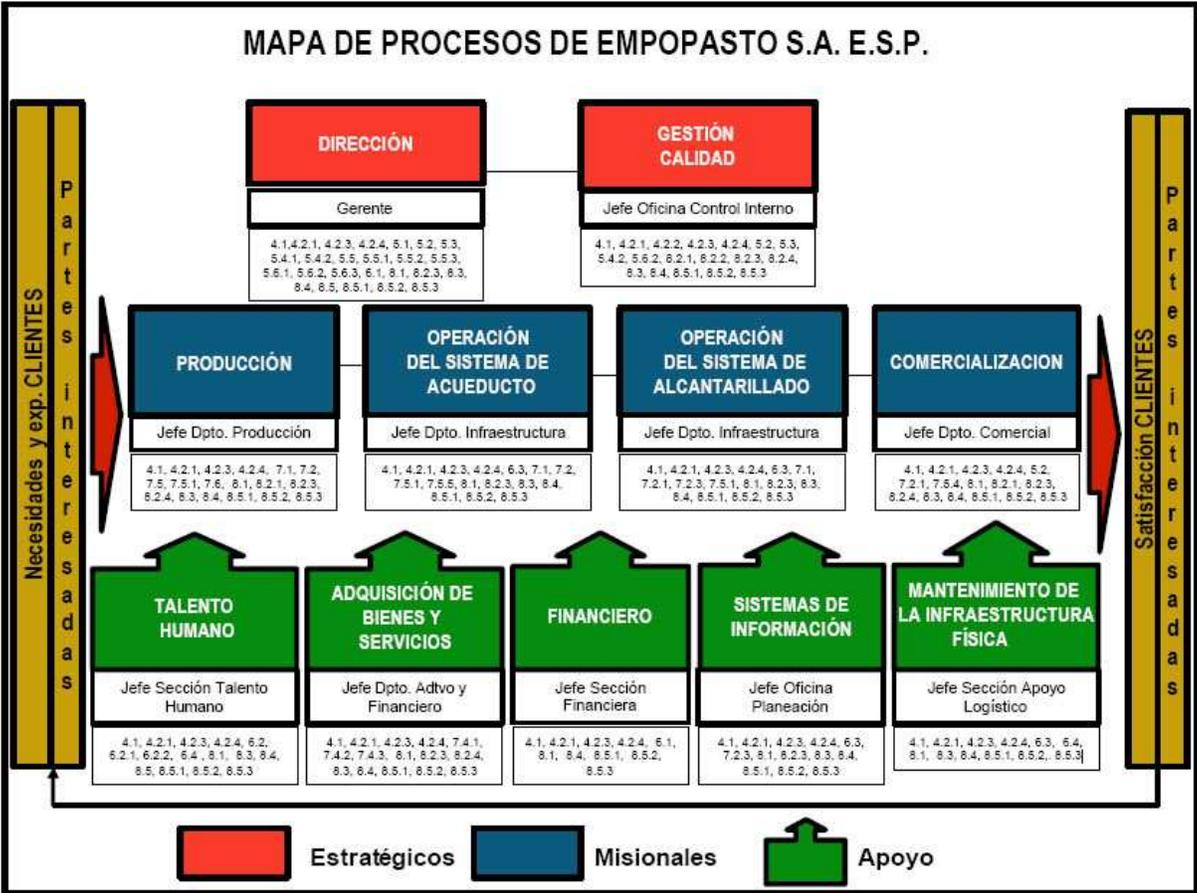


Figura 1. Mapa de procesos de EMPOPASTO S.A. E.S.P.

5. ANTECEDENTES

La empresa de obras sanitarias de Pasto EMPOPASTO S.A. E.S.P. ha tomado como primera medida preventiva para controlar la infracción de horarios de los empleados, leyes acordadas en el contrato de trabajo sancionatorias a aquellos trabajadores que incumplan los horarios establecidos.

Por lo anteriormente expuesto, se implementó un programa creado en el sistema gestor de bases de datos Access llamado SYREM, que en un principio se calculaba facilitaría el control del ingreso y salida de los empleados. Dicho sistema no contempló que fácilmente los trabajadores pueden registrar a otros compañeros utilizando el carné sin ningún tipo de control.

Este sistema tampoco tiene en cuenta los distintos horarios del personal, los reportes se limitan a informar los retrasos sin tomar en cuenta los diferentes segmentos de labores que hacen que determinados grupos manejen otro tipo de hora de entrada y salida, por ejemplo los encargados de atender emergencias, trabajos de mantenimiento, imprevistos, etc.

Existe un alto grado de inseguridad debido a que el programa no posee control de usuario y fácilmente es asequible por carecer de contraseña de entrada, lo que permite acceder a este libremente corriendo el riesgo de presentarse una alteración. De igual manera la aplicación no está unificada, en el momento las sedes de Mijitayo, Centenario, San Felipe y Centro manejan el programa de manera independiente lo que impide el control y la ejecución de reportes.

El aplicativo existente no está diseñado para colaborar en la liquidación de la nómina del personal. Su gestión se limita a reportar la hora de entrada, salida y retrasos generados por los trabajadores. En los casos que por motivos laborales los empleados permanecen más tiempo del horario estipulado en la empresa, realizando tareas extras el sistema no lo toma como horas extras ni tampoco existe el módulo para interactuar con la nómina.

6. MARCO TEORICO

6.1. ANALISIS Y DISEÑO ORIENTADO A OBJETOS

Actualmente, el enfoque orientado a objetos forma parte de la tendencia principal para el desarrollo de software, porque ha demostrado ser válido en la construcción de sistemas en toda clase de dominios de problemas, abarcando todo el abanico de tamaños y complejidades.

Los métodos orientados a objetos para el análisis de requerimientos de software (AOO) permiten al analista obtener el modelo de un problema representando clases, objetos, atributos y operaciones como componentes principales de modelización. Los objetos modelizan casi cualquier aspecto identificable del ámbito del problema: entidades externas, cosas, sucesos, papeles, unidades organizativas, lugares y estructuras. Como punto importante, los objetos encapsulan datos y procesos. Las operaciones de procesamiento son parte del objeto y son iniciadas pasando un mensaje al objeto. Una definición de una clase forma la base para la reusabilidad en los niveles de modelización, diseño e implementación. Las actividades de diseño orientado a objetos (DOO) están agrupadas en los cuatro componentes principales del sistema final: el componente de problema, el componente de interfaz humana, el componente de manejo de datos y el componente de manejo de tareas. Toda la documentación del análisis debe llevar directamente hacia la etapa del diseño. En este punto se necesitan pocas herramientas nuevas. El diseño orientado a objetos crea un modelo de mundo real que puede ser realizado en software. Los objetos proporcionan un mecanismo para representar el ámbito de información.

6.2. HERRAMIENTA UML

Lenguaje Unificado de Modelado¹ (**UML**, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

¹ http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa (Lengua de Modelación Unificada), no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la orientación a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de estados
- Diagrama de secuencias
- Diagrama de actividades
- Diagrama de colaboraciones
- Diagrama de componentes
- Diagrama de distribución

6.2.1 Cómo nació UML. Durante los ochenta y principios de los noventa Grady Booch, James Rumbaugh, e Ivar Jacobson trabajaban por separado en desarrollo de notaciones para el análisis y diseño de sistemas orientados a objetos. Los tres llegaron por separado a obtener bastante reconocimiento.

Booch había escrito "Object-Oriented Analysis and Design with Applications" un libro de referencia en el análisis y diseño orientado a objetos desarrollando su propia notación.

Por su parte **James Rumbaugh** había desarrollado su propia notación de diseño orientado a objetos llamada OMT (Object Modeling Technique) en su libro "Object-Oriented Modeling and Design".

Por otro lado **Jacobson** se había revelado como un visionario del análisis (padre de los casos de uso) y sobre todo del diseño orientado a objetos, sorprendiendo a todo el mundo en "Object-Oriented Software Engineering: A Use Case Driven Approach".

A mediados de los noventa empezaron a intercambiar documentos y trabajar en conjunto produciendo grandes avances en el modelado de sistemas orientados a objetos.

En 1994 Rational contrató a Rumbaugh en donde ya trabajaba Booch, un año después Jacobson se unía a ellos en Rational.

En 1997 salió a la luz la versión 1.0 de UML.

6.3. JAVA²

6.3.1 Descripción y características de este potente y moderno lenguaje de programación. Java es un lenguaje de programación con el se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

6.3.2 Pasado y presente de Java. Fue pensado originalmente para utilizarse en cualquier tipo de electrodoméstico pero la idea fracasó. Uno de los fundadores de Sun rescató la idea para utilizarla en el ámbito de Internet y convirtieron a Java en un lenguaje potente, seguro y universal gracias a que lo puede utilizar todo el mundo y es gratuito. Una de los primeros triunfos de Java fue que se integró en el navegador Netscape y permitía ejecutar programas dentro de una página web, hasta entonces impensable con el HTML.

Actualmente, Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas. Para lo que interesa, con Java se puede programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desea hacer con acceso a través web se puede hacer utilizando Java.

² <http://www.desarrolloweb.com/articulos/497.php>

6.4. PHP³

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa "**PHP Hypertext Pre-processor**" (inicialmente PHP Tools, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdof en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha declinado desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La más reciente versión principal del PHP fue la versión 5.2.5 de 8 de noviembre de 2007.

6.5. PostgreSQL⁴

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado *Ingres* en la Universidad Berkeley de California. *Ingres* fue más tarde desarrollado comercialmente por la *Relational Technologies/Ingres Corporation*.

En 1986 otro equipo dirigido por *Michael Stonebraker* de Berkeley continuó el desarrollo del código de *Ingres* para crear un sistema de bases de datos objeto-relacionales llamado *Postgres*. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, *Postgres* fue renombrado a *PostgreSQL*, tras un breve periplo como *Postgres95*. El proyecto *PostgreSQL* sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características (las cuales serán discutidas en detalle en la sección denominada ``*Juego de*

³ <http://es.wikipedia.org/wiki/.php>

⁴ <http://www.sobl.org/traduccion/practical-postgres/node12.html>

Características de PostgreSQL”) que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

6.6. XAJAX⁵

A pesar de que el término «AJAX» fuese creado en 2005, la historia de las tecnologías que permiten AJAX se remonta a una década antes con la iniciativa de Microsoft en el desarrollo de Scripting Remoto. Sin embargo, las técnicas para la carga asíncrona de contenidos en una página existente sin requerir recarga completa remontan al tiempo del elemento `iframe` (introducido en Internet Explorer 3 en 1996) y el tipo de elemento `layer` (introducido en Netscape 4 en 1997, abandonado durante las primeras etapas de desarrollo de Mozilla). Ambos tipos de elemento tenían el atributo `src` que podía tomar cualquier dirección URL externa, y cargando una página que contenga javascript que manipule la página paterna, pueden lograrse efectos parecidos al AJAX.

El **Microsoft's Remote Scripting** (o MSRS, introducido en 1998) resultó un sustituto más elegante para estas técnicas, con envío de datos a través de un applet Java el cual se puede comunicar con el cliente usando JavaScript. Esta técnica funcionó en ambos navegadores, Internet Explorer versión 4 y Netscape Navigator versión 4. Microsoft la utilizó en el Outlook Web Access provisto con la versión 2000 de Microsoft Exchange Server.

La comunidad de desarrolladores web, primero colaborando por medio del grupo de noticias *microsoft.public.scripting.remote* y después usando blogs, desarrollaron una gama de técnicas de scripting remoto para conseguir los mismos resultados en diferentes navegadores. Los primeros ejemplos incluyen la librería JSRS en el año 2000, la introducción a la técnica imagen/cookie en el mismo año y la técnica JavaScript bajo demanda (JavaScript on Demand) en 2002. En ese año, se realizó una modificación por parte de la comunidad de usuarios al *Microsoft's Remote Scripting* para reemplazar el applet Java por XMLHttpRequest. Frameworks de Scripting Remoto como el ARSCIF aparecieron en 2003 poco antes de que Microsoft introdujera Callbacks en ASP. NET.

Desde que XMLHttpRequest está implementado en la mayoría de los navegadores, raramente se usan técnicas alternativas. Sin embargo, todavía se utilizan donde se requiere una mayor compatibilidad, una reducida implementación, o acceso cruzado entre sitios web. Una alternativa, el Terminal_SVG (basado en SVG), emplea una conexión persistente para el intercambio continuo entre el navegador y el servidor.

⁵ <http://es.wikipedia.org/wiki/AJAX>

6.7. Aplicación Cliente Servidor⁶

Esta arquitectura consiste básicamente en que un programa -el cliente- realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

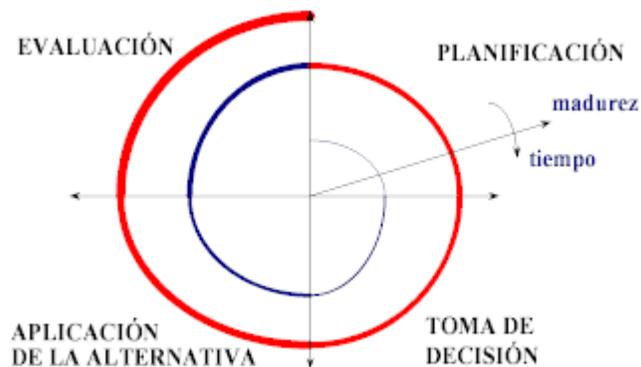
La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

⁶ <http://es.wikipedia.org/wiki/Cliente-servidor>

7. METODOLOGIA

Se va a utilizar el ciclo de Vida en Espiral⁷. El ciclo de vida en espiral puede considerarse como una generalización del anterior para los casos en que **no basta con una sola evaluación de un prototipo** para asegurar la desaparición de incertidumbres y/o ignorancias. El propio producto a lo largo de su desarrollo puede así considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente.

A menudo la **f fuente de incertidumbres es el propio cliente**, que aunque sepa en términos generales lo que quiere, no es capaz de definirlo en todos sus aspectos sin ver como unos influyen en otros. En estos casos la evaluación de los resultados por el cliente no puede esperar a la entrega final y puede ser necesaria repetidas veces.



El esquema del ciclo de vida para estos casos puede representarse por un bucle en espiral, donde los cuadrantes son, habitualmente, fases de **especificación, diseño, realización y evaluación** (o conceptos y términos análogos).

En cada vuelta el producto gana en "madurez" (aproximación al final deseado) hasta que en una vuelta la evaluación lo apruebe y el bucle pueda abandonarse.

Diagrama 1. Ciclo de Vida Espiral

7.1 VENTAJAS CICLO DE VIDA EN ESPIRAL

- La planificación es sencilla.
- La calidad del producto resultante es alta.
- Permite trabajar con personal poco cualificado.

7.2. INCONVENIENTES CICLO DE VIDA EN ESPIRAL

- Lo peor es la necesidad de tener todos los requisitos al principio. Lo normal es que el cliente no tenga perfectamente definidas las especificaciones del sistema, o puede ser que surjan necesidades imprevistas.
- Si se han cometido errores en una fase es difícil volver atrás.

⁷ <http://www.getec.etsit.upm.es/docencia/gproyectos/planificacion/cvida.htm>

- No se tiene el producto hasta el final, esto quiere decir que:
 - Si se comete un error en la fase de análisis no lo descubrimos hasta la entrega, con el consiguiente gasto inútil de recursos.
 - El cliente no verá resultados hasta el final, con lo que puede impacientarse.

El proyecto está planificado en dos ciclos cada uno de tres meses (doce semanas).

ACTIVIDADES LÓGICAS ANTERIORES	ACTIVIDADES PLANIFICADAS			ACTIVIDADES LÓGICAS POSTERIORES
	ORDEN	DETALLE	DURACIÓN EN SEMANAS	
INICIO-H	A	Comunicación con el cliente	1	B
A	B	Planificación	1	C
B	C	Análisis de riesgos	1	D
C	D	Ingeniería	4	E
D	E	Construcción y adaptación	3	F
E	F	Evaluación del cliente	1	G
F	G	Toma de decisiones	1	Si (Cliente da conformidad) → Fin de proyecto. Sino → A
		TOTAL:	24	

Tabla 1. Cronograma de actividades del proyecto

8. RESULTADOS DE LA INVESTIGACIÓN

8.1 MODELO DE CASOS DE USO

8.1.1 Listado de características.

ID	CARACTERISTICA_1
DESCRIPCION	El sistema debe permitir administrar Sedes. Ingresar, Modificar, Listar, Eliminar las diferentes Sedes de la Empresa
ESTADO	Incorporada

ID	CARACTERISTICA_2
DESCRIPCION	El sistema debe permitir administrar Cargos. Ingresar, Modificar, Listar, Eliminar los diferentes Cargos de la Empresa
ESTADO	Incorporada

ID	CARACTERISTICA_3
DESCRIPCION	El sistema debe permitir administrar Dependencias. Ingresar, Modificar, Listar, Eliminar las diferentes Dependencias de la Empresa
ESTADO	Incorporada

ID	CARACTERISTICA_4
DESCRIPCION	El sistema debe permitir administrar Horarios. Ingresar, Modificar, Listar, Eliminar los diferentes Horarios de la Empresa
ESTADO	Incorporada

ID	CARACTERISTICA_5
DESCRIPCION	El sistema debe permitir administrar Empleados. Ingresar, Modificar, Listar, Eliminar los diferentes Empleados de la Empresa con sus respectivos atributos (nombre, sede, cedula, correo, dirección, teléfono, foto, cargo, dependencia, horario).
ESTADO	Incorporada

ID	CARACTERISTICA_6
DESCRIPCION	Debe existir un reporte de ingreso y salida de empleados en la jornada laboral que visualice de forma gráfica (con colores dependiendo la hora de llegada o salida) para poder identificar de forma rápida las faltas del reglamento. Estos parámetros de las alertas deben ser dinámicos parametrizables.
ESTADO	Incorporada

ID	CARACTERISTICA_7
DESCRIPCION	El sistema debe tener un usuario y contraseña, la cual se puede cambiar por políticas de seguridad
ESTADO	Incorporada

ID	CARACTERISTICA_8
DESCRIPCION	El sistema debe registrar la fecha y hora de ingreso y salida del empleado en su jornada laboral, el sistema debe obtener la fecha y hora del servidor, para evitar que se altere la información.
ESTADO	Incorporada

ID	CARACTERISTICA_9
DESCRIPCION	El sistema debe reconocer en qué sede ingreso a su jornada laboral el empleado.
ESTADO	Incorporada

ID	CARACTERISTICA_10
DESCRIPCION	El reporte de ingresos y salidas debe ser dinámico y parametrizable (fecha inicio, fecha fin, por sedes, dependencias, cargos, empleados,)
ESTADO	Incorporada

ID	CARACTERISTICA_11
DESCRIPCION	Debe existir un reporte de ingresos y salidas donde indique las horas laboradas en un intervalo de tiempo (fecha inicio, fecha final).
ESTADO	Incorporada

ID	CARACTERISTICA_12
DESCRIPCION	Debe existir un reporte en donde indique que días no marcó en un intervalo de tiempo (fecha inicio, fecha final).
ESTADO	Incorporada

ID	CARACTERISTICA_13
DESCRIPCION	La información de los registros de las diferentes sedes se debe manejar de manera centralizada en un servidor, para poder tener acceso a ella de manera inmediata desde cualquier lugar.
ESTADO	Incorporada

8.1.2 Listado de actores.

NOMBRE	Administrador del Sistema
DESCRIPCION	Se encarga de administrar todos los parámetros del sistema y sacar reportes de los registros de ingresos y salida. Puede ser el jefe de Talento Humano o el Asistente <ul style="list-style-type: none">• Necesita ingresar, modificar, eliminar Sedes• Necesita ingresar, modificar, eliminar Dependencias• Necesita ingresar, modificar, eliminar Cargos• Necesita ingresar, modificar, eliminar Empleados• Necesita modificar Alertas• Necesita ingresar, modificar, eliminar Horarios• Necesita modificar Contraseña• Necesita generar Reportes

NOMBRE	Empleado
DESCRIPCION	Se encarga de registrar la hora de ingreso o salida en su respectiva sede.

8.1.3 Listado de casos de uso.

ID	CU_INI_SESION_1
CASO DE USO	Iniciar Sesión Administrativa
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrados del Sistema Ingresar en el navegador a: http://www.empopasto.com.co/ingreso * El Administrador del Sistema debe tener un usuario con su respectiva contraseña
POSTCONDICIONES	Iniciar Sesión en el Sistema
FLUJO BASICO	
1. El administrador digita el Usuario y Contraseña 2. El sistema Verifica el usuario y Contraseña 3. El sistema carga el menú para administrar el sistema	
FLUJOS ALTERNATIVOS	
2.a Si el Usuario y Contraseña no son correctos -> alerta "Clave o usuario Incorrectos"	

USUARIO:	<input type="text"/>
PASSWORD:	<input type="password"/>
INGRESAR	CANCELAR

Figura 2. Caso de uso iniciar sesión

ID	CU_ADM_EMPLEADO_2
CASO DE USO	Administrar Empleado
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema debe haber seleccionado la opción Empleado
POSTCONDICIONES	* Habilitar las opciones Adicionar y Actualizar Empleados * Listar todos los Empleados
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema debe Seleccionar el botón Empleado 2. El Sistema lista todos los Empleados y habilita la Opción Actualizar , Adicionar y en cada Empleado las opciones de Eliminar , Modificar y Estado del Empleado. 	
FLUJOS ALTERNATIVOS	
<p>2.a Si el Administrador del Sistema selecciona la opción Eliminar, entonces: CU_ELIMINA_EMPLEADO_3</p> <p>2.b Si el Administrador del Sistema selecciona la opción Adicionar , entonces: CU_ADICIONA_EMPLEADO_4</p> <p>2.c Si el Administrador del Sistema selecciona la opción Modificar , entonces: CU_MODIFICA_EMPLEADO_4</p> <p>2.d Si el Administrador del Sistema selecciona la opción Estado, entonces: CU_ESTADO_EMPLEADO_5</p>	



Figura 3. Caso de uso administrar empleado

ID	CU_ELIMINA_EMPLEADO_3
CASO DE USO	Eliminar Empleado
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe Seleccionar la opción Empleado
POSTCONDICIONES	* El sistema Elimina el empleado de la base de datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Eliminar del Empleado. 2. El sistema alerta para confirmar la eliminación. 3. El sistema elimina el empleado de la base de datos. 	
FLUJOS ALTERNATIVOS	
2.a Si el usuario Cancela, entonces: El sistema aborta el proceso de eliminación.	

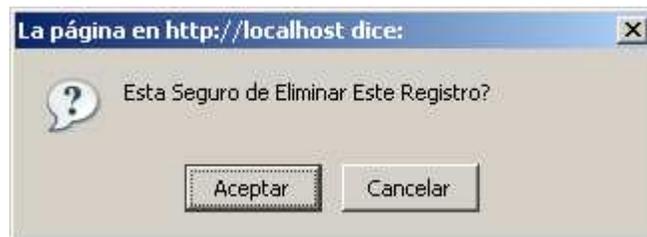


Figura 4. Caso de uso eliminar empleado

ID	CU_ADICIONA_EMPLEADO_4
CASO DE USO	Adicionar Empleado
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* Administrador del Sistema debe Seleccionar la opción Empleado
POSTCONDICIONES	* El sistema Adiciona un nuevo Empleado en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema ingresa los datos en el formulario 2. El Administrador del Sistema selecciona la opción Agregar 3. El sistema valida los datos. 4. El sistema ingresa un nuevo empleado en la base de datos 	
FLUJOS ALTERNATIVOS	
3.a. Si los datos no son correctos el sistema indica un mensaje de "Error Agregando Registro"	



Primer Apellido: arturo
 Segundo Apellido:
 Nombres: paola
 Cedula: 5123456
 Telefono: 300123568
 e-mail: paola@udenar.edu.co
 Dirección: cra 21#13-56
 Sede: MIJITAYO
 Dependencia: COORDINACION DE TECNOLOGIA
 Cargo: PROFESIONAL UNIVERSITARIO I
 Horario: ADMINISTRATIVA
 Foto: D:\CARLOS\Mis imagen

Activo: Si
 No

Figura 5. Caso de uso adicionar empleado

ID	CU_MODIFICA_EMPLEADO_5
CASO DE USO	Modificar Empleado
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe Seleccionar la opción Modificar
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación el Sistema modifica los datos del Empleado en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema Ingresa los datos en el formulario 2. El Administrador del Sistema selecciona el botón Modificar 3. El sistema valida los datos. 4. El sistema la información del empleado modifican en la base de datos. 	
FLUJOS ALTERNATIVOS	
3.a. Si los datos no son correctos el sistema indica un mensaje de "Error Modificando Registro"	

Primer Apellido:

Segundo Apellido:

Nombres:

Cedula:

Telefono:

e-mail:

Dirección:

Sede:

Dependencia:

Cargo:

Horario:

Foto:

Activo: Si
 No

Figura 6. Caso de uso modificar empleado

ID	CU_ESTADO_EMPLEADO_6
CASO DE USO	Cambiar Estado Empleado
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe Seleccionar la opción Estado
POSTCONDICIONES	* Si el empleado estaba activado quedara desactivado y viceversa
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona el botón Estado 2. El sistema cambia el estado del empleado. 3. El sistema lista empleados con el estado (activo - inactivo) actualizados 	
FLUJOS ALTERNATIVOS	



Figura 7. Caso de uso estado empleado

ID	CU_ADM_SEDE_7
CASO DE USO	Administrar Sede
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	<ul style="list-style-type: none"> * El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema debe haber seleccionado la opción Sede
POSTCONDICIONES	<ul style="list-style-type: none"> * El sistema habilita las opciones Adicionar y Actualizar Sedes * El sistema lista todas las Sedes
FLUJO BASICO	1. El sistema lista todas las Sedes y habilita la Opción Actualizar , Adicionar y en cada registro las opciones de Eliminar y Modificar
FLUJOS ALTERNATIVOS	2.a Si el Administrador del Sistema selecciona la opción Eliminar, entonces: CU_ELIMINA_SEDE_8 2.b Si el Administrador del Sistema selecciona la opción Adicionar , entonces: CU_ADICIONA_SEDE_9 2.c Si el Administrador del Sistema selecciona la opción Modificar , entonces: CU_MODIFICA_SEDE_10

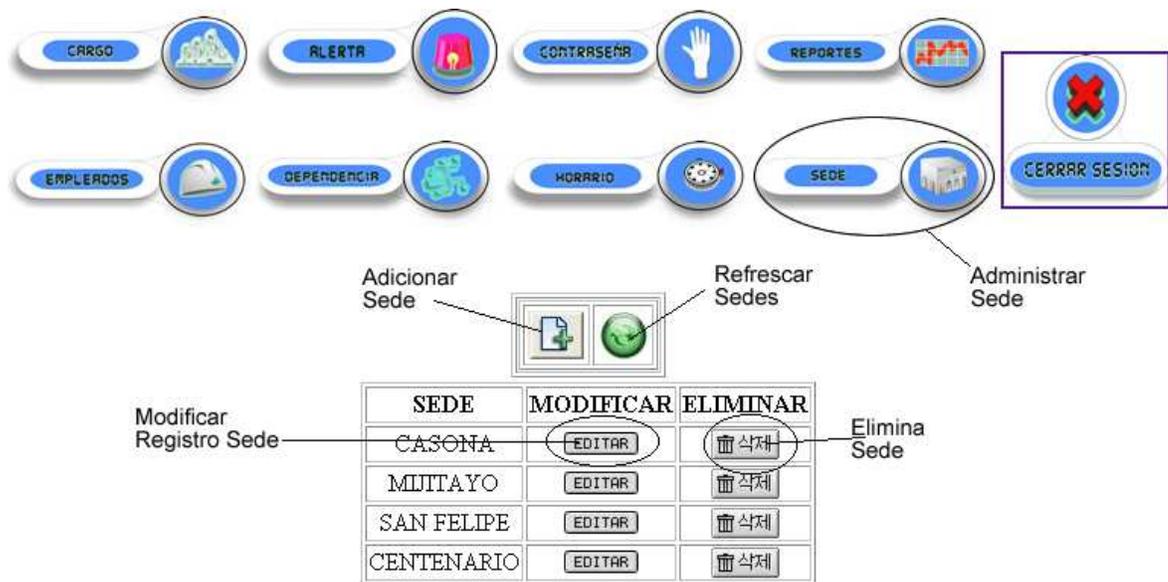


Figura 8. Caso de uso administrar sede

ID	CU_ELIMINA_SEDE_8
CASO DE USO	Eliminar Sede
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe Seleccionar la opción Sede
POSTCONDICIONES	* El sistema Elimina la Sede de la base de datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema debe Seleccionar la opción Eliminar 2. El Sistema alerta para confirmar la eliminación 3. El Sistema Elimina la Sede de la Base de Datos 	
FLUJOS ALTERNATIVOS	
2.a. Si el Administrador del Sistema Cancela la eliminación el sistema aborta el proceso de eliminación.	



Figura 9. Caso de uso eliminar sede

ID	CU_ADICIONA_SEDE_9
CASO DE USO	Adiciona Sede
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El administrador del sistema debe haber Seleccionado la Opción Sede
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación se Adiciona una nueva Sede en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema debe Seleccionar la opción adicionar 2. El Administrador del Sistema Ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Agregar 4. El sistema valida los datos. 5. El sistema ingresa una nueva Sede en la base de datos 	
FLUJOS ALTERNATIVOS	
3.a Si los datos no son correctos el sistema indica un mensaje de “Error Agregando Registro”	

Sede:

Figura 10. Caso de uso adicionar sede

ID	CU_MODIFICA_SEDE_10
CASO DE USO	Modifica Sede
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El administrador del sistema debe haber Seleccionado la Opción Sede
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación se Modifica la Sede en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Modificar 2. El Administrador del Sistema ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Modificar 4. El sistema valida los datos. 5. El sistema modifica la Sede en la base de datos 	
FLUJOS ALTERNATIVOS	
3.a. Si los datos no son correctos el sistema indica un mensaje de “Error Modificando Registro”	

Sede:

Figura 11. Caso de uso modificar sede

ID	CU_ADM_DEPENDENCIA_11
CASO DE USO	Administrar Dependencia
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema debe haber seleccionado la opción Dependencia
POSTCONDICIONES	* El sistema habilita las opciones Adicionar y Actualizar Dependencias * El sistema lista todas las Dependencias
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona el botón Sede 2. El sistema Lista todas las Dependencias y habilita la Opción Actualizar , Adicionar y en cada registro las opciones de Eliminar y Modificar 	
FLUJOS ALTERNATIVOS	
<p>2.a Si el Administrador del Sistema selecciona la opción Eliminar, entonces: CU_ELIMINA_DEPENDENCIA_12</p> <p>2.b Si el Administrador del Sistema selecciona la opción Agregar , entonces: CU_ADICIONA_DEPENDENCIA_13</p> <p>2.c Si el Administrador del Sistema selecciona la opción Editar , entonces: CU_MODIFICA_DEPENDENCIA_14</p>	

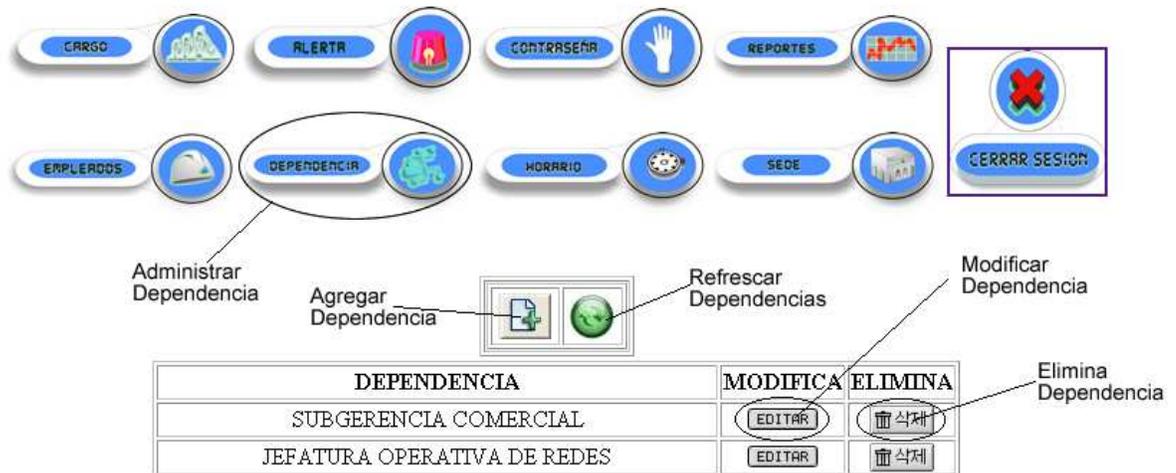


Figura 12. Caso de uso administrar dependencia

ID	CU_ELIMINA_DEPENDENCIA_12
CASO DE USO	Eliminar Dependencia
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber seleccionado la opción Dependencia
POSTCONDICIONES	* El sistema Elimina una Dependencia de la base de datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema debe haber seleccionado la opción Eliminar. 2. El sistema alerta para confirmar la eliminación. 3. El sistema elimina la dependencia de la base de datos. 	
FLUJOS ALTERNATIVOS	
2.a. Si el Administrador del Sistema Cancela la operación entonces el sistema aborta el proceso de eliminación.	

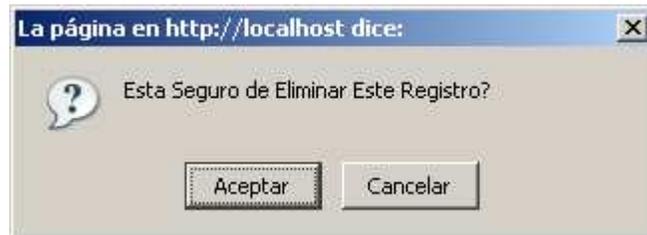


Figura 13. Caso de uso eliminar dependencia

ID	CU_ADICIONA_DEPENDENCIA_13
CASO DE USO	Adiciona Dependencia
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber Seleccionado la opción Dependencia
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación el sistema adiciona una nueva Dependencia en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción adicionar 2. El Administrador del Sistema Ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Agregar 4. El sistema valida los datos. 5. El sistema ingresa una nueva Dependencia en la base de datos 	
FLUJOS ALTERNATIVOS	
4.a. Si los datos no son correctos el sistema indica un mensaje de "Error Agregando Registro"	

Dependencia:

Figura 14. Caso de uso agregar dependencia

ID	CU_MODIFICA_DEPENDENCIA_14
CASO DE USO	Modifica Dependencia
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber Seleccionado la opción Dependencia
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación se Modifica la Dependencia en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Modificar 2. El Administrador del Sistema ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Modificar 4. El sistema valida los datos. 5. El sistema modifica la Dependencia en la base de datos 	
FLUJOS ALTERNATIVOS	
4.b. Si los datos no son correctos el sistema indica un mensaje de "Error Modificando Registro"	

Dependencia:

Figura 15. Caso de uso modificar dependencia

ID	CU_ADM_CARGO_15
CASO DE USO	Administrar Cargo
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema debe haber seleccionado la opción Cargo
POSTCONDICIONES	* El sistema habilita las opciones Adicionar y Actualizar Cargo * El sistema lista todos Cargo
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona el botón Cargo 2. El sistema Lista todas las Dependencias y habilita la Opción Actualizar , Adicionar y en cada registro las opciones de Eliminar y Modificar 	
FLUJOS ALTERNATIVOS	
<p>2.a Si el Administrador del Sistema selecciona la opción Eliminar, entonces: CU_ELIMINA_CARGO_16</p> <p>2.b Si el Administrador del Sistema selecciona la opción Agregar , entonces: CU_ADICIONA_CARGO_17</p> <p>2.c Si el Administrador del Sistema selecciona la opción Editar , entonces: CU_MODIFICA_CARGO_18</p>	

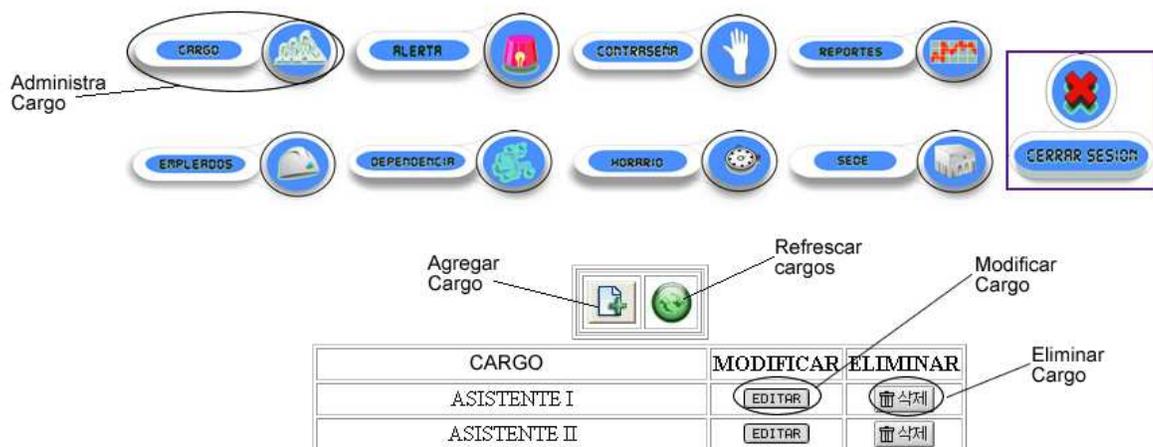


Figura 16. Caso de uso administrar cargo

ID	CU_ELIMINA_CARGO_16
CASO DE USO	Eliminar Cargo
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber seleccionado la opción Cargo
POSTCONDICIONES	* El sistema Elimina un Cargo de la base de datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema debe haber seleccionado la opción Eliminar. 2. El sistema alerta para confirmar la eliminación. 3. El sistema elimina el cargo de la base de datos. 	
FLUJOS ALTERNATIVOS	
2.a. Si el Administrador del Sistema Cancela la operación entonces el sistema aborta el proceso de eliminación	

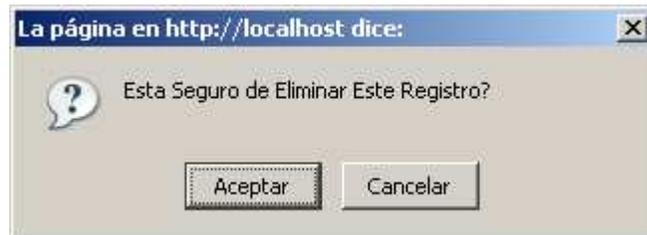


Figura 17. Caso de uso eliminar cargo

ID	CU_ADICIONA_CARGO_17
CASO DE USO	Adiciona Cargo
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber Seleccionado la opción Cargo
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación el sistema se Adiciona una nuevo Cargo en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción adicionar 2. El Administrador del Sistema Ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Agregar 4. El sistema valida los datos. 5. El sistema ingresa un nuevo Cargo en la base de datos. 	
FLUJOS ALTERNATIVOS	
4.a. Si los datos no son correctos el sistema indica un mensaje de "Error Agregando Registro"	

Cargo:

Figura 18. Caso de uso adicionar cargo

ID	CU_MODIFICA_CARGO_18
CASO DE USO	Modifica Cargo
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber Seleccionado la opción Cargo
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación se Modifica el Cargo en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Modificar 2. El Administrador del Sistema ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Modificar 4. El sistema valida los datos. 5. El sistema modifica el cargo en la base de datos 	
FLUJOS ALTERNATIVOS	
4.b. Si los datos no son correctos el sistema indica un mensaje de "Error Modificando Registro"	

Cargo:

Figura 19. Caso de uso modificar cargo

ID	CU_ADM_HORARIO_19
CASO DE USO	Administrar Horario
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema debe haber seleccionado la opción Horario
POSTCONDICIONES	* El sistema habilita las opciones Adicionar y Actualizar Horarios * El sistema lista todos los horarios
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Horario 2. El sistema Lista todas las Dependencias y habilita la Opción Actualizar , Adicionar y en cada registro los botones de Eliminar y Modificar 	
FLUJOS ALTERNATIVOS	
<p>2.a Si el Administrador del Sistema selecciona la opción Eliminar, entonces: CU_ELIMINA_HORARIO_20</p> <p>2.b Si el Administrador del Sistema selecciona la opción Agregar , entonces: CU_ADICIONA_HORARIO_21</p> <p>2.c Si el Administrador del Sistema selecciona la opción Editar , entonces: CU_MODIFICA_HORARIO_22</p>	

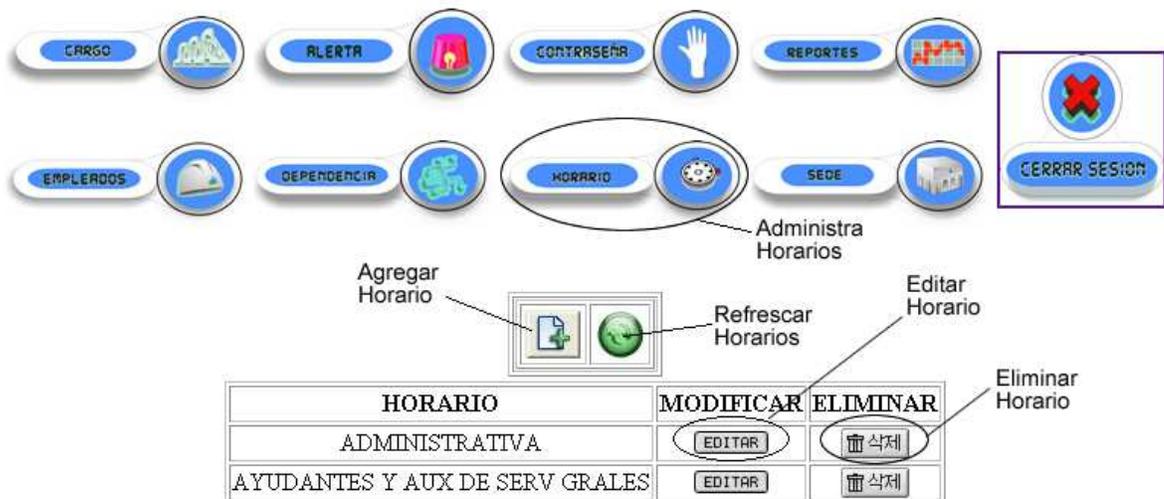


Figura 20. Caso de uso administrar horario

ID	CU_ELIMINA_HORARIO_20
CASO DE USO	Eliminar Horario
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber seleccionado la opción
POSTCONDICIONES	* El sistema Elimina Un Horario de la base de datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema debe haber seleccionado la opción Eliminar. 2. El sistema alerta para confirmar la eliminación. 3. El sistema elimina el Horario de la base de datos. 	
FLUJOS ALTERNATIVOS	
2.a. Si el Administrador del Sistema Cancela la operación entonces el sistema aborta el proceso de eliminación.	



Figura 21. Caso de uso eliminar horario

ID	CU_ADICIONA_HORARIO_21
CASO DE USO	Adiciona Horario
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber Seleccionado la opción
POSTCONDICIONES	* Si los datos son consistentes y confirma la operación el sistema adiciona un nuevo Horario en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción adicionar 2. El Administrador del Sistema Ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Agregar 4. El sistema valida los datos. 5. El sistema ingresa un nuevo Horario en la base de datos. 	
FLUJOS ALTERNATIVOS	
4.a. Si los datos no son correctos el sistema indica un mensaje de "Error Agregando Registro"	

Adiciona Horario



Nombre Horario:

Repetir	Día	Entrada1	Salida1	Entrada2	Salida2
<input type="checkbox"/>	Lunes:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Martes:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Miercoles:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Jueves:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Viernes:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Sabado:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Descripción:

Figura 22. Caso de uso adicionar horario

ID	CU_MODIFICA_HORARIO_22
CASO DE USO	Modifica Horario
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber Seleccionado la opción modificar Horario
POSTCONDICIONES	* Si los datos son consistentes y confirma el sistema Modifica el horario en la Base de Datos
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Modificar 2. El Administrador del Sistema ingresa los datos en el formulario 3. El Administrador del Sistema selecciona la opción Modificar 4. El sistema valida los datos. 5. El sistema modifica la Horario en la base de datos 	
FLUJOS ALTERNATIVOS	
4.b. Si los datos no son correctos el sistema indica un mensaje de "Error Modificando Registro"	



HORARIO	MODIFICAR	ELIMINAR
ADMINISTRATIVA	<input type="button" value="EDITAR"/>	<input type="button" value="삭제"/>
AYUDANTES Y AUX DE SERV GRALES	<input type="button" value="EDITAR"/>	<input type="button" value="삭제"/>

Editar Horario

Figura 23. Caso de uso modificar horario

ID	CU_ADM_ALERTAS_23
CASO DE USO	Administrar Alertas
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema seleccionar la opción Alertas
POSTCONDICIONES	* Listar todas las Alertas (Amarillo, Naranja Rojo) * El sistema actualiza los parámetros de tiempo para listar los reportes con colores.
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Alerta 2. El sistema lista las Alertas y habilita la Opción Enviar y Cancelar. 3. El Administrador del Sistema Ingresar nuevos datos en los parámetros de Alertas 4. El sistema valida datos 5. El Sistema actualiza los parámetros de colores 	
FLUJOS ALTERNATIVOS	
4.a. Si los datos no son consistentes, entonces: El sistema alerta que no puede terminar el proceso.	

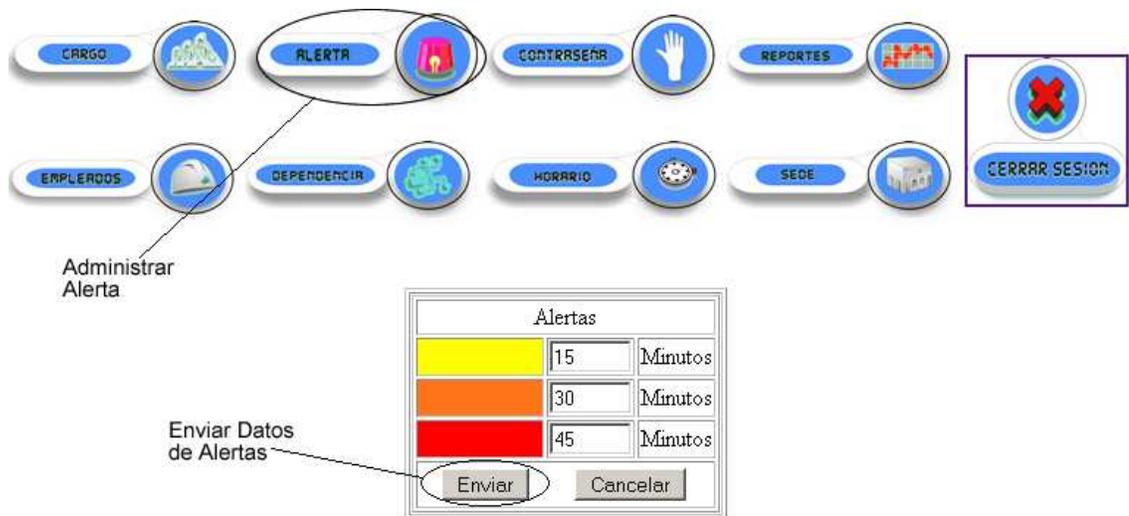


Figura 24. Caso de uso administrar alertas

ID	CU_ADM_CONTRASEÑA_24
CASO DE USO	Administrar Contraseña
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema seleccionar la opción Contraseña
POSTCONDICIONES	El sistema Actualiza la Contraseña.
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Contraseña 2. Ingresar nueva contraseña y la confirmación en el formulario 3. Enviar la nueva contraseña 4. El sistema valida datos 5. El sistema Actualiza la contraseña 	
FLUJOS ALTERNATIVOS	
4.a Si los datos no son consistentes entonces, El sistema alerta que no puede terminar el proceso.	

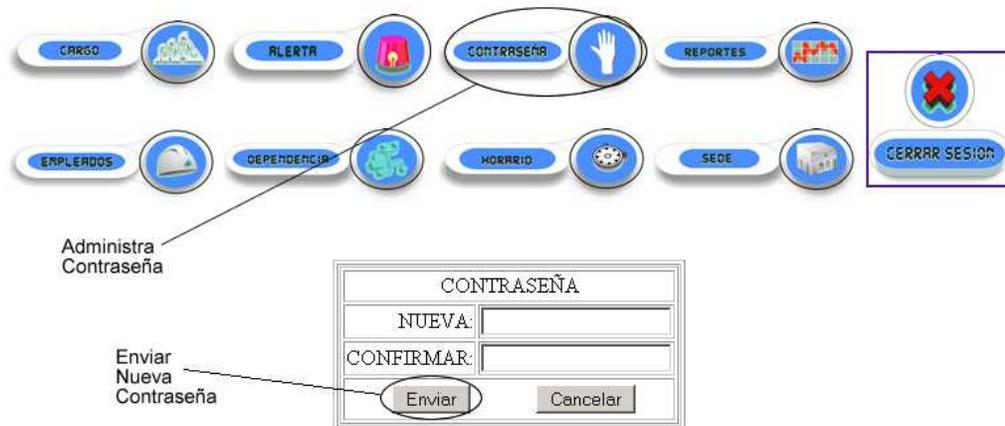


Figura 25. Caso de uso administrar contraseña

ID	CU_ADM_REPORTES_25
CASO DE USO	Administrar Reportes
ACTOR PRINCIPAL	Administrador del Sistema
PRECONDICIONES	* El Administrador del Sistema debe haber iniciado Sesión * El Administrador del Sistema seleccionar la opción Reportes
POSTCONDICIONES	El sistema Genera Reporte Personalizado
FLUJO BASICO	
<ol style="list-style-type: none"> 1. El Administrador del Sistema selecciona la opción Reporte 2. El Administrador del Sistema selecciona las opciones del Reporte que Necesite Generar 3. El Administrador del Sistema Envía los datos 4. El Sistema genera el Reporte 	
FLUJOS ALTERNATIVOS	
<ol style="list-style-type: none"> 2.a. Si selecciona la opción Fecha Inicio y Fecha Fin: esta opción es obligatoria para limitar la búsqueda de datos 2.b. Si selecciona la opción Empleado: el sistema va a generar un reporte con toda la información del empleado seleccionado de la lista. 2.c. Si selecciona la opción Sede: el sistema va a generar un reporte con toda la información de la sede seleccionada de la lista. 2.d. Si selecciona la opción Dependencia: el sistema va a generar un reporte con toda la información de la Dependencia seleccionada de la lista. 2.e. Si selecciona la opción Cargo: el sistema va a generar un reporte con toda la información del Cargo seleccionado de la lista. 2.f. Si selecciona la opción No Marcan: el sistema va a generar un reporte el(los) empleados que no han marcado en el rango de fechas. 2.g. Si selecciona la opción Suma de Horas: el sistema va a generar un reporte con la sumatoria de horas trabajadas en el rango de fechas. 2.h. Si selecciona la opción Discriminado: el sistema va a generar un reporte detallado con la fecha, hora, color de alerta de ingreso y salida. 	

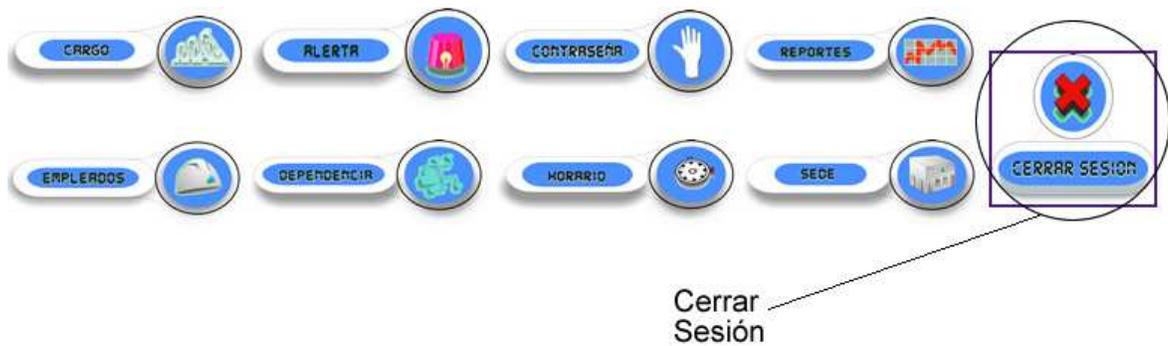


Figura 27. Caso cerrar sesión

ID	CU_REGISTRAR_27
CASO DE USO	Registrar fecha y hora de entrada y salida
ACTOR PRINCIPAL	Empleado
PRECONDICIONES	* Los datos del empleado deben estar Registrados en la Base de Datos(cedula, nombre, etc.)
POSTCONDICIONES	El registro queda ingresado en la base de datos con fecha, hora y entrada o salida
FLUJO BASICO	
1. El empleado pasa el Carné por el lector de código de barras o Digita la Cedula. 2. El Sistema Muestra un Mensaje indicando si el Registro fue Ingresado con éxito, nombre, fecha y hora.	
FLUJOS ALTERNATIVOS	
2.a si hay un error, el sistema informa que no fue posible realizar el registro.	



Figura 28. Caso de uso registrar fecha y hora de entrada y salida

8.1.4 Diagrama de casos de uso.

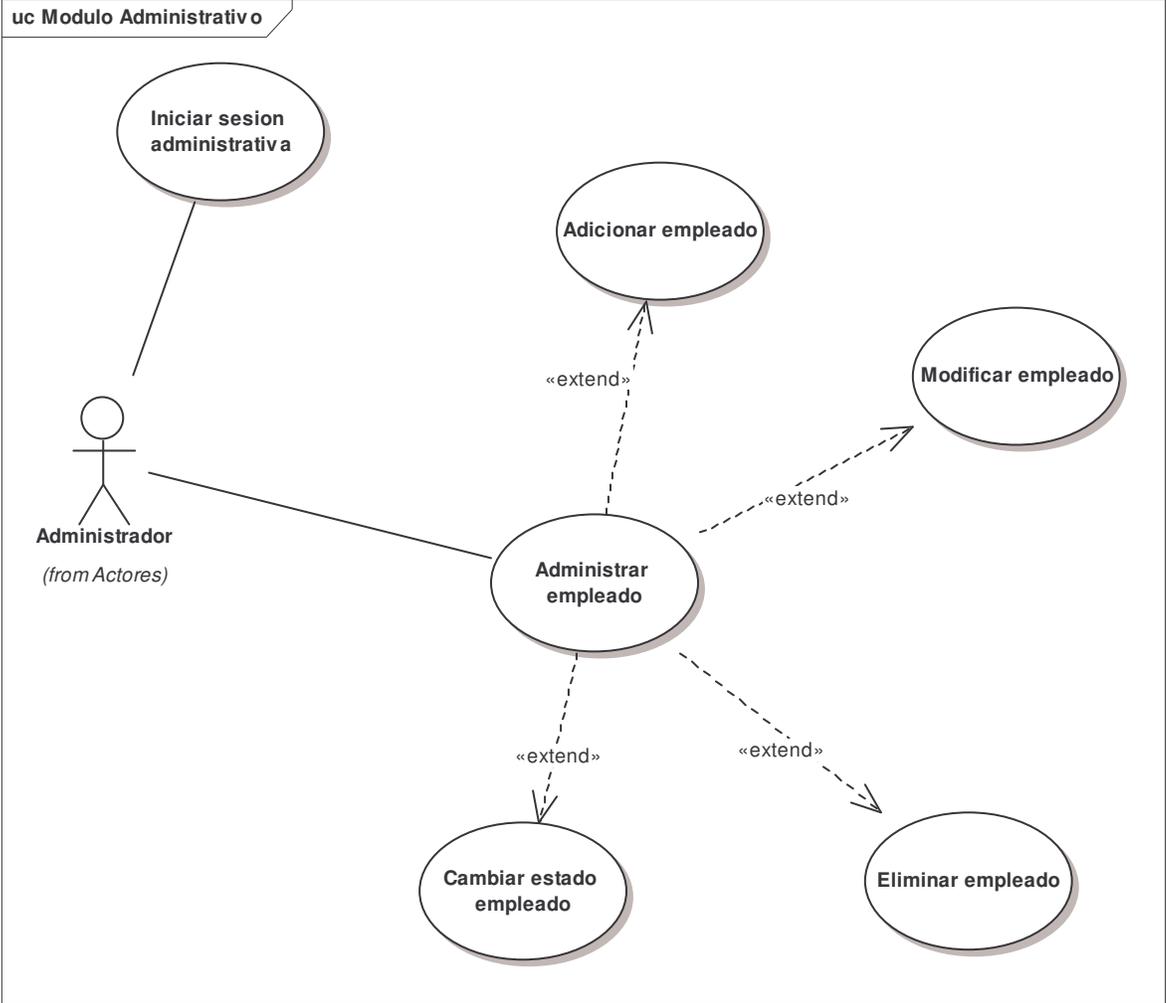


Diagrama 2. Casos de uso modulo administrativo

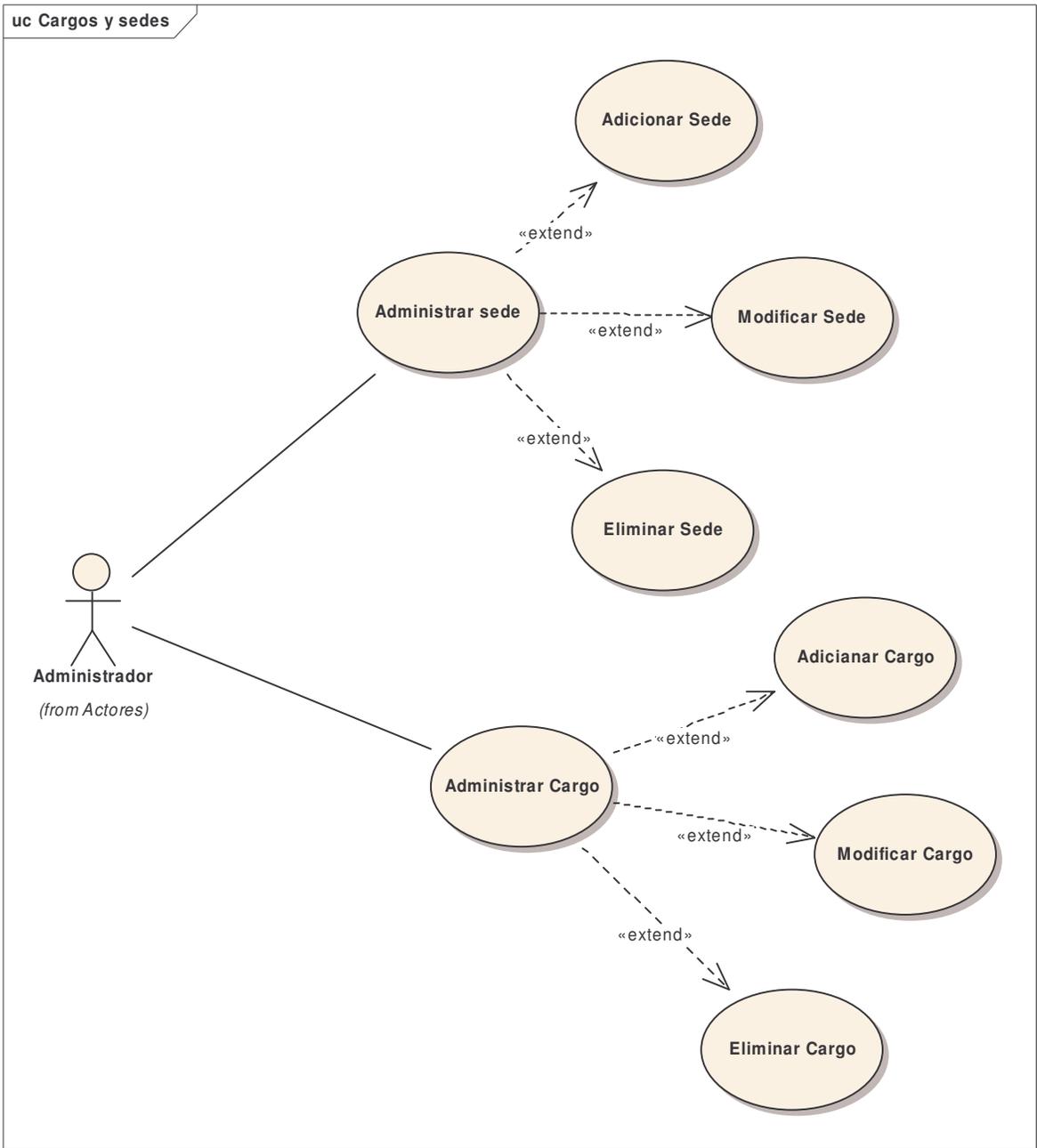


Diagrama 3. Casos de uso cargos y sedes

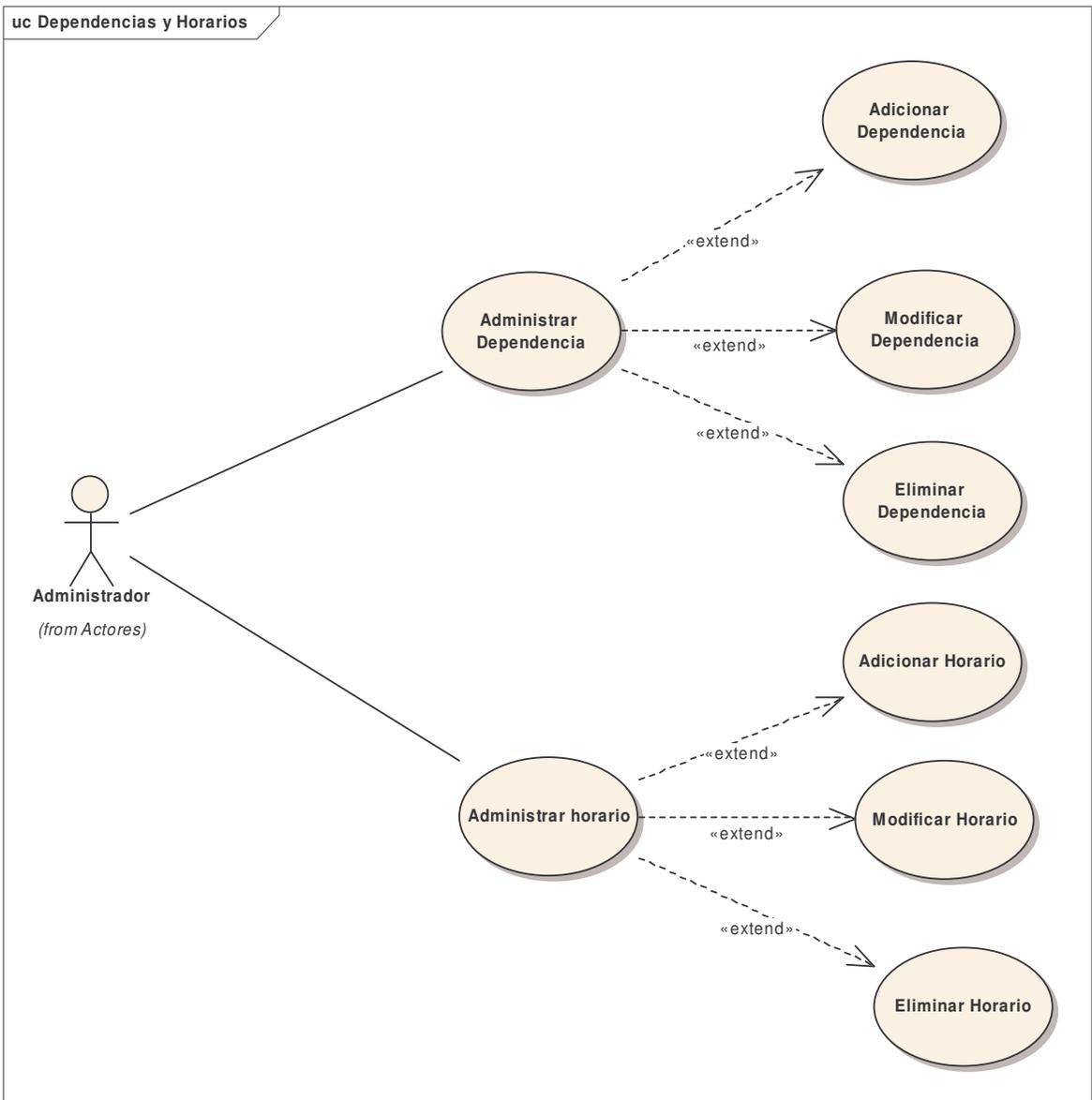


Diagrama 4. Casos de uso dependencias y horario

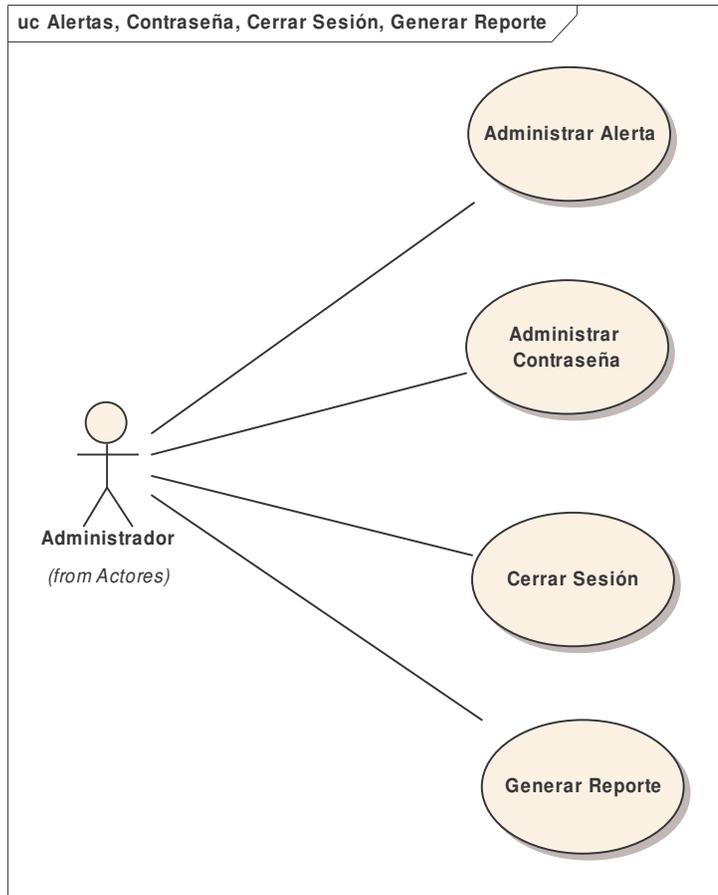


Diagrama 5. Casos de uso alertas, contraseña, cerrar sesión, reporte



Diagrama 6. Casos de uso registrar fecha y hora

8.1.5 Diagramas de secuencia del sistema.

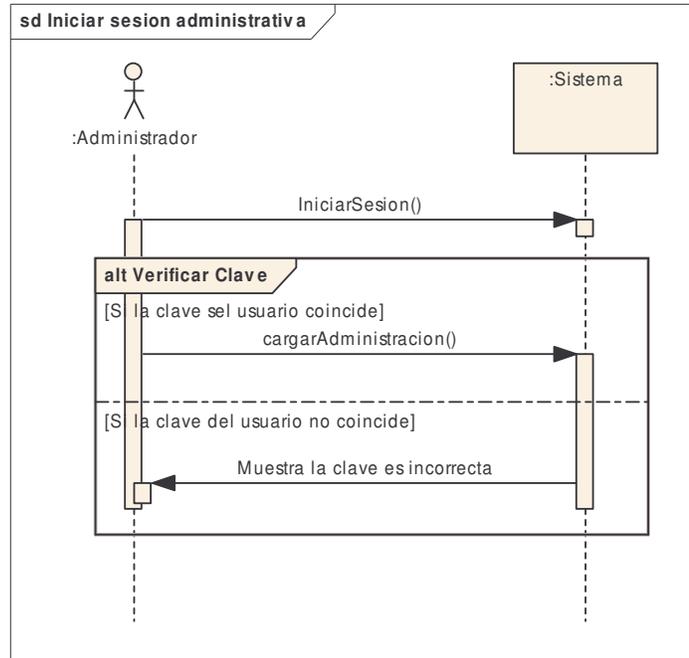


Diagrama 7. Secuencia iniciar sesión

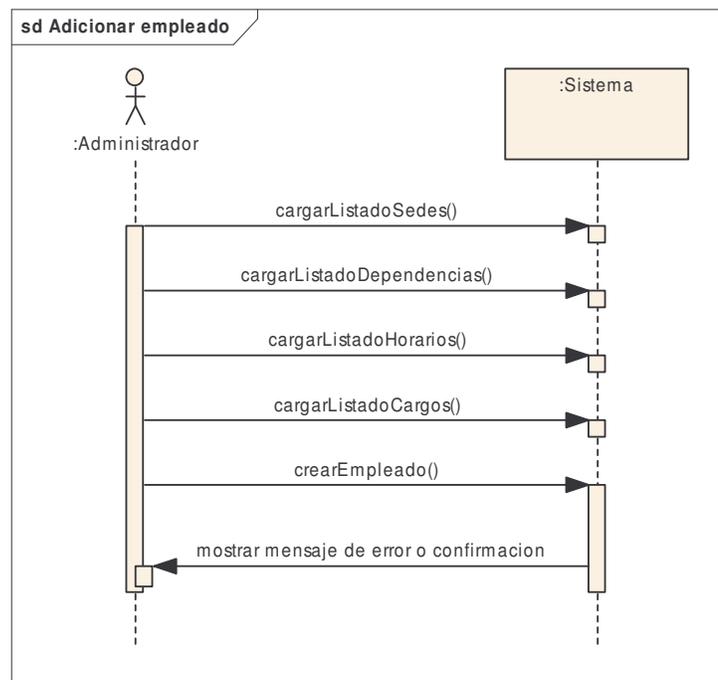


Diagrama 8. Secuencia adicionar empleado

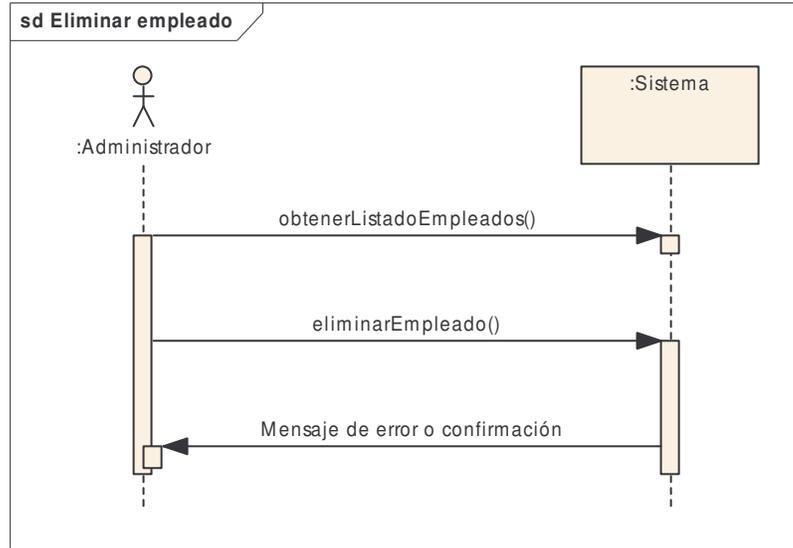


Diagrama 9. Secuencia eliminar empleado

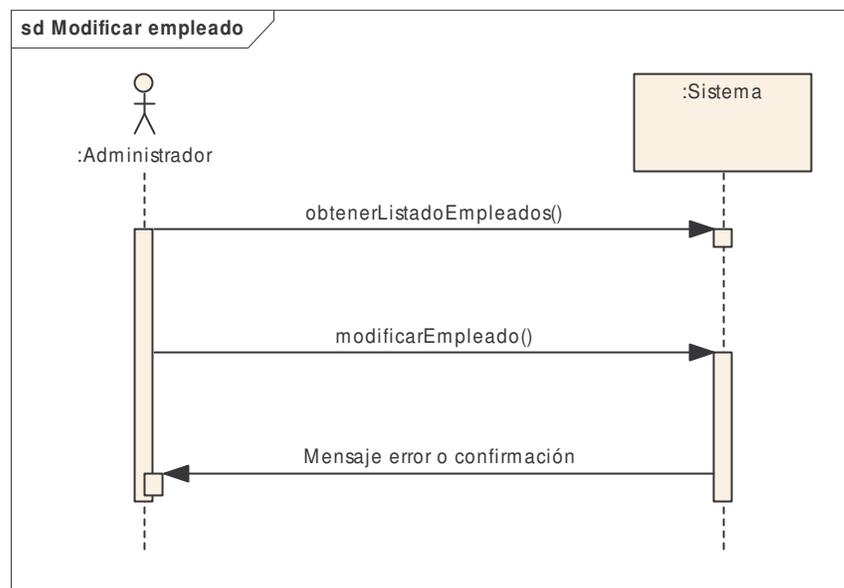


Diagrama 10. Secuencia modificar empleado

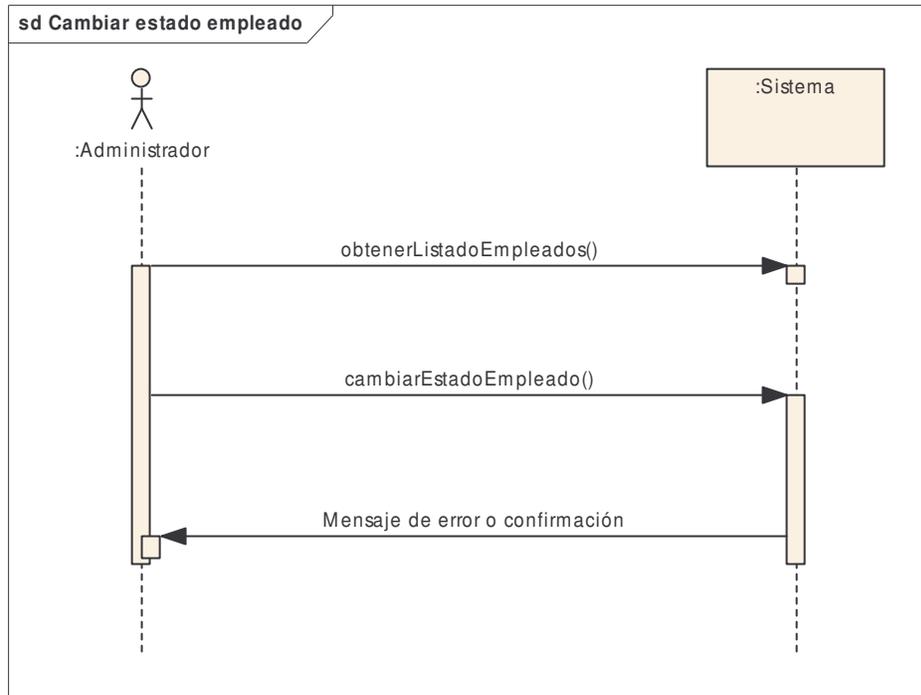


Diagrama 11. Secuencia estado empleado

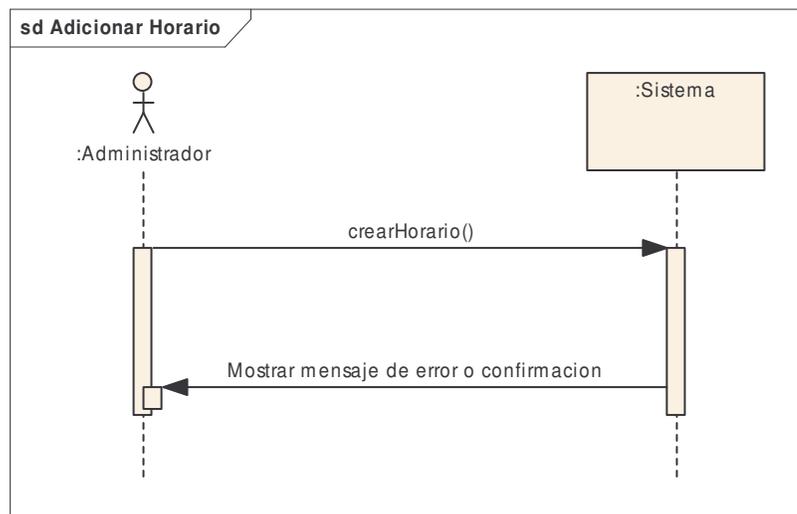


Diagrama 12. Secuencia adicionar horario

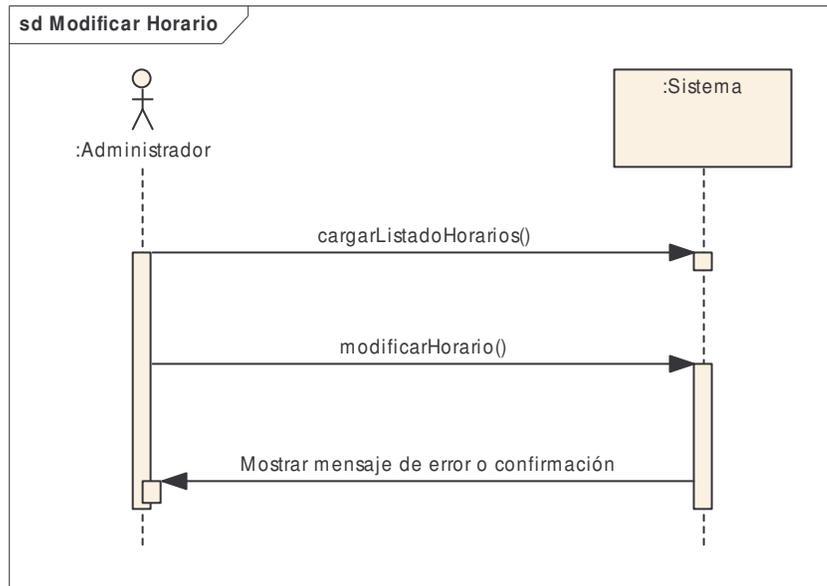


Diagrama 13. Secuencia modificar horario

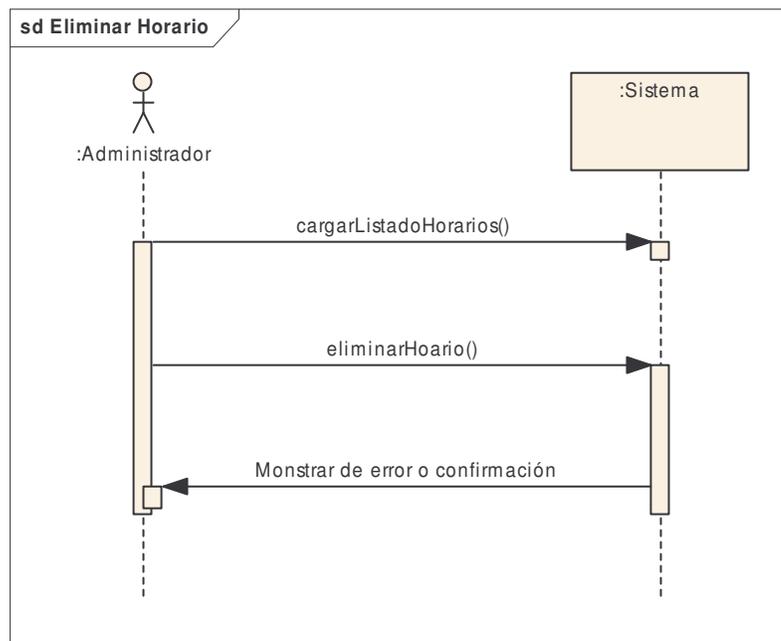


Diagrama 14. Secuencia eliminar horario

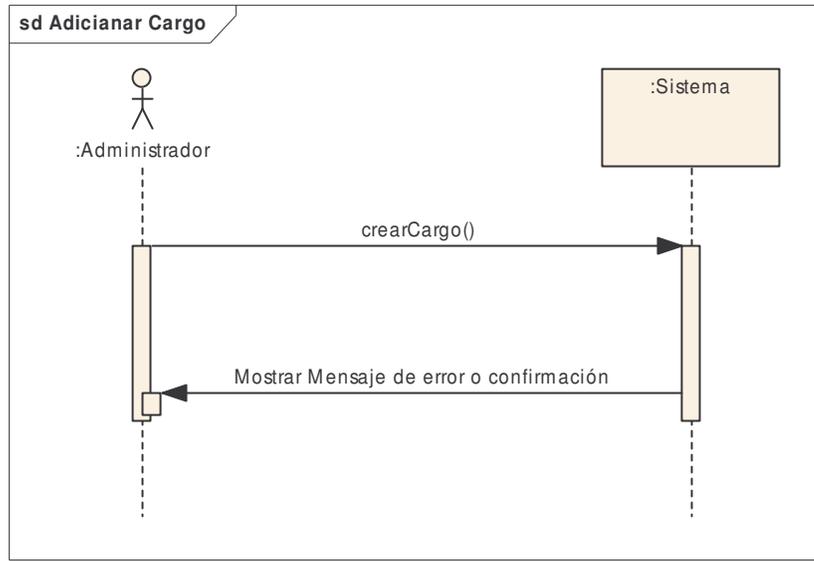


Diagrama 15. Secuencia adicionar cargo

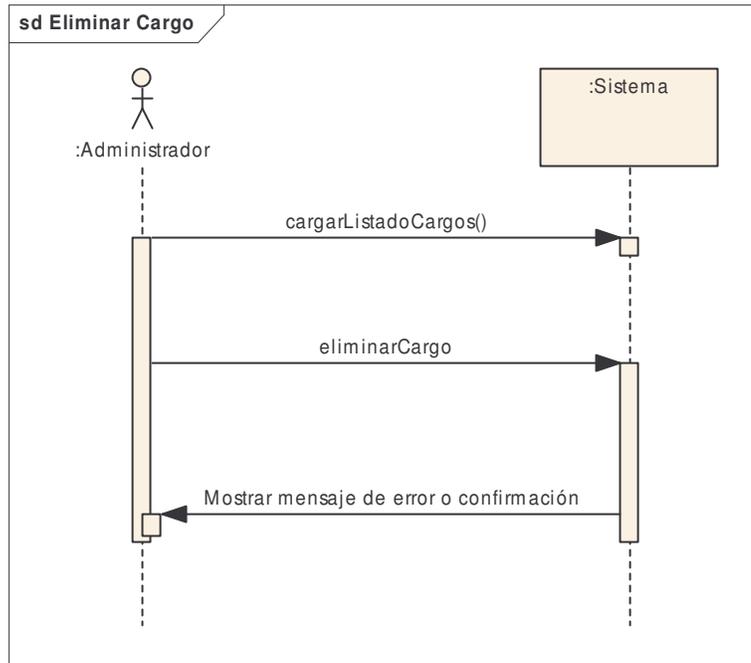


Diagrama 16. Secuencia eliminar cargo

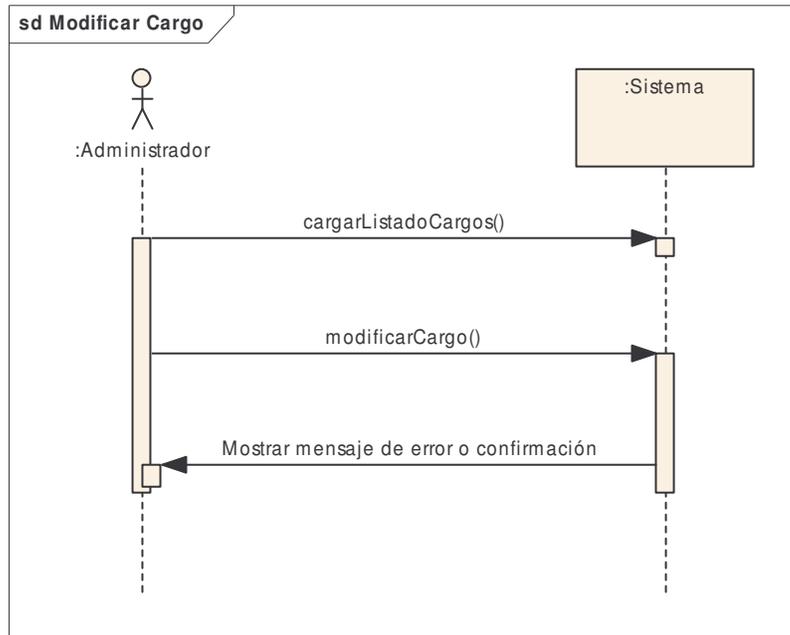


Diagrama 17. Secuencia modificar cargo

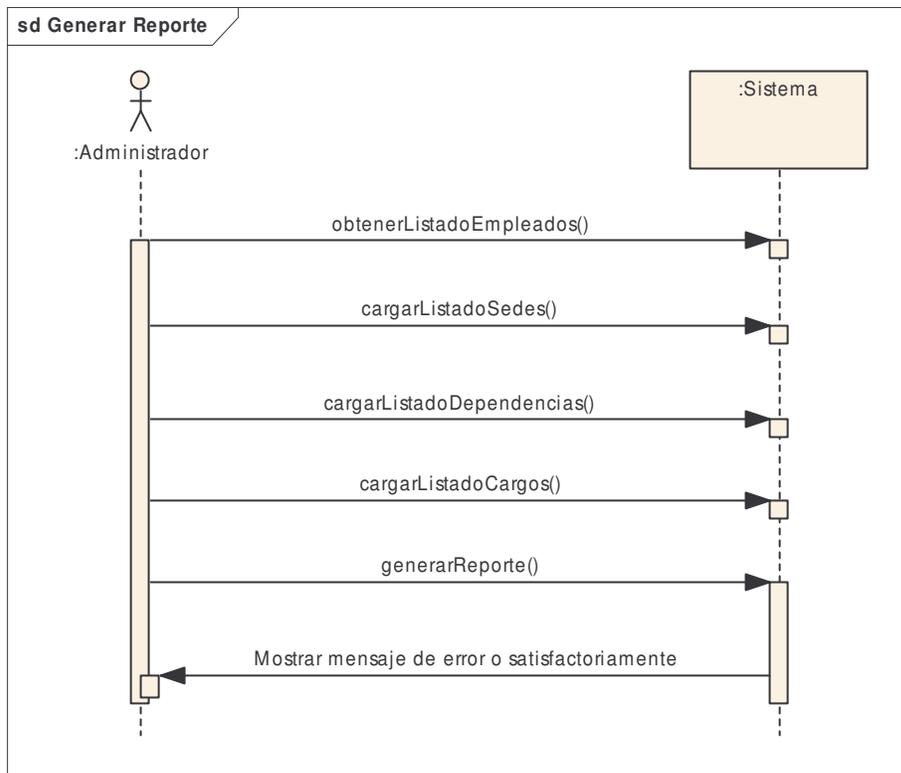


Diagrama 18. Secuencia generar Reporte

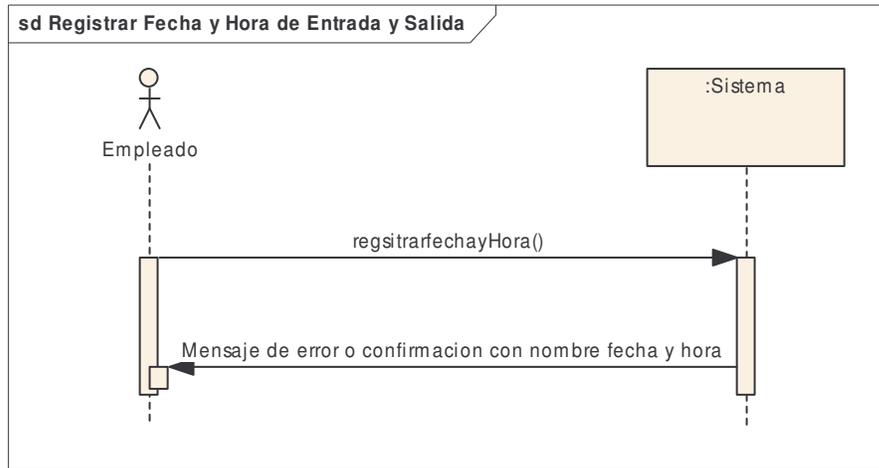


Diagrama 19. Secuencia registrar fecha y hora de entrada y salida

8.2 MODELO DE ANÁLISIS

8.2.1 Listado de conceptos.

Tabla 2. Listado de conceptos

Concepto	Descripción
Empleado	Se encarga de todos procesos relacionados con el empleado como registrar, modificar, eliminar, cambiar estado entre otros.

Concepto	Descripción
Sede	Se encarga de todos procesos relacionados con la Sede: registrar, modificar, eliminar.

Concepto	Descripción
Dependencia	Se encarga de todos procesos relacionados con la dependencia: registrar, modificar, eliminar.

Concepto	Descripción
8Cargo	Se encarga de todos procesos relacionados con el Cargo: registrar, modificar, eliminar.

Concepto	Descripción
Horario	Se encarga de todos procesos relacionados con las Horarios de las diferentes Dependencias y Cargos de los empleados de la empresa: registrar, modificar, eliminar.

Concepto	Descripción
Alerta	Se encarga de administrar los parámetros para indicar de manera gráfica que tan tarde llega o que tan temprano sale un empleado y hay tres escalas de alerta: amarillo, naranja y Rojo.

Concepto	Descripción
Contraseña	Se encarga de modificar la contraseña de un usuario.

Concepto	Descripción
Sede	Se encarga de todos procesos relacionados con la Sede: registrar, modificar, eliminar.

8.2.2 Diagrama de clases del análisis.

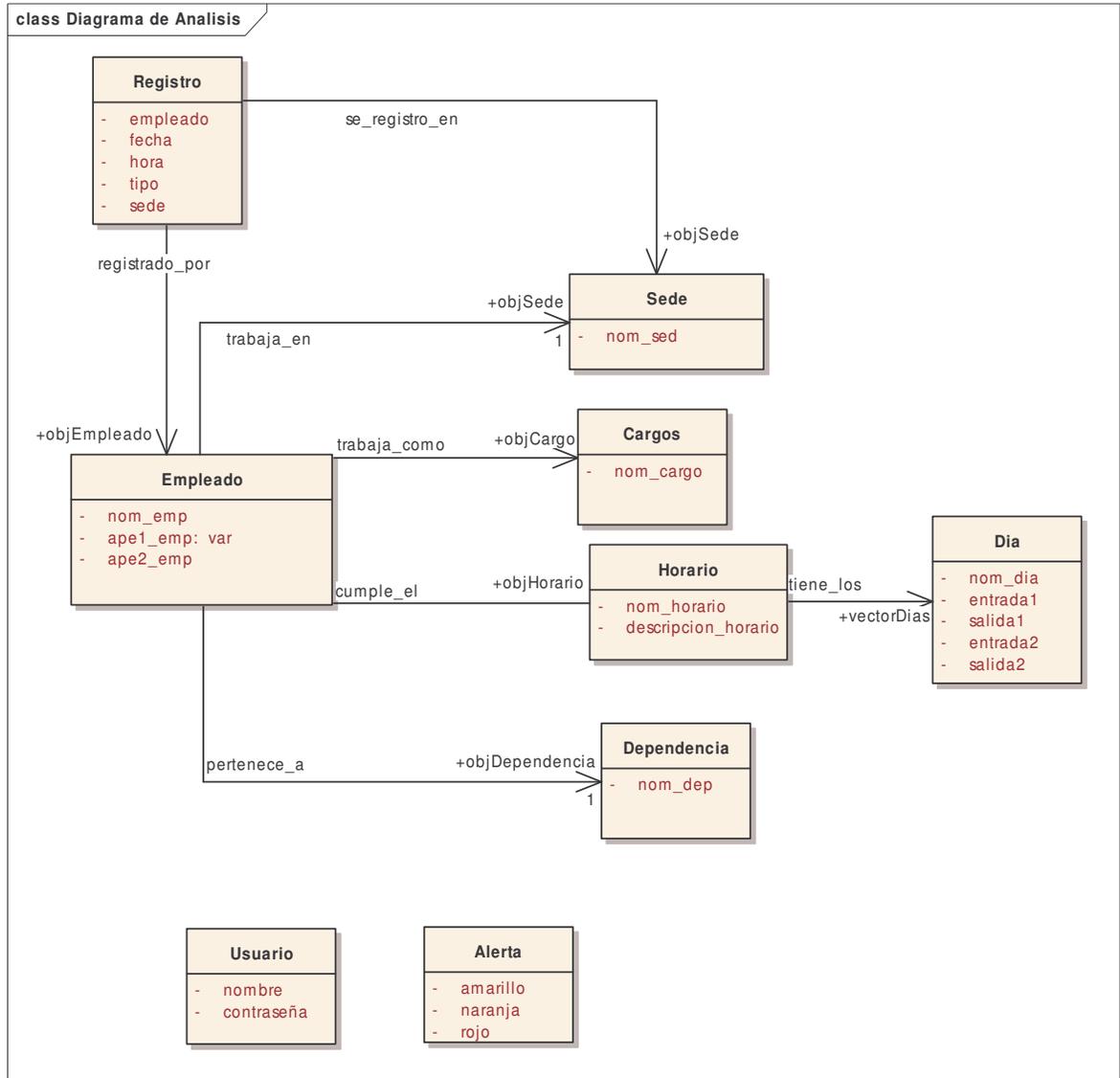


Diagrama 20. Análisis del sistema

8.3 MODELO DE DISEÑO

8.3.1 Diagramas de secuencia.

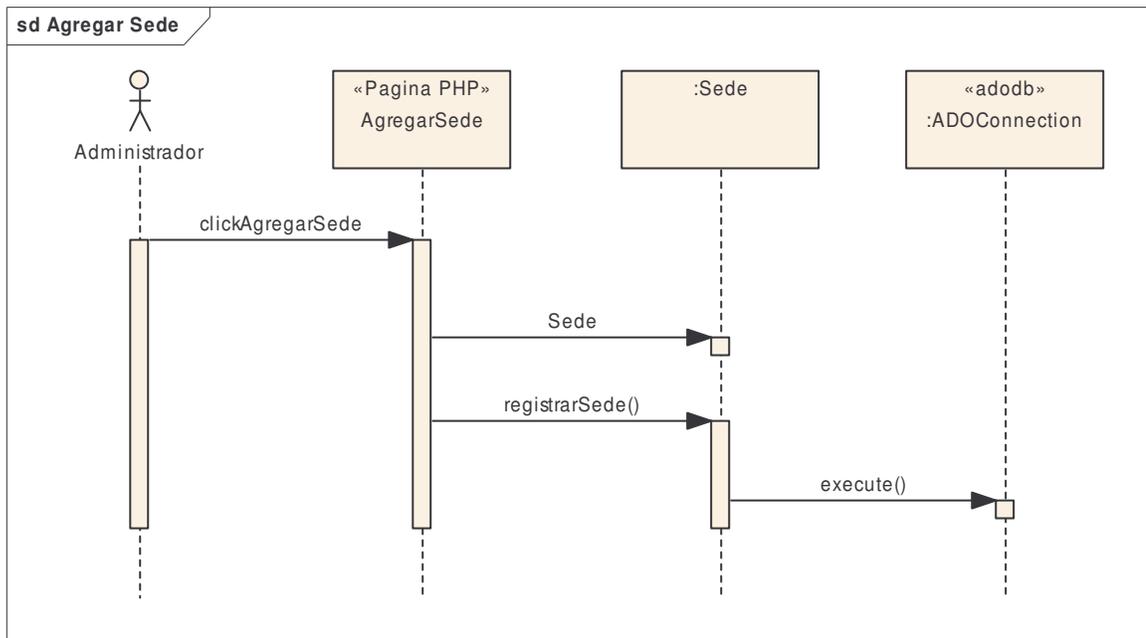


Diagrama 21. Secuencia agregar sede

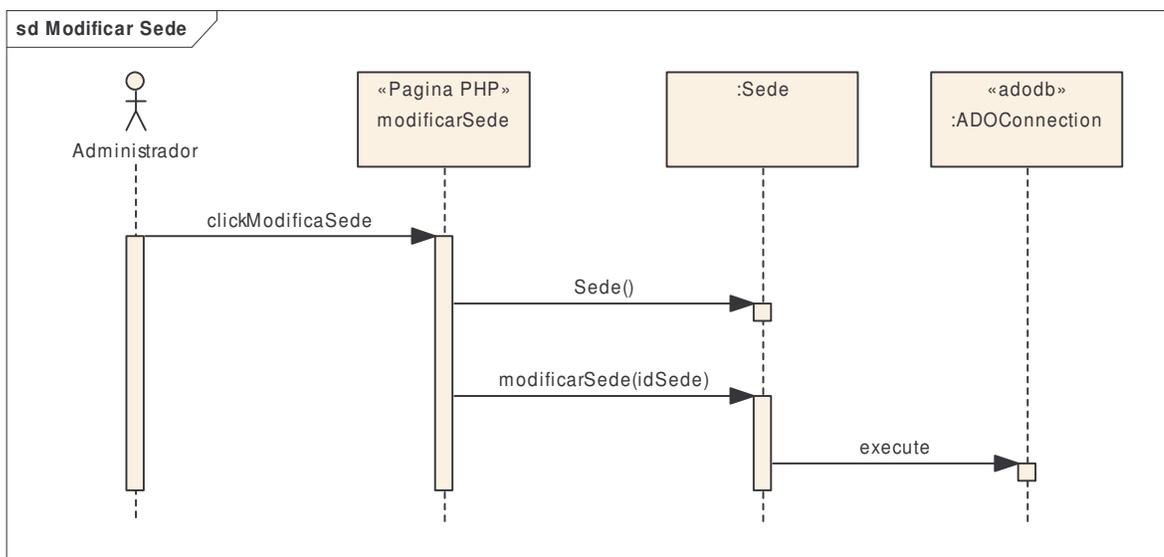


Diagrama 22. Secuencia modificar sede

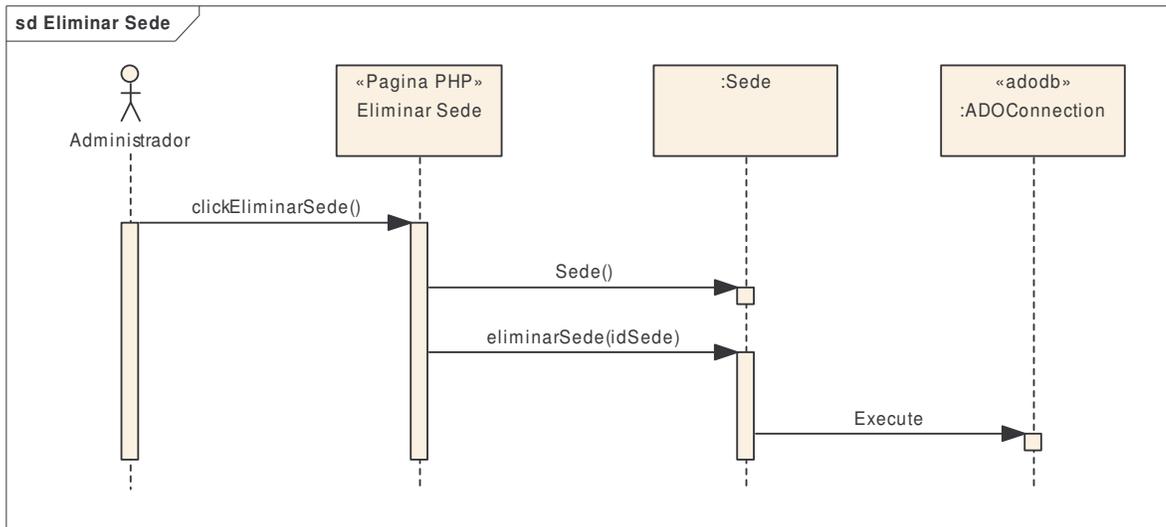


Diagrama 23. Secuencia eliminar sede

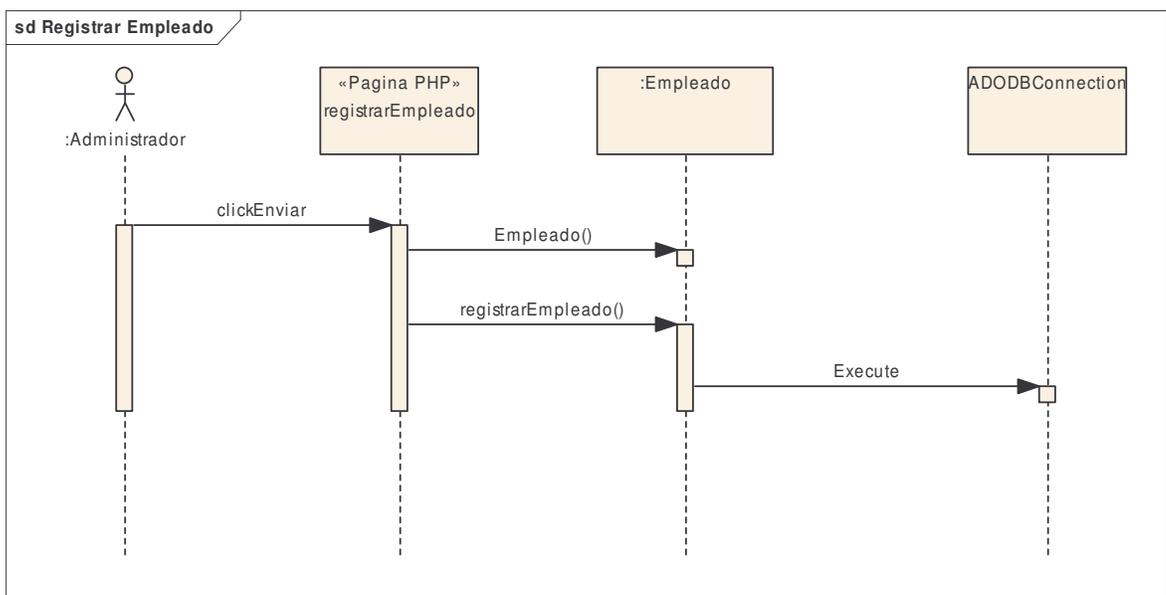


Diagrama 24. Secuencia registrar empleado

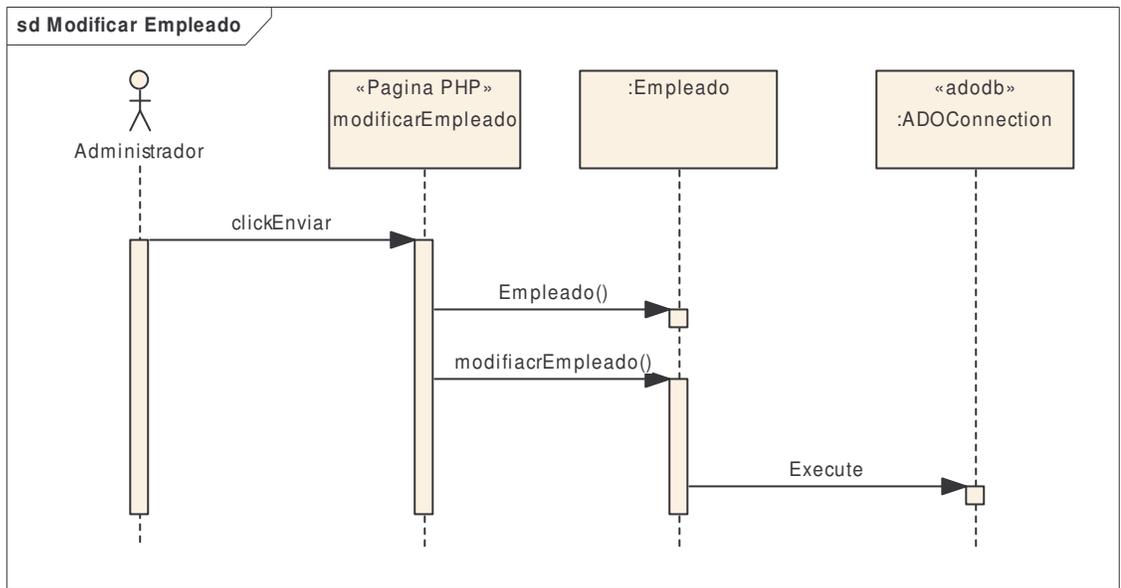


Diagrama 25. Secuencia modificar empleado

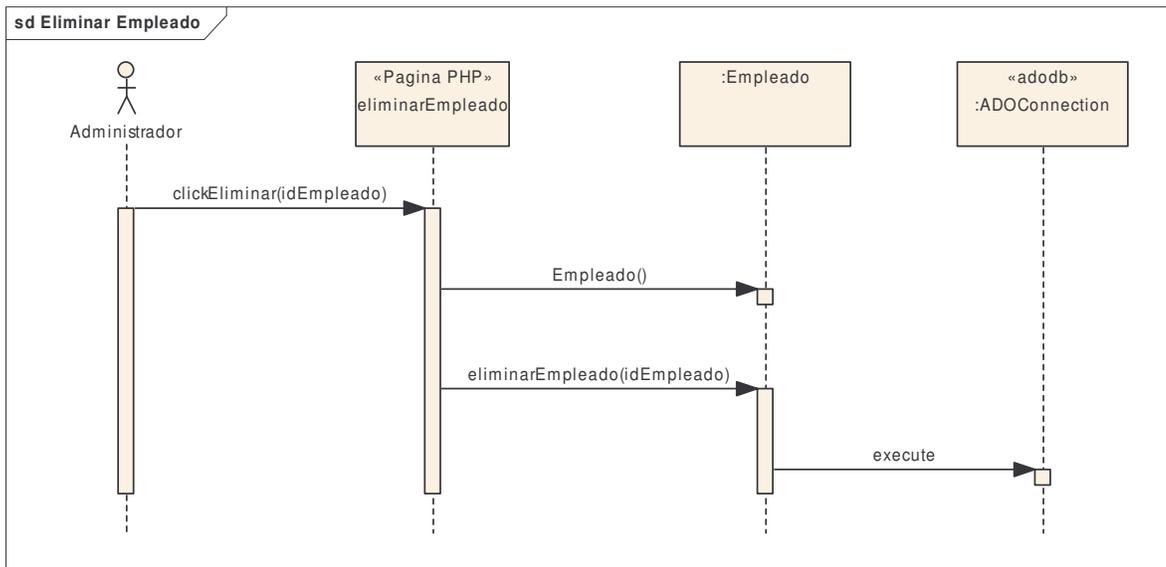


Diagrama 26. Secuencia eliminar empleado

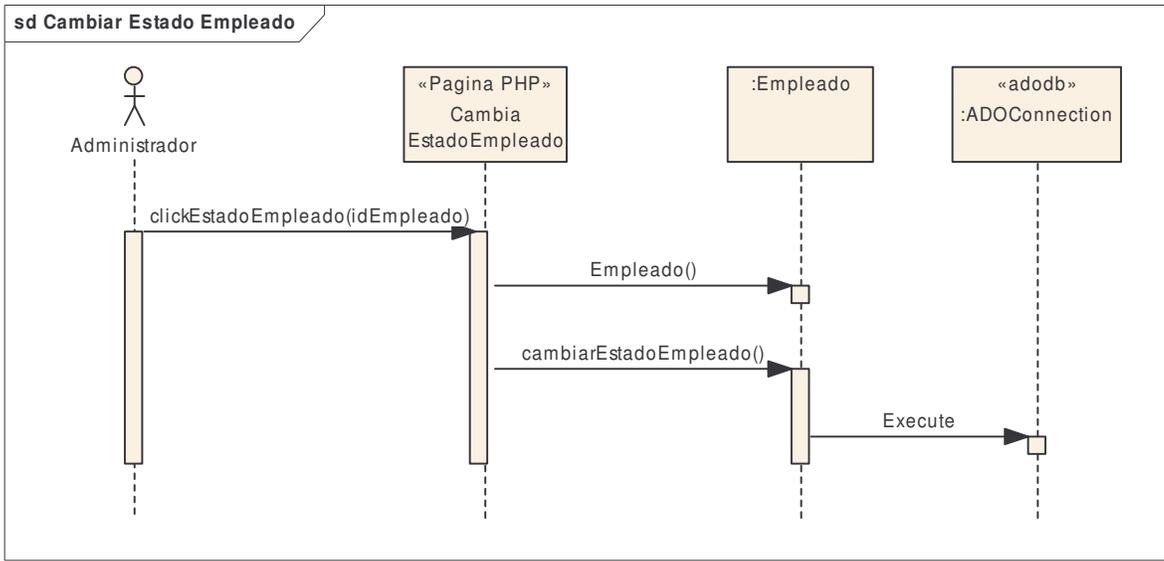


Diagrama 27. Secuencia estado empleado

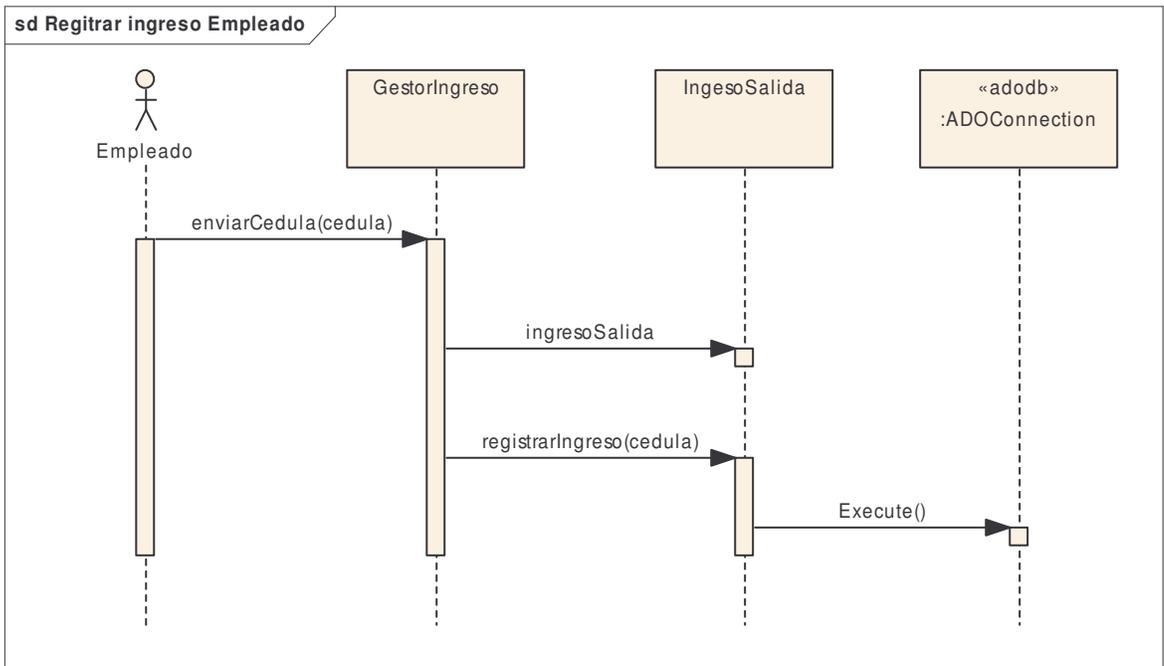


Diagrama 28. Secuencia registrar ingreso empleado

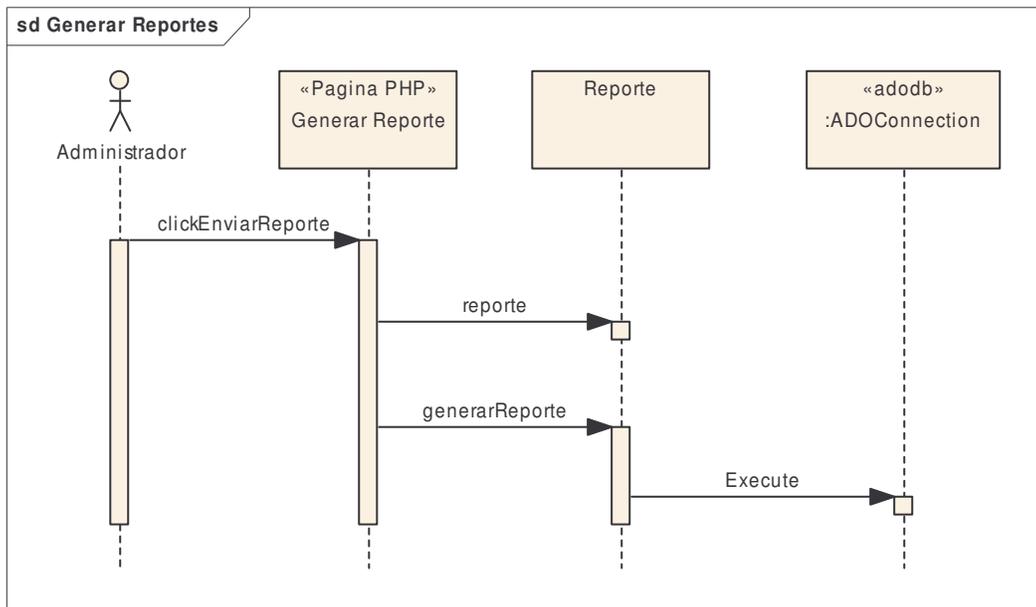


Diagrama 29. Secuencia generar reportes

8.3.2 Diagrama de clases del diseño.

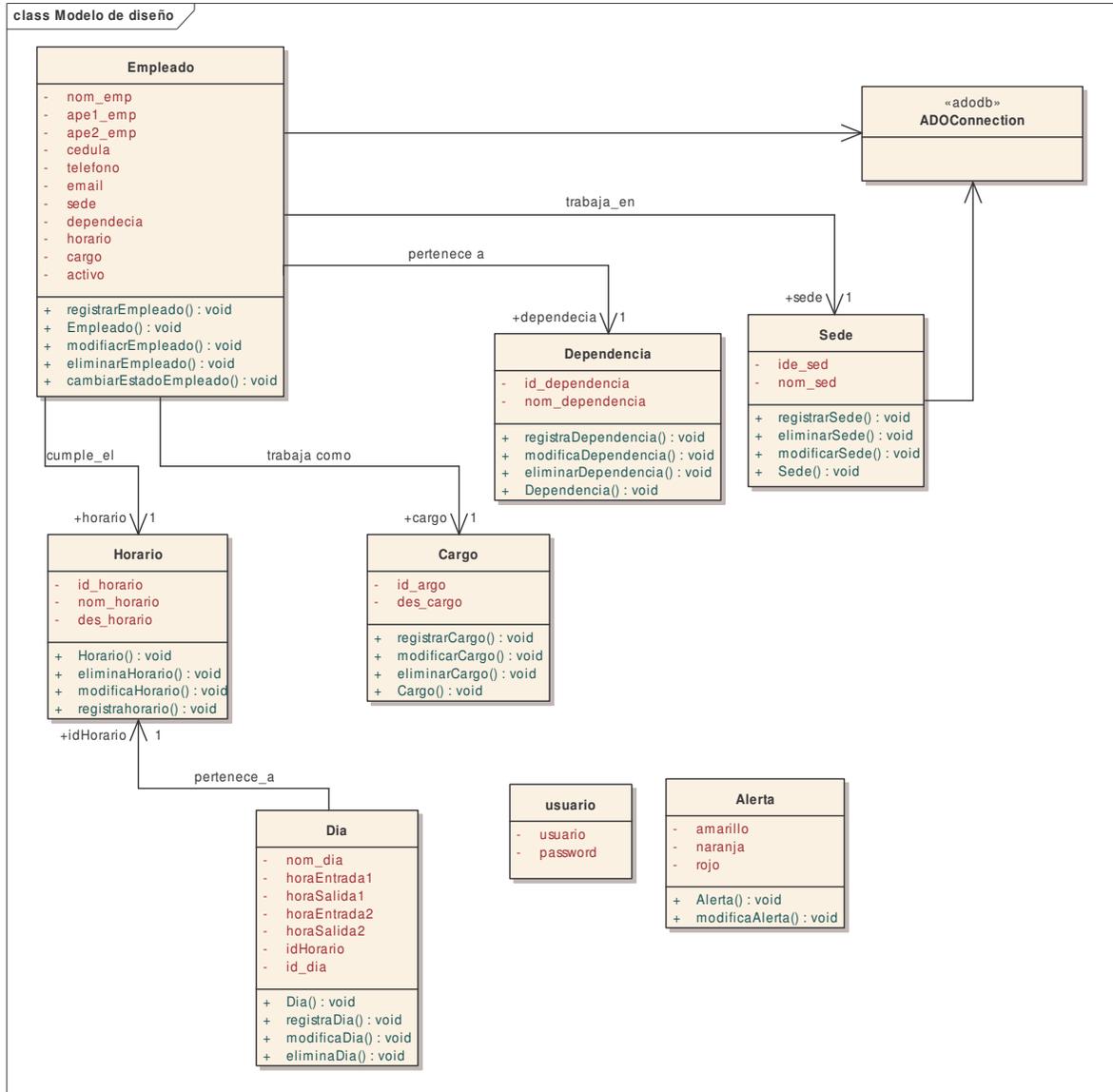


Diagrama 30. Clases del diseño

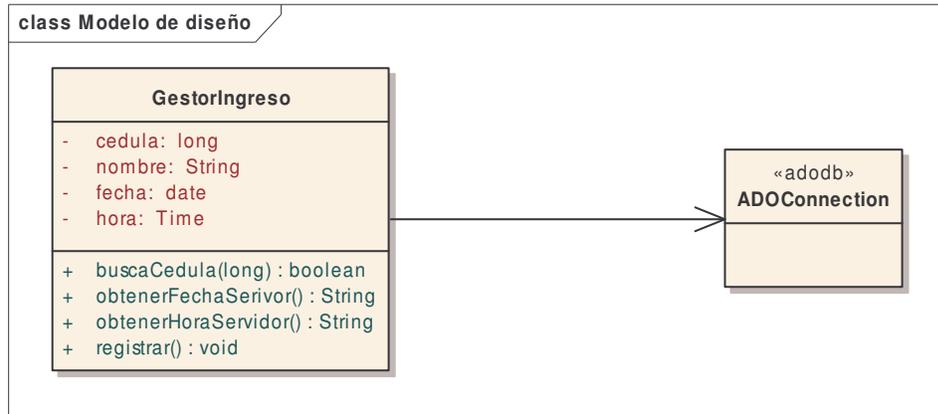


Diagrama 31. Clases de diseño 2

8.3.3 Diseño de Datos.

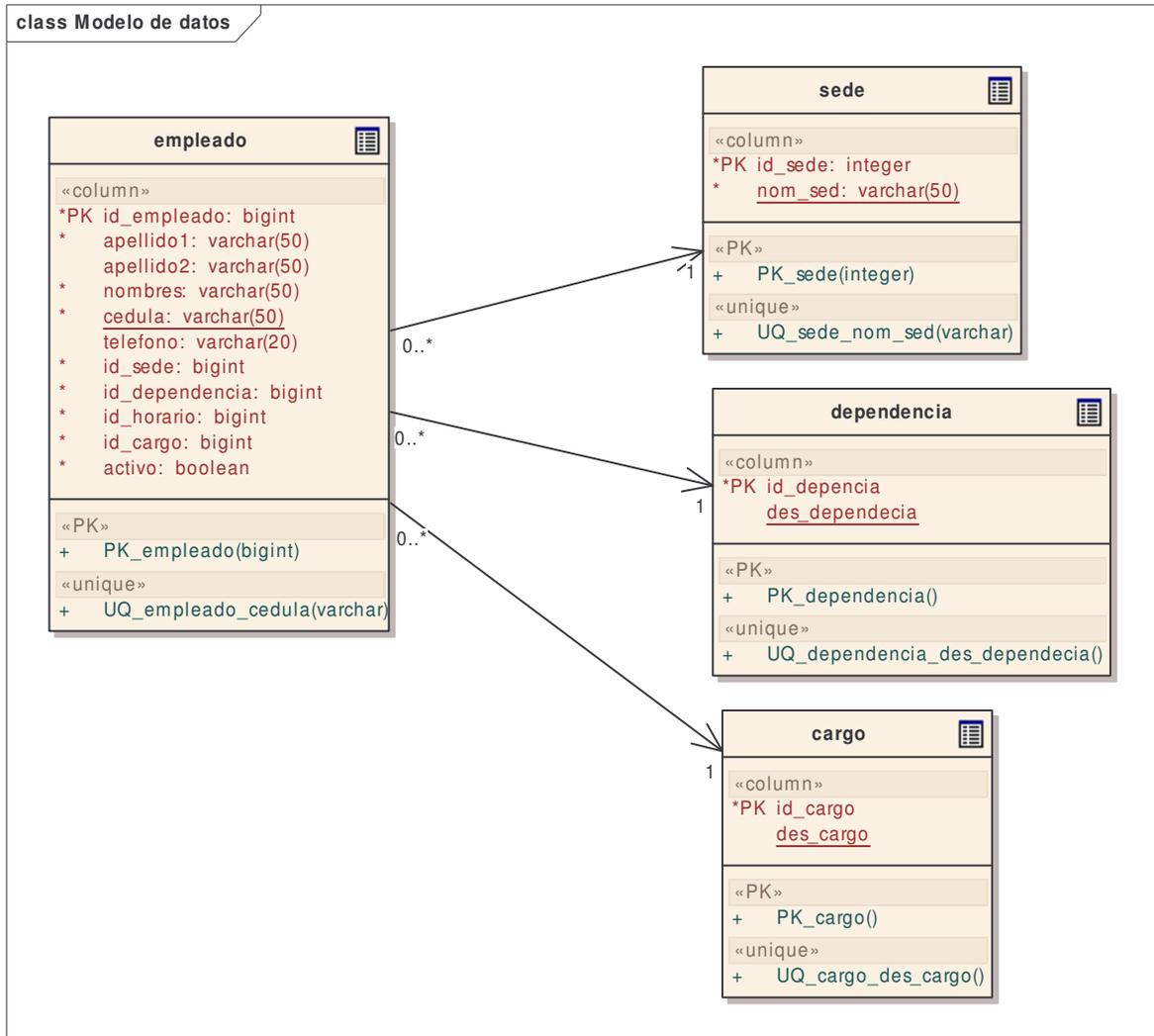


Diagrama 32. Diseño de datos 1

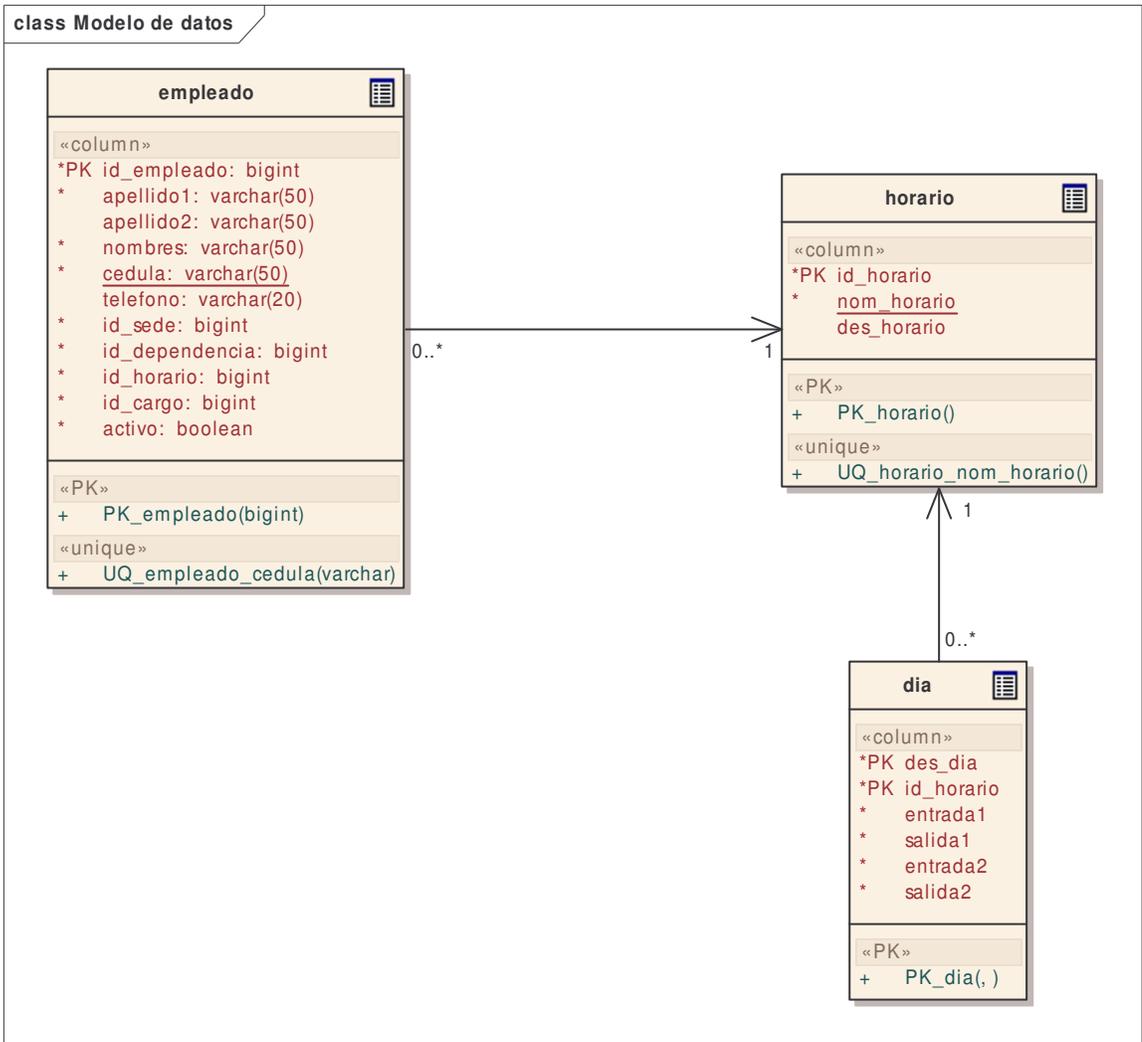


Diagrama 33. Diseño de datos 2

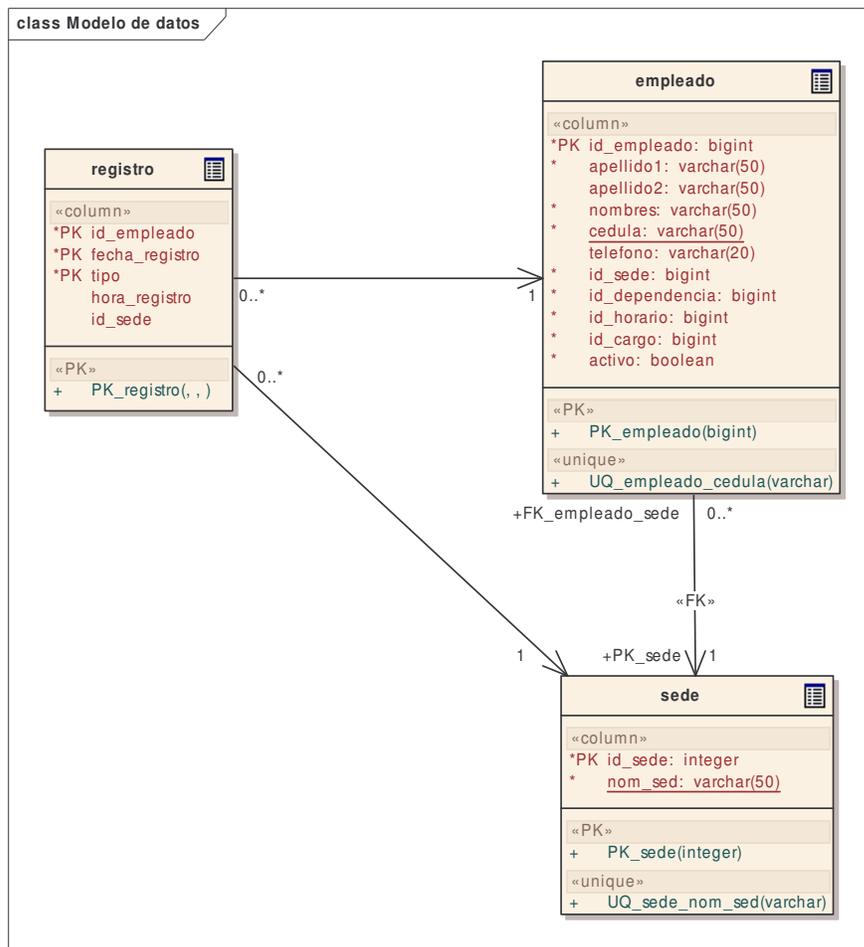


Diagrama 34. Diseño de datos 3

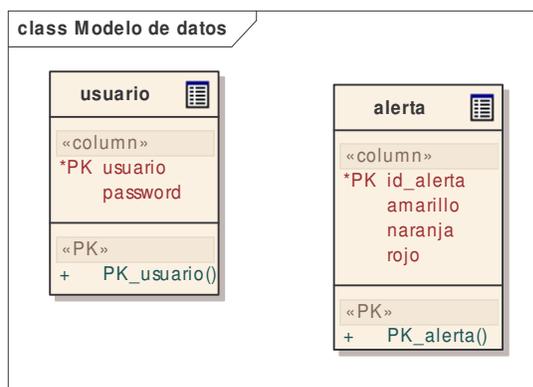


Diagrama 35. Diseño de datos 4

8.4 MODELO DE IMPLEMENTACION

8.4.1 Diagrama de paquetes

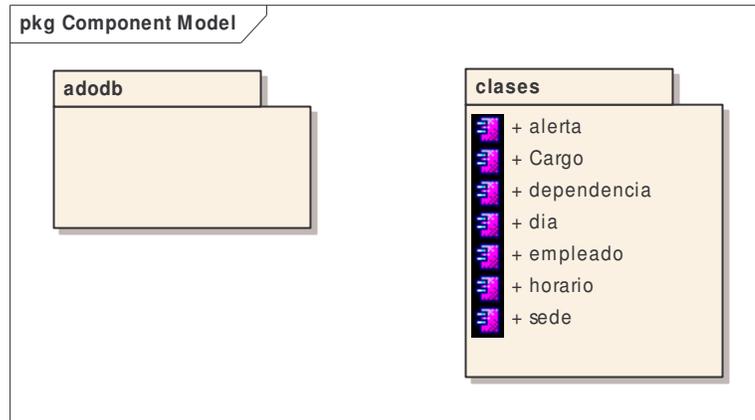


Diagrama 36. Diagrama de paquetes

8.4.1.1 Paquete clases

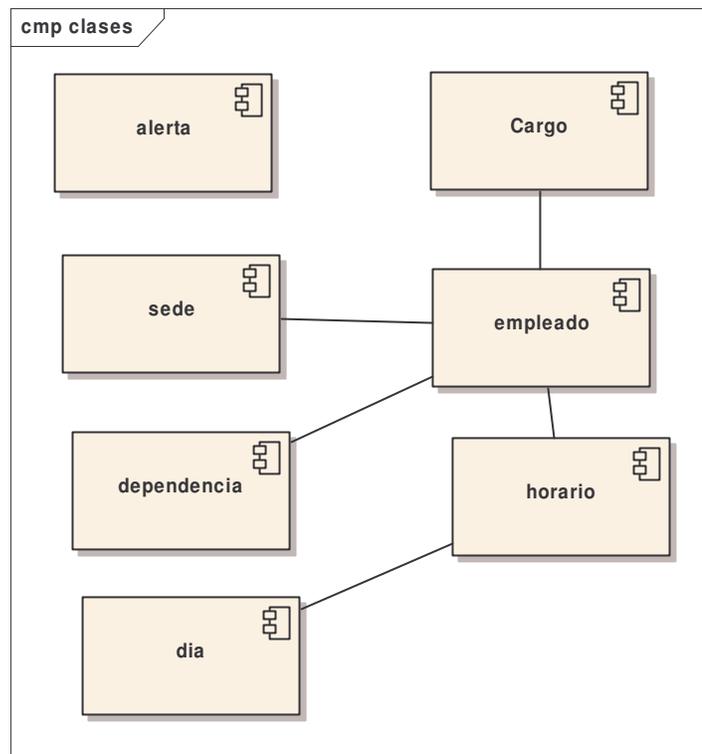


Diagrama 38. Paquetes de clase

8.5 PRUEBAS DEL SISTEMA

Tabla 3. Pruebas del sistema

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P01	Agregar Sede	Revisar si la nueva sede aparece en todas las listas donde está relacionada. (Empleado, Reportes). Dejar el campo "Nombre Sede" en blanco y dar clic en el botón agregar. Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Funciona correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P02	Modificar Sede	Dejar el campo "Nombre Sede" en blanco y dar clic en el botón Modificar Revisar si la sede aparece modificada en todas las listas donde está relacionada la sede (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Corregido

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P03	Eliminar Sede	Oprimir el botón Eliminar. Revisar si la sede desaparece de todas las listas donde está relacionada. (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Corregido

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P04	Agregar Dependencia	Dejar el campo "Nombre Dependencia" en blanco y dar clic en el botón agregar. Revisar si la nueva Dependencia aparece en todas las listas donde está relacionada. (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Funciona Correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P05	Modificar Dependencia	Dejar el campo "Nombre Dependencia" en blanco y dar clic en el botón Modificar Revisar si la Dependencia modificada aparece en todas las listas donde está relacionada la Dependencia (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Funciona Correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P06	Eliminar Dependencia	Oprimir el botón Eliminar. Revisar si la Dependencia desaparece de todas las listas donde está relacionada. (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Funciona correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P07	Agregar Cargo	Digitar nombre de Cargo en la caja de texto. Oprimir el botón Agregar Revisar si la nueva Cargo aparece en todas las listas donde está relacionada. (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P08	Modificar Cargo	Digitar nombre de Cargo en la caja de texto. Oprimir el botón Modificar Revisar si la nuevo Cargo aparece en todas las listas donde está relacionado el Cargo (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Funciona correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P09	Eliminar Cargo	Oprimir el botón Eliminar. Revisar si el Cargo desaparece de todas las listas donde está relacionada. (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Funciona correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P10	Actualizar Alertas	Se digitó caracteres erróneos (letras, etc.). Después se envió la información para que el sistema la procese y no acepto caracteres diferentes a números ni campos vacíos.	Corregido

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P11	Actualizar Contraseña	Se digitó contraseña diferente a la confirmación. Se dejaron los campos de contraseña y confirmación vacíos.	Corregido

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P12	Agregar Horario	Se dejó el nombre del horario en blanco Se digitó horas erróneas y caracteres inválidos en cada caja de texto de todos los días. Se repitió el nombre de otro horario existente.	Funciona Correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P13	Modificar Horario	Se dejó el nombre del horario en blanco Se digitó horas erróneas y caracteres inválidos en cada caja de texto de todos los días.	Funciona correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P14	Eliminar Horario	El sistema dejó eliminar el horario cuando ningún empleado estaba asociado a el	Funciona correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P15	Agregar Empleado	Dejar en blanco los datos relevantes (primer apellido, nombres, cedula) Revisar si el nuevo empleado aparece en todas las listas donde está relacionado. (Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito. Verificar si la foto queda guardada en el servidor con el nombre correcto. Verificar si las alertas de los reportes corresponden al horario asignado	Corregido

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P16	Modificar Empleado	Dejar en blanco los datos relevantes (primer apellido, nombres, cedula). Revisar si el empleado aparece en todas las listas donde está relacionado. (Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito. Verificar si la foto queda guardada en el servidor con el nombre correcto. Verificar si las alertas de los reportes corresponden al horario asignado	Corregido

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P17	Eliminar Empleado	Revisar si el empleado desaparece de todas las listas donde está relacionada. (Empleado, Reportes). Oprimir el botón Cancelar para verificar si aborta la operación con éxito.	Funciona correctamente

ID _ PRUEBA	NOMBRE PRUEBA	DESCRIPCION PRUEBA	ESTADO
P18	Generar Reportes	Revisar si los reportes generan datos coherentes y confiables. Revisar si las horas de ingreso y salida corresponden a los registros existentes. Realizar el cálculo de la suma de horas. revisar los si los sql ejecutados en el sistema gestor de bases de datos son iguales a los reportes generados en html.	Corregido

9. CONCLUSIONES

- El trabajo demostró que implementando una metodología pre establecida - UML, Ciclo de Vida en Espiral -, es posible realizar la implementación de un sistema que permite establecer un mecanismo de control en el ingreso y salida del personal de EMPOPASTO S.A. E.S.P. de manera eficiente y eficaz.
- El sistema permite el registro entrada y salida de las personal al servicio de la Empresa de manera fácil y controlada.
- Al generar reportes actualizados del ingreso y salida de los empleados de EMPOPASTO S.A. E.S.P., de acuerdo a las necesidades del área de Talento Humano, se mejora el control del personal, establece parámetros para crear una tabla estadística facilitando la toma de decisiones, y por lo tanto, permite optimizar la productividad de la empresa.
- Se contribuyó al forjamiento de una cultura de cumplimiento de los horarios establecidos por la empresa de obras sanitarias EMPOPASTO S.A. E.S.P. que mejora el ambiente laboral dentro de la empresa y el servicio al usuario.
- La arquitectura CLIENTE – SERVIDOR es muy útil y eficiente en la práctica; con la aplicación en entorno Web no es necesaria la instalación de ningún programa en el cliente para que el sistema funcione. Con un navegador, un usuario y una contraseña usted tendrá acceso al sistema desde cualquier estación de trabajo desde cualquier lugar del mundo, lo cual lo hace muy atractivo a los usuarios finales.

10. RECOMENDACIONES

- Mantener un contacto permanente entre el analista y el usuario final del programa con el objeto de crear un canal de comunicación que permita una información oportuna y consistente de los requerimientos administrativos del software.
- Implementar un Lector Biométrico de huella digital para garantizar la efectividad del control de ingresos y salidas.
- Llevar una metodología estricta del desarrollo de software, conlleva a la construcción de herramientas con alto nivel de calidad satisfaciendo las necesidades puntuales de una organización de acuerdo a los estándares administrativos.
- Elegir el lenguaje de programación con base a los requerimientos de hardware y software y la habilidad del programador con el lenguaje establecido para desarrollar el programa es una decisión muy importante, ya que de utilizar un lenguaje no adecuado conllevaría a generar un conflicto o inconsistencias en el sistema y a retrasar el cronograma de actividades en un tiempo significativo del proyecto.

REGISTRO BIBLIOGRAFIA

- Larman, Craig. UML y Patrones – Introducción al Análisis y Diseño Orientado a Objetos, Prentice Hall, México, 1999
- Reglamento Interno de Trabajo de EMPOPASTO SA ESP. Acuerdo No 12 de 2005.

WEBGRAFIA

- http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado
- <http://es.wikipedia.org/wiki/.php>
- <http://www.sobl.org/traduccion/practical-postgres/node12.html>
- <http://www.getec.etsit.upm.es/docencia/gproyectos/planificacion/cvida.htm>
- <http://es.wikipedia.org/wiki/Cliente-servidor>
- <http://es.wikipedia.org/wiki/AJAX>
- <http://www.desarrolloweb.com/articulos/497.php>

ANEXOS

ANEXO A. MANUAL DEL SISTEMA

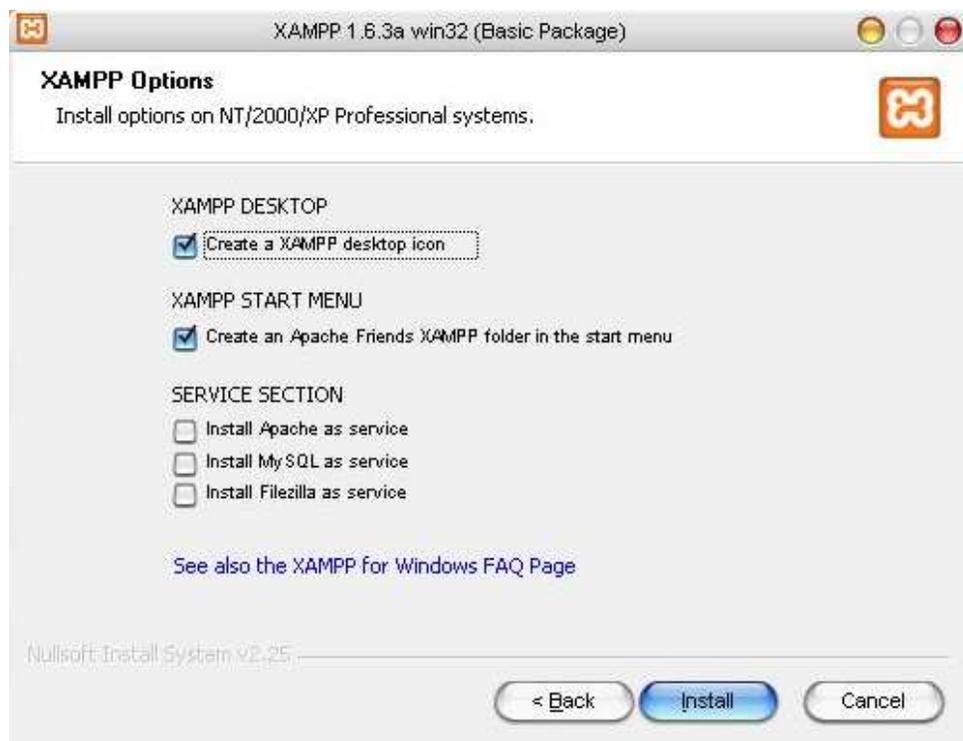
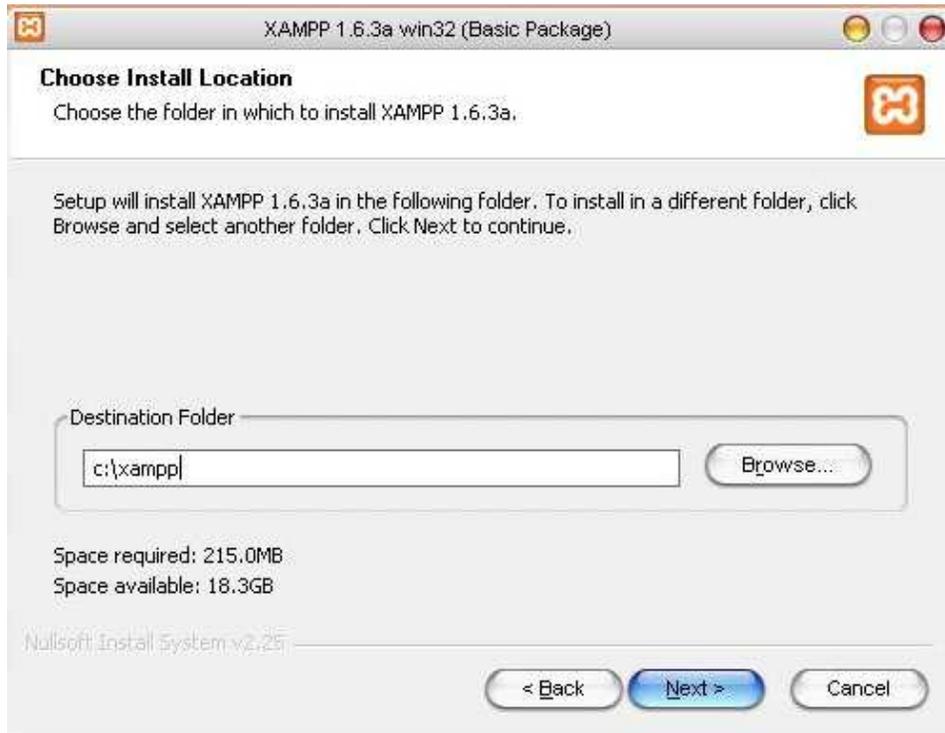
1. Instalar APACHE

1.1. Ejecutar el programa

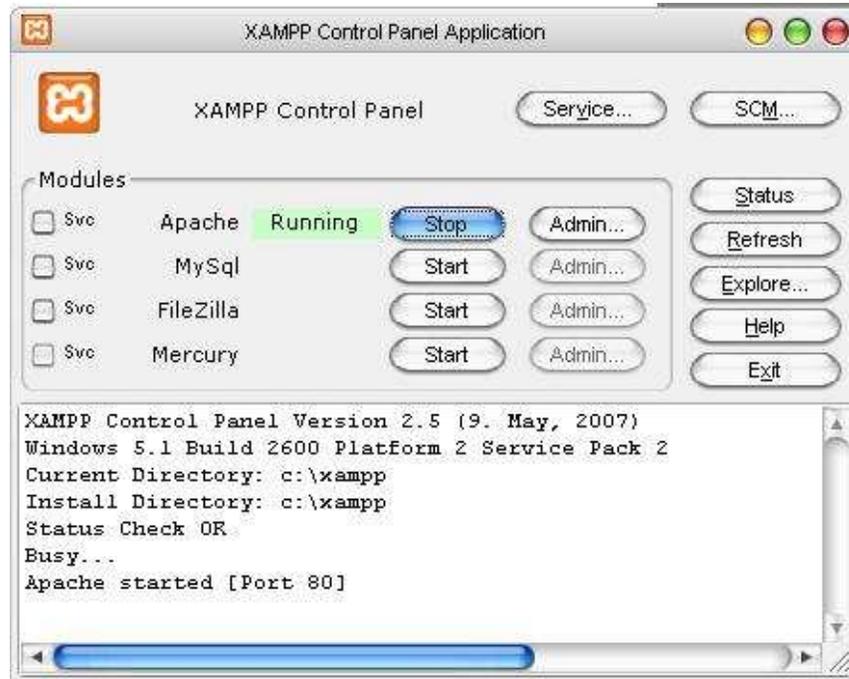
xampp-win32-1.6.3a-installer 34.258 KB Aplicación

Seguir las siguientes instrucciones.





Iniciar el servicio (Start) APACHE



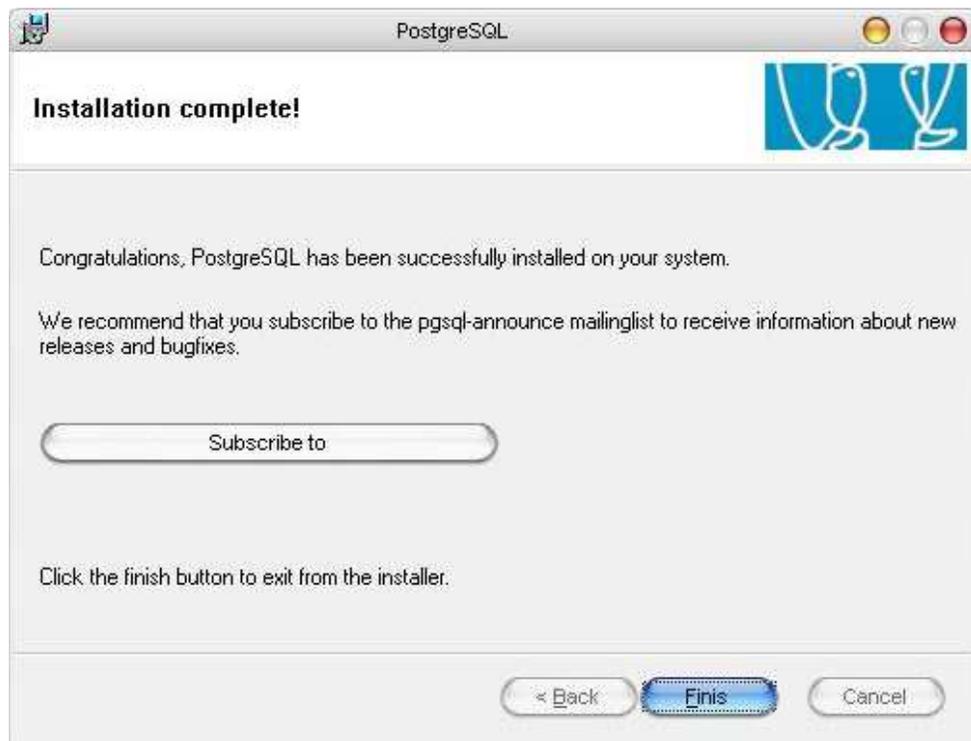
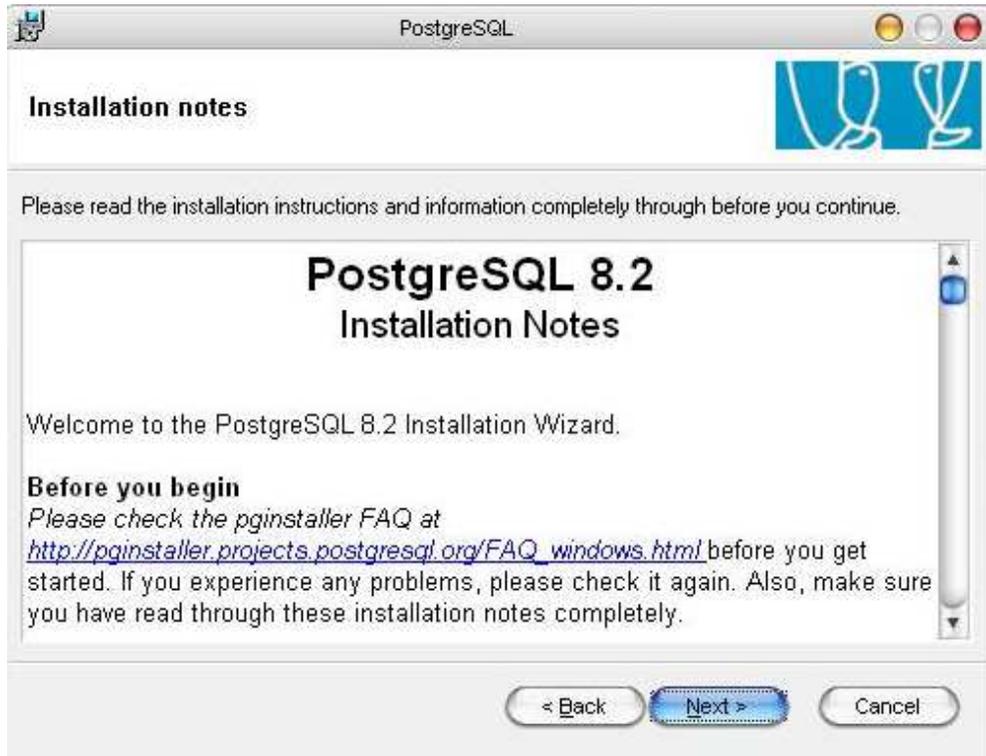
2. Instalar POSTGRES

Leer archivo readme.txt

postgresql-8.2	134 KB	Paquete de Window...	20/04/2007 12:06
postgresql-8.2-int	26.973 KB	Paquete de Window...	20/04/2007 12:06
README	1 KB	Documento de texto	20/04/2007 9:24
upgrade	1 KB	Archivo por lotes M...	04/12/2006 11:02

Ejecutar el archivo:

postgresql-8.2	134 KB	Paquete de Window...	20/04/2007 12:06
----------------	--------	----------------------	------------------



Ejecutar los otros archivos.

 postgresql-8.2-int	26.973 KB	Paquete de Window...	20/04/2007 12:06
 README	1 KB	Documento de texto	20/04/2007 9:24
 upgrade	1 KB	Archivo por lotes M...	04/12/2006 11:02

3. Instalar Base de datos.

Descomprimir el archivo base.zip

 BASE.zip	38 KB	WinZip File	22/04/2008 04:07 p.m.
--	-------	-------------	-----------------------

Abrir pgadmin (Administrador de postgres)

Crear una conexión

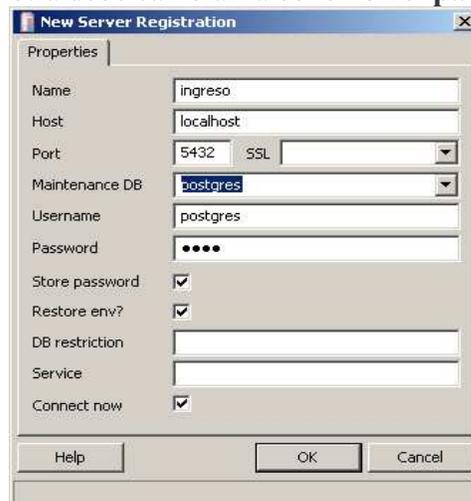


Nombre conexión: ingreso (o cualquiera es indiferente)

Host: localhost (obligatorio)

Username: postgres

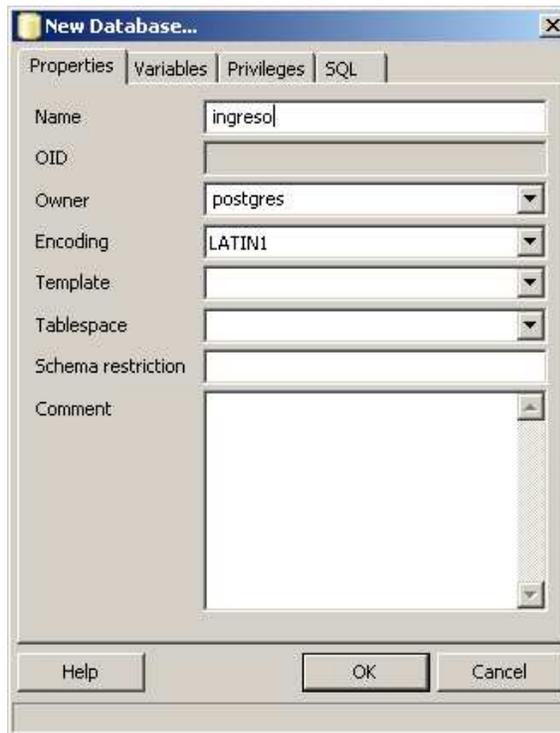
Password: 1234 (si ingresa otra debe cambiar la conexión el **paquete ingreso**)



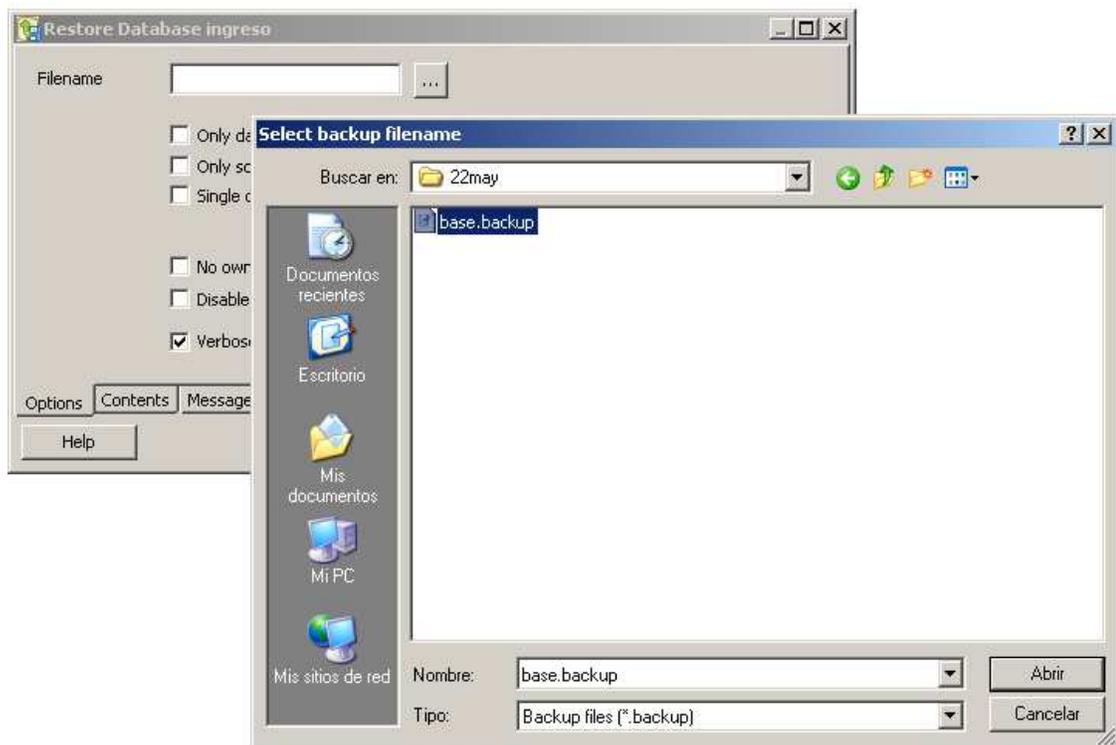
Crear Base de datos ingreso



name: ingreso
Owner: Postgres
Encoding: Latin1



Restaurar la base de datos con el scrip base.backup



Actualizar los cambios realizados.

4. Montar programa administrador en el APACHE

Descomprimir el archivo ingresoPHP.zip

 ingresoPHP.zip	3.851 KB	WinZip File	18/05/2008 09:18 p.m.
--	----------	-------------	-----------------------

Copiar la carpeta ya descomprimida **ingreso** en c:\xampp\htdocs\

En el navegador entrar a <http://localhost/ingreso/>

Usuario: empopasto

Contraseña: talento humano

NOTA: si la conexión de la aplicación con la base de datos no se establece, verificar si el usuario y la contraseña del archivo ubicado en ./ingreso/conexion.php coincidan con el usuario y contraseña con que se creo la base de datos en postgres.

6. Instalar Java

Ejecutar el archivo:

 jdk-1_5_0_11-nb-5_5-win-mi	138.676 KB	Aplicación
--	------------	------------

Se instala el java y el NETBEANS

Descomprimir el archivo IngresosJAVA.zip

Y ejecutar el archivo: Ingresos\dist\ingresos.jar que es el programa de registro de horas y salidas de los empleados de EMPOPASTO SA ESP

ANEXO B. MANUAL DE USUARIO

1. INICIAR SESION

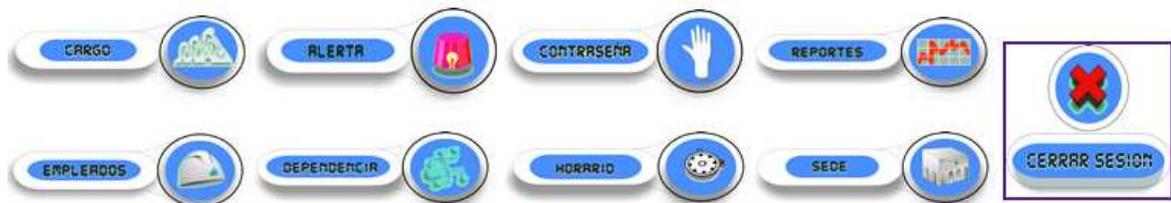
Ingresar en el navegador la página www.empopasto.com.co/ingreso

USUARIO:	<input type="text"/>
PASSWORD:	<input type="password"/>
INGRESAR	CANCELAR

Debe digitar el Nombre de Usuario y la Contraseña, luego oprimir el botón Ingresar (no importa que sean mayúsculas o minúsculas).

Si los datos son correctos, el sistema mostrará un mensaje de bienvenida, de lo contrario aparecerá una alerta indicando el error.

2. MENU PRINCIPAL



Después de ingresar al sistema, aparecerá el menú principal que permite administrar los diferentes elementos del sistema tales como sedes, dependencias, cargos, empleados, contraseña, horarios, alertas, sacar reportes y cerrar sesión.

3. SEDE

Al dar clic en el botón Sede, permite administrar las sedes: crear, modificar, eliminar. Tal y como se muestra a continuación.

Adicionar Sede

Actualizar Sedes

Administrar Sede

Modificar Registro Sede

SEDE	MODIFICAR	ELIMINAR
CASONA	EDITAR	Eliminar Sede
MITTAYO	EDITAR	Eliminar Sede
SAN FELIPE	EDITAR	Eliminar Sede
CENTENARIO	EDITAR	Eliminar Sede

4. DEPENDENCIA

Al dar clic en el botón Dependencia, permite administrar las Dependencias: crear, modificar, eliminar. Tal y como se muestra a continuación.

Administrar Dependencia

Agregar Dependencia

Actualizar Dependencias

Modificar Dependencia

DEPENDENCIA	MODIFICA	ELIMINA
SUBGERENCIA COMERCIAL	EDITAR	Elimina Dependencia
JEFATURA OPERATIVA DE REDES	EDITAR	Elimina Dependencia

5. CARGO.

Al dar clic en el botón Cargo, permite administrar los Cargos: crear, modificar, eliminar. Tal y como se muestra a continuación.

CARGO	MODIFICAR	ELIMINAR
ASISTENTE I	EDITAR	삭제
ASISTENTE II	EDITAR	삭제

6. HORARIO.

Al dar clic en el botón Horario, permite administrar los Horarios: crear, modificar, eliminar. Tal y como se muestra a continuación.

HORARIO	MODIFICAR	ELIMINAR
ADMINISTRATIVA	EDITAR	삭제
AYUDANTES Y AUX DE SERV GRALES	EDITAR	삭제

Adicionar Horario: Para crear un nuevo horario digite el nombre del horario, las horas de ingreso y salida de los días en la semana en que tiene que trabar (las horas se marcan de manera militar ej: las dos de la tarde se digitan 14:00) y por ultimo una descripción del horario (opcional). Lugo oprima el botón Agregar, si los datos son correctos aparece un mensaje de confirmación de la operación, de lo contrario el sistema mostrara un mensaje de alerta.



Nombre Horario: PRUEBA

Repetir	Día	Entrada1	Salida1	Entrada2	Salida2
<input type="checkbox"/>	Lunes:	08:00	12:00	14:00	18:00
<input type="checkbox"/>	Martes:				
<input type="checkbox"/>	Miercoles:	07:30	12:15	13:45	18:15
<input type="checkbox"/>	Jueves:				
<input type="checkbox"/>	Viernes:	08:10	11:50	14:10	17:50
<input type="checkbox"/>	Sabado:				

Descripción: PRUEBA DE INGRESO DE HORARIOS

7. EMPLEADO

Al dar clic en el botón Empleado, permite administrar los Empleados: crear, modificar, eliminar. Tal y como se muestra a continuación.

PRIMER APELLIDO	SEGUNDO APELLIDO	NOMBRES	SEDE	DEPENDENCIA	MODIFICA	ELIMINA	ACTIVO
ACOSTA	GOMEZ	JAIME	CASONA	GERENCIA	EDITAR	삭제	✓
ANDRADE	AREVALO	LILIANA	CASONA	OFICINA ASESORA JURIDICA	EDITAR	삭제	✓
ARCOS	BOLAÑOS	EVELIO	CASONA	SUBGERENCIA COMERCIAL	EDITAR	삭제	✓

Agregar Empleado:

Form fields and controls:

- 1) Primer Apellido: [Input field]
- 2) Segundo Apellido: [Input field]
- 3) Nombres: [Input field]
- 4) Cedula: [Input field]
- 5) Telefono: [Input field]
- 6) e-mail: [Input field]
- 7) Dirección: [Input field]
- 8) Sede: [Dropdown menu, selected: CASONA]
- 9) Dependencia: [Dropdown menu, selected: CONTADOR]
- 10) Cargo: [Dropdown menu, selected: ASISTENTE I]
- 11) Horario: [Dropdown menu, selected: ADMINISTRATIVA]
- 12) Foto: [Input field] [Examinar...]
- 13) Activo: Si No

Buttons: [Agregar] [Cancelar]

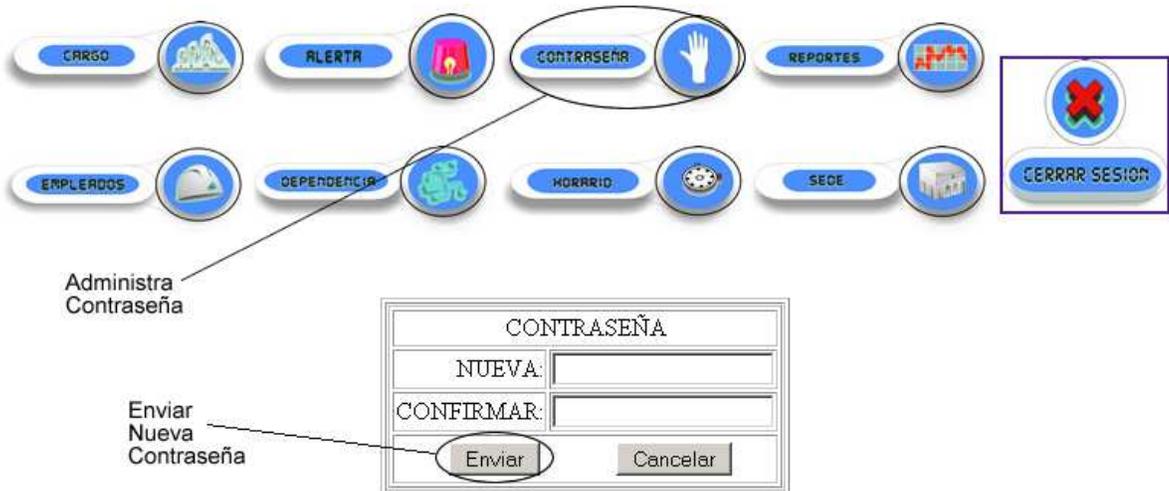
- 1) Primer Apellido: este campo es obligatorio (Letras)
- 2) Segundo Apellido: este campo es opcional (Letras)
- 3) Nombres: este campo es obligatorio (Letras)
- 4) Cedula: este campo es obligatorio (Números)
- 5) Teléfono: este campo es opcional (Números)
- 6) E-mail: este campo es opcional (Letras, números, símbolos)
- 7) Dirección: este campo es opcional (Letras, números, símbolos)
- 8) Sede: Debe seleccionar una sede de la lista.
- 9) Dependencia: Debe seleccionar una dependencia de la lista.
- 10) Cargo: Debe seleccionar un cargo de la lista.

- 11) Horario: Debe seleccionar un horario de la lista.
- 12) Foto: Debe seleccionar una foto de 200x200 px aproximadamente (Opcional)
- 13) Debe seleccionar la opción si el empleado esta activo o inactivo, por defecto aparece Activo.

Por ultimo debe oprimir el botón **Agregar** y si los datos están bien diligenciados el sistema indicará un mensaje de confirmación, de lo contrario indicará un mensaje de alerta.

8. CONTRASEÑA

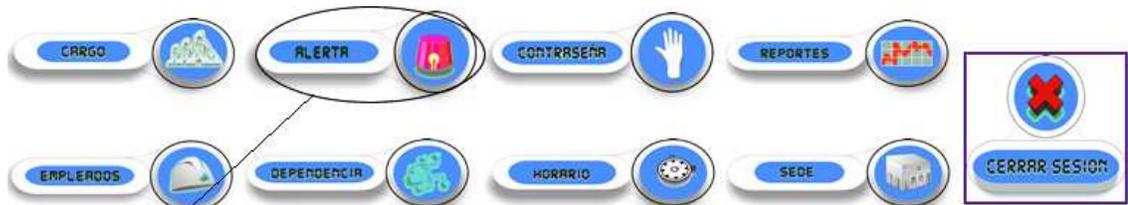
Para el cambio de contraseña debe entrar a la opción contraseña, digitar la nueva contraseña, confirmarla y oprimir el botón **Enviar**. Si los datos son correctos la contraseña será actualizada inmediatamente.



9. ALERTA

Parámetro que permite visualizar de forma gráfica que tan tarde llegó o que tan temprano salió un empleado de la Empresa.

Por ejemplo:



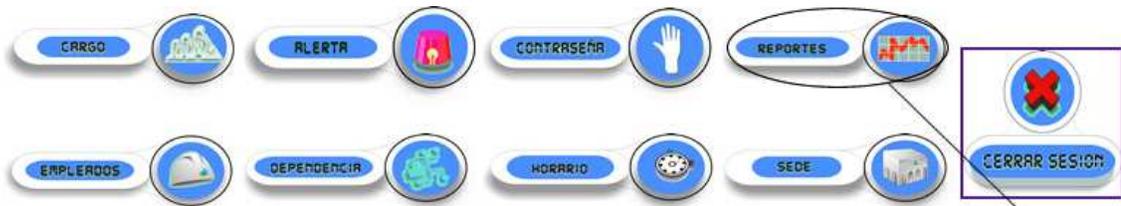
Administrar Alerta

Enviar Datos de Alertas

Alertas		
	15	Minutos
	30	Minutos
	45	Minutos
<input type="button" value="Enviar"/>		<input type="button" value="Cancelar"/>

Aquí indica que si el empleado llega entre 1 y 15 minutos tarde en el reporte va a apareces en amarillo, si llega entre 16 y 30 minutos tarde en el reporte va a apareces en naranja y si se demora mas de 30 minutos aparece en rojo.

10. GENERAR REPORTES



Administrador de Reportes

REPORTES	
<input checked="" type="checkbox"/> Fecha	Fecha Inicio: <input type="text"/> ...
	Fecha Fin: <input type="text"/> ...
<input type="checkbox"/> Empleado:	ANDRADE AREVALO LILIANA
<input type="checkbox"/> Sede:	CASONA
<input type="checkbox"/> Dependencia:	CONTADOR
<input type="checkbox"/> Cargo:	ASISTENTE I
<input type="checkbox"/> No marcan	
<input type="checkbox"/> Suma de Horas	
<input type="checkbox"/> Discriminado	
<input type="button" value="Enviar"/> <input type="button" value="Cancelar"/>	

Para poder generar un reporte es necesario seleccionar obligatoriamente la **“Fecha de inicio” y “Fecha fin”**.

Opciones de Reportes:

El sistema puede generar reportes personalizados dependiendo las necesidades del usuario.

Si desea el reporte de **un empleado en particular** es necesario escoger el Check Box Empleado y seleccionarlo de la lista de empleados.

Si desea el reporte de los **empleados que pertenecen a una sede** especifica se debe escoger el Check Box Sede y seleccionarla de la lista de Sedes.

Si desea el reporte de los **empleados que pertenecen a una Dependencia** especifica se debe escoger el Check Box dependencia y seleccionarla de la lista de Dependencias.

Si desea el reporte de los **empleados que pertenecen a un Cargo** especifico se debe escoger el Check Box cargo y seleccionarlo de la lista de Cargos.

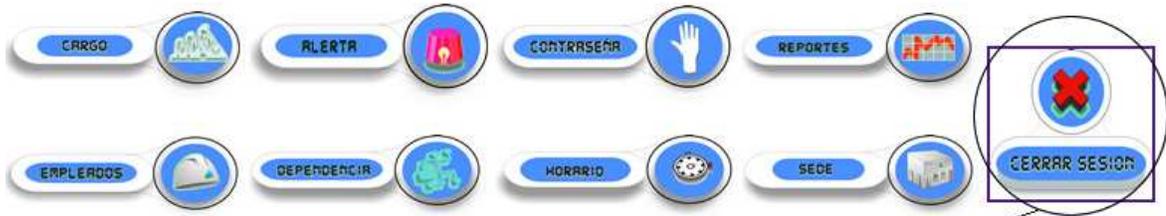
Para generar el reportes de los **empleados que no marcan** seleccionando el **check box No Marcan**.

Para generar el reportes de la **suma de horas trabajadas de los empleados** seleccionando el **check box Suma de Horas**.

Para generar el reportes de las **Horas de Ingresos y salidas de los empleado** seleccionando el **check box Discriminado**.

11. CERRAR SESIÓN.

Permite salir del sistema de forma segura.



Cerrar
Sesión