

**SISTEMA DE INFORMACIÓN WEB DE REGISTRO Y CONTROL PARA LA
PROGRAMACIÓN DE CURSOS EN EL SERVICIO NACIONAL DE APRENDIZAJE
SENA - CENTRO MULTISECTORIAL LOPE**

JUAN PABLO GOMEZ

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2007**

**SISTEMA DE INFORMACIÓN WEB DE REGISTRO Y CONTROL PARA
PROGRAMACIÓN DE CURSOS EN EL SERVICIO NACIONAL DE APRENDIZAJE
SENA - CENTRO MULTISECTORIAL LOPE**

JUAN PABLO GOMEZ

**Trabajo de Grado presentado como requisito parcial para optar el título de
INGENIERIO DE SISTEMAS**

**Asesor
ING. FRANKLIN JIMENEZ**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2007**

Nota de aceptación:

El trabajo de grado desarrollado bajo el nombre de "SISTEMA DE INFORMACIÓN WEB DE REGISTRO Y CONTROL PARA LA PROGRAMACIÓN DE CURSOS EN EL SERVICIO NACIONAL DE APRENDIZAJE SENA - CENTRO MULTISECTORIAL LOPE", presentado como requisito parcial para optar el título de Ingeniero de Sistemas, fue aprobado por su jurado calificador.

Atentamente,

Ing. Franklin Jiménez
Jurado

Ing. Oscar Ceballos
Jurado

San Juan de Pasto, Noviembre 15 de 2007

AGRADECIMIENTOS

Expreso mis sinceros agradecimientos a Dios por su fidelidad y respaldo a mi trabajo, a mi padre HERNANDO GÓMEZ, a mi madre ALBA LUCIA CALVACHE y a mi hermana CATALINA GÓMEZ por su comprensión y apoyo incondicional.

También agradezco al ING. ROBERTO PANTOJA y los Jurados OSCAR CEBALLOS Y FRANKLIN JIMENEZ por su colaboración y disposición a lo largo del desarrollo de este trabajo.

CONTENIDO

	pag.
INTRODUCCIÓN	13
1. ANTECEDENTES	15
2. RECURSOS	16
2.1 RECURSOS HUMANOS.....	16
2.2 RECURSOS TECNOLOGICOS	16
2.2.1 Hardware.....	16
2.2.2 Software	16
2.3 RECURSOS MATERIALES	16
2.4 RECURSOS FINANCIEROS.....	16
2.5 RECURSOS OPERATIVOS.....	17
3. MARCO TEÓRICO.....	18
3.1 RED DE ÁREA LOCAL (LAN).....	18
3.2 BASE DE DATOS	19
3.3 APLICACIÓN WEB	19
3.4 INTERNET INFORMATION SERVICES (IIS).....	20
3.5 ASP.NET	21
4. METODOLOGÍA.....	22

4.1	PROGRAMACIÓN EXTREMA (<i>EXTREME PROGRAMMING</i> , XP)	22
4.1.1	Las Historias de Usuario	23
4.1.2	Roles XP	23
4.1.3	Proceso XP	24
4.1.4	Fase I: Exploración.....	24
4.1.5	Fase II: Planificación de la Entrega	25
4.1.6	Fase III: Iteraciones	25
4.1.7	Fase IV: Producción	25
4.1.8	Fase V: Mantenimiento	26
4.1.9	Fase VI: Muerte del Proyecto	26
4.1.10	Prácticas XP	26
4.1.11	El juego de la planificación	26
4.1.12	Entregas pequeñas	26
4.1.13	Metáfora	26
4.1.14	Diseño simple.....	27
4.1.15	Pruebas	27
4.1.16	Refactorización (<i>Refactoring</i>).....	27
4.1.17	Programación en parejas	27
4.1.18	Propiedad colectiva del código.....	28
4.1.19	Integración continua.....	28

4.1.20 40 horas por semana	28
4.1.21 Cliente in-situ.....	28
4.1.22 Estándares de programación	29
4.1.23 Comentarios respecto de las prácticas	29
5. DESARROLLO DEL APLICATIVO	30
5.1 HISTORIAS DE USUARIO Y TAREAS DE PROGRAMACIÓN	31
5.2 CASOS DE USO DEL SISTEMA	48
5.2.1 Lista de actores	48
5.3 DIAGRAMA DE CASOS DE USO	58
5.4 MODELO CONCEPTUAL	59
5.5 DIAGRAMA DE CLASES	60
5.6 DISEÑO DE LA BASE DE DATOS	63
5.6.1 Diagrama entidad relación.....	64
5.6.2 Tablas de la base de datos	65
5.7 GLOSARIO.....	69
5.7.1 Casos de uso	69
5.7.2 Conceptos	70
5.7.3 Clases métodos y atributos	71
5.7.4 Parámetros.....	82
6. CONCLUSIONES.....	85

7. RECOMENDACIONES	86
BIBLIOGRAFÍA.....	87

LISTA DE FIGURAS

	pag.
Figura 1. Diagrama de casos de uso.....	58
Figura 2. Modelo conceptual.....	59
Figura 3. Diagrama de clases.....	60
Figura 4. Diagrama entidad relación.....	64

LISTA DE ANEXOS

	pag.
Anexo A. MANUAL DEL SISTEMA.....	89

RESUMEN

El Sena - centro multisectorial Lope, como entidad educativa, realiza un proceso de programación de cursos que consiste esencialmente en asignar instructores, aulas, fechas y horarios de manera previa al inicio de los cursos planeados para el año, sin embargo, las condiciones tecnológicas de Sena relacionadas con el hardware y el software, no se han usado para solucionar las dificultades y optimizar el proceso. El trabajo presente consistió en tomar el proceso de programación de cursos y llevarlo a una aplicación Web aprovechando la red local y la plataforma .Net de Microsoft, legalmente disponible por Sena.

En la construcción de software es importante definir la metodología de desarrollo y conocer los conceptos teóricos que envuelven al proyecto, este trabajo habla claramente sobre ello.

ABSTRACT

The Sena – Lope Multicectorial Center as an educative entity, make a programming process of curses that consists essentially assigning instructors, classrooms, dates and schedules in a previous manner to the opening of the courses planed to the year, nevertheless the Sena's technological conditions related with hardware and software, have not been used to solve the difficulties and to optimize the process. The present work consisted in to take the programming process of curses to a web application taking advantage the local network and the platform .Net of Microsoft, legally available by The Sena.

In the construction of software is important to define the methodology of development and to know the theoretical concepts that wrap the project; this work speaks clearly about it.

INTRODUCCIÓN

Hoy en día el manejo de la información se ha constituido en uno de los aspectos más importantes en relación con la eficiencia que las empresas buscan para alcanzar sus objetivos. La incorporación de sistemas computacionales y software en las empresas, sin lugar a duda ha demostrado ser una excelente herramienta para la administración de la información, garantizando la reducción de esfuerzos y recursos.

Con base en lo anterior y frente a las necesidades manifiestas en el área de coordinación académica del SENA se propone el trabajo Sistema de información Web de registro y control para la programación de cursos en el servicio nacional de aprendizaje SENA – centro multisectorial Lope en modalidad de pasantía, con el fin de solucionar las deficiencias y optimizar el manejo que se le da a la información en dicho proceso.

El equipo de coordinación académica del centro multisectorial Lope está encargado de elaborar anualmente la programación de cursos, es decir, llevar a cabo una planificación de fechas y horarios de los cursos que se van a dictar en el año según la disponibilidad de aulas e instructores, para ello, era preciso contar con el catálogo de cursos o estructura curricular de cursos. La manipulación de estos documentos presentaba dificultad particularmente realizando búsquedas pues el tamaño de la información es muy grande (se encuentra registrada en archivos de Excel), así mismo, cuando empezaba a aumentar los cursos que se programaban era difícil validar la disponibilidad de aulas e instructores y que no se presenten cruces de horario. El centro multisectorial Lope está dividido en tres sectores ubicados en áreas físicas distantes unos de otros, de tal manera que los instructores de cada sector realizaban la programación de cursos de forma diferente y el proceso no se estandarizaba, dificultando la generación de reportes, es por esto que se planteó un sistema Web que realice el proceso de programación de cursos. El ámbito de funcionamiento del sistema de programación de cursos es el centro multisectorial Lope del Sena, en donde se encuentra instalada una red local que conecta los equipos de las diferentes oficinas. El sistema Web incluye la documentación correspondiente al catálogo de cursos, un calendario anual que determina las fechas establecidas de actividades extraordinarias y datos particulares de aulas e instructores.

El sistema quedó integrado de la siguiente manera: módulo de registro y control de cursos, módulo de registro y control de instructores, módulo de registro y control de aulas, módulo de registro y control de usuarios, módulo de programación de cursos y módulo de reportes. Para la construcción de la aplicación se utilizó el entorno .NET FrameWork 1.1 y el gestor de bases de datos MySql 5.0.

Este documento contiene la información relativa al desarrollo del proyecto, el marco teórico, la metodología de desarrollo utilizada, y la documentación generada tras la construcción de la aplicación web.

1. ANTECEDENTES

La programación de cursos en el Sena no se ha formalizado como un procedimiento sistemático, sin embargo los coordinadores académicos se han apoyado en herramientas software como Project y Excel haciendo uso de estos de una forma rudimentaria.

El catálogo de cursos se encuentra registrado en archivos de Excel, de los cuales se adquiere la información necesaria para la apertura de cursos. Por cada curso abierto, por cada aula asignada y por cada instructor asignado se construye un diagrama de Gantt teniendo en cuenta el calendario de días extraordinarios en los cuales no se dictan clases, en base a esto se determina la fecha de inicio y se calcula la fecha de terminación de cada materia del curso correspondiente y se definen los horarios de clase. Estos diagramas de Gantt se convierten en los reportes de la programación de cursos de los cuales se obtiene la información que el sistema de gestión e información académica a nivel nacional del Sena requiere.

2. RECURSOS

2.1 RECURSOS HUMANOS

Teniendo en cuenta los roles descritos en la metodología de desarrollo programación extrema se define:

- Ing. Roberto Pantoja Instructor de teleinformática (Asesor SENA).
Gestor (Big Boss), Cliente
- Rosario Moreno Torres (coordinadora académica).
Cliente
- Luis Eduardo Enríquez (Instructor Área Agropecuaria).
Cliente
- Ing. Franklin Jiménez (Asesor de proyecto).
Consultor
- Juan Pablo Gómez egresado Universidad de Nariño, desarrollador del proyecto.
Programador, Tester.

2.2 RECURSOS TECNOLOGICOS

2.2.1 Hardware. En las diferentes oficinas del Sena – Centro multisectorial Lope se encuentran equipos con diferentes características, entre ellas: procesadores Pentium III y Pentium IV, memorias Ram de 128 y 256 Mb, discos duros de 40 y 80 Gb, y tarjeta de red. Para un funcionamiento tipo cliente en un ambiente web las características de los equipos del centro multisectorial Lope son adecuados.

Para el desarrollo del aplicativo el Sena facilitó un equipo marca DELL con procesador Intel Celeron CPU 2,8 Ghz, memoria 256 Mb DDR, disco duro 80 Gb, unidad de CD, mouse óptico DELL, monitor 17" DELL, teclado DELL, unidad Diskette 3/2, características apropiadas para ejecutar las herramientas de programación utilizadas en la construcción del sistema.

2.2.2 Software. Sistema Operativo Windows XP SP II, Internet information server II 5.1, Microsoft Visual Studio.Net 2003, .Net FrameWork 1.1, Microsoft Office 2003, SGBD Msq1 5.0. El Sena cuenta con las licencias para usar el software descrito.

2.3 RECURSOS MATERIALES

Materiales de oficina

2.4 RECURSOS FINANCIEROS

Los recursos financieros para este proyecto los asume el desarrollador.

2.5 RECURSOS OPERATIVOS

El sistema se va a montar en uno de los tres servidores ubicados en la oficina de sistemas, se tendrá acceso a la aplicación a través de la red local desde las oficinas de coordinación académica, la sala de instructores y subdirección de centros. Será manipulado por los coordinadores académicos e instructores asignados.

3. MARCO TEÓRICO

Un sistema de información es el conjunto de procesos que, operando sobre un conjunto de datos estructurados de acuerdo a una empresa, recopila, elabora y distribuye la información necesaria para las actividades de dirección y control correspondientes, apoyando la toma de decisiones necesarias para desempeñar las funciones y procesos de negocios de la empresa.

El desarrollo tecnológico en la actualidad en cuanto a hardware y software permite construir sistemas de información de alto rendimiento; por esta razón y debido a la importancia que tiene la información, las empresas adoptan tecnologías a su alcance que les permitan optimizar recursos y desempeñarse ágilmente a fin de generar sus productos o servicios con calidad y constituirse en entes productivos y competitivos.

En este sentido a continuación se definen los temas y conceptos que enmarcan el sistema de información Web (aplicación Web) planteado en este proyecto.

3.1 RED DE ÁREA LOCAL (LAN)¹

LAN es la abreviatura de Local Area Network (Red de Área Local o simplemente Red Local). Una red local es la interconexión de varios ordenadores y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de unos pocos kilómetros. Su aplicación más extendida es la interconexión de ordenadores personales y estaciones de trabajo en oficinas, fábricas, etc; para compartir recursos e intercambiar datos y aplicaciones. En definitiva, permite que dos o más máquinas se comuniquen.

El término red local incluye tanto el hardware como el software necesario para la interconexión de los distintos dispositivos y el tratamiento de la información.

En una empresa suelen existir muchos ordenadores, los cuales necesitan de su propia impresora para imprimir informes (redundancia de hardware), los datos almacenados en uno de los equipos es muy probable que sean necesarios en otro de los equipos de la empresa por lo que será necesario copiarlos en este, pudiéndose producir desfases entre los datos de un usuario y los de otro, la ocupación de los recursos de almacenamiento en disco se multiplican (redundancia de datos), los ordenadores que trabajen con los mismos datos tendrán que tener los mismos programas para manejar dichos datos (redundancia de software) etc... La solución a estos problemas se llama red de área local. La red de área local permite compartir bases de datos (se elimina la redundancia de datos), programas (se elimina la redundancia de software) y periféricos como puede ser un módem, una tarjeta RDSI, una impresora, etc... (se elimina la redundancia de hardware); poniendo a nuestra disposición otros medios de comunicación como pueden ser el correo electrónico y el Chat. Nos permite realizar un proceso distribuido, es decir, las tareas se pueden

¹ LAN Red de área local. Disponible en web http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_local. Consulta: marzo 6 de 2007.

repartir en distintos nodos y nos permite la integración de los procesos y datos de cada uno de los usuarios en un sistema de trabajo corporativo. Tener la posibilidad de centralizar información o procedimientos facilita la administración y la gestión de los equipos. Además una red de área local conlleva un importante ahorro, tanto de dinero, ya que no es preciso comprar muchos periféricos, se consume menos papel, y en una conexión a Internet se puede utilizar una única conexión telefónica compartida por varios ordenadores conectados en red; como de tiempo, ya que se logra gestión de la información y del trabajo.

3.2 BASE DE DATOS²

Una base de datos o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

3.3 APLICACIÓN WEB³

Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet (red local). Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal. En contraste, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web individual es

² Bases de datos. Disponible en web: http://es.wikipedia.org/wiki/Base_de_datos. Consulta marzo 6 de 2007.

³ Aplicación Web. Disponible en web: http://es.wikipedia.org/wiki/Aplicacion_web. Consulta: marzo 6 de 2007.

enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

Una ventaja significativa en la construcción de aplicaciones web que soporten las características de los browsers estándar es que deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada casi en todos lados. Sin embargo, aplicaciones inconsistentes de HTML, CSS, DOM y otras especificaciones de browsers pueden causar problemas en el desarrollo y soporte de aplicaciones web. Adicionalmente, la habilidad de los usuarios a personalizar muchas de las características de la interfaz (como tamaño y color de fuentes, tipos de fuentes, inhabilitar Javascript) puede interferir con la consistencia de la aplicación web.

Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web es la primera capa, un motor usando alguna tecnología web dinámica (ejemplo: CGI, PHP, Java Servlets o ASP) es la capa de en medio, y una base de datos como última capa. El navegador web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

3.4 INTERNET INFORMATION SERVICES (IIS)⁴

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego, fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Existen otros servidores web que pueden usarse como alternativa a éste que es de propiedad de Microsoft Corporation, por ejemplo Apache, Cherokee que son desarrollados en Software Libre y otros muchos.

⁴ Internet Information Services (IIS). Disponible en web: <http://es.wikipedia.org/wiki/IIS>. Consulta: marzo 6 de 2007.

3.5 ASP.NET⁵

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server.

Microsoft introdujo esta tecnología llamada Active Server Pages (ASP) en diciembre de 1996, por lo que no es nada nueva. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. La definición contextual de Microsoft es que "Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web".

Microsoft desarrolló una nueva tecnología denominada ASP.NET como parte de su estrategia .NET para el desarrollo Web, con el objetivo de resolver las limitaciones de ASP y posibilitar la creación de software como servicio. ASP.NET es la plataforma unificada de desarrollo web que proporciona a los desarrolladores los servicios necesarios para crear aplicaciones web empresariales.

⁵ Tecnología ASP.NET. Disponible en web: <http://es.wikipedia.org/wiki/ASP.NET>. Consulta marzo 6 de 2007.

4. METODOLOGÍA

La metodología de desarrollo elegida para la construcción del sistema descrito en este proyecto es programación extrema (Xp), pues, se han encontrado características que se adaptan a las condiciones del proyecto. Una de ellas y quizás la más importante de esta metodología es la llamada cliente in-situ, es decir, el cliente viene a ser parte del equipo de desarrollo acompañándolo durante todo el proceso de manera que toda mala interpretación y duda puede corregirse a tiempo a través de una comunicación oral. El Ing. Roberto Pantoja (funcionario del Sena) conoce y ha realizado la programación de cursos, además se dispuso a colaborar en la construcción de historias de usuario y solucionar toda inquietud.

Xp demuestra sencillez tanto en su aprendizaje como en su aplicación, es una metodología ágil y flexible pues es fácilmente adaptable al contexto en proyectos pequeños y con requisitos cambiantes, igualmente el esfuerzo no está concentrado en la construcción de largos y complejos documentos, más bien demuestra gran practicidad.

Es importante aclarar que este proyecto no fue desarrollado tratando de adaptarlo a la metodología, sino más bien, fueron los aportes significativos de la metodología la asistencia para el desarrollo de proyecto y satisfacer las necesidades del cliente.

4.1 PROGRAMACIÓN EXTREMA (*EXTREME PROGRAMMING, XP*)⁶

XP es una metodología ágil centrada en potenciar las relaciones interpersonales. Como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación se presentan las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.

⁶ Letelier, Patricio y Penadés, M^a Carmen. Metodologías ágiles para el desarrollo de software, Universidad Politécnica de Valencia. Disponible en web: <http://www.willydev.net/descargas/masyxp.pdf>. Consulta: Noviembre 15 de 2007.

4.1.1 Las Historias de Usuario. Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción o sólo una descripción, más quizás una estimación de esfuerzo en días. Beck en su libro presenta un ejemplo de ficha (*customer story and task card*) en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado, cosas por terminar y comentarios. No hay que preocuparse si en un principio no se identifican todas las historias de usuario. Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

4.1.2 Roles XP. Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado se describe los roles de acuerdo con la propuesta original de Beck.

- **Programador:** El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- **Cliente:** El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto, pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.
- **Encargado de pruebas (Tester):** El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento (Tracker):** El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado,

comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

- **Entrenador (*Coach*):** Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- **Gestor (*Big boss*):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

4.1.3 Proceso XP. Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma, el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. El ciclo de vida ideal de XP consiste de seis fases: exploración, planificación de la entrega (*Release*), iteraciones, producción, mantenimiento y muerte del proyecto.

4.1.4 Fase I: Exploración. En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

4.1.5 Fase II: Planificación de la Entrega. En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

4.1.6 Fase III: Iteraciones. Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

4.1.7 Fase IV: Producción. La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

4.1.8 Fase V: Mantenimiento. Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

4.1.9 Fase VI: Muerte del proyecto. Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

4.1.10 Prácticas XP. La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que se describen a continuación.

4.1.11 El juego de la planificación. Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. Esta práctica se puede ilustrar como un juego, donde existen dos tipos de jugadores: Cliente y Programador. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes. Este juego se realiza durante la planificación de la entrega, en la planificación de cada iteración y cuando sea necesario reconducir el proyecto.

4.1.12 Entregas pequeñas. La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero si que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

4.1.13 Metáfora. En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema.

4.1.14 Diseño simple. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

4.1.15 Pruebas. La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

4.1.16 Refactorización (*Refactoring*). La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo. Robert Martin⁷ señala que el diseño del sistema de software es una cosa viviente. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema. Para mantener un diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto. De hecho, este cuidado continuo sobre el diseño es incluso más importante que el diseño inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

4.1.17 Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Según Cockburn y Williams en un estudio realizado para identificar los costos y beneficios de la programación en parejas⁸, las principales ventajas de introducir este estilo de programación, son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutaban más su trabajo. Dichos beneficios se consiguen después de varios meses de practicar la programación en parejas. En los

⁷ Martin, R. "Continuous Care vs. Initial Design". 2002. Disponible en web: www.objectmentor.com/resources/articles/Continuous_Care.pdf.

⁸ Cockburn, A., Williams, L. "The Costs and Benefits of Pair Programming". Humans and Technology Technical Report. 2000.

estudios realizados por Cockburn y Williams este lapso de tiempo varía de 3 a 4 meses.

4.1.18 Propiedad colectiva del código. Cualquier programador puede cambiar parte del código. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

4.1.19 Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido. Martin Fowler⁹ afirma que el desarrollo de un proceso disciplinado y automatizado es esencial para un proyecto controlado, el equipo de desarrollo está más preparado para modificar el código cuando sea necesario, debido a la confianza en la identificación y corrección de los errores de integración.

4.1.20 40 horas por semana. Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

4.1.21 Cliente in-situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada. Jeffries¹⁰ indica que se debe pagar un precio por perder la oportunidad de un cliente con alta disponibilidad. Algunas recomendaciones propuestas para dicha situación son las siguientes: intentar conseguir un representante que pueda estar siempre disponible y que actúe como interlocutor del cliente, contar con el cliente al menos en las reuniones de planificación, establecer visitas frecuentes de los programadores al cliente para validar el sistema, anticiparse a los problemas asociados estableciendo llamadas telefónicas frecuentes y conferencias, reforzando el compromiso de trabajo en equipo.

⁹ Fowler, M., Foemmel M. "Continuous Integration". 2001. Disponible en web: www.martinfowler.com/articles/designDead.html.

¹⁰ Jeffries, R., Anderson, A., Hendrickson, C. "Extreme Programming Installed". Addison-Wesley. 2001

4.1.22 Estándares de programación. XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

4.1.23 Comentarios respecto de las prácticas. El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras, donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

5. DESARROLLO DEL APLICATIVO

El sistema web se encuentra dividido de la siguiente manera:

Módulo de cursos: Teniendo en cuenta que un curso está constituido por módulos y un módulo está constituido por instruccionales o materias, este módulo del sistema realiza el proceso de inserción, modificación y eliminación de registros correspondiente a cursos, módulos e instruccionales. También permite asignar los módulos pertenecientes a un curso y los instruccionales pertenecientes a un módulo.

Módulo de instructores: Permite realizar el proceso de inserción, modificación y eliminación de registros correspondiente a los instructores o docentes.

Módulo de aulas: Permite realizar el proceso de inserción y eliminación de registros correspondientes a las aulas.

Módulo de programación de cursos: Permite realizar el proceso de programación de cursos por medio de la asignación de instructores, aulas, fechas y horarios a los módulos e instruccionales de los cursos disponibles. También realiza al manejo de fechas en las cuales no hay actividades educativas.

Módulo de reportes: Realiza la construcción de los reportes del estado de programación de cursos, instructores y aulas.

Módulo de usuarios del sistema: Permite realizar la administración de los usuarios que manejan el sistema.

Para la adquisición de los requerimientos y construcción de los módulos del sistema se utilizó las historias de usuario propuestas en la metodología de desarrollo extreme programming, sin embargo, para un mejor detalle en este trabajo también se aplicó la técnica de casos de uso, se elaboró el modelo conceptual y el diagrama de clases propio del lenguaje unificado de modelado UML.

5.1 HISTORIAS DE USUARIO Y TAREAS DE PROGRAMACIÓN

Historia de Usuario	
Número: 1	Usuario: Programador de cursos
Nombre historia: Control de cursos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados:	Iteración asignada: 1
Programador responsable: Juan Pablo Gómez	
Descripción: Se lee del catálogo de cursos desde archivos de Excel o archivos impresos y se introduce la información de los cursos, los módulos del curso, y los instruccionales de cada módulo. Esto debe ser ingresado en la base de datos y debe quedar coherentemente relacionado.	
Observaciones: <ul style="list-style-type: none"> - <i>El catálogo de cursos conforma todos los cursos que brinda el Sena.</i> - <i>Un cursos se conforma de varios módulos</i> - <i>Un módulo se conforma de varios instruccionales</i> - <i>Un instruccional es una materia.</i> - <i>Un módulo es llamado también curso corto</i> 	

Tarea de programación de la historia de usuario #1	
Número tarea: 1	Número historia: 1
Nombre tarea: Interfaz de control de instruccionales o materias	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Construir una ventana que permita ingresar el código, nombre y duración de un instruccional, igualmente que permita buscar y modificar un registro.	
Observaciones:	

Tarea de programación de la historia de usuario #1	
Número tarea: 2	Número historia: 1
Nombre tarea: Control de la base de datos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Definir la codificación que permita acceder a una tabla especificada de la base de datos realizando las operaciones de consulta, inserción, modificación y eliminación de registros.	
Observaciones: <ul style="list-style-type: none"> - <i>La base de datos se construirá paralelamente con el desarrollo de las historias de usuario definidas, este código irá cambiando en todo ese proceso, sin embargo, ese cambio no se lo tomara como tarea de programación pues son cambios no significativos.</i> 	

Tarea de programación de la historia de usuario #1	
Número tarea: 3	Número historia: 1
Nombre tarea: Manejo de operaciones del catálogo de cursos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Determinar las sentencias SQL y la codificación para el manejo de registros del catálogo de cursos (Cursos, módulos, instruccionales).	
Observaciones:	

Tarea de programación de la historia de usuario #1	
Número tarea: 4	Número historia: 1
Nombre tarea: Interfaz de control de módulos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
<p>Descripción: Construir una ventana que permita ingresar el código y nombre de un módulo, también debe permitir buscar y modificar un módulo. Esta ventana debe cargar los instruccionales y asignar aquellos que pertenezcan al módulo creado. De esta manera, un módulo quedará conformado de varios instruccionales. La duración total para el módulo es igual a la suma de la duración de todos los instruccionales asignados, este dato deberá ser registrado en la base de datos como parte de la información del módulo.</p>	
Observaciones:	

Tarea de programación de la historia de usuario #1	
Número tarea: 5	Número historia: 1
Nombre tarea: Interfaz de control de cursos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
<p>Descripción: Construir una ventana que permita ingresar el código, versión, nombre y sector de un curso, igualmente debe tener la posibilidad de buscar y modificar un curso. En esta ventana se debe cargar los módulos y asignar aquellos que pertenezcan al curso creado. De esta manera, un curso quedará conformado de varios módulos. La duración total para el curso es igual a la suma de la duración de todos los módulos asignados, este dato deberá ser registrado en la base de datos como parte de la información del curso.</p>	
Observaciones:	

Historia de Usuario	
Número: 2	Usuario: Programador de cursos
Nombre historia: Control de instructores	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados:	Iteración asignada: 1
Programador responsable: Juan Pablo Gómez	
Descripción: Se ingresa la información básica, de los instructores tanto de planta como de contrato en la base de datos.	
Observaciones: - <i>La información básica corresponde a datos personales, y tipo de vinculación.</i>	

Tarea de programación de la historia de usuario #2	
Número tarea: 1	Número historia: 2
Nombre tarea: Interfaz de control de instructores	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Construir una ventana que permita ingresar la cédula, nombre, vinculación (planta o contrato), definir si el instructor pertenece al sindicato de trabajadores, potencial de trabajo, profesión, teléfono y dirección.	
Observaciones:	

Tarea de programación de la historia de usuario #2	
Número tarea: 2	Número historia: 2
Nombre tarea: Manejo de operaciones para instructores	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha fin: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Determinar las sentencias SQL y la codificación para el manejo de registros de los instructores.	
Observaciones:	

Historia de Usuario	
Número: 3	Usuario: Programador de cursos
Nombre historia: Control de calendario	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados:	Iteración asignada: 2
Programador responsable: Juan Pablo Gómez	
Descripción: La dirección de centros entrega un documento en el que define por cada mes del año lectivo las fechas en que no habrá labores académicas por eventos extraordinarios en los que solo participan los instructores de planta. Igualmente, hay unas fechas que solo están reservadas para las reuniones del sindicato en las cuales solo participan instructores de planta pertenecientes al sindicato. También se debe tener en cuenta aquellas fechas que afectan a todos los instructores, incluidos los de contrato. Estas fechas deberán estar registradas en la base de datos	
Observaciones:	

Tarea de programación de la historia de usuario #3	
Número tarea: 1	Número historia: 3
Nombre tarea: Interfaz de control para el calendario de instructores de contrato - planta - sindicato	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
<p>Descripción: Construir una ventana que permita ingresar las fechas extraordinarias correspondientes a los días en que no habrá actividades lectivas durante todo el año para los instructores de contrato, planta y los que pertenecen al sindicato.</p> <p>Se debe tener en cuenta lo siguiente: Las fechas definidas para los instructores de contrato afectan a los instructores de planta, pero los de planta no afectan a los de contrato.</p> <p>Los instructores que pertenecen al sindicato siempre son de planta, por lo tanto, cuando se define fechas para los instructores de planta afecta al sindicato, pero cuando se establece fechas para los instructores del sindicato este no afecta ni a los de contrato ni a los de planta.</p> <p>Deberá permitir eliminar fechas específicas. Todo debe quedar registrado en la base de datos.</p>	
Observaciones:	

Tarea de programación de la historia de usuario #3	
Número tarea: 2	Número historia: 3
Nombre tarea: Manejo de operaciones del calendarios	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
<p>Descripción: Determinar las sentencias SQL y la codificación para el manejo de registros del calendario.</p>	
Observaciones:	

Historia de Usuario	
Número: 4	Usuario: Programador de cursos
Nombre historia: Control de aulas	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados:	Iteración asignada: 2
Programador responsable: Juan Pablo Gómez	
Descripción: Se ingresa la información básica de las aulas que maneja el Sena en la base de datos.	
Observaciones:	

Tarea de programación de la historia de usuario #4	
Número tarea: 1	Número historia: 4
Nombre tarea: Interfaz de control de aulas	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Construir una ventana que permita ingresar la identificación de aula, el sector al cual pertenece y una descripción.	
Observaciones:	

Tarea de programación de la historia de usuario #4	
Número tarea: 2	Número historia: 4
Nombre tarea: Manejo de operaciones para aulas	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Determinar las sentencias SQL y la codificación para el manejo de registros de aulas.	
Observaciones:	

Historia de Usuario	
Número: 5	Usuario: Programador de cursos
Nombre historia: Control de instruccionales fuera del catálogo	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados:	Iteración asignada: 2
Programador responsable: Juan Pablo Gómez	
Descripción: Se ingresa la información de los instruccionales que no pertenecen al catálogo, estos instruccionales poseen los mismo atributos de los instruccionales pertenecientes al catálogo de cursos, sin embargo, en el caso de que se quiera extender un módulo o instruccional programado, el sistema carga los campos de número de orden, año, y el código del módulo o instruccional programado. Debe quedar registrado en la base de datos.	
Observaciones:	

Tarea de programación de la historia de usuario #5	
Número tarea: 1	Número historia: 5
Nombre tarea: Interfaz de control de instruccionales fuera de catálogo	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Construir una ventana que permita ingresar el código, nombre y duración del instruccional, en el caso de que se extienda un módulo o instruccional programado el sistema carga los campos numero de orden, año y código del módulo o instruccional programado, igualmente debe permita buscar y modificar un registro. Registrarlo en la base de datos.	
Observaciones:	

Historia de Usuario	
Número: 6	Usuario: Programador de cursos
Nombre historia: Apertura de cursos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados:	Iteración asignada: 2
Programador responsable: Juan Pablo Gómez	
Descripción: Se lee de la base de datos los cursos y los instructores disponibles. Se asigna un número de orden al curso a programar, el año de programación, la jornada, la intensidad horaria, el número de alumnos matriculados al curso, el tipo de curso y un director de grupo. Esto debe actualizarse en la base de datos. Se debe tener a mano los cursos que ya están abiertos identificando si han sido completamente programados o parcialmente programados.	
Observaciones: - <i>El número de orden es una identificación al curso una vez se programa.</i>	

Tarea de programación de la historia de usuario #6	
Número tarea: 1	Número historia: 6
Nombre tarea: Interfaz de manejo para abrir cursos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Construir una ventana que permita ingresar un numero de orden, la jornada, Intensidad horaria, el número de alumnos, tipo de formación (titulada, complementaria), seleccionar un director de grupo (instructor). Cuando el tipo de formación sea titulada deben cargarse los cursos, y cuando el tipo de formación sea complementaria deben cargarse los módulos. Esto se almacena en la base de datos. Debe mostrar los cursos que ya han sido abiertos y si están completa o parcialmente programados. Una vez abierto el curso se procede a programarlo.	
Observaciones:	

Tarea de programación de la historia de usuario #6	
Número tarea: 2	Número historia: 6
Nombre tarea: Manejo de operaciones para abrir cursos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Determinar las sentencias SQL y la codificación para el manejo de la apertura de cursos.	
Observaciones:	

Historia de Usuario	
Número: 7	Usuario: Programador de cursos
Nombre historia: Programación de cursos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:	Iteración asignada: 2
Programador responsable: Juan Pablo Gómez	
<p>Descripción: Se lee de la base de datos el curso abierto y sus correspondientes módulos e instruccionales. Para cada módulo se asigna un instructor, aula, intensidad horaria, se fijan los días de la semana en los que se lo dictará y se establece la fecha de inicio. Teniendo en cuenta estos datos más la información del <u>calendario de trabajo</u> se calcula la fecha de terminación del módulo. El sistema debe validar el cruce de horarios de aulas e instructores. Por cada uno de los módulos se realiza este proceso hasta que el curso quede totalmente programado. Esto debe quedar actualizado en la base de datos.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> - <i>El calendario de trabajo se define en la historia de usuario # 3.</i> - <i>Un módulo corresponde a varios instruccionales o materia.</i> - <i>Si se programa un módulo por completo todos sus instruccionales quedan sujetos a esa programación, sin embargo también debe ofrecer la opción de programar cada instruccional de manera independiente, como si se tratase de un módulo.</i> 	

Tarea de programación de la historia de usuario #7	
Número tarea: 1	Número historia: 7
Nombre tarea: Interfaz de control para los horarios de cursos, aulas e instructores	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
<p>Descripción Construir 3 tablas para cursos, instructores y aulas a manera de diagrama de gantt en las cuales se plasme el horario y los días de trabajo.</p>	
Observaciones:	

Tarea de programación de la historia de usuario #7	
Número tarea: 2	Número historia: 7
Nombre tarea: Interfaz para el manejo de parámetros de la programación y validación de la programación	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
<p>Descripción:</p> <p>Construir una ventana que visualice el número de orden, el nombre del curso, el código del curso, la versión (si tiene), y el tipo de formación (titulada, complementaria) del curso que se está programando. En la ventana se deben cargar los módulos del curso y se deben cargar los instruccionales del módulo. Se debe definir la hora de inicio, la hora de terminación y la fecha de inicio del módulo. Se debe establecer un listado de instructores filtrados dependiendo si son de contrato, de planta o temporales, también se debe establecer un listado de todas las aulas. Se debe definir los días de la semana en los que se desea que el módulo quede programado. Si el usuario lo desea puede ir programando instruccional por instruccional hasta que todo el módulo quede programado.</p> <p>Hay unos instruccionales que están fuera del catálogo de cursos, estos también deben tener la posibilidad de ser programados.</p> <p>La duración del módulo o instruccional puede redefinirse manualmente para el efecto de la programación y sin afectar el catálogo de cursos.</p> <p>Definidos estos parámetros se realiza el cálculo de la fecha de terminación del módulo o instruccional dependiendo de los calendarios de instructores de planta y/o pertenecientes al sindicato registrados en la base de datos. Se valida el cruce de horarios para aulas, instructores y la duplicación de elementos ya programados.</p> <p>Si la validación es favorable los datos de programación se pasan a la instancia de manejo de la base de datos para que sean registrados.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> - <i>Cuando no se halle el instructor adecuado para algún módulo o instruccional, el sistema define un instructor temporal o virtual y en el momento en que pueda asignarse un instructor real, este tomara el lugar del virtual. Este proceso se realizara desde el control de instructores.</i> 	

Tarea de programación de la historia de usuario #7	
Número tarea: 3	Número historia: 7
Nombre tarea: Manejo de operaciones de la programación de cursos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Determinar las sentencias SQL y la codificación para el manejo de la programación de cursos.	
Observaciones:	

Historia de Usuario	
Número: 8	Usuario: Programador de cursos
Nombre historia: Reportes	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados:	Iteración asignada: 3
Programador responsable: Juan Pablo Gómez	
Descripción: El sistema debe elaborar reportes del estado de la programación.	
Observaciones:	

Tarea de programación de la historia de usuario #8	
Número tarea: 1	Número historia: 8
Nombre tarea: Control excel	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Desarrollar la codificación que permita la ejecución y el manejo de la aplicación MS Excel. debe controlar la apertura y almacenamiento en disco de archivos *.xls.	
Observaciones:	

Tarea de programación de la historia de usuario #8	
Número tarea: 2	Número historia: 8
Nombre tarea: Reporte de cursos	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Desarrollar la codificación que permita generar un informe general de cursos programados en un archivo *.xls especificando los que han terminado y los que pasan al siguiente año. Para un curso programado definido debe generar un archivo *.xls en el que se especifique a través de un diagrama de Gantt los módulos e instruccionales programados dejando notar la fecha de inicio, fecha de terminación, el aula, y el instructor asociado.	
Observaciones:	

Tarea de programación de la historia de usuario #8	
Número tarea: 3	Número historia: 8
Nombre tarea: Reporte de Instructor	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin:
Programador responsable: Juan Pablo Gómez	
Descripción: Desarrollar la codificación que permita generar por instructor un informe en un archivo *.xls, se debe definir un diagrama de Gantt con los módulos e instruccionales que dicta con su respectivas fecha de inicio y de terminación. También debe informar las horas que ha dictado, y las horas que le faltan para alcanzar el potencial en el año.	
Observaciones:	

Tarea de programación de la historia de usuario #8	
Número tarea: 4	Número historia: 8
Nombre tarea: Reporte de aula	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin:
Programador responsable: Juan Pablo Gómez	
Descripción: Desarrollar la codificación que permita generar por aula un informe en un archivo *.xls, se debe definir un diagrama de Gantt con los módulos e instruccionales que se dictan en el aula con su respectivas fecha de inicio y de terminación.	
Observaciones:	

Historia de Usuario	
Número: 9	Usuario: Administrador del sistema, programador de cursos
Nombre historia: Restricción de uso del sistema	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados:	Iteración asignada: 3
Programador responsable: Juan Pablo Gómez	
Descripción: Para ingresar al sistema se debe solicitar un nombre de usuario y contraseña Se restringe a los usuarios el acceso a reportes y a la actualización de registros.	
Observaciones:	

Tarea de programación de la historia de usuario #9	
Número tarea: 1	Número historia: 9
Nombre tarea: Interfaz de cuentas de usuario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Construir una ventana que permita ingresar un ID y nombre del usuario nuevo. Debe cargar todos los usuarios del sistema, y por cada uno de ellos definir si puede insertar, modificar, eliminar o tener acceso a reportes. Debe permitir restablecer la contraseña, activar desactivar y eliminar cualquier cuenta de usuario.	
Observaciones:	

Tarea de programación de la historia de usuario #9	
Número tarea: 2	Número historia: 9
Nombre tarea: Manejo de operaciones de cuentas de usuario	
Tipo de tarea : Desarrollo	Puntos estimados:
Fecha inicio: -	Fecha fin: -
Programador responsable: Juan Pablo Gómez	
Descripción: Determinar las sentencias SQL y la codificación para el manejo de registros de cuentas de usuario. Debe encriptar las contraseñas.	
Observaciones:	

5.2 CASOS DE USO DEL SISTEMA

5.2.1 Lista de actores

Nombre:	Administrador del sistema
Descripción :	Es la persona encargada del manejo de usuarios del sistema. Tiene acceso a toda la funcionalidad del sistema.
Objetivos:	Crea, y administra los usuarios otorgando o denegando privilegios de acceso a los recursos del sistema, como reportes y actualización de datos.

Nombre:	Programador de cursos
Descripción :	Es la persona encargada de realizar la programación de cursos, es asignada por el Sena y debe tener conocimientos básicos del funcionamiento de la programación de cursos.
Objetivos:	Accede a todos los recursos del sistema menos a aquellos restringidos por el administrador del sistema.

5.2.2 Casos de uso

CASO DE USO: Ingresar al sistema
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: tener acceso a los módulos del sistema por medio de una identificación y contraseña.
RESUMEN: Un programador de cursos ingresa al sistema.
CURSO NORMAL DE LOS EVENTOS 1. Un programador de cursos ingresa su nombre de usuario, su contraseña al sistema y luego pulsa en el botón aceptar. El sistema carga una ventana de inicio.
FLUJOS ALTERNATIVOS Línea 1: El nombre de usuario o la contraseña es inválido, y el sistema muestra un error.
PRECONDICIONES
POSTCONDICIONES ▪ Se tiene acceso a los diferentes módulos del sistema.

CASO DE USO: Controlar usuarios
ACTORES: Administrador del sistema.
PROPOSITO: Administrar los usuarios del sistema.
RESUMEN: El administrador del sistema ingresa al sistema, en este caso de uso puede crear y eliminar usuarios y otorgarles privilegios.
CURSO NORMAL DE LOS EVENTOS 1. El sistema muestra un listado con los usuarios del sistema registrados 2. El administrador del sistema puede crear, eliminar, activar, desactivar usuarios, además asignar privilegios de acceso a reportes y actualización de datos de instruccionales, módulos, cursos, instructores, aulas, fechas y programación de cursos. 2.1 Para crear un usuario el administrador del sistema digita un nombre de identificación y su nombre real, el sistema genera por contraseña el mismo nombre de identificación. 2.2 Para eliminar un usuario el administrador del sistema selecciona el usuario y luego

<p>pulsa el botón eliminar.</p> <p>2.3 Para permitirle a un usuario el ingreso al sistema el administrador del sistema selecciona el usuario y pulsa el botón activar.</p> <p>2.4 Para denegar a un usuario el ingreso al sistema el administrador del sistema selecciona el usuario y pulsa el botón desactivar.</p> <p>2.5 Para conceder o denegar un privilegio de actualización de datos o acceso a reportes en alguno de los módulos del sistema el administrador del sistema elige el módulo, selecciona cualquiera de los dos privilegios y define un valor positivo o negativo.</p>
<p>FLUJOS ALTERNATIVOS Línea 2.1: Si el nombre de usuario ya esta registrador el sistema muestra un mensaje de error.</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Administrador del sistema debe haber ingresado al sistema con su identificación y contraseña.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se tiene acceso a los diferentes módulos del sistema.

<p>CASO DE USO: Controlar instruccionales fuera de catálogo</p>
<p>ACTORES: Programador de cursos, Administrador del sistema.</p>
<p>PROPOSITO: Administrar la información correspondiente a los instruccionales fuera del catálogo y extensión de módulos e instruccionales.</p>
<p>RESUMEN: Un Programador de cursos ingresa al sistema, lee de un archivo de Excel los datos correspondientes a un instruccional fuera de catálogo y los registra en el sistema. Tiene la posibilidad de modificar y eliminar instruccionales.</p>
<p>CURSO NORMAL DE LOS EVENTOS</p> <ol style="list-style-type: none"> 1. El sistema muestra un listado con los instruccionales registrados. 2. El programador de cursos puede insertar modificar o eliminar un instruccional. <ol style="list-style-type: none"> 2.1 Para insertar un instruccional el programador de cursos digita el código, el nombre y la duración del instruccional y luego presiona el botón insertar. En el caso de que el instruccional sea una extensión de un módulo instruccional programado, los campos año, numero de orden, y código de programación son cargados por el sistema. 2.2 Para modificar un instruccional el programador de cursos selecciona el instruccional deseado digita los nuevos datos, y luego presiona el botón modificar. 2.3 Para eliminar un instruccional el programador de cursos selecciona el instruccional deseado y luego presiona el botón eliminar.
<p>FLUJOS ALTERNATIVOS Línea 2.1: Si el instruccional que se ingresa ya se encuentra almacenado el sistema genera un error. Línea 2.3: Si el instruccional que se desea eliminar esta programado, el sistema genera un error.</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña.
<p>POSTCONDICIONES.</p> <ul style="list-style-type: none"> ▪ Se actualiza el sistema con un nuevo instruccional. ▪ Se elimina del sistema un instruccional. ▪ Se extiende un instruccional o módulo programado.

CASO DE USO: Controlar instruccionales
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Administrar la información correspondiente a los instruccionales.
RESUMEN: Un Programador de cursos ingresa al sistema, lee de un archivo de Excel los datos correspondientes a un instruccional y los registra en el sistema. Tiene la posibilidad de modificar y eliminar instruccionales.
CURSO NORMAL DE LOS EVENTOS <ol style="list-style-type: none"> 1. El sistema muestra un listado con los instruccionales registrados. 2. El programador de cursos puede insertar modificar o eliminar un instruccional. <ol style="list-style-type: none"> 2.1 Para insertar un instruccional El programador de cursos digita el código, el nombre y la duración del instruccional y luego presiona el botón insertar. 2.2 Para modificar un instruccional el programador de cursos selecciona el instruccional deseado digita los nuevos datos, y luego presiona el botón modificar. 2.3 Para eliminar un instruccional el programador de cursos selecciona el instruccional deseado y luego presiona el botón eliminar.
FLUJOS ALTERNATIVOS Línea 2.1: Si el instruccional que se ingresa se encuentra almacenado el sistema genera un error. Línea 2.3: Si el instruccional que se desea eliminar esta programado, el sistema genera un error.
PRECONDICIONES <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña.
POSTCONDICIONES <ul style="list-style-type: none"> ▪ Se actualiza el sistema con un nuevo instruccional. ▪ Se elimina del sistema un instruccional.

CASO DE USO: Controlar módulos
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Administrar la información correspondiente a los módulos.
RESUMEN: Un Programador de cursos ingresa al sistema, lee de un archivo de Excel los datos correspondientes a los módulos y los registra en el sistema. En este caso de uso se asigna a cada módulo los instruccionales correspondientes. Tiene la posibilidad de modificar y eliminar módulos.
CURSO NORMAL DE LOS EVENTOS <ol style="list-style-type: none"> 1. El sistema muestra un listado con los módulos e instruccionales registrados. 2. El programador de cursos puede insertar modificar o eliminar un módulo. <ol style="list-style-type: none"> 2.1 Para insertar un módulo el programador de cursos digita el código y el nombre del módulo y luego presiona el botón insertar. 2.2 Para modificar un módulo el programador de cursos selecciona el módulo deseado, digita los nuevos datos, y luego presiona el botón modificar. 2.3 Para eliminar un módulo el programador de cursos selecciona el módulo deseado y luego presiona el botón eliminar. 3. El programador de cursos selecciona un módulo y el sistema carga en la pantalla sus instruccionales asignados. 4. El programador de cursos puede asignar o eliminar la asignación de instruccionales pertenecientes a un módulo. <ol style="list-style-type: none"> 4.1 Para asignar un instruccional a un módulo el programador de cursos selecciona el módulo, selecciona el instruccional y pulsa en el botón adicionar. El sistema actualiza la duración para el módulo y carga en pantalla el instruccional asignado.

4.2 Para eliminar un instruccional asignado a un módulo el programador de cursos selecciona el instruccional asignado y pulsa el botón eliminar.
<p>FLUJOS ALTERNATIVOS</p> <p>Línea 2.1: Si el módulo que se ingresa ya se encuentra almacenado el sistema genera un error.</p> <p>Línea 2.3: Si el módulo que se desea eliminar esta programado, el sistema genera un error.</p> <p>Línea 4.1: Si el instruccional seleccionado ya ha sido asignado el sistema muestra por pantalla un mensaje de error.</p> <p>Línea 4.2: Si el módulo seleccionado está programado se muestra por pantalla un mensaje de error.</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El sistema debe tener registrado al menos un instruccional.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se actualiza el sistema con un nuevo módulo. ▪ Se elimina del sistema un módulo. ▪ Se actualiza el sistema con los instruccionales que debe tener un módulo específico.

CASO DE USO: Controlar cursos
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Administrar la información correspondiente a los cursos.
RESUMEN: Un Programador de cursos ingresa al sistema, lee de un archivo de Excel los datos correspondientes a los cursos y los registra en el sistema. En este caso de uso se asigna a cada curso los módulos correspondientes. Tiene la posibilidad de modificar y eliminar cursos.
<p>CURSO NORMAL DE LOS EVENTOS</p> <ol style="list-style-type: none"> 1. El sistema muestra un listado con los cursos y módulos registrados. 2. El programador de cursos puede insertar modificar o eliminar un curso. <ol style="list-style-type: none"> 2.1 Para insertar un curso el programador de cursos digita el código, la versión, el nombre del curso, el sector y luego presiona el botón insertar. 2.2 Para modificar un curso el programador de cursos selecciona el curso deseado, digita los nuevos datos, y luego presiona el botón modificar. 2.3 Para eliminar un curso el programador de cursos selecciona el curso deseado y luego presiona el botón eliminar. 3. El programador de cursos selecciona un curso y el sistema carga en la pantalla sus módulos asignados. 4. El programador de cursos puede asignar o eliminar la asignación de módulos pertenecientes a un curso. <ol style="list-style-type: none"> 4.1 Para asignar un módulo a un curso el programador de cursos selecciona el curso, selecciona el módulo y pulsa en el botón adicionar. El sistema actualiza la duración para el curso y carga en pantalla el módulo asignado. 4.2 Para eliminar un módulo asignado a un curso el programador de cursos selecciona el módulo asignado y pulsa el botón eliminar.
<p>FLUJOS ALTERNATIVOS</p> <p>Línea 2.1: Si el curso que se ingresa ya se encuentra almacenado el sistema genera un error.</p> <p>Línea 2.3: Si el curso que se desea eliminar esta programado, el sistema genera un error.</p> <p>Línea 4.1: Si el módulo seleccionado ya ha sido asignado el sistema muestra por pantalla un mensaje de error.</p> <p>Línea 4.2: Si el curso seleccionado está programado se muestra por pantalla un mensaje de error.</p>

<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El sistema debe tener registrado al menos un módulo.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se actualiza el sistema con un nuevo curso. ▪ Se elimina del sistema un curso. ▪ Se actualiza el sistema con los módulos que debe tener un curso específico.

<p>CASO DE USO: Controlar Instructores</p>
<p>ACTORES: Programador de cursos, Administrador del sistema.</p>
<p>PROPOSITO: Administrar la información correspondiente a los instructores.</p>
<p>RESUMEN: Un Programador de cursos ingresa al sistema, luego al módulo de instructores, y registra los datos que corresponden al instructor vinculado a la empresa. En este caso de uso se puede modificar y eliminar instructores, y generar reportes de la carga laboral de los instructores durante el año.</p>
<p>CURSO NORMAL DE LOS EVENTOS</p> <ol style="list-style-type: none"> 1. El sistema muestra un listado con los instructores registrados. 2. El programador de cursos puede insertar modificar o eliminar un instructor. <ol style="list-style-type: none"> 2.1 Para insertar un instructor el programador de cursos selecciona el tipo de vinculación, digita la cedula, el nombre, selecciona si pertenece o no al sindicato, digita el potencial de trabajo, y opcionalmente puede ingresar la profesión, teléfono y dirección. Luego presiona el botón insertar. 2.2 Para modificar un instructor el programador de cursos selecciona el instructor deseado, digita los nuevos datos, y luego presiona el botón modificar. 2.3 Para eliminar un instructor el programador de cursos selecciona el instructor deseado y luego presiona el botón eliminar. 3. El programador de cursos de cursos selecciona un instructor, selecciona el año y ejecuta el caso de uso Generar reporte de instructor.
<p>FLUJOS ALTERNATIVOS</p> <p>Línea 2.1: Si el instructor que se ingresa ya se encuentra almacenado el sistema genera un error.</p> <p>Línea 2.3: Si el instructor que se desea eliminar tiene asignado al menos un módulo o instruccional, el sistema genera un error.</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se actualiza el sistema con un nuevo instructor. ▪ Se elimina del sistema un instructor. ▪ Se modifica un instructor temporal, por un instructor con vinculación tipo contrato.

<p>CASO DE USO: Controlar Aulas</p>
<p>ACTORES: Programador de cursos, Administrador del sistema.</p>
<p>PROPOSITO: Administrar la información correspondiente a las aulas.</p>
<p>RESUMEN: Un Programador de cursos ingresa al sistema, luego al módulo de aulas, y registra</p>

los datos que corresponden al aula. En este caso de uso también se puede eliminar aulas, y generar reportes de la asignación de cursos a las aulas.
<p>CURSO NORMAL DE LOS EVENTOS</p> <ol style="list-style-type: none"> 1. El sistema muestra un listado con las aulas registrados. 2. El programador de cursos puede insertar o eliminar un aula. <ol style="list-style-type: none"> 2.1 Para insertar un aula el programador de cursos digita el número de aula, selecciona el sector al que pertenece y opcionalmente puede digitar una descripción. Luego presiona el botón insertar. 2.2 Para eliminar un aula el programador de cursos selecciona el aula deseada y luego presiona el botón eliminar. 3. El programador de cursos de cursos selecciona un aula, selecciona el año y ejecuta el caso de uso Generar reporte de aula.
<p>FLUJOS ALTERNATIVOS</p> <p>Línea 2.1: Si el aula que se ingresa ya se encuentra almacenada el sistema genera un error.</p> <p>Línea 2.2: Si el aula que se desea eliminar tiene asignado al menos un módulo o instruccional, el sistema genera un error.</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se actualiza el sistema con una nueva aula. ▪ Se elimina del sistema un aula.

CASO DE USO: Controlar Fechas no lectivas
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Administrar la información de las fechas no lectivas para los cursos durante el año.
RESUMEN: Un Programador de cursos ingresa al sistema, luego al módulo de control de fechas, y registra las fechas no lectivas durante todo el año. En este caso de uso también se puede eliminar fechas, y generar un reporte general del calendario para el año establecido.
<p>CURSO NORMAL DE LOS EVENTOS</p> <ol style="list-style-type: none"> 1. El sistema muestra un listado con las fechas no lectivas registradas. 2. El programador de cursos puede insertar o eliminar una fecha. <ol style="list-style-type: none"> 2.1 Para insertar una fecha el programador de cursos selecciona si la fecha restrictiva es general, solo para instructores de planta, o solo para instructores del sindicato, selecciona el Año, mes, día y luego presiona el botón insertar. 2.2 Para eliminar una fecha el programador de cursos selecciona la fecha deseada y luego presiona el botón eliminar. 3. El programador de cursos selecciona un año y ejecuta el caso de uso Generar reporte de calendario.
<p>FLUJOS ALTERNATIVOS</p> <p>Línea 2.1 Si la fecha que se ingresa ya se encuentra almacenada el sistema genera un error.</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se actualiza el sistema con fechas no lectivas. ▪ Se elimina del sistema una fecha.

CASO DE USO: Realizar Apertura de cursos
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Abrir cursos del catálogo de cursos.
RESUMEN: Un Programador de cursos ingresa al sistema, luego al módulo de apertura de cursos, selecciona el curso que se necesita abrir asignándole un número de orden. En este caso de uso también se puede eliminar cursos abiertos, generar el reporte de la programación específica de un curso y generar el reporte general de cursos abiertos en un año.
CURSO NORMAL DE LOS EVENTOS <ul style="list-style-type: none"> 1. El sistema muestra un listado con los cursos y los módulos registrados. (Un módulo puede ser un curso también). 2. La pantalla muestra un listado con los cursos abiertos definiendo en qué estado se encuentra (iniciado, terminado o pasa al siguiente año). 3. El programador de cursos selecciona un curso/módulo que se necesita abrir, selecciona el año, digita el numero de orden, número de alumnos, intensidad horaria, jornada, y el director de grupo (instructor). Y pulsa en el botón abrir curso. Luego el sistema carga en pantalla el curso con los datos ingresados especificando que ha sido abierto. 4. El programador de cursos selecciona un curso abierto y ejecuta el caso de uso Generar reporte de curso. 5. El programador selecciona un año y ejecuta el caso de uso Generar reporte general de cursos. 6. El programador de cursos selecciona un curso abierto y ejecuta el caso de uso Programar cursos.
FLUJOS ALTERNATIVOS Línea 3: Si el curso que se está ingresando tiene un número de orden y año ya existente, el sistema genera un error.
PRECONDICIONES <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El sistema debe tener registrado al menos un curso o módulo. ▪ El sistema debe tener registrado al menos un instructor.
POSTCONDICIONES <ul style="list-style-type: none"> ▪ Se actualiza el sistema con cursos abiertos. ▪ Se elimina del sistema un curso abierto, eliminando también toda su programación.

CASO DE USO: Programar cursos
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Programar los módulos e instruccionales de un curso abierto.
RESUMEN: Un Programador de cursos ingresa al sistema, luego al módulo de apertura de cursos, selecciona un curso abierto y luego pulsa en el botón programar curso. En este caso de uso se asignan fechas, horarios, aulas e instructores para cada módulo e instruccional de un curso abierto, también se puede eliminar una programación registrada.
CURSO NORMAL DE LOS EVENTOS <ul style="list-style-type: none"> 1. El sistema muestra la información general del curso proporcionada en el caso de uso Realizar apertura de cursos. 2. El sistema muestra un listado con los módulos del curso y los instruccionales de cada módulo mostrando cuales han sido programados. 3. El programador de cursos selecciona un módulo, instruccional o instruccional fuera de

<p>catálogo según como quiera programar. (cuando elige programar un módulo, los instruccionales asignados a ese módulo quedan igualmente programados).</p> <ol style="list-style-type: none"> 4. El programador de cursos selecciona la fecha de inicio. 5. El programador de cursos selecciona el instructor que se va asignar al módulo/instruccional y el sistema carga en pantalla la disponibilidad de horarios y fechas de dicho instructor según sea su tipo de vinculación. 6. El programador de cursos selecciona el aula que se va asignar al módulo/instruccional y el sistema carga en pantalla la disponibilidad de horarios y fechas de dicha aula. 7. El programador de cursos selecciona la hora de inicio y de terminación, los días de la semana en los que se dictará el módulo/instruccional seleccionado. 8. El programador de cursos pulsa el botón programar. 9. El programador de cursos elimina la programación de un módulo/instruccional pulsando el botón eliminar programación.
<p>FLUJOS ALTERNATIVOS Línea 3: El programador de cursos extiende un módulo/instruccional y ejecuta el caso de uso Controlar instruccionales fuera de catálogo. Línea 8:</p> <ol style="list-style-type: none"> a. Si el módulo/instruccional ya ha sido programado, el sistema muestra mensaje de error. b. Si la fecha de inicio se ingreso incorrectamente, el sistema muestra mensaje de error. c. Si se presenta un cruce de horario en un día con el módulo/instruccional, aula o instructor seleccionado, el sistema muestra el error correspondiente. d. Si el módulo/instruccional se programa correctamente, el sistema muestra por pantalla que el módulo/instruccional está programado.
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El programador de cursos debe haber ingresado en el caso de uso Realizar apertura de cursos.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se actualiza el sistema con la programación de un módulo/instruccional. ▪ Se actualiza el sistema con la asignación de trabajo para instructores y aulas.

CASO DE USO: Generar reporte de instructor.
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Obtener el reporte de la asignación de módulos e instruccionales a un instructor en un año determinado.
RESUMEN: Un Programador de cursos ingresa al sistema, luego al caso de uso Controlar instructores y obtiene el reporte del instructor seleccionado en el año deseado.
CURSO NORMAL DE LOS EVENTOS
<ol style="list-style-type: none"> 1. El sistema genera un archivo XLS definiendo los módulos e instruccionales asignados en un año a un instructor, además establece los horarios, fecha de inicio y de terminación de cada módulo e instruccional programado.
FLUJOS ALTERNATIVOS
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El programador de cursos debe haber ejecutado el caso de uso Controlar

<i>instructores.</i>
POSTCONDICIONES <ul style="list-style-type: none"> ▪ Se descarga un archivo XLS con el reporte de un instructor.

CASO DE USO: Generar reporte de aula
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Obtener el reporte de la asignación de módulos e instruccionales a un aula en un año determinado.
RESUMEN: Un Programador de cursos ingresa al sistema, luego al caso de uso Controlar aulas y obtiene el reporte del aula seleccionado en el año deseado.
CURSO NORMAL DE LOS EVENTOS <ol style="list-style-type: none"> 1. El sistema genera un archivo XLS definiendo los módulos e instruccionales asignados en un año a un aula, además establece los horarios, fecha de inicio y de terminación de cada módulo e instruccional programado.
FLUJOS ALTERNATIVOS
PRECONDICIONES <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El programador de cursos debe haber ejecutado el caso de uso Controlar aulas.
POSTCONDICIONES <ul style="list-style-type: none"> ▪ Se descarga un archivo XLS con el reporte de un aula

CASO DE USO: Generar reporte de calendario
ACTORES: Programador de cursos (iniciador).
PROPOSITO: Obtener el reporte de las fechas no lectivas en un año determinado.
RESUMEN: Un Programador de cursos ingresa al sistema, luego al caso de uso Controlar fechas no lectivas y obtiene el reporte correspondiente.
CURSO NORMAL DE LOS EVENTOS <ol style="list-style-type: none"> 1. El sistema genera un archivo XLS definiendo las fechas no lectivas para todos los instructores, para los instructores de planta, y para los instructores que pertenecen al sindicato.
FLUJOS ALTERNATIVOS
PRECONDICIONES <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El programador de cursos debe haber ejecutado el caso de uso Controlar fechas no lectivas.
POSTCONDICIONES <ul style="list-style-type: none"> ▪ Se descarga un archivo XLS con el reporte de un calendario con las fechas no lectivas.

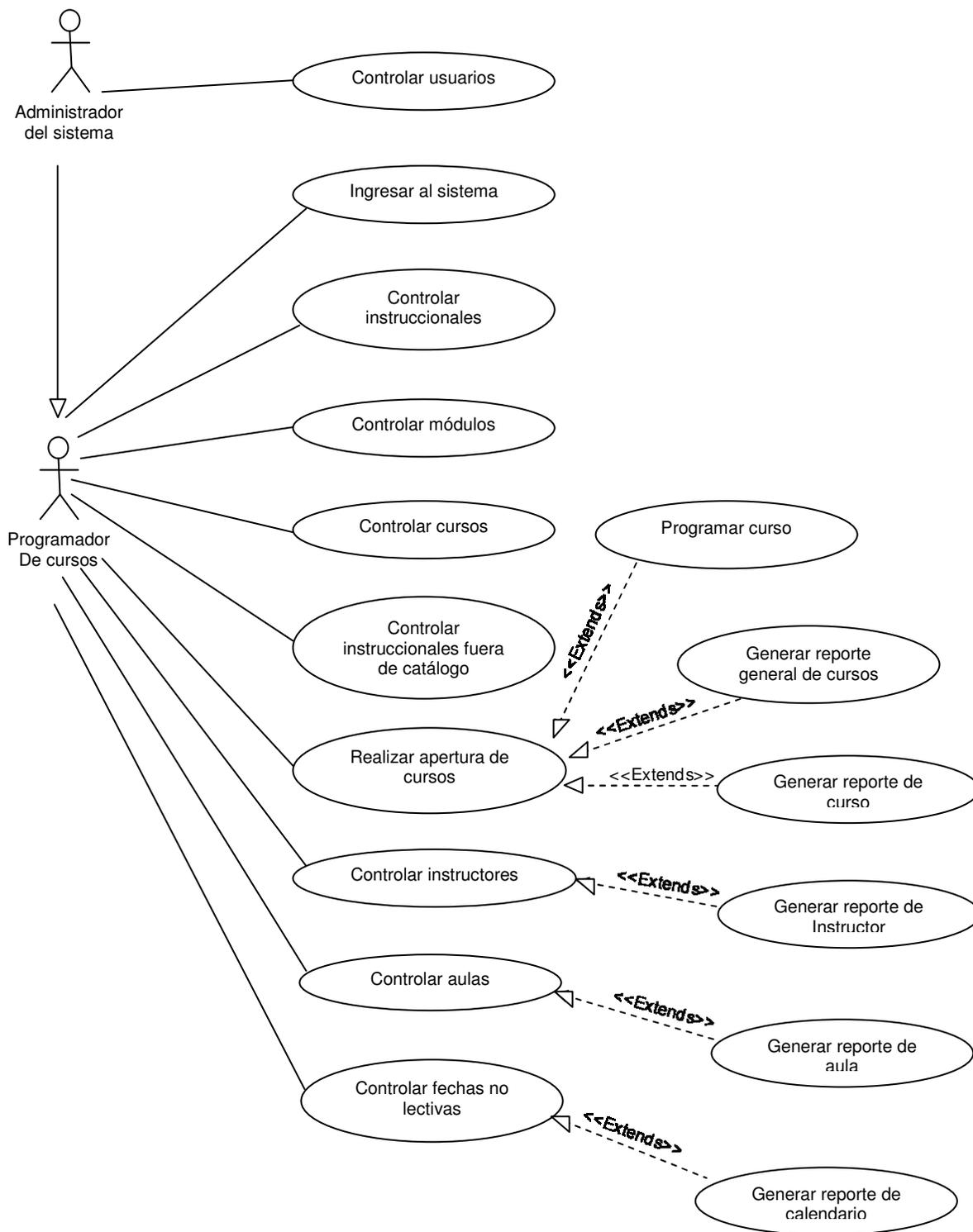
CASO DE USO: Generar reporte de curso
ACTORES: Programador de cursos, Administrador del sistema.
PROPOSITO: Obtener el reporte de un curso programado en un año específico.
RESUMEN: Un Programador de cursos ingresa al sistema, luego al caso de uso Realizar Apertura de cursos y obtiene el reporte correspondiente.

<p>CURSO NORMAL DE LOS EVENTOS</p> <ol style="list-style-type: none"> 1. El sistema genera un archivo XLS definiendo la fecha de inicio, fecha de terminación el horario y los instructores asignados a cada módulo o instruccional del curso programado.
<p>FLUJOS ALTERNATIVOS</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El programador de cursos debe haber ejecutado el caso de uso Realizar apertura de cursos.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se descarga un archivo XLS con el reporte de un curso.

<p>CASO DE USO: Generar reporte general de cursos</p>
<p>ACTORES: Programador de cursos, Administrador del sistema.</p>
<p>PROPOSITO: Obtener el reporte de cursos abiertos en un año específico.</p>
<p>RESUMEN: Un Programador de cursos ingresa al sistema, luego al caso de uso Realizar Apertura de cursos y obtiene el reporte general de cursos en un año determinado.</p>
<p>CURSO NORMAL DE LOS EVENTOS</p> <ol style="list-style-type: none"> 1. El sistema genera un archivo XLS definiendo el número de orden, el código, la versión, el nombre, el número de alumnos, la intensidad horaria, la jornada, el director de grupo, la fecha de inicio y la fecha de terminación de cada curso abierto en un año específico.
<p>FLUJOS ALTERNATIVOS</p>
<p>PRECONDICIONES</p> <ul style="list-style-type: none"> ▪ El Programador de cursos debe haber ingresado al sistema con su Identificación y contraseña. ▪ El programador de cursos debe haber ejecutado el caso de uso Realizar apertura de cursos.
<p>POSTCONDICIONES</p> <ul style="list-style-type: none"> ▪ Se descarga un archivo XLS con el reporte de todos los cursos abiertos.

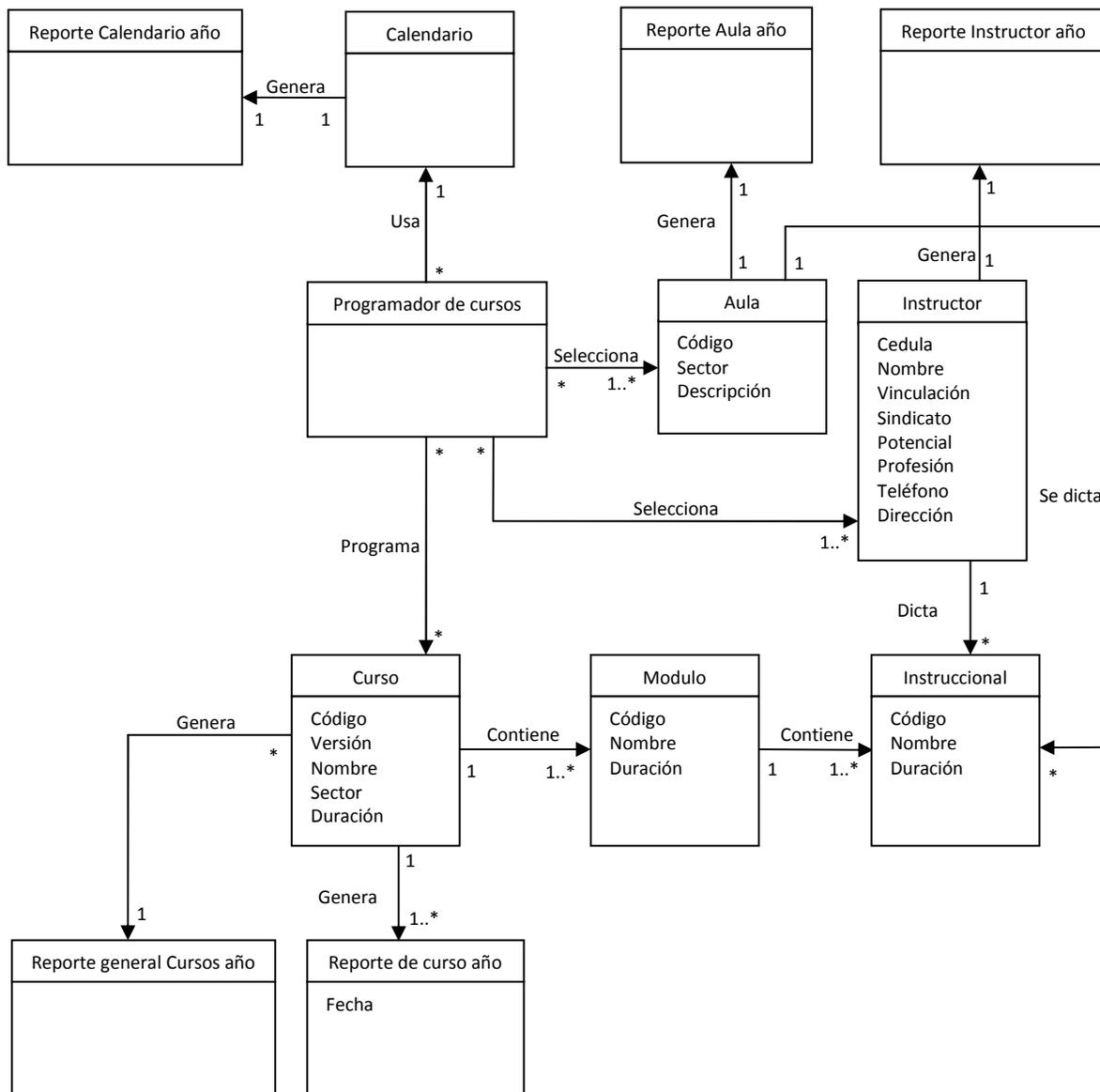
5.3 DIAGRAMA DE CASOS DE USO

Figura 1. Diagrama de casos de uso



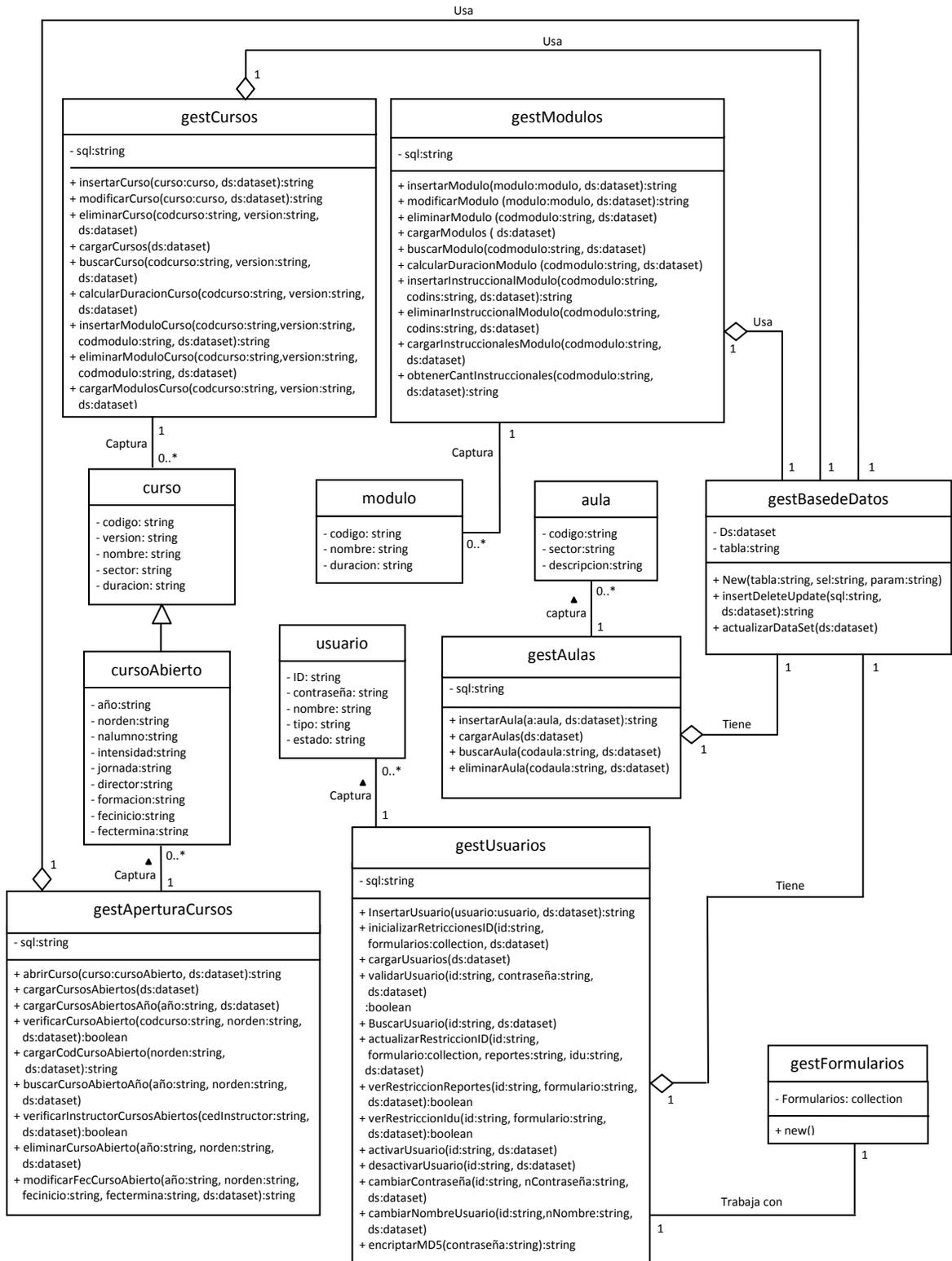
5.4 MODELO CONCEPTUAL

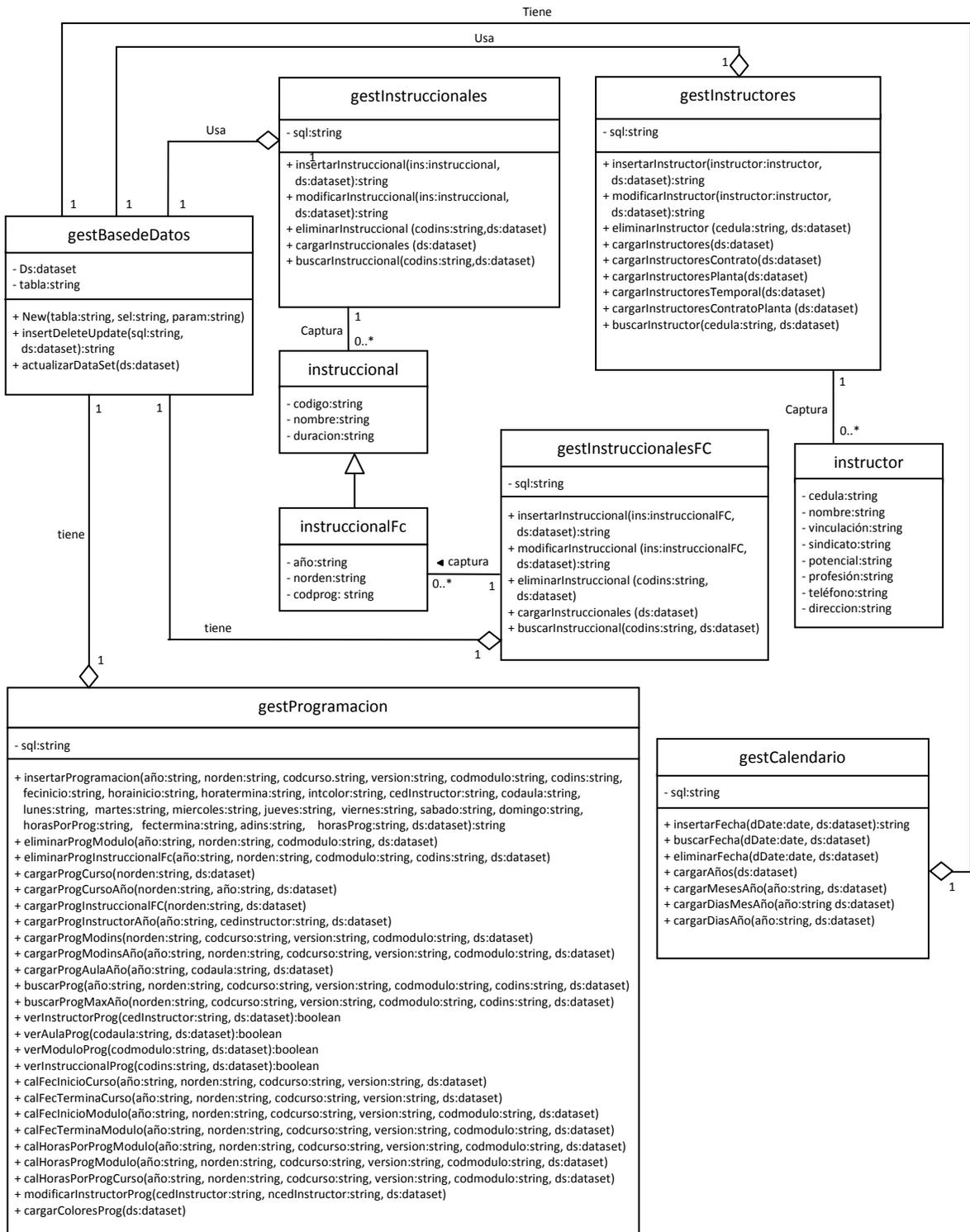
Figura 2. Modelo conceptual

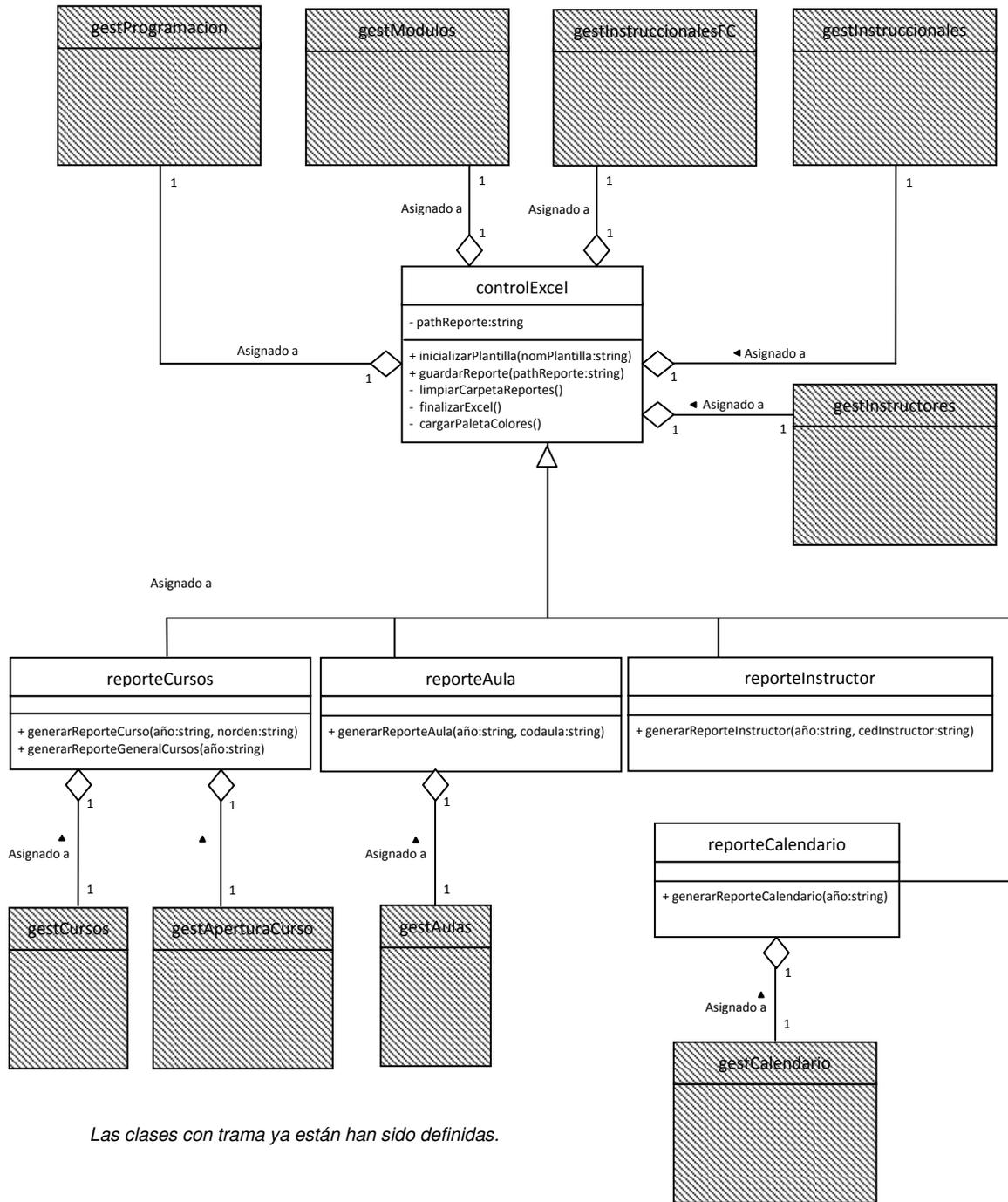


5.5 DIAGRAMA DE CLASES

Figura 3. Diagrama de clases







5.6 DISEÑO DE LA BASE DE DATOS

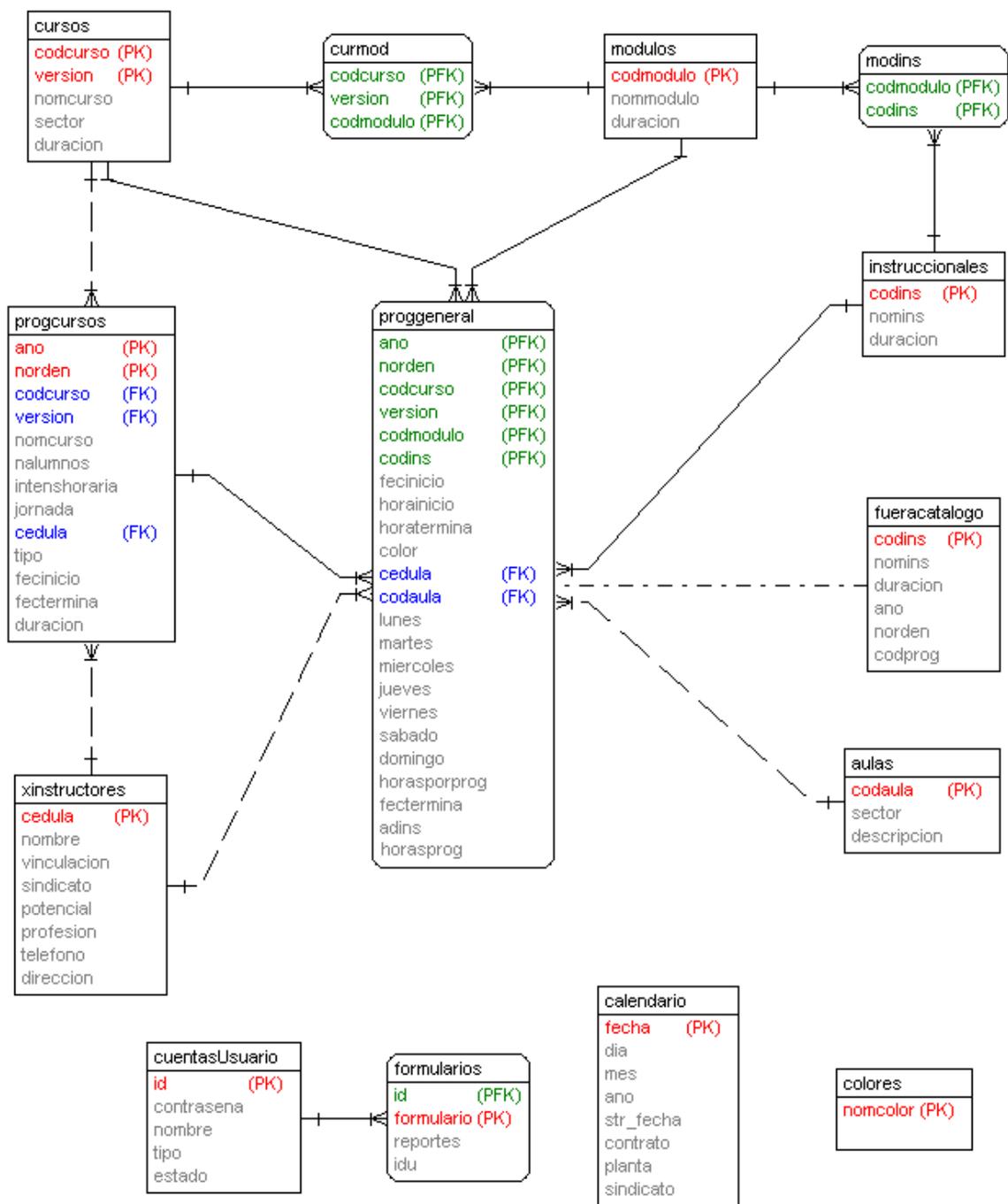
Las bases de datos relacionales son el tipo de bases de datos actualmente más difundido. Los motivos de este éxito son debido a que ofrecen sistemas simples y eficaces para representar y manipular los datos.

La estructura fundamental del modelo relacional es la "relación", es decir una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad. Por ejemplo, si en la base de datos se tienen que representar personas, se podrá definir una relación llamada "Personas", cuyos atributos describen las características de las personas. Cada tupla de la relación "Personas" representará una persona concreta. Para distinguir una tupla de otra, se recurre al concepto de "llave primaria" es decir, un atributo o conjunto de atributos que identifican de forma segura y exclusiva una fila o tupla en una relación. Cuando una relación debe hacer referencia a otra relación, se recurre al concepto de "llave foránea" es decir, un atributo o conjunto de atributos de una relación cuyos valores deben de concordar con los de una llave primaria de alguna otra relación, el objetivo es tener concordancia entre datos de dos relaciones. A esta correspondencia se le conoce como integridad referencial.

La base de datos del sistema web desarrollado en este proyecto se diseño teniendo en cuenta los conceptos de bases de datos relacionales y de manera paralela a la codificación de las historias de usuario y tareas de programación definidas.

5.6.1 Diagrama entidad relación

Figura 4. Diagrama entidad relación



5.6.2 Tablas de la base de datos

Fueracatalogo				
Descripcion: Esta tabla almacena los instruccionales fuera de catálogo de cursos.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Codins	Carácter	4	Primaria	Código del instruccional
Nomins	Carácter	100		Nombre del instruccional
Duracion	Numérico	4		Duración del instruccional
Año	Carácter	4		Año de programación
Norden	Carácter	5		Numero de orden
Codprog	Carácter	10		Código programado

Instruccionales				
Descripcion: Esta tabla almacena los instruccionales que se asignan al catálogo de cursos				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Codins	Carácter	10	Primaria	Código del instruccional
Nomins	Carácter	100		Nombre del instruccional
Duración	Numérico	4		Duración del instruccional

Modins				
Descripcion: Esta tabla almacena permite el almacenamiento de los instruccionales asignados a los módulos				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Codmodulo	Carácter	8	Primaria, Foránea	Código del módulo
Codins	Carácter	10	Primaria, Foránea	Código del instruccional

Módulos				
Descripcion: Esta tabla almacena los módulos que se asignan al catálogo de cursos				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Codmodulo	Carácter	8	Primaria	Código del módulo
Nommodulo	Carácter	100		Nombre del módulo
Duración	Numérico	4		Duración del módulo

Curmod				
Descripción: Esta tabla almacena permite el almacenamiento de los módulos asignados a los cursos				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Codcurso	Carácter	6	Primaria, Foránea	Código del curso
Version	Carácter	2	Primaria, Foránea	Versión del curso
Codmodulo	Carácter	8	Primaria, Foránea	Código del módulo

Cursos				
Descripción: Esta tabla almacena los cursos que se asignan al catálogo de cursos				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Codcurso	Carácter	6	Primaria	Código del curso
Version	Carácter	2	Primaria	Versión del curso
Nomcurso	Carácter	100		Nombre del curso
Sector	Carácter	2		Sector al que pertenece el curso
Duración	Numerico	4		Duración del curso

xInstructores				
Descripción: Esta tabla almacena los instructores activos.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Cedula	Fecha	8	Primaria	Cedula del instructor
Nombre	Carácter	45		Nombre del instructor
Vinculación	Carácter	8		Tipo de vinculación laboral
Sindicato	Carácter	2		Pertenece al sindicato
Potencial	Numerico	4		Horas que debe trabajar en el año
Profesión	Carácter	45		Profesión del instructor
Teléfono	Carácter	20		Teléfono del instructor
Dirección	Carácter	60		Dirección del instructor

Aulas				
Descripción: Esta tabla almacena las aulas de clase.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Codaula	Carácter	7	Primaria	Código del aula
Str_sector	Carácter	12		Sector al que pertenece el aula
Descripcion	Carácter	100		Descripción del aula

Calendario				
Descripción: Esta tabla almacena las fechas no lectivas para instructores de contrato, instructores de planta e instructores que pertenecen al sindicato.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Fecha	Fecha	8	Primaria	Fecha dd-mm-aa
Día	Carácter	2		Día
Mes	Carácter	2		Mes
Año	Carácter	4		Año
Str_fecha	Carácter	40		Fecha en formato de cadena
Contrato	Booleano	1		Contrato
Planta	Booleano	1		Planta
Sindicato	Booleano	1		Sindicato

Progursos				
Descripción: Esta tabla almacena los cursos que se han abierto.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Ano	Carácter	4	Primaria	Año de programación
Norden	Carácter	5	Primaria	Numero de orden
Codcurso	Carácter	8	Foránea	Código del curso o módulo
Versión	Carácter	2	Foránea	Versión del curso
Nomcurso	Carácter	100		Nombre del curso
Nalumnos	Numérico	4		Número de alumnos
intenshoraria	Numérico	4		Intensidad horaria
Jornada	Carácter	1		Jornada
Cedula	Carácter	8	Foránea	Cedula del director de grupo
Tipo	Booleano	1		Tipo de formación
Fecinicio	Fecha	8		Fecha de inicio
Fectermina	Fecha	8		Fecha de terminación
Duración	Numérico	4		Duración del curso

Proggeneral				
Descripción: Esta tabla almacena la programación de módulos, instruccionales, aulas e instructores de los cursos abiertos.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Ano	Carácter	4	Primaria	Año de programación
Norden	Carácter	5	Primaria	Numero de orden
Codcurso	Carácter	8	Primaria, Foránea	Código del curso
Versión	Carácter	2	Primaria, Foránea	Versión del curso
Codmodulo	Carácter	8	Primaria, Foránea	Código del módulo
Codins	Numérico	10	Primaria, Foránea	Código del instruccional
Fecinicio	Fecha	8		Fecha de inicio
Horainicio	Carácter	4		Hora de inicio
Horatermina	Carácter	4		Hora de terminación

Intcolor	Numérico	4		Color
Cedula	Carácter	8	Foránea	Cedula del instructor
Codaula	Carácter	7	Foránea	Código del aula
Lunes	Carácter	1		Lunes
Martes	Carácter	1		Martes
Miércoles	Carácter	1		Miércoles
Jueves	Carácter	1		Jueves
Viernes	Carácter	1		Viernes
Sabado	Carácter	1		Sábado
Domingo	Carácter	1		Domingo
Horasporprog	Numérico	4		Horas por programar
Fectermina	Fecha	8		Fecha de terminación
Adins	Carácter	1		Adición automática de instruccional
horasProg	Numérico	4		Horas programadas reales.

Cuentasusuario				
Descripcion: Esta tabla almacena los usuarios con acceso al sistema.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
id	Carácter	15	Primaria	Identificación
Contraseña	Carácter	32		Contraseña MD5
Nombre	Carácter	45		Nombre de usuario
Tipo	Carácter	20		Tipo de usuario
Estado	Carácter	20		Estado de cuenta

Formularios				
Descripcion: Esta tabla almacena los privilegios que tienen los usuarios en los formularios del sistema				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
id	Carácter	15	Primaria, Foránea	Identificación
Formulario	Carácter	25	Primaria	Formulario (pagina web)
Reportes	Booleano	1		Reportes
ldu	Booleano	1		Inserción, eliminación actualización

Colores				
Descripcion: Esta tabla almacena los colores que se manejan para identificar los módulos e instruccionales que se han programado.				
Atributo	Tipo	Tamaño (bytes)	Llave	Descripción
Nomcolor	Carácter	20	Primaria	Nombre del color

5.7 GLOSARIO

5.7.1 Casos de uso

Término	Categoría	Descripción
Controlar aulas	Caso de uso	Descripción del proceso de inserción y eliminación de aulas.
Controlar cursos	Caso de uso	Descripción del proceso de inserción modificación eliminación y administración de cursos.
Controlar fechas no lectivas	Caso de uso	Descripción del proceso de inserción y eliminación de fechas no lectivas para instructores de contrato, planta y pertenecientes al sindicato.
Controlar Instruccionales fuera de catalogo	Caso de uso	Descripción del proceso de inserción modificación eliminación y administración de instruccionales que no se asignaran a ningún módulo.
Controlar instruccionales	Caso de uso	Descripción del proceso de inserción modificación eliminación y administración de instruccionales.
Controlar instructores	Caso de uso	Descripción del proceso de inserción modificación eliminación y administración de instructores
Controlar módulos	Caso de uso	Descripción del proceso de inserción modificación eliminación y administración de módulos.
Controlar usuarios	Caso de uso	Descripción del proceso de inserción modificación eliminación y otorgamiento de privilegios a los usuarios del sistema.
Ingresar al sistema	Caso de uso	Descripción del proceso de ingreso al sistema.
Programar cursos	Caso de uso	Descripción del proceso de programación de módulos, instruccionales, aulas e instructores de los cursos abiertos.
Realizar apertura de cursos	Caso de uso	Descripción del proceso de apertura de cursos.

5.7.2 Conceptos

Término	Categoría	Descripción
Aula	Concepto	Espacio físico donde se dicta un módulo o instruccional.
Calendario	Concepto	Conjunto de fechas de un año organizadas.
Curso	Concepto	Conjunto de módulos organizados.
Instruccional	Concepto	Contenido temático o materia.
Instructor	Concepto	Persona encargada de dictar un módulo o instruccional.
Módulo	Concepto	Conjunto de instruccionales organizados.
Programador de cursos	Concepto	Persona encargada de la programación de cursos.
Reporte aula año	Concepto	Diagrama de Gantt que determina la asignación de módulos e instruccionales a un aula en un año.
Reporte calendario año	Concepto	Listado de fechas no lectivas en un año.
Reporte de curso año	Concepto	Diagrama de Gantt que determina la programación de módulos e instruccionales de un curso en un año.
Reporte general cursos año	Concepto	Listado de los cursos abiertos en un año.
Reporte instructor año	Concepto	Diagrama de Gantt que determina la asignación de módulos e instruccionales a un instructor en un año.

5.7.3 Clases métodos y atributos

Término	Categoría	Descripción
aula	Clase	Establece los datos específicos que tiene un aula.
aula.codigo	Atributo	Define el código de aula.
aula.descripcion	Atributo	Define una descripción general del aula.
aula.sector	Atributo	Define uno de los tres sectores del Sena. (Comercio, servicios, agropecuario, Industrial) en donde se encuentra ubicada el aula.
boolean	Clase	Define un objeto tipo booleano.
collection	Clase	Define una colección de objetos.
controlExcel	Clase	Define las operaciones para el control la aplicación Microsoft Excel 2003.
controlExcel.cargarPaletaColores(..)	Método	Carga la paleta de colores para el manejo de relleno de celdas en los reportes generados.
controlExcel.finalizarExcel(..)	Método	Cierra el proceso EXCEL en el servidor.
controlExcel.guardarReporte(..)	Método	Guarda el reporte generado en la ruta especificada.
controlExcel.inicializarPlantilla(..)	Método	Carga en el servidor la aplicación Excel con la plantilla o formato de reporte elegido.
controlExcel.limpiarCarpetaReportes(..)	Método	Elimina todos los reportes almacenados en el servidor.
controlExcel.pathReporte	Atributo	Define la ruta de acceso y nombre de archivo del reporte que se genera.
curso	Clase	Establece los datos específicos que tiene un curso.
curso.codigo	Atributo	Define el código del curso.
curso.duracion	Atributo	Define la duración en horas del curso.
curso.nombre	Atributo	Define el nombre del curso.
curso.sector	Atributo	Define uno de los tres sectores del Sena. (Comercio, servicios,

		agropecuario, Industrial) al que pertenece el curso.
curso.version	Atributo	Define la versión del curso.
cursoAbierto	Clase	Establece los datos específicos que tiene un curso cuando se lo abre.
cursoAbierto.año	Atributo	Define el año de apertura del curso.
cursoAbierto.director	Atributo	Define la cédula del director de grupo asignado al curso abierto.
cursoAbierto.fecinicio	Atributo	Define la fecha de inicio del curso abierto.
cursoAbierto.fectermina	Atributo	Define la fecha de terminación del curso abierto.
cursoAbierto.formacion	Atributo	Define el tipo de formación del curso abierto (Titulada=Curso, Complementaria=Módulo).
cursoAbierto.intensidad	Atributo	Define la intensidad horaria del curso abierto.
cursoAbierto.jornada	Atributo	Define la jornada que tiene el curso abierto (Diurna, Nocturna).
cursoAbierto.nalumnos	Atributo	Define el número de alumnos que maneja el curso abierto.
cursoAbierto.norden	Atributo	Define el número de orden asignado al curso abierto.
dataset	Clase	Define la representación de un conjunto completo de datos. (Permite la manipulación de los datos de la base de datos).
date	Clase	Define un objeto tipo fecha.
gestAperturaCursos	Clase	Define las operaciones para el manejo de los cursos abiertos.
gestAperturaCursos.abrirCurso(..)	Método	Determina en el sistema que un curso se ha abierto.
gestAperturaCursos.buscarCursoAbiertoAño(..)	Método	Carga en un dataset un curso abierto en un año definido.
gestAperturaCursos.cargarCodCursoAbierto(..)	Método	Devuelve el código de un curso abierto registrado en el sistema.
gestAperturaCursos.cargarCursosAbiertos(..)	Método	Carga en un dataset los cursos abiertos registrados en el sistema.
gestAperturaCursos.cargarCursosAbiertosAño(..)	Método	Carga en un dataset los cursos abiertos de un año en particular registrados en el sistema.

gestAperturaCursos.eliminarCursoAbierto(..)	Método	Elimina un curso abierto y toda su programación del sistema.
gestAperturaCursos.modificarFecCursoAbierto(..)	Método	Modifica la fecha de inicio y fecha de terminación de un curso abierto.
gestAperturaCursos.verificarCursoAbierto(..)	Método	Verifica si un curso o un módulo se encuentra abierto.
gestAperturaCursos.verificarInstructorCursosAbiertos(..)	Método	Verifica si un instructor está comprometido como director de grupo en un curso abierto.
gestAulas	Clase	Define las operaciones para el manejo de las aulas.
gestAulas.buscarAula(..)	Método	Carga en un dataset un aula en particular.
gestAulas.cargarAulas(..)	Método	Carga en un dataset las aulas registradas en el sistema.
gestAulas.eliminarAula(..)	Método	Elimina un aula del sistema.
gestAulas.insertarAula(..)	Método	Inserta un aula en el sistema.
gestAulas.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestBasedeDatos	Clase	Permite manejar las operaciones sobre las tablas de la base de datos.
gestBasedeDatos.actualizarDataSet(..)	Método	Actualiza el dataset con los datos generados por una operación realizada sobre la base de datos.
gestBasedeDatos.ds	atributo	Representación de un conjunto de datos.
gestBasedeDatos.insertDeleteUpdate(..)	Método	Realiza la operación de inserción eliminación o actualización sobre la base de datos según una instrucción SQL.
gestBasedeDatos.new(..)	Método	Inicializa
gestBasedeDatos.tabla	atributo	Cadena de caracteres que define una tabla de la base de datos.
gestCalendario	Clase	Define las operaciones para el manejo de las fechas no lectivas.
gestCalendario.buscarFecha(..)	Método	Carga en un dataset una fecha no lectiva en particular.
gestCalendario.cargarAños(..)	Método	Carga en un dataset los años en donde se han determinado fechas no lectivas.
gestCalendario.cargarDiasAño(..)	Método	Carga en un dataset los días de

		un año específico en donde se han determinado fechas no lectivas.
gestCalendario.cargarDiasMesAño(..)	Método	Carga en un dataset los días de un mes y año específicos en donde se han determinado fechas no lectivas.
gestCalendario.cargarMesesAño(..)	Método	Carga en un dataset los meses de un año específico en donde se han determinado fechas no lectivas.
gestCalendario.eliminarFecha(..)	Método	Elimina una fecha no lectiva del sistema.
gestCalendario.insertarFecha(..)	Método	Inserta una fecha no lectiva en el sistema.
gestCalendario.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestCursos	Clase	Define las operaciones para el manejo de los cursos.
gestCursos.buscarCurso(..)	Método	Carga en un dataset un curso en particular.
gestCursos.calcularDuracionCurso(..)	Método	Establece la duración de un curso.
gestCursos.cargarCursos(..)	Método	Carga en un dataset los cursos registrados en el sistema.
gestCursos.cargarModulosCurso(..)	Método	Carga en un dataset los módulos asignados a un curso.
gestCursos.eliminarCurso(..)	Método	Elimina un curso del sistema.
gestCursos.eliminarModuloCurso(..)	Método	Elimina del sistema la asignación de un módulo a un curso.
gestCursos.insertarCurso(..)	Método	Inserta un curso en el sistema.
gestCursos.insertarModuloCurso(..)	Método	Inserta en el sistema la asignación de un módulo a un curso.
gestCursos.modificarCurso(..)	Método	Modifica un curso en el sistema.
gestCursos.sql	atributo	Cadena de caracteres que define una instrucción SQL.
gestFormularios	Clase	Define las operaciones para el manejo de los formularios del sistema.
gestFormularios.formularios	Atributo	Colección de cadenas de caracteres con los nombre de los formularios del sistema.

gestFormularios.new(..)	Método	Instancia los formularios del sistema en una colección.
gestInstruccionales	Clase	Define las operaciones para el manejo de los instruccionales.
gestInstruccionales.buscarInstruccional(..)	Método	Carga en un dataset un instruccional en particular.
gestInstruccionales.cargarInstruccionales(..)	Método	Carga en un dataset los instruccionales registrados en el sistema.
gestInstruccionales.eliminarInstruccional(..)	Método	Elimina un instruccional del sistema.
gestInstruccionales.insertarInstruccional(..)	Método	Inserta un instruccional en el sistema.
gestInstruccionales.modificarInstruccional(..)	Método	Modifica un instruccional en el sistema.
gestInstruccionales.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestInstruccionalesFC	Clase	Define las operaciones para el manejo de los instruccionales fuera de catálogo.
gestInstruccionalesFC.buscarInstruccional(..)	Método	Carga en un dataset un instruccional fuera de catálogo en particular.
gestInstruccionalesFC.cargarInstruccionales(..)	Método	Carga en un dataset los instruccionales fuera de catálogo registrados en el sistema.
gestInstruccionalesFC.eliminarInstruccional(..)	Método	Elimina un instruccional fuera de catálogo del sistema.
gestInstruccionalesFC.insertarInstruccional(..)	Método	Inserta un instruccional fuera de catálogo en el sistema.
gestInstruccionalesFC.modificarInstruccional(..)	Método	Modifica un instruccional fuera de catálogo en el sistema.
gestInstruccionalesFC.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestInstructor.insertarInstructor(..)	Método	Inserta un instructor en el sistema.
gestInstructores	Clase	Define las operaciones para el manejo de los instructores.
gestInstructores.buscarInstructor(..)	Método	Carga en un dataset un instruccional en particular.
gestInstructores.cargarInstructores(..)	Método	Carga en un dataset los instructores registrados en el sistema.
gestInstructores.cargarInstructoresContrato(..)	Método	Carga en un dataset los instructores de contrato registrados en el sistema.

gestInstructores.cargarInstructoresContratoPlanta(..)	Método	Carga en un dataset los instructores de contrato y de planta registrados en el sistema.
gestInstructores.cargarInstructoresPlanta(..)	Método	Carga en un dataset los instructores de planta registrados en el sistema.
gestInstructores.cargarInstructoresTemporal(..)	Método	Carga en un dataset los instructores de vinculación temporal registrados en el sistema.
gestInstructores.eliminarInstructor(..)	Método	Elimina un instructor del sistema.
gestInstructores.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestModulos	Clase	Define las operaciones para el manejo de los módulos.
gestModulos.buscarModulo(..)	Método	Carga en un dataset un módulo en particular.
gestModulos.calcularDuraciónModulo(..)	Método	Establece la duración de un módulo.
gestModulos.cargarInstruccionalesModulo(..)	Método	Carga en un dataset los instruccionales asignados a un módulo.
gestModulos.cargarModulos(..)	Método	Carga en un dataset los módulos registrados en el sistema.
gestModulos.eliminarInstruccionalesModulo(..)	Método	Elimina del sistema la asignación de un instruccional a un módulo.
gestModulos.eliminarModulo(..)	Método	Elimina un módulo del sistema.
gestModulos.insertarInstruccionalesModulo(..)	Método	Inserta en el sistema la asignación de un instruccional a un módulo.
gestModulos.insertarModulo(..)	Método	Inserta un módulo en el sistema.
gestModulos.modificarModulo(..)	Método	Modifica un módulo en el sistema.
gestModulos.obtenerCantInstruccionales(..)	Método	Carga la cantidad de instruccionales de un módulo.
gestModulos.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestProgramacion	Clase	Define las operaciones para el manejo de programación de módulos, instruccionales, aulas e instructores asignados a un curso abierto.
gestProgramacion.buscarProg(..)	Método	Carga en un dataset la

		programación de un módulo o instruccional definido.
gestProgramacion.buscarProgMaxAño(..)	Método	Carga en un dataset la programación del último año de un módulo o instruccional definido.
gestProgramacion.calFeclniciaModulo(..)	Método	Calcula la fecha de inicio de un módulo.
gestProgramacion.calFeclnicioCurso(..)	Método	Calcula la fecha de inicio de un curso.
gestProgramacion.calFecTerminaCurso(..)	Método	Calcula la fecha de terminación de un curso.
gestProgramacion.calFecTerminaModulo(..)	Método	Calcula la fecha de terminación de un módulo
gestProgramacion.calHorasPorProgCurso(..)	Método	Calcula las horas por programar de un curso.
gestProgramacion.calHorasPorProgModulo(..)	Método	Calcula las horas por programar de un módulo.
gestProgramacion.calHorasProgModulo(..)	Método	Calcula las horas programadas de un módulo cuando este no ha sido programado totalmente.
gestProgramacion.cargarProgAulaAño(..)	Método	Carga en un dataset la programación de un aula en un año.
gestProgramacion.cargarProgCurso(..)	Método	Carga en un dataset la programación de los módulos e instruccionales de un curso.
gestProgramacion.cargarProgCursoAño(..)	Método	Carga en un dataset la programación de los módulos e instruccionales de un curso de un año específico.
gestProgramacion.cargarProgInstruccionalFC(..)	Método	Carga en un dataset la programación de los instruccionales fuera de catálogo de un curso.
gestProgramacion.cargarProgInstructorAño(..)	Método	Carga en un dataset la programación de un instructor en un año.
gestProgramacion.cargarProgModins(..)	Método	Carga en un dataset la programación de los instruccionales de un módulo.
gestProgramacion.cargarProgModinsAño(..)	Método	Carga en un dataset la programación de los instruccionales de un módulo en un año definido.
gestProgramacion.eliminarProgInstruccionalFC(. .)	Método	Elimina la programación de un instruccional fuera del catálogo del sistema.
gestProgramacion.eliminarProgModulo(..)	Método	Elimina la programación de un instruccional o módulo del sistema.

gestProgramacion.insertarProgramacion(..)	Método	Inserta la programación de un instruccional o módulo en el sistema.
gestProgramacion.modificarInstructorProg(..)	Método	Modifica la cédula de un instructor que tiene asociado módulos o instruccionales.
gestProgramacion.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestProgramacion.verAulaProg(..)	Método	Verifica si un aula tiene asignado al menos un módulo o instruccional.
gestProgramacion.verInstruccionalProg(..)	Método	Verifica si un instruccional se encuentra completamente programado.
gestProgramacion.verInstructorProg(..)	Método	Verifica si un instructor tiene asignado al menos un módulo o instruccional.
gestProgramacion.verModuloProg(..)	Método	Verifica si un módulo se encuentra completamente programado.
gestUsuario.cambiarContraseña(..)	Método	Cambia la contraseña de un usuario del sistema.
gestUsuarios	Clase	Define las operaciones para el manejo de los usuarios del sistema.
gestUsuarios.activarUsuario(..)	Método	Activa un usuario del sistema.
gestUsuarios.actualizarRestriccionID(..)	Método	Actualiza las restricciones de uso de los formularios del sistema para un usuario.
gestUsuarios.buscarUsuario(..)	Método	Carga en un dataset un usuario del sistema en particular.
gestUsuarios.cambiarNombreUsuario(..)	Método	Cambia el nombre de un usuario del sistema.
gestUsuarios.cargarUsuarios(..)	Método	Carga en un dataset los usuarios registrados en el sistema.
gestUsuarios.desactivarUsuario(..)	Método	Desactiva un usuario del sistema.
gestUsuarios.encryptedMD5(..)	Método	Encriptar contraseñas con el algoritmo MD5 (Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5, algoritmo de reducción criptográfico).
gestUsuarios.InicializarRestriccionesID(..)	Método	Inicializa en el sistema las restricciones de formularios para un usuario. (Define al usuario sin restricciones de uso).
gestUsuarios.insertarUsuario(..)	Método	Inserta un usuario en el sistema.

gestUsuarios.sql	Atributo	Cadena de caracteres que define una instrucción SQL.
gestUsuarios.validarUsuario(..)	Método	Valida la correspondencia del ID y contraseña de un usuario existente en el sistema.
gestUsuarios.verRestriccionIdu(..)	Método	Verifica si un usuario tiene la restricción de inserción modificación o eliminación de registros en un formulario del sistema.
gestUsuarios.verRestriccionReportes(..)	Método	Verifica si un usuario tiene la restricción de acceso a reportes en un formulario del sistema.
instruccional	Clase	Establece los datos específicos que tiene un instruccional.
instruccional.codigo	Atributo	Define el código del instruccional.
instruccional.codprog	Atributo	Código del módulo o instruccional que se extiende.(Un módulo o instruccional se extiende cuando requieren de una programación adicional.
instruccional.duracion	Atributo	Define la duración en horas del instruccional.
instruccional.nombre	Atributo	Define el nombre del instruccional.
instruccionalFC	Clase	Establece los datos específicos que tiene un instruccional que no pertenece a ningún módulo. (Fuera del catálogo de cursos).
instruccionalFC.año	Atributo	Año en el que está programado el módulo o instruccional que se extiende.
instruccionalFC.norden	Atributo	Número de orden del curso programado al cual pertenece el módulo o instruccional que se extiende
instructor	Clase	Establece los datos específicos que tiene un instructor.
instructor.cedula	Atributo	Define la cédula de ciudadanía del instructor.
instructor.direccion	Atributo	Define la dirección del instructor
instructor.modificarInstructor(..)	Método	Modifica un instructor en el sistema.
instructor.nombre	Atributo	Define el nombre y apellido del instructor.
instructor.potencial	Atributo	Define el número de horas de

		trabajo establecidas para un instructor en un año.
instructor.profesion	Atributo	Define la profesión del instructor
instructor.sindicato	Atributo	Define si el instructor pertenece o no al sindicato.
instructor.telefono	Atributo	Define el teléfono del instructor.
instructor.vinculacion	Atributo	Define el tipo de vinculación laboral del instructor (Contrato, Planta).
modulo	Clase	Establece los datos específicos que tiene un módulo.
modulo.codigo	Atributo	Define el código del módulo.
modulo.duracion	Atributo	Define la duración en horas del módulo.
modulo.nombre	Atributo	Define el nombre del módulo.
reporteAula	Clase	Define las operaciones para generar el reporte de aulas programadas.
reporteAula.generarReporteAula(..)	Método	Genera el reporte de la programación de un aula en un año.
reporteCalendario	Clase	Define las operaciones para generar el reporte de las fechas no lectivas.
reporteCalendario.generarReporteCalendario(..)	Método	Genera el reporte de fechas no lectivas en un año.
reporteCursos	Clase	Define las operaciones para generar los reportes de los cursos programados.
reporteCursos.generarReporteCurso(..)	Método	Genera el reporte de la programación de un curso en un año.
reporteCursos.generarReporteGeneralCursos(..)	Método	Genera un reporte con el listado de los cursos abiertos en un año.
reporteInstructor	Clase	Define las operaciones para generar los reportes de los instructores programados.
reporteInstructor.generarReporteInstructor(..)	Método	Genera el reporte de la programación de un aula en un año.
string	Clase	Define un objeto tipo cadena de caracteres.
usuario	Clase	Establece los datos específicos

		que tiene un usuario del sistema.
usuario.contraseña	Atributo	Define la contraseña de acceso al usuario del sistema.
usuario.estado	Atributo	Define el estado del usuario(Activo, Desactivo)
usuario.ID	Atributo	Define la identificación (login) del usuario del sistema.
usuario.nombre	Atributo	Define el nombre y apellido del usuario del sistema.
usuario.tipo	Atributo	Define el tipo de usuario (Administrador, Usuario).

5.7.4 Parámetros

Término	Categoría	Descripción
adins	Parámetro	Define si un instruccional ha sido programado de forma automática tras la programación del módulo al cual se encuentra asociado.
año->gestAperturaCursos	Parámetro	Define el año en que se abre un curso.
año->gestCalendario	Parámetro	Define el año de una fecha no lectiva.
año->gestProgramacion	Parámetro	Define el año en que se programa un módulo o instruccional.
año->reporteAula	Parámetro	Define el año de programación del aula.
año->reporteCalendario	Parámetro	Define el año del calendario.
año->reporteCursos	Parámetro	Define el año del curso(s) abierto del cual se va a generar el reporte.
año->reporteInstructor	Parámetro	Define el año de programación del instructor.
cedInstructor	Parámetro	Cédula del instructor que dicta un módulo o instruccional.
codAula	Parámetro	Código de aula en la que se dicta un módulo o instruccional.
codcurso	Parámetro	Define el código de un curso
codins	Parámetro	Define el código de un instruccional
codmodulo	Parámetro	Define el código de un módulo
dDate	Parámetro	Una fecha del calendario.
domingo	Parámetro	Define si el módulo o instruccional programado se dicta los días domingo.
ds	Parámetro	Instancia de un dataset.
duracion	Parámetro	Duración (horas) de un módulo o instruccional.
fecinicio	Parámetro	Fecha de inicio un módulo, instruccional o curso abierto.
fectermina	Parámetro	Fecha de terminación de un módulo, instruccional curso abierto.

formulario	Parámetro	Define un formulario (pagina web) del sistema.
horalnicio	Parámetro	Hora de inicio de un módulo o instruccional.
horasPorProg	Parámetro	Horas por programar de un módulo o instruccional.
horasProg	Parámetro	Horas reales de programación de un módulo o instruccional.
horaTermina	Parámetro	Hora de terminación de un módulo o instruccional.
ID	Parámetro	Identificación (login) de un usuario del sistema,.
idu	Parámetro	Define si un usuario tiene o no restricciones de inserción modificación o eliminación en un formulario del sistema.
intColor	Parámetro	Numero de un color asociado a un módulo o instruccional programado.
jueves	Parámetro	Define si el módulo o instruccional programado se dicta los días jueves.
lunes	Parámetro	Define si el módulo o instruccional programado se dicta los días lunes.
martes	Parámetro	Define si el módulo o instruccional programado se dicta los días martes.
miercoles	Parámetro	Define si el módulo o instruccional programado se dicta los días miércoles.
ncedInstructor	Parámetro	Nueva cédula de instructor asignada aun módulo o instruccional programado.
nContraseña	Parámetro	Define la nueva contraseña para un usuario del sistema.
nNombre	Parámetro	Define el nuevo nombre de usuario para un usuario del sistema.
nomPlantilla	Parámetro	Define el nombre de archivo de la plantilla de reporte.
norden	Parámetro	Numero de orden de un curso. (Código asignado a un curso abierto)
param	Parámetro	Define los parámetros de una operación que se realiza sobre la base de datos.
pathReporte	Parámetro	Define la ruta de acceso y nombre de archivo del reporte que se va guardar.
reportes	Parámetro	Define si un usuario tiene o no restricciones de acceso a reportes en un formulario del sistema.
sabado	Parámetro	Define si el módulo o instruccional programado se

		dicta los días sábado.
sel	Parámetro	Define la operación de consulta que se va a realizar sobre la base de datos.
tabla	Parámetro	Define una tabla de la base de datos.
version	Parámetro	Define la versión de un curso
viernes	Parámetro	Define si el módulo o instruccional programado se dicta los días viernes.

6. CONCLUSIONES

El sistema web se instaló satisfactoriamente en un equipo servidor del centro multisectorial Lope, brindando a todas sus dependencias la posibilidad de ejecutar la aplicación valiéndose de la red local, esta condición permite que el proceso de programación de cursos se estandarice para todos los programadores.

El módulo de reportes genera de manera muy ágil los horarios de clase de alumnos e instructores, permite conocer el estado de la programación de módulos e instruccionales de un curso abierto, define la disponibilidad de las aulas y el calendario de fechas no lectivas para instructores.

El módulo de programación de cursos permite realizar el proceso de programación de módulos e instruccionales permitiendo al usuario validar fechas, horarios, asignación de instructores y aulas muy eficientemente.

Durante el transcurso de construcción del software se pudo notar las bondades de la plataforma .Net particularmente en la construcción de aplicaciones web y manejo de bases de datos. La plataforma brinda un contexto coherente, rico y ordenado de programación orientada objetos lo que hace del software implementado una herramienta escalable y de fácil mantenimiento.

La metodología extreme programming ofrece gran agilidad y velocidad en el desarrollo de software, sin embargo, genera aplicaciones de carente documentación, en este caso es aconsejable apoyarse de herramientas como el lenguaje de modelado unificado UML.

7. RECOMENDACIONES

Asignar un webmaster responsable del sistema web para mantenimiento.

Realizar una capacitación sobre el manejo del sistema a los programadores de cursos asignados.

Dar a conocer el sistema web desarrollado y proponer un proyecto de actualización del sistema web para que funcione a nivel nacional.

BIBLIOGRAFÍA

- Larman, Craig. UML Y Patrones Introducción al análisis y diseño orientado a objetos. PRENTICE HALL, México, 1999. ISBN: 970-17-0261-1.
- Extreme programming. Disponible en web: <http://www.extremeprogramming.org>. Consulta: Noviembre 15 de 2006.
- Letelier, Patricio y Penadés, M^a Carmen. Metodologías ágiles para el desarrollo de software, Universidad Politécnica de Valencia. Disponible en web: <http://www.willydev.net/descargas/masyxp.pdf>. Consulta: Noviembre 15 de 2007.
- Ejemplo de desarrollo de software utilizando la metodología Xp, Departamento de sistemas, información y computación, Universidad politécnica de Valencia. Disponible en web: <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemploxp/index.html>. Consulta: Febrero 2 de 2007.
- La World Wide Web, la Web o WWW. Disponible en web: http://es.wikipedia.org/wiki/World_Wide_Web. Consulta: marzo 6 de 2007.
- LAN Red de área local. Disponible en web http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_local. Consulta: marzo 6 de 2007.
- Aplicación Web. Disponible en web: http://es.wikipedia.org/wiki/Aplicacion_web. Consulta: marzo 6 de 2007.
- Internet Information Services (IIS). Disponible en web: <http://es.wikipedia.org/wiki/IIS>. Consulta: marzo 6 de 2007.
- Tecnología ASP.NET. Disponible en web: <http://es.wikipedia.org/wiki/ASP.NET>. Consulta marzo 6 de 2007.
- Bases de datos. Disponible en web: http://es.wikipedia.org/wiki/Base_de_datos. Consulta marzo 6 de 2007.
- El modelo relacional. Disponible en web: http://www.htmlpoint.com/sql/sql_03.htm Consulta: Junio 7 de 2007.