Universidad de las Ciencias Informáticas

Facultad 15

Título: Extensión de la aplicación informática

Coliseo Virtual.

Trabajo de Diploma para optar por el título de

Ingeniero Informático

Autor(es): Yusleiny Castro Suazo

José Antonio Ruz Polanco

Tutor(es): Lic. Rolan Rober Bullain Dieguez

Ing. Abdiel Matos Nieto

Ciudad de La Habana

Junio del 2010

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de este trabajo y autorizamos al Departamento de Especialidades (Facultad 15), de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los __ días del mes de ___ del año ____.

Yusleiny Castro Suazo

_____

José Antonio Ruz Polanco

_____

Lic. Rolan R. Bullain Dieguez

_____

Ing. Abdiel Matos Nieto

_____

AGRADECIMIENTOS

Agradecemos por la realización de este trabajo a nuestros tutores Rolan Bullain Dieguez y Abdiel Matos Nieto por confiar en nosotros y por prestarnos su ayuda en todo momento. A nuestros amigos por apoyarnos ante todos los problemas y en los momentos en los que parecíamos desfallecer. A todas las personas que de una forma u otra nos brindaron su consejo y soporte en las ocasiones en las que el camino a tomar no parecía claro. A todos los que nos aportaron experiencias y conocimientos que permitieron la culminación de este trabajo.

DEDICATORIA

A nuestros padres por su confianza y cariño incondicionales que nos permitió afrontar las mayores dificultades. A nuestros amigos de toda la carrera por su apoyo y sus consejos que fortalecieron nuestro espíritu.

RESUMEN

El presente trabajo brinda continuidad al trabajo de diploma ¨Coliseo Virtual¨ defendido por los ya ingenieros Abdiel Matos e Ismaray Núñez en el cual se realiza la primera fase de desarrollo de la aplicación informática del mismo nombre. Esta investigación tiene como objetivo extender las funcionalidades de la aplicación, de manera que posibilite establecer un vínculo entre estudiantes y profesores a través de la misma, con el cual los docentes puedan establecer mejores vías de intercambio con sus alumnos. Se pretende que mediante el empleo de esta solución de software se pueda realizar un trabajo diferenciado con los estudiantes, buscando que se potencie el desarrollo de habilidades en la asignatura de programación. Para su desarrollo se identificaron los requisitos del sistema, se realiza el diseño e implementación de la solución propuesta, permitiendo presentar una aplicación que puede ser utilizada como apoyo al proceso de aprendizaje.

ÍNDICE DE IMÁGENES

ÍNDICE DE TABLAS

TABLA DE CONTENIDOS

INTRODUCCIÓN

Las ciencias informáticas poseen numerosas aplicaciones en las diferentes esferas de la sociedad actual, transformándola en un entorno cada vez más dependiente de su uso y abriendo nuevas fronteras a un amplio campo de posibilidades para su desarrollo. Ante la creciente demanda de software a nivel mundial, se ha hecho indispensable la formación de especialistas en las diferentes áreas de la informática que sean capaces de desarrollar sistemas que respondan a las necesidades de los sectores empresariales y sociales y que soporten la infraestructura de un país.

En los últimos años, Cuba ha llevado a cabo un gran proceso de formación de profesionales en el área de la informática con el fin de introducirse en el mercado del desarrollo de software y contribuir a la informatización de la sociedad. Con este propósito se han creado un gran número de politécnicos para potenciar el estudio de esta ciencia y la Universidad de las Ciencias Informáticas (UCI) donde se forman ingenieros calificados para asumir el reto que la sociedad y el país imponen.

En la UCI se implementa un sistema de enseñanza que integra el estudio, la producción y la investigación, esto potencia que los estudiantes puedan ejercitar los conocimientos aplicándolos a situaciones reales, lo que permite que al graduarse posean competencias y habilidades profesionales sólidas para desempeñarse en un ambiente productivo. Con el afán de obtener cada vez mejores resultados, la universidad se auxilia de herramientas educativas que ayudan a desarrollar habilidades necesarias en un ingeniero informático como el conocimiento de las técnicas de programación y el trabajo con algoritmos entre otras disciplinas de gran importancia para la formación de un profesional capacitado. Entre las herramientas que responden a estas características en el centro, se encuentra la aplicación web de la *Cátedra de Programación Avanzada* (http://cpav.uci.cu), donde se publican ejercicios de programación para los estudiantes y se realizan diferentes competencias a nivel de universidad, el *Jurado Xtreme* perteneciente a la facultad 8 (http://onlinejudgef8.uci.cu) que también posee problemas algorítmicos disponibles para su resolución y ofrece opciones para la realización de competencias, y el *Coliseo Virtual*, perteneciente a la facultad 15, sobre la cual estará centrada esta investigación.

El *Coliseo Virtual* es una aplicación web de apoyo para el aprendizaje de las técnicas de programación. Esta herramienta posibilita que los estudiantes asimilen los conocimientos mediante la realización de diferentes estilos de competencias para la resolución de problemas asociados a la programación, entre los estilos existentes se encuentran: las pruebas en línea, los test, retos y competencias estilo libre y

personalizadas. Además, brinda posibilidades para consultar cursos y conformar equipos para la realización de los problemas.

A pesar de toda la gama de funcionalidades que posee la herramienta Coliseo Virtual, aún carece de otras que mejorarían su funcionamiento y ofrecerían mayores posibilidades a los usuarios que interactúan con ella. Actualmente no cuenta con un módulo para profesores y estudiantes que permita a estos últimos la realización de actividades asignadas, garantice que sus resultados puedan ser supervisados y que los profesores puedan realizar un trabajo diferenciado, aumentando el nivel de interacción entre estos roles, lo cual facilitaría la atención individual y la detección de los principales problemas en la asimilación de los conocimientos.

Teniendo en cuenta lo anteriormente expuesto se definió el siguiente **problema científico**:

- La ausencia de opciones para el seguimiento, control y trabajo diferenciado con los estudiantes en la aplicación informática Coliseo Virtual.

Para solucionar el problema científico se plantea como **objetivo de la investigación**: realizar un proceso de extensión de la aplicación informática Coliseo Virtual.

El problema tiene como **objeto de estudio**: el proceso de desarrollo de software, y como **campo de acción**: el software de gestión y los jueces en línea.

Se plantea como **hipótesis** de la investigación: que con el desarrollo de un módulo para profesores y estudiantes, la aplicación informática Coliseo Virtual potenciará el trabajo diferenciado educador-educando y ofrecerá mejores vías de intercambio entre estos roles.

De esta forma se plantean las siguientes **tareas de la investigación** para dar cumplimiento al objetivo:

- Estudio de diferentes jueces en línea con el fin de identificar características que puedan ser adaptadas a la herramienta Coliseo Virtual.
- Análisis de las diferentes herramientas a utilizar en el proceso de desarrollo.
- Realización del levantamiento de requisitos para identificar las funcionalidades que requiere el módulo.
- Inclusión de las nuevas funcionalidades dentro de: el modelo del sistema, modelo de diseño, modelo de implementación y la base de datos del sistema.

- Implementación de la solución propuesta.
- Realización de pruebas al sistema.

El documento se estructuró en tres capítulos para dar solución al problema planteado en la investigación:

- Capítulo No. 1: constituye la base teórica del presente trabajo. En él se realiza un estudio de las aplicaciones jueces en línea, además se describen y justifican las tecnologías, herramientas, metodologías y lenguajes a utilizar para el proceso de desarrollo.

- Capítulo No. 2: describe la propuesta de solución para la problemática presentada en la introducción del trabajo. Se lleva a cabo un análisis del sistema existente. Se definen los requisitos funcionales y no funcionales con los que debe cumplir la aplicación. Se identifican actores y describen los casos de uso del sistema. Se realiza el diseño del sistema y se presenta el diagrama de clases persistentes de la base de datos.

- Capítulo No. 3: realiza una descripción de la implementación del sistema y de las pruebas aplicadas al mismo. Se incluyen el modelo de implementación, los diagramas de componentes y de despliegue seguidos de las pruebas aplicadas y se ilustran los diferentes casos de pruebas seleccionados.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

El presente capítulo incluye un análisis de las diferentes aplicaciones y jueces en línea existentes dentro y fuera del país, que ofrecen elementos para el desarrollo de nuevas funcionalidades para el Coliseo Virtual. Es mostrado un estudio de las diferentes metodologías de desarrollo y los paradigmas de programación. Se lleva a cabo un análisis de las diferentes herramientas, tecnologías y plataformas de desarrollo que pueden servir para alcanzar los objetivos propuestos.

1.1 JUECES EN LÍNEA

Durante los últimos años se ha fomentado a nivel mundial el desarrollo de concursos de programación que utilizan como escenario de competencia a Internet. Son disímiles las aplicaciones que se pueden encontrar en la web que están implementadas sobre la base de un juez en línea, algunas están patrocinadas por diferentes universidades de mundo y otras por la Asociación para la Maquinaria de la Computación (ACM, por sus siglas en inglés). Aunque el principal objetivo de la gran mayoría de estas competencias está asociado a la búsqueda de talentos por parte de grandes compañías, mediante estas competiciones se propicia el desarrollo del pensamiento lógico, algorítmico, matemático y posibilitan estimular el aprendizaje de la programación.

A pesar de que muchas de estas herramientas no persiguen el mismo objetivo con el cual fue creado desde sus inicios el Coliseo Virtual, sí brindan una interesante fuente de ideas que se pueden adecuar a las características del sistema logrando así enriquecer la aplicación y de esta forma proporcionar mayores beneficios al trabajo docente. Es precisamente por estas características que serán analizados los siguientes jueces en línea:

- http://acm.timus.ru/: patrocinado por la Universidad Estatal de Ural ubicada en Rusia. Posee gran cantidad de ejercicios organizados en 8 volúmenes de entre 50 y 90 ejercicios cada uno. Al usuario registrado se le llevará una estadística de su desempeño en la competencia según los siguientes aspectos: cantidad de problemas, lenguaje de programación utilizado, tiempo de ejecución y memoria usada, entre otros.

- http://acm.tju.edu.cn/toj/: patrocinado por la Universidad de Tianjin y por la ACM. Permite organizar nuevas competencias además de las que ya propone. Se lleva una estadística de los jugadores según la competencia en la que estén registrados (cantidad de problemas resueltos, cantidad de problemas presentados, porcentaje de aciertos). Posee una gran colección de ejercicios (25 volúmenes de entre 80 y 100 ejercicios cada uno).

- http://imaginecup.com: concurso que se desarrolla a nivel mundial y está patrocinado por Microsoft. Dentro de los estilos de competencias que más se destacan en el mismo se encuentran: diseño de software, desarrollo empotrado, desarrollo web, algoritmos, cortos en arte digital y diseño de interfaz. Tiene como característica relevante que incluye varios estilos de competencia y en cada uno de ellos varía la cantidad de integrantes por equipos.

- Copa Pascal: evento competitivo organizado por el Departamento de Técnicas de Programación y patrocinado por la Universidad de Ciencias Informáticas. No posee una dirección url específica, sólo está disponible durante el período que dure la competencia. Cuando se ha empleado un juez en línea ha variado la cantidad de lenguajes de programación que acepta. Permite llevar un ranking con las soluciones de los problemas resueltos por los equipos.

- http://cpav.uci.cu: aplicación web de la cátedra de programación avanzada en la Universidad de Ciencias Informáticas. Se necesita previa autenticación (usuario UCI) para poder enviar los resultados de los ejercicios, también se deberá elegir el lenguaje de programación a utilizar (Pascal, Perl, Python, C#, Java, C++, C) y se subirá el código fuente del mismo. Los ejercicios están organizados en 14 volúmenes de entre 27 y 30 problemas cada uno. Se mantiene un ranking de los jugadores según los puntos acumulados y la cantidad de ejercicios aceptados.

- http://onlinejudgef8.uci.cu:5800/JudgeOnline: juez online organizado por la facultad 8 de la Universidad de Ciencias Informáticas. Su objetivo principal reside en servir de jurado a las competencias de programación desarrolladas en la UCI tales como: Copa Void de Programación, Copa Pascal, Copa GrundyPC.

- Coliseo Virtual: aplicación web creada para apoyar la enseñanza de la programación en la facultad 15 que permite el desarrollo de competencias online en diferentes estilos. Requiere que el usuario se encuentre autenticado. Posee opciones que permiten la conformación de equipos y la

realización de retos y dispone de documentación y materiales de consulta para complementar el proceso de aprendizaje. Esta herramienta aún no cuenta con una serie de características que serían importantes para mejorar su nivel de eficacia en el proceso de apoyo a la enseñanza, como opciones que permitan personalizar las formas de aprendizaje y dar seguimiento a las actividades que realizan los usuarios.

1.2 METODOLOGÍAS DE DESARROLLO

En los últimos tiempos se requiere cada vez más de la construcción de sistemas complejos y rápidos, que se ajusten a las crecientes necesidades de los usuarios. Para esto se requiere lograr una integración de las múltiples facetas del desarrollo de software. Por eso, se hizo necesario realizar un estudio de las metodologías de ingeniería del software que pudieran servir para guiar el proceso de automatización de las operaciones informáticas que se proponen en este trabajo.

Las metodologías de desarrollo de software son el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a producir un nuevo producto. La finalidad de una metodología de desarrollo es garantizar la eficacia (cumplir los requisitos iniciales) y la eficiencia (minimizar las pérdidas de tiempo) en el proceso de generación de software (Del Castillo Morán, 2008).

A lo largo de todo el ciclo evolutivo de la ingeniería de software, han sido creadas muchas metodologías que estandarizan el proceso de desarrollo de un software, y que serán útiles en mayor o menor grado según las características del sistema que se desee construir. Existen dos clasificaciones fundamentales en las cuales son agrupadas estas metodologías: metodologías ágiles y metodologías robustas.

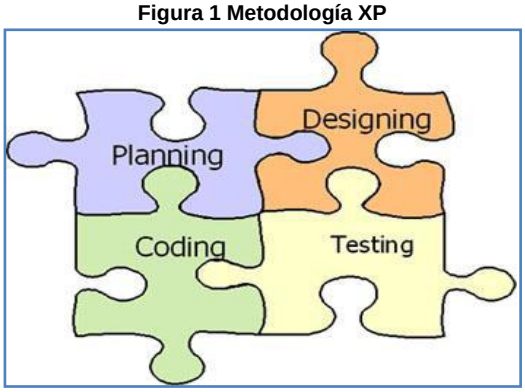
1.2.1 METODOLOGÍAS ÁGILES

Están especialmente orientadas para proyectos pequeños, constituyen una solución hecha a la medida para ese entorno, aportando una elevada simplificación en el proceso de desarrollo de software, que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto (Canós, y otros). La diferencia inmediata con las metodologías robustas es que son menos orientadas a la documentación, la cantidad exigida de la misma para una tarea dada es considerablemente menor que en las metodologías tradicionales. Están pensadas para trabajar con incertidumbre por lo que ofrecen una solución factible para proyectos que se encuentran en un ambiente inestable.

Dentro de ellas podemos encontrar el eXtreme Programming (XP), Scrum y Feature Driven Development, entre otras. En esta sección se hará referencia a XP por ser una de las que posee mayor éxito y aceptación en el mundo de los desarrolladores de software.

Extreme Programming

Figura 1 Metodología XP



Fuente: Tomado de (Sánchez, 2004)

Extreme Programming es un conjunto de técnicas y prácticas para el desarrollo de software. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Tiene dos principios básicos: la mejora de la comunicación con los usuarios, para retroalimentar el proceso de desarrollo; y obtener cuanto antes un programa que haga algo, partiendo de esto para ir añadiendo incrementalmente nuevas características. Algunas de las medidas que proponen no tienen sentido para proyectos pequeños (Wells, 2009).

Como características más relevantes de XP se encuentran (Kent, 1999):

- Es una metodología creada a base de prueba y error.
- Hace énfasis en el desarrollo del software más que una buena documentación.
- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- No introduce funcionalidades antes de que sean necesarias.
- El cliente o usuario se convierte en miembro del mismo equipo.

Esta metodología no es apropiada para el desarrollo de las funcionalidades en las cuales se basa este trabajo debido a que las mismas poseen requisitos estables y que no tendrán cambios significativos

durante el proceso de desarrollo; además se necesita generar una buena documentación para los futuros hitos de desarrollo.

1.2.2 METODOLOGÍAS ROBUSTAS

Las metodologías robustas son empleadas para dar soluciones empresariales muy complejas. Se basan en el uso de tecnología orientada a objetos. Guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado.

Dentro de las metodologías robustas se destacan el Proceso Unificado del Software (Rational Unified Process RUP), Microsoft Solutions Framework (MSF) y Métrica 3.0, entre otras. Siendo RUP la que, por ser una de las más representativas dentro de este grupo, será objeto de estudio del presente trabajo.

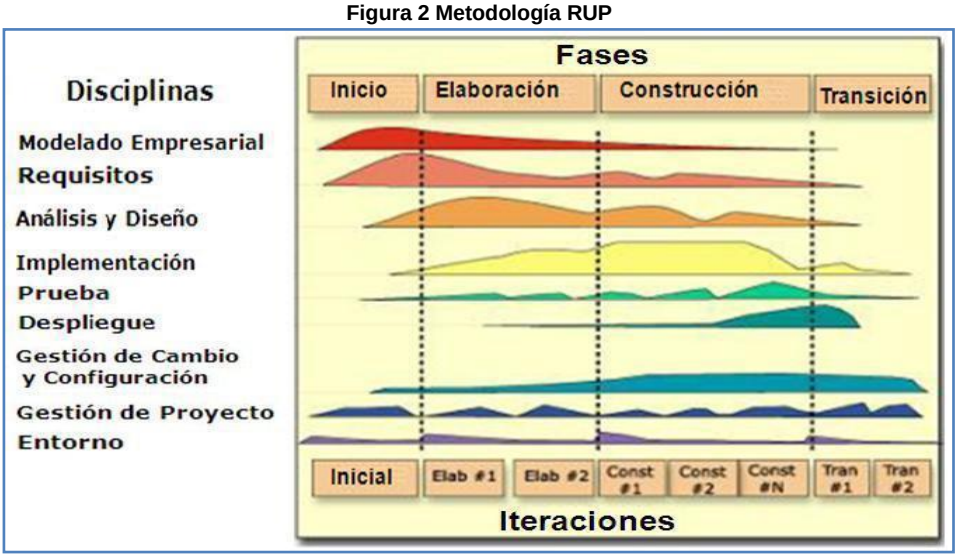
Rational Unified Process

El proceso Unificado del Software proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales dentro de un calendario previsible. RUP captura la mayoría de las mejores prácticas dentro del desarrollo del software moderno en una forma que es adecuado para una amplia gama de proyectos y organizaciones (Kruchten, 2004); estas son: desarrollo incremental, administración de requisitos, uso de arquitectura basada en componentes, modelamiento visual, verificación continua de la calidad, control de cambios.

Los objetivos principales que se plantean son (Fernández, 2000):

1. Proporcionar una guía del orden de las actividades de los equipos
2. Especificar cuáles artefactos deben ser desarrollados y cuándo.
3. Dirigir las tareas de desarrolladores individuales y en equipo como una sola.
4. Ofrecer criterios para monitorear y medir los productos y actividades del proyecto.

Figura 2 Metodología RUP



Fuente: Tomado de (Kruchten, 2004)

La figura 1-2 muestra la estructura general de RUP, el proceso tiene dos dimensiones: el eje horizontal representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hilos. El eje vertical representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de procesos, disciplinas, flujos de trabajo, actividades, artefactos y roles (Letelier, 2009).

El Proceso Unificado tiene tres características distintivas (Booch, 1999):

- **Dirigido por casos de Uso:** emplea casos de uso para impulsar el proceso de desarrollo desde su inicio hasta la implementación. Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.

- **Centrado en la Arquitectura:** el proceso trata de entender la mayoría de los importantes aspectos estáticos y dinámicos en términos de arquitectura de software. La arquitectura está en función de las necesidades de los usuarios y es capturada en los casos de uso básicos.

- **Iterativo e incremental:** el proceso reconoce que es práctico dividir los proyectos grandes en proyectos más pequeños o mini-proyectos. Cada mini-proyecto comprende una iteración que resulta en un incremento. Una iteración puede abarcar a todos los flujos de trabajo en el proceso.

Las características que posee RUP, a las cuales se hizo referencia en este epígrafe, demuestran que esta metodología robusta es la más eficiente para la planificación, desarrollo y validación del sistema. Además de que permite la respectiva generación de artefactos y documentación para cada etapa del proceso lo que conlleva a obtener un producto final enmarcado dentro de los estándares de calidad establecidos.

1.3 PARADIGMAS DE PROGRAMACIÓN

Un paradigma de programación es un modelo básico de diseño e implementación de programas. Un modelo que permite desarrollar programas conforme a ciertos principios o fundamentos específicos que se aceptan como válidos. En otras palabras, es una colección de modelos conceptuales que juntos modelan el proceso de diseño, orientan la forma de pensar y solucionar los problemas y, por lo tanto, determinan la estructura final de un programa (Facultad Regional de Buenos Aires, 2005).

Existen diferentes paradigmas de programación como la Programación Lógica, Modular, Funcional, Concurrente, Estructurada, Orientada a Objetos y Orientada a Aspectos. Teniendo en cuenta las características del sistema solo serán analizadas la Programación Orientada a Objetos y la Programación Orientada a Aspectos debido a que son las que más se ajustan a las necesidades existentes para el proceso de desarrollo.

1.3.1 PROGRAMACIÓN ORIENTADA A OBJETOS

La Programación Orientada a Objetos (OOP según sus siglas en inglés) es un paradigma de programación que define los programas en términos de “clases de objetos”, objetos que son entidades que combinan estado (propiedades o datos), comportamiento (procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto) (Weitzenfeld, 2004).

La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

Un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, dispone de mecanismos de interacción (los métodos) que favorecen la comunicación entre objetos de una misma clase o de distintas, y en consecuencia, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las

que no se separan, ni deben separarse, información (datos) y procesamiento (métodos) (Winblad, 1993). Las clases son declaraciones o abstracciones de objetos, lo que significa, que una clase es la definición de un objeto.

Por las características anteriormente expuestas y teniendo en cuenta las ventajas de seguridad y la fortaleza en los planteamientos lógicos en los que se basa y las facilidades que ofrece el trabajo con objetos es el paradigma seleccionado para la construcción del sistema.

1.3.2 PROGRAMACIÓN ORIENTADA A ASPECTOS

La Programación Orientada a Aspectos (AOP, por las siglas de Aspect-Oriented Programming) es un paradigma de programación relativamente joven cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de conceptos. Gracias a la AOP se pueden encapsular los diferentes conceptos que componen una aplicación en entidades bien definidas, eliminando las dependencias entre cada uno de los módulos. De esta forma se consigue razonar mejor sobre los conceptos, se elimina la dispersión del código y las implementaciones resultan más comprensibles, adaptables y reusables. Varias tecnologías con nombres diferentes se encaminan a la consecución de los mismos objetivos y así, el término AOP el cual es usado para referirse a varias tecnologías relacionadas como los métodos adaptivos, los filtros de composición, la programación orientada a sujetos o la separación multidimensional de competencias (Asteasuain, 2002).

La Programación Orientada a Aspectos complementa la Programación Orientada a Objetos permitiendo que el desarrollador modifique dinámicamente los modelos estáticos Orientados a Objetos para incluir el código requerido y crear un sistema que pueda crecer con el fin de alcanzar nuevos requerimientos. Así como los objetos en el mundo real pueden cambiar sus estados durante su ciclo de vida, una aplicación puede adoptar nuevas características a medida que se desarrolla.

A pesar de las ventajas que posee la utilización de este paradigma, no se utilizará en el proceso de desarrollo debido a que se considera que aún faltan muchos años para que se consolide. Además es posible agregar que todavía necesitan incrementarse el número de herramientas que soportan su uso, lo cual justifica que no sea seleccionado para el desarrollo del presente trabajo.

1.4 LENGUAJES

A continuación se procede a describir los lenguajes, tanto de programación como de modelado, que serán utilizados en el proceso de desarrollo de la aplicación.

1.4.1 LENGUAJE UNIFICADO DE MODELADO

El Lenguaje Unificado de Modelado (UML según sus siglas en inglés) es un lenguaje gráfico para visualizar, construir, especificar, y documentar las partes o artefactos de un sistema de software o aplicación. Este ofrece un estándar para describir un “plano” del sistema (Modelo) incluyendo aspectos conceptuales tales como procesos de negocio y funcionalidades del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de Base de Datos y componentes de software reutilizables. Pueden ser artefactos un modelo o una descripción que comprenda el desarrollo de software (García, 2005).

UML centra el desarrollo en tres modelos diferentes:

- **Modelo funcional:** diagramas de casos de uso, describen el sistema desde la perspectiva del usuario.
- **Modelo objeto:** diagrama de clases, describen la estructura de un sistema en términos de objetos, atributos, asociaciones y operaciones.
- **Modelo dinámico:** diagramas de secuencia y de estados, describen el comportamiento del sistema (UML, 2008).

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Desarrollo), pero no especifica en sí mismo qué metodología o proceso usar (Craig, 1999).

1.4.2 PHP

PHP es un lenguaje de programación de alto nivel embebido en páginas HTML y ejecutado en el servidor, que se utiliza para la programación de páginas web dinámicas. Habitualmente se combina con el motor de

bases de datos MySQL, aunque cuenta con soporte para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión. Permite la creación de aplicaciones gráficas independientes del navegador, por medio de la combinación con GTK (Belando, 2007).

Ventajas de PHP:

- Es un lenguaje de alto nivel que no requiere definición de tipos de variables.
- Es libre y de código abierto, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Es un lenguaje multiplataforma ya que tiene la capacidad de ser ejecutado en la mayoría de sistemas operativos.
- Puede interactuar con los servidores web más populares ya existe en versión CGI, módulo para Apache e ISAPI.
- Dispone de una gran capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, aunque destaca su conectividad con MySQL.
- Permite las técnicas de programación orientada a objetos.

Se seleccionó el lenguaje de programación PHP para implementar la solución propuesta por sus características de ser multiplataforma, integrarse mejor al servidor web y su capacidad para incluir scripts de manera sencilla dentro del protocolo HTML. Además también fue considerado el hecho de que la aplicación del Coliseo Virtual se encuentra implementada ya en este lenguaje por lo que se hace recomendable seguir utilizándolo.

1.4.3 JAVASCRIPT

El lenguaje de JavaScript permite extender los límites del lenguaje HTML, enlazando los scripts a la página Web, para que respondan los eventos provocados por el usuario ejecutando un código. Es un lenguaje compacto, está basado en los objetos y es empleado en el desarrollo Web con el fin de mejorar la presentación y la interactividad de las páginas HTML (Aumaille, 2000).

Sus características más importantes son:

- Es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente.
- Es un lenguaje orientado a eventos. Cuando un usuario presiona sobre un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante JavaScript se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos.

Los elementos mencionados que aporta Javascript a las aplicaciones Web hacen que este lenguaje sea necesario para el desarrollo. Además de que es el complemento ideal del HTML, al permitir a la página realizar algunas tareas por sí misma, sin necesidad de estar sobrecargando el servidor del cual depende y el lenguaje de scripting es seguro y fiable.

1.4.4 SQL

El Lenguaje de Consulta Estructurado (Structured Query Language), es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre la misma (Casares, 2008).

Debido a las ventajas que ofrece para el manejo y consulta de datos se selecciona este lenguaje para el trabajo con la base de datos de la aplicación.

1.5 HERRAMIENTAS CASE

Se pueden definir las Herramientas CASE (Computer Aided Software Engineering) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (Vivas White, y otros, 2009).

Actualmente existen Herramientas CASE tales como: Visual Paradigm para UML, Rational Rose, Enterprise Architect, Umbrello, ArgoUML, etc.

Para el desarrollo de este trabajo sólo serán analizadas Rational Rose y Visual Paradigm debido a que son las de mayor éxito y popularidad.

1.5.1 RATIONAL ROSE

El Rational Rose es una herramienta CASE privativa que proporciona un lenguaje de modelado común y el medio ambiente para acelerar la creación de software de calidad. Es una herramienta de diseño de software destinado a modelado visual y construcción de componentes de aplicaciones empresariales de software a nivel (IBM, 2006).

Dos características populares de Rational Rose es su capacidad para proporcionar el desarrollo iterativo y la ingeniería de viaje redondo. También mantiene la consistencia de los modelos del sistema software, realiza chequeo de la sintaxis UML, generación de documentación de manera automática, generación de código a partir de los modelos, ingeniería inversa (crear modelo a partir código).

Rational Rose ofrece representaciones del sistema desde distintos puntos de vista:

- **Casos de Uso**: Define la interacción entre actores y casos de uso. Los diagramas que se realizan son de casos de uso, de colaboración, de secuencia y de actividad.
- **Lógica:** Define fundamentalmente las clases del sistema y sus relaciones. Los diagramas principales son de clases (estático) y de estados (dinámicos).
- **Componentes:** Contiene información sobre ficheros, ejecutables y librerías del sistema. Acepta diagramas de componentes.
- **Despliegue**: Muestra la asignación de procesos al hardware. Sólo admite diagramas de despliegue.

Pese a su popularidad y amplia gama de funcionalidades posee la deficiencia de ser software propietario y sus versiones solo pueden ser utilizadas en Windows, por lo cual no se selecciona como herramienta CASE.

1.5.2 VISUAL PARADIGM

Visual Paradigm para UML es una herramienta UML fácil de usar que soporta captura de requisitos, diseño de base de datos, ingeniería inversa, generación de código, importación desde Rational Rose, exportación/importación XMI, generador de informes, editor de figuras, integración con MS Visio, plug-in, integración IDE como Visual Studio, IntelliJ IDEA, Java, C++, CORBA IDL, PHP, y **Python** entre otros. Se

encuentra disonible en multiples plataformas y entre sus nuevas características se incluyen el modelado colaborativo con CVS y Subversion e interoperabilidad con modelos UML2 a través de XMI (Visual Paradigm International, 2007).

Posee disponibles distintas versiones: Enterprise, Professional, Standard, Modeler y Personal y Comunitaria (que es gratuita). La compañía facilita licencias especiales para fines académicos.

Al realizar una comparación entre estas dos herramientas CASE se decidió utilizar el Visual Paradigm para modelar el sistema que se propone en el presente trabajo debido a las posibilidades que ofrece de ser un sistema multiplataforma, disponible para varios sistemas operativos como Linux y Windows y a sus facilidades de uso.

1.6 SISTEMAS GESTORES DE CONTENIDOS

Los sistemas gestores de contenido (CMS: Content Management System) se caracterizan por ofrecer soluciones para el diseño, la maquetación, la publicación, los flujos de trabajo y el control de derechos de autor de los contenidos que se generan (Lara, y otros, 2005). Permiten la automatización de la gestión, generación y recuperación de información digital, lo que brinda la posibilidad de entregar el contenido adecuado a las personas correspondientes en su oportuno momento. Ofrecen la posibilidad de editar texto y poseen la capacidad de procesar documentos en XML. El CMS crea una plantilla o conjunto de plantillas que gestionan estructura, formato, aspecto, patrones. Una característica fundamental de las herramientas CMS es la de facilitar el ciclo de trabajo (Ruíz, 2000).

La utilidad del empleo de este tipo de herramientas viene justificada por varios motivos, dentro de ellos podemos mencionar: inclusión de nuevas funcionalidades en la web, reutilización de objetos o componentes, páginas interactivas, cambios del aspecto de la web, consistencia de la web, control de acceso.

Algunas de las herramientas CMS más populares son: Drupal, EZ-Publish, Joomla, Mambo pero de ellas solo se realizará el análisis correspondiente a Drupal y Joomla por ser herramientas que ofrecen una gran cantidad de prestaciones, son de código abierto y multiplataforma, por lo tanto están en concordancia con las necesidades del sistema que se desea implementar.

1.6.1 DRUPAL

Drupal es una plataforma de administración de contenidos de código abierto bajo la licencia GPL. Equipado con una variada gama de funcionalidades, Drupal puede soportar distintos tipos de sitios web como blogs personales, proyectos corporativos y grandes sitios web impulsados por la comunidad. En esta aplicación los textos y enlaces a otros tipos de contenido se almacenan en una base de datos, recuperándose e integrándose de manera dinámica para ser presentada a un usuario en respuesta a una solicitud enviada a través de un navegador web (Drupal.org, 2009).

Ventajas de Drupal:

- Configuración de Idiomas.
- Fácil configuración de temas y una gran cantidad de plantillas (aunque todas muy parecidas).
- Gran cantidad de opciones de configuración con una interfaz amigable para el administrador.
- Fácilmente extensible a través de módulos que permiten “prototipar” o adaptar rápidamente la aplicación a necesidades específicas o cambiantes.

A pesar de las ventajas que ofrece el uso de este CMS, no es el gestor de contenidos sobre el que se encuentra implementada la aplicación Coliseo Virtual y realizar un proceso de migración resultaría demasiado engorroso, por lo cual no se selecciona para el proceso de desarrollo.

1.6.2 JOOMLA

Joomla es un sistema gestor de contenidos dinámico que brinda la posibilidad de crear aplicaciones web de alta interactividad, profesionalidad y eficiencia. Está implementado en PHP y SQL y utiliza base de datos relacionales específicamente MySQL. El CMS Joomla es de uso gratuito, de libre distribución, y de código abierto (OpenSource). El mismo se usa y distribuye bajo licencia pública general (GNU/GLP) y a pesar de que corre mejor en servidores Unix/Apache la administración y edición de contenidos desde la computadora que accede el administrador puede funcionar tanto con sistemas Unix o Windows (Trevejo Alonso, 2008).

Dentro de las características de Joomla se destacan:

- Versiones imprimibles de páginas en distintos formatos.
- Integración simple de animaciones flash, videos, imágenes y otros elementos gráficos.

- Permite creación e integración con blogs, foros, comunidades virtuales y otras aplicaciones web
- Tiene muchas aplicaciones y módulos que se pueden incorporar, calendarios de eventos, múltiples idiomas, comercio electrónico, boletines, gestores de galerías.
- Administración múltiple y escalonada mediante contraseñas.

El sistema de gestión de contenidos Joomla es poderoso y fácil de utilizar. Diseñado para gestionar numerosas configuraciones de sitios Web, Joomla es lo bastante simple para sitios pequeños y lo suficientemente robusto para aplicaciones profesionales a gran escala (Díaz, y otros, 2009). Este Gestor de Contenidos es el seleccionado para el desarrollo debido a las características anteriormente expuestas y a que la aplicación Coliseo Virtual se encuentra implementada con su uso por lo tanto se hace recomendable su utilización en el desarrollo de la aplicación.

1.7 SISTEMAS GESTORES DE BASES DE DATOS

Un Sistema Gestor de Bases de Datos (SGBD en lo adelante) o DBMA (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (Martínez, 2006).

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Para llevar a cabo el objetivo propuesto se utilizará el SGBD relacional MySQL, su elección se encuentra relacionada a la definición del CMS seleccionado expresada en el *epígrafe 1.6.2*.

1.7.1 MYSQL

MySQL es un Sistema Gestor de Bases de Datos Relacionales (SGBDR) que opera a través de un intérprete de comandos. Los comandos de MySQL siguen el estándar de SQL (Structured Query Language), que es un lenguaje normalizado para operar con Bases de Datos Relacionales. Es desarrollado sobre la filosofía de software libre. Fue implementado por la empresa MySQL AB, que

también le brinda soporte, pero puede utilizarse gratuitamente y su código fuente está disponible para los usuarios. La base de datos MySQL se ha convertido en la base de datos de código abierto más popular del mundo, debido a su gran rendimiento consistente, de alta fiabilidad y facilidad de uso (Oracle Corporation, 2010).

La arquitectura que posee MySQL lo convierte en un software extremadamente rápido, fácil de personalizar. Posee una extensiva reutilización de código dentro del software que le ha dado lugar a un sistema de administración de Bases de Datos con una significativa velocidad, compactación, estabilidad y facilidad de despliegue. Además de poseer una infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación. Sin contar su fácil instalación y configuración (Toledo, y otros, 2002).

Luego de realizar un análisis de las características del MySQL se decide utilizar este gestor pues es multiplataforma, se puede adquirir libremente y se complementa con el lenguaje seleccionado que ofrece numerosas funcionalidades para trabajar con este gestor de bases de datos.

1.8 BIBLIOTECAS DE CLASES JAVASCRIPT

Las bibliotecas de clases son un conjunto de rutinas de software pre-elaborados (definiciones de clase) que los programadores utilizan para escribir programas orientados a objetos. Estas definiciones de clase también incluyen las características de la herencia (PC Magazine, 2010).

Existen muchas bibliotecas para el trabajo con javascript, destacándose entre las más populares Dojo, jQuery, MooTools, Portotype y Ext JS. Para el proceso de desarrollo fue seleccionada la biblioteca Ext JS, debido a que posee una API (Interfaz de Programación de Aplicaciones en español) extensa, de gran facilidad de uso y con buena documentación y un gran número de componentes para la personalización de la interfaz gráfica de una aplicación web. A continuación es descrita con mayor detalle la biblioteca de clases seleccionada.

1.8.1 EXT.JS

Ext.JS es una biblioteca Javascript para la creación de aplicaciones enriquecidas del lado del cliente. Sus características principales son: gran desempeño, componentes de interfaz de usuario personalizables, con buen diseño y documentación (Ext JS Inc., 2010).

Esta biblioteca originalmente surgió como una extensión de la librería de Yahoo UI. En Ext.JS el manejo del DOM, AJAX y otras funciones básicas son hechas por la librería YUI y se enfoca en componentes para incluir dentro de una aplicación web, varios de estos componentes están capacitados para comunicarse con el servidor usando AJAX.

Otros elementos relevantes son:

- Compatibilidad con los principales navegadores web: Internet Explorer, Firefox, Safari, Chrome y Opera.
- API extensa, intuitiva y fácil de utilizar
- Licencia Libre y Comercial disponibles.

1.9 ENTORNOS DE DESARROLLO INTEGRADO PARA APLICACIONES WEB

Un entorno de desarrollo integrado o IDE (Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación.

Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, en ocasiones, un sistema de control de versiones.

Los IDEs proveen un marco de trabajo amigable para los lenguajes de programación como pudieran ser C++, Python, Java, C#, Delphi, Visual Basic, entre otros. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

1.9.1 NET BEANS

NetBeans es un entorno de desarrollo integrado de código abierto y de distribución gratuita que proporciona herramientas muy cómodas y de fácil uso para el desarrollo de aplicaciones sobre la plataforma JAVA (PC Magazine, 2010). Entre las características que proporciona tenemos:

- Desarrollo de aplicaciones multiplataforma sobre: MacOS, Windows, Linux.

- Add-ons para desarrollo Móvil, desarrollo Web gráfico, integración con SOA, optimización de aplicaciones y desarrollo con C y C++.
- Cliente CVS integrado.
- Instalación y actualización simple.
- Características visuales para el desarrollo web.
- Mejoras en el editor de código (Completa el código más inteligentemente).
- Permite el desarrollo colaborativo.

El entorno de desarrollo NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las API de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

A pesar de que NetBeans es uno de los IDEs más potentes que existen en la actualidad para el desarrollo web, no será utilizado para la construcción de la solución propuesta debido a que consume una gran cantidad de recursos.

1.9.2 ZEND STUDIO

Es un programa construido por Zend, que han sido los impulsores de la tecnología de servidor PHP, orientado a desarrollar aplicaciones Web. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, y Zend ha lanzado varias versiones del producto para Windows, Linux, y MacOs. Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. La parte el cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, pero para aprovechar toda la potencia de la herramienta de depuración, habrá que disponer de la parte del servidor que instala Apache y el módulo PHP. Zen Studio posee el Zend Framework, un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios Web modernos, robustos y seguros (Álvarez, 2003).

Zend Studio no se distribuye de forma gratuita ni pertenece a la comunidad de código abierto y se caracteriza por:

- Integrar Java fácilmente en su código utilizando las características del completamiento de código.
- Integración del uso y completamiento de código personalizado de Zend Framework y vista de la lista de las funciones del framework desde la Visualización de Funciones PHP.
- Visualizar los eventos de Zend Platform en una ventana de lista de eventos personalizada y dedicada.
- Documentar el código de forma más sencilla, aplicaciones, y proyectos con PHP Documentor, la herramienta de documentación estándar para PHP.
- Simplificar el despliegue con la integración FTP y SFTP de forma tal que permita a los programadores en forma segura subir y descargar archivos de proyectos de modo transparente hacia y desde servidores remotos.

Debido a que esta herramienta no se distribuye de forma gratuita ni es open source no es seleccionada para el proceso de desarrollo.

1.9.3 APTANA STUDIO

El entorno de desarrollo Aptana Studio incluye soporte para PHP. Esta aplicación gratuita, se deriva de Eclipse, y permite ahorrar mucho trabajo a la hora de desarrollar o modificar aplicaciones y sitios web realizados con PHP gracias al completamiento/sugerencia automática de código así como a la facilidad para previsualizar y ejecutar los sitios directamente desde el mismo entorno de desarrollo. Permite manejar bases de datos directamente e incluye soporte para otras tecnologías como Ajax, Jquery, Ruby on Rails, Python o productos como el novedoso Adobe Air. Al estar desarrollado en Java, funcionará en virtualmente cualquier sistema operativo, incluyendo los más utilizados como Windows, Mac OS y Linux. Aptana está basado en el conocido entorno de desarrollo Eclipse, también de código abierto. Pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo web, con soporte a HTML, CSS y Javascript.

Las características principales de Aptana Studio son:

- Ayudas visuales para la escritura de scripts en diversos lenguajes, como coloreado y auto escritura del código, ayudas contextuales de referencia a medida que se escribe.

- Visualización de errores de sintaxis a medida que se escribe.
- Soporte para hacer FTP a servidores remotos, con herramientas para sincronización.
- Debug en Firefox (Debug Internet Explorer también con la versión Profesional)
- Librerías de funciones en Javascript populares en Ajax/Javascript para utilizar en los proyectos.
- Ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
- Previsualización de estilos CSS con el editor CSS.
- Extensible a partir de plug-ins que puede crear Aptana u otras empresas y herramientas para estar al tanto de cualquier nuevo añadido.
- Extensible por Javascript. Los usuarios pueden escribir scripts para realizar acciones y macros.

Además, este IDE provee chequeo de sintaxis, organización estructural del código fuente, depuración, autocompletamiento integrado con el DOM y diferentes frameworks. Permite optimizar el código para los más renombrados navegadores Web del momento (Internet Explorer, Firefox, Opera, Nestcape) gracias a la gran cantidad de información contenida para este fin. Soporta las bibliotecas de clases más populares: Prototype, Dojo, MochiKit, Yahoo UI, Aflax, Ext JS, HTML, CSS y JavaScript.

Las múltiples funcionalidades que caracterizan a este IDE que hacen que su uso sea muy sencillo e intuitivo, lo cual ligado a que es multiplataforma, es un software libre, posee un bajo consumo de recursos de la computadora, ofrece muchas facilidades para el trabajo con Javascript, Ajax y que además soporta la biblioteca javascript Ext JS, justifican que sea el IDE seleccionado para el desarrollo del presente trabajo.

CONCLUSIONES

- El estudio de los jueces en línea de mayor relevancia en la actualidad permitió identificar en ellos la carencia de opciones similares a las que se desean implementar en el Coliseo Virtual.

- El análisis de las metodologías de desarrollo de software existentes posibilitó la selección de RUP como la más adecuada para ser utilizada durante el proceso de desarrollo.

- La investigación realizada sobre las diferentes herramientas y lenguajes disponibles mediante procesos de comparación y valoración hizo posible la selección de las que más se ajustaban a las necesidades del grupo de desarrollo, entre las que más se destacan el CMS Joomla, el lenguaje PHP, la herramienta para el modelamiento en UML Visual Paradigm y el IDE Aptana Studio.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

INTRODUCCIÓN

En el presente capítulo se realiza una descripción de la propuesta de solución para la problemática presentada en la introducción del trabajo. Se inicia realizando un análisis del sistema existente. Se definen los requisitos funcionales y no funcionales con los que debe cumplir la aplicación. Se identifican actores, se describen los casos de uso del sistema y se lleva a cabo el diseño del sistema. Por último se presenta el diagrama de clases persistentes de la base de datos.

2.1 ANÁLISIS CRÍTICO DEL SISTEMA ANTERIOR.

El sistema Coliseo Virtual es una aplicación web construida con el objetivo de implementar un juez en línea para desarrollar competencias de programación mediante el uso de los juegos didácticos.

Para la modelación del sistema propuesto no se lograron definir procesos específicos en el entorno, solamente identificar conceptos y objetos relacionados con el mismo, por ello se realizó un modelo de dominio que fue especificado en iteraciones anteriores.

Actualmente la aplicación no cuenta con una serie de características que permitan darle un mayor grado de cumplimiento a sus objetivos iniciales como la carencia de un módulo que les permita a los profesores mantener un control, seguimiento y trabajo diferenciado sobre sus estudiantes mediante la definición de rutas de aprendizaje y exámenes personalizados; posibilitando la aplicación de un sistema de evaluación, garantizando la realización de las tareas asignadas a los estudiantes y que permita la generación de reportes para comprobar los resultados obtenidos.

2.2 PROPUESTA DEL SISTEMA

El sistema que se propone pretende dar cumplimiento a la situación que se plantea implementando un grupo de funcionalidades como las que se exponen a continuación:

La aplicación debe ser capaz de permitir a los profesores gestionar el trabajo con los estudiantes y de ver su desempeño en las tareas propuestas, debe ofrecer la posibilidad de crear exámenes y otros métodos de evaluación por parte de los docentes. También debe permitir que los estudiantes sean capaces de desarrollar los diferentes tipos de evaluaciones asignadas dentro de un plazo definido de tiempo.

2.3 REQUISITOS DE LA SOLUCIÓN PROPUESTA

A continuación se procederá a especificar los requisitos funcionales y no funcionales donde se explican las características que se desean incluir al sistema propuesto.

2.3.1 REQUISITOS FUNCIONALES

RF1: Gestionar Grupo:

El usuario registrado como profesor podrá gestionar (Actualizar, Adicionar, Eliminar) los grupos docentes de estudiantes brindando la información necesaria en cada caso, lo cual será descrito a continuación:

> **RF1.1**: Actualizar datos del grupo: el profesor podrá actualizar los datos del grupo que fue previamente registrado.
>
> **RF1.2**: Adicionar nuevo grupo: el profesor podrá adicionar un nuevo grupo proporcionando el nombre del grupo y seleccionando los estudiantes que formarán parte del mismo.
>
> **RF1.3**: Eliminar grupo: el profesor podrá eliminar un grupo registrado para lo cual deberá seleccionar uno de los grupos previamente listado.

RF2: Gestionar rutas de aprendizaje.

El profesor podrá gestionar (Actualizar, Eliminar, Insertar) el conjunto de rutas de aprendizaje, proporcionando la información pertinente en cada caso y que se describe a continuación:

> **RF2.1.** Actualizar rutas de aprendizaje: el profesor podrá actualizar las rutas de aprendizaje, pudiendo modificar el límite de tiempo de la misma, cambiando sus ejercicios y nombre.
>
> **RF2.2.** Eliminar rutas de aprendizaje: el profesor podrá eliminar la ruta completa de aprendizaje
>
> **RF2.3.** Insertar ruta de aprendizaje: el profesor podrá insertar una nueva ruta para lo cual debe seleccionar los problemas, test o exámenes que formarán parte de la ruta, una vez que estos sean listados. Además deberán especificar el nombre de la ruta, la fecha de inicio y la fecha de fin de ésta.

RF3. Asignar ruta de aprendizaje

El profesor podrá asignar una ruta de aprendizaje a los grupos o estudiantes que determine. Para ello debe de estar autenticado como usuario Profesor y debe seleccionar la opción ¨Asignar ruta¨ y seleccionar a quien desea asignarle la ruta.

RF4.Realizar ruta de aprendizaje.

Un estudiante podrá realizar una ruta de aprendizaje, para ello debe estar autenticado y acceder al menú "Tareas" y seleccionar la opción "Rutas de aprendizaje" y elegir la ruta que desea realizar entre las asignadas.

RF5. Verificar rutas de aprendizaje.

El profesor, una vez autenticado en el sistema, tendrá la posibilidad de verificar la realización de una ruta de aprendizaje para lo cual primero debe acceder al menú de "Profesor" donde al seleccionar la opción "Verificar ruta", debe seleccionar una ruta y el sistema le mostrará el nombre de los estudiantes que la han realizado y la puntuación obtenida por los mismos en la realización de la actividad.

RF6. Gestionar examen personalizado.

El profesor podrá gestionar (Actualizar, Eliminar, Insertar) exámenes para lo cual deberá ingresar los datos correspondientes en cada caso:

> **RF6.1.** Actualizar examen: el profesor podrá actualizar un examen teniendo la posibilidad de modificar los siguientes datos: nombre del examen, contraseña de acceso, tiempo de duración del examen, fecha de inicio, fecha de fin, así como variar los ejercicios que forman parte del mismo.
>
> **RF6.2.** Eliminar examen: el profesor podrá eliminar un examen para lo cual deberá seleccionar un examen entre los existentes, los cuales serán previamente mostrados en una lista.
>
> **RF6.3.** Insertar examen: el profesor podrá insertar un nuevo examen para lo cual primero debe seleccionar: el nombre del examen, la contraseña, tiempo de inicio, tiempo de finalización, fecha de inicio y fecha final. Además debe seleccionar los ejercicios que formarán parte del examen, los cuales se mostraran según el criterio de selección (año, materia, tema) gestionado por el mismo profesor.

RF7. Asignar examen.

El profesor podrá asignar un examen a los estudiantes o grupos que él determine, para lo cual deberá estar previamente autenticado y deberá seleccionar la opción ¨ Asignar examen ¨ y escoger quién desea asignarle el examen.

RF8. Realizar examen personalizado.

El estudiante podrá realizar un examen mediante la selección de uno de los exámenes disponibles que aparecerán al marcar la opción “Exámenes” disponible en el menú “Tareas”.

RF9.Consultar reporte de examen.

El profesor podrá obtener un reporte de los exámenes realizados mediante el acceso al menú “Profesor” y la selección de la opción “Reporte de examen” donde el profesor deberá seleccionar un examen de los que se encuentran disponibles y el sistema le mostrará a todos los estudiantes que participaron en la misma con la cantidad de ejercicios realizados y la puntuación obtenida.

RF.10 Ver ranking de las rutas.

El usuario previamente autenticado podrá ver el ranking de los estudiantes que han realizado rutas de aprendizaje.

2.3.2 REQUISITOS NO FUNCIONALES

RNF1. Apariencia o interfaz externa.

La interfaz deberá ser de fácil uso y amigable para sus usuarios, de manera que no ocasione ninguna molestia o incomodidad. Estará estructurada de manera tal que no será ningún impedimento para cualquier tipo de usuario navegar fácilmente por la aplicación.

RNF2. Rendimiento.

El módulo estará diseñado sobre la base de una arquitectura cliente/servidor. Este requerirá de una alta capacidad de procesamiento con el fin de poder ejecutar algoritmos complejos cuando se requiera enviar la solución de algún ejercicio. Además los tiempos de respuesta han de ser cortos, aproximadamente hasta 4 segundos, tanto para el envío de respuesta como para la calificación de problemas.

RNF3. Seguridad

Se requiere previa autenticación por parte del usuario antes de hacer uso de las principales funcionalidades del módulo, deberán prevenirse los ataques por inyección de código SQL.

RNF4. Legales

El módulo estará desarrollado con el uso de tecnologías y herramientas de software libre.

RNF5. Software.

Cliente: Sistema operativo Windows o Linux

Servidor: Sistema Operativo Windows o Linux, gestor de base de datos MySQL 5.1 y servidor para aplicaciones web Apache 2.2.

RNF6.Hardware

Computadora cliente: 512Mb de memoria RAM.

Computadora servidor: entre 512M RAM y 1Gb.

2.4 MODELO DEL SISTEMA

2.4.1 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA

A continuación se procede a la descripción de los actores que forman parte del sistema que son las personas encargadas de darle cumplimiento a los requisitos definidos anteriormente.

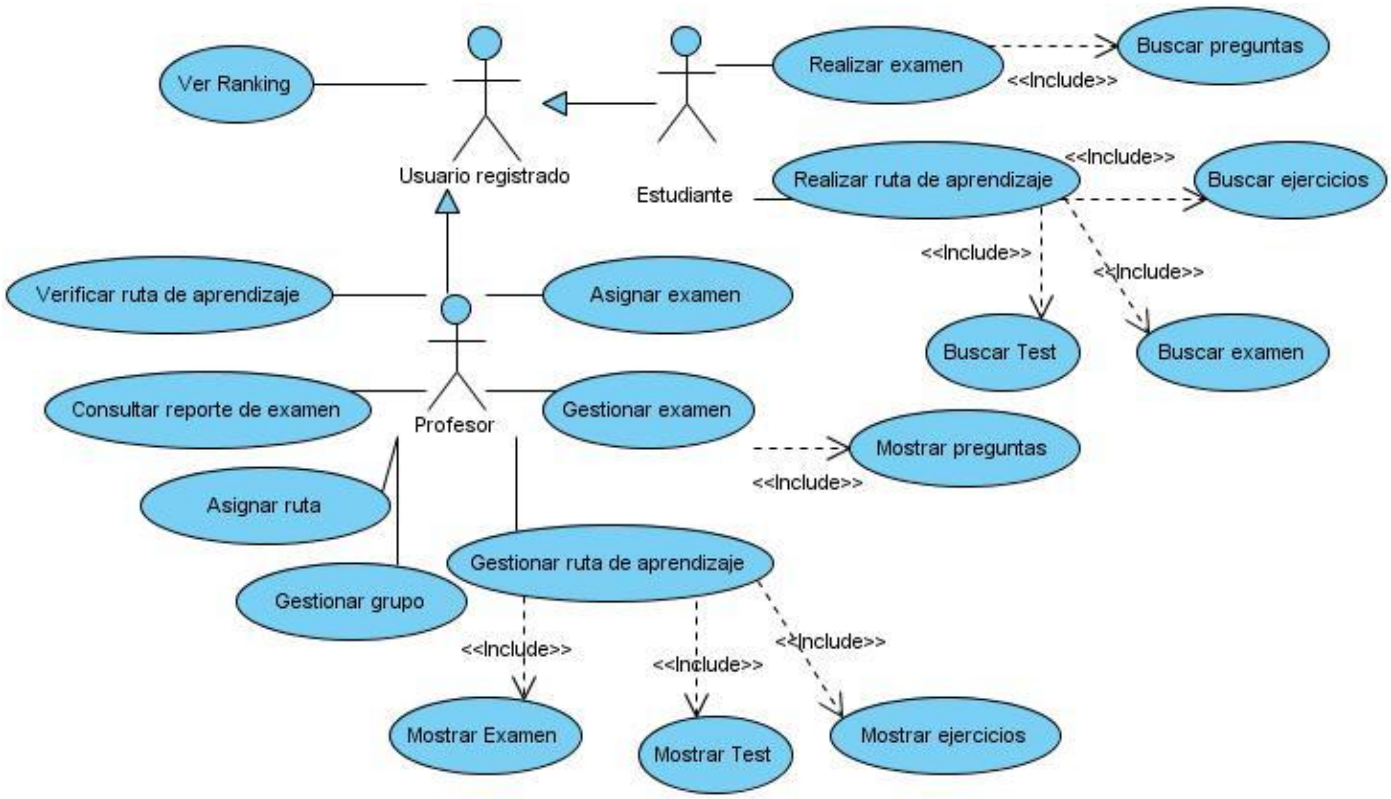
Tabla 1 Actores del Sistema

Actores del sistema	Descripción
Profesor	Es el encargado de gestionar todas las funcionalidades relacionadas con el control a los estudiantes.
Estudiante	El estudiante puede realizar los exámenes y las rutas de aprendizaje que son designadas por sus profesores.
Usuario registrado	El usuario registrado podrá ver el ranking de las rutas de aprendizaje

2.4.2 DIAGRAMA DE CASOS DE USO DEL SISTEMA

A continuación es presentado el diagrama de casos de uso del sistema.

Figura 3 Diagrama de casos de uso del sistema



2.4.3 DESCRIPCIÓN DE UN CASO DE USO ARQUITECTÓNICAMENTE SIGNIFICATIVO

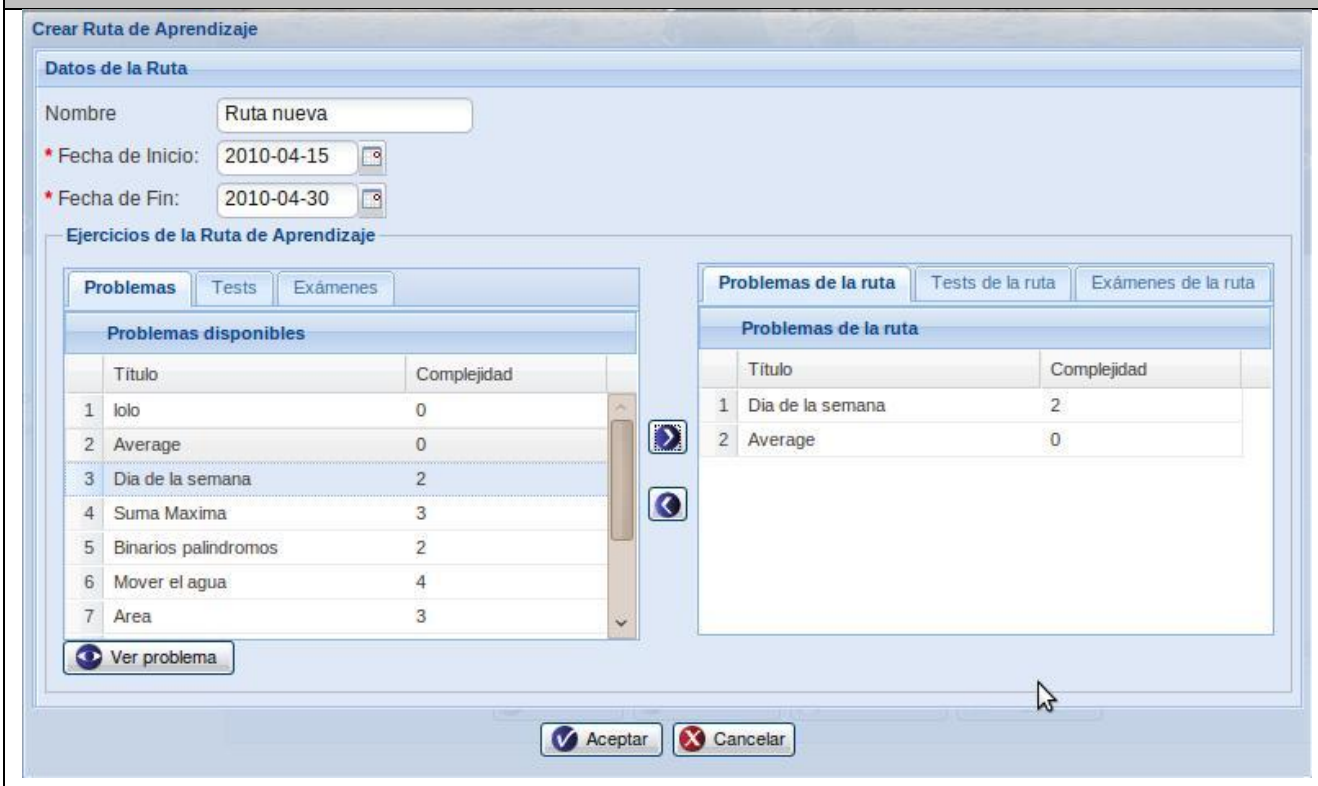
A continuación se muestra la descripción textual del Caso de Uso Gestionar Ruta de Aprendizaje que es inicializado por el actor Profesor, el resto de los casos de uso del sistema están disponibles en el documento Modelo de Casos de Uso del Sistema.

Tabla 2 Descripción textual del CU Gestionar Rutas de aprendizaje

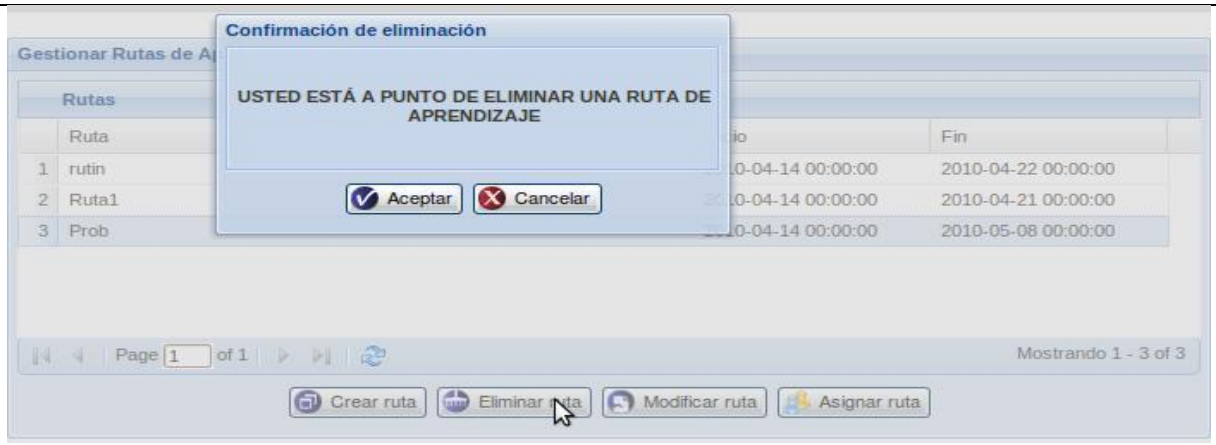
Caso de Uso	
Nombre del caso de uso:	Gestionar rutas de aprendizaje
Actores:	Profesor
Propósito:	Gestionar (Actualizar, Eliminar, Insertar) las rutas de aprendizaje

	para ser utilizadas por los estudiantes.
Resumen: el caso de uso inicia cuando el profesor selecciona del menú “Profesor” la opción “Gestionar rutas de aprendizaje” algunas de las funcionalidades presentadas son (Actualizar rutas de aprendizaje, Eliminar rutas de aprendizaje, Insertar rutas de aprendizaje).	
Referencias:	RF5
Precondiciones	El profesor debe de estar registrado en el sistema
Poscondiciones	El profesor actualiza, elimina o inserta una ruta de aprendizaje.
Flujo Normal de Eventos	
Acciones del actor	**Respuesta del sistema**
1. El CU inicia cuando el profesor selecciona la siguiente opción del menú “Profesor”; “Rutas de aprendizaje” de la cual puede: o Insertar ruta de aprendizaje (Ver sección **“Adicionar ruta de aprendizaje”**) o Eliminar ruta de aprendizaje (Ver sección **“Eliminar ruta de aprendizaje”**) o Actualizar ruta de aprendizaje (Ver sección **“Actualizar ruta de aprendizaje”**)	
Sección “Adicionar ruta de aprendizaje”	
	2. El sistema muestra la interfaz correspondiente a ”Adicionar rutas de aprendizaje” que contiene los siguientes campos: o Nombre de la ruta o Fecha de inicio o Fecha final Además muestra los ejercicios disponibles de : o Problemas o Test o Examen
3. El actor llena los campos solicitados.	
4. El actor selecciona los ejercicios que desea.	5. El sistema va agregando los ejercicios a la ruta de aprendizaje.
6. El actor presiona el botón “Adicionar ruta”	7. El sistema verifica que no exista una ruta con el mismo nombre. (Ver **flujo alterno 1: ‘Ruta existente”**)
	8. El sistema adiciona la ruta y cierra la interfaz. Termina el

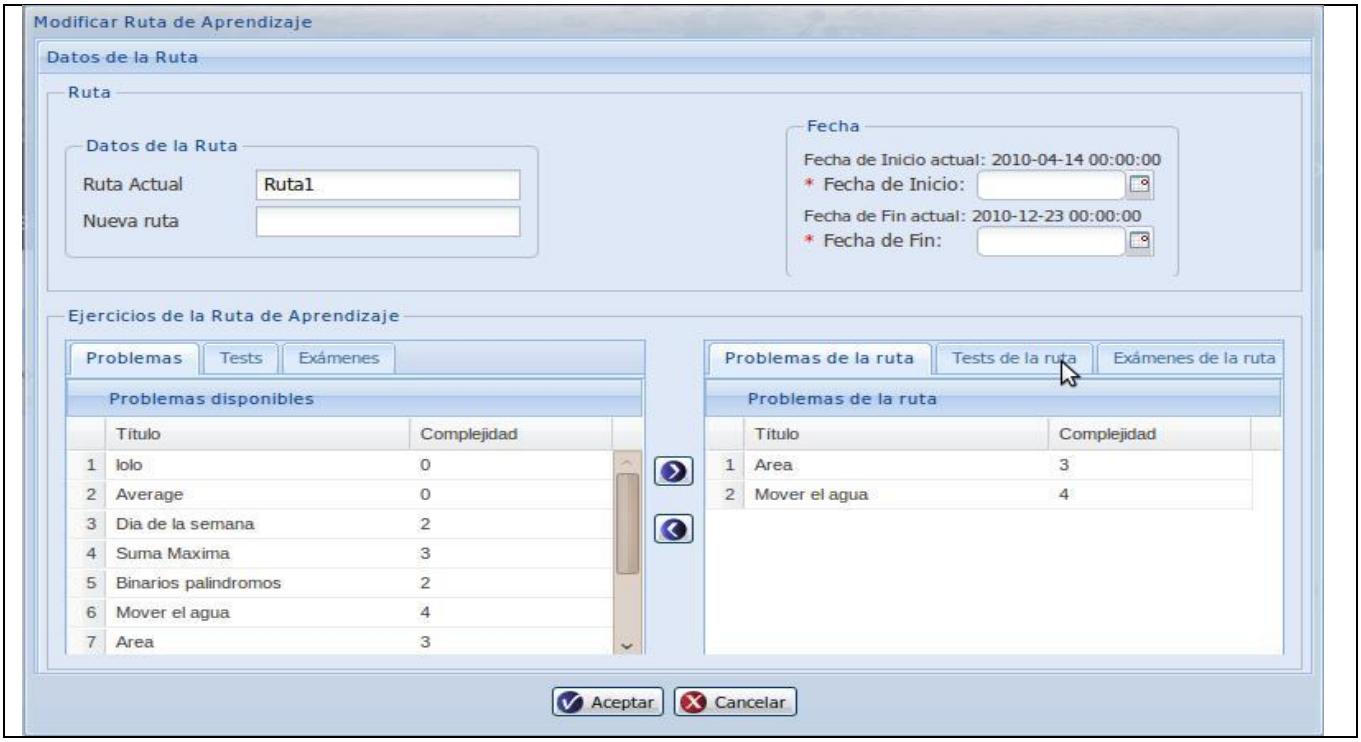
	CU.
Flujo Alterno 1: "Ruta existente"	
Acciones del actor	**Respuesta del sistema**
	1. El sistema muestra un mensaje notificando que ya existe una ruta con ese nombre.
2. El profesor cierra el mensaje	3. Continuar con el **flujo normal de eventos paso 2**
Prototipo de interfaz	



Sección "Eliminar ruta de aprendizaje"	
Acciones del actor	**Respuesta del sistema**
2. El actor selecciona la ruta que desea eliminar en la interfaz principal.	
3. El actor presiona el botón "Eliminar"	4. El sistema elimina la ruta de aprendizaje y muestra un mensaje de notificación. Termina el CU
Prototipo de Interfaz de Usuario	



Sección: "Actualizar ruta de aprendizaje"	
Acciones del actor	**Respuesta del sistema**
	2. El sistema muestra la interfaz correspondiente a "Actualizar rutas de aprendizaje" donde se visualizan todas las rutas de aprendizaje existentes.
3. El actor selecciona la ruta que desea actualizar y presiona el botón "Modificar".	4. El sistema muestra los campos modificables que son los siguientes: o Nombre de la ruta o Fecha de Inicio o Fecha de Fin o Los ejercicios de la ruta y carga los ejercicios disponibles.
5. Si el actor modifica los datos deseados y presiona el botón" Aceptar".	6. El sistema modifica la ruta de aprendizaje y cierra la interfaz. Termina el CU.
Prioridad	Crítico
Prototipo de Interfaz de Usuario	



1.6 ARQUITECTURA DEL SISTEMA

La arquitectura del sistema fue definida en iteraciones anteriores de desarrollo en la cual se especifica que se hace uso de estilos arquitectónicos de ¨Llamada y Retorno¨ empleando para su descripción y representación el modelo de 4+1 vistas propuesto por RUP, de las cuales se seleccionaron para describir la arquitectura: la vista lógica, despliegue e implementación. Para el desarrollo del módulo que se propone en el presente trabajo, el uso del patrón de diseño Modelo-Vista-Controlador (MVC) es uno de los elementos definidos dentro de la arquitectura que adquiere gran importancia debido a que el CMS Joomla se encuentra implementado con su utilización y por las facilidades que ofrece para la detección de errores y reutilización de código. Por estas razones es descrito a continuación.

1.6.1 PATRÓN DE DISEÑO: MODELO VISTA CONTROLADOR

Modelo-Vista-Controlador (MVC) es un patrón de desarrollo que separa la parte lógica de una aplicación de su presentación. Básicamente sirve para separar el lenguaje de programación del HTML lo máximo posible y para poder reutilizar componentes fácilmente. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas (Capuz, 2008).

El Modelo es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos (Catalani, 2007).

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al mismo (Catalani, 2007).

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo (Catalani, 2007).

Específicamente el CMS Joomla se implementa utilizando este patrón a través de 3 clases principales: JModel, JView, JController.

JController es una clase abstracta que proporciona la funcionalidad básica para las clases de su propio controlador que crear para sus componentes. Tiene un método para cada tarea con la que el controlador tenga que tratar: mostrar un elemento, crear un elemento, borrar un elemento. En estos métodos el controlador analiza la entrada del usuario y lleva a cabo los pasos necesarios para generar la respuesta (Dique, 2010).

JModel es una clase abstracta que proporciona la funcionalidad básica de los objetos modelo concreto en relación con el patrón de Joomla MVC. Los modelos que amplían la clase JModel se definen a menudo en los componentes de Joomla y definir la lógica de negocio del componente específico. Poseen distintos métodos para realizar distintas acciones sobre los datos (Dique, 2010).

JView es una clase abstracta que proporciona la funcionalidad básica para el patrón de diseño MVC de Joomla. Se necesita para escribir sus propias clases de vista concreta, a hacer pleno uso de la funcionalidad (Dique, 2010).

1.7 MODELO DE DISEÑO

A continuación se procederá a la descripción del diseño del sistema. En el libro El proceso unificado de desarrollo de software se plantea por sus autores que se puede realizar el análisis de los requisitos como parte integrada en el diseño siempre y cuando:

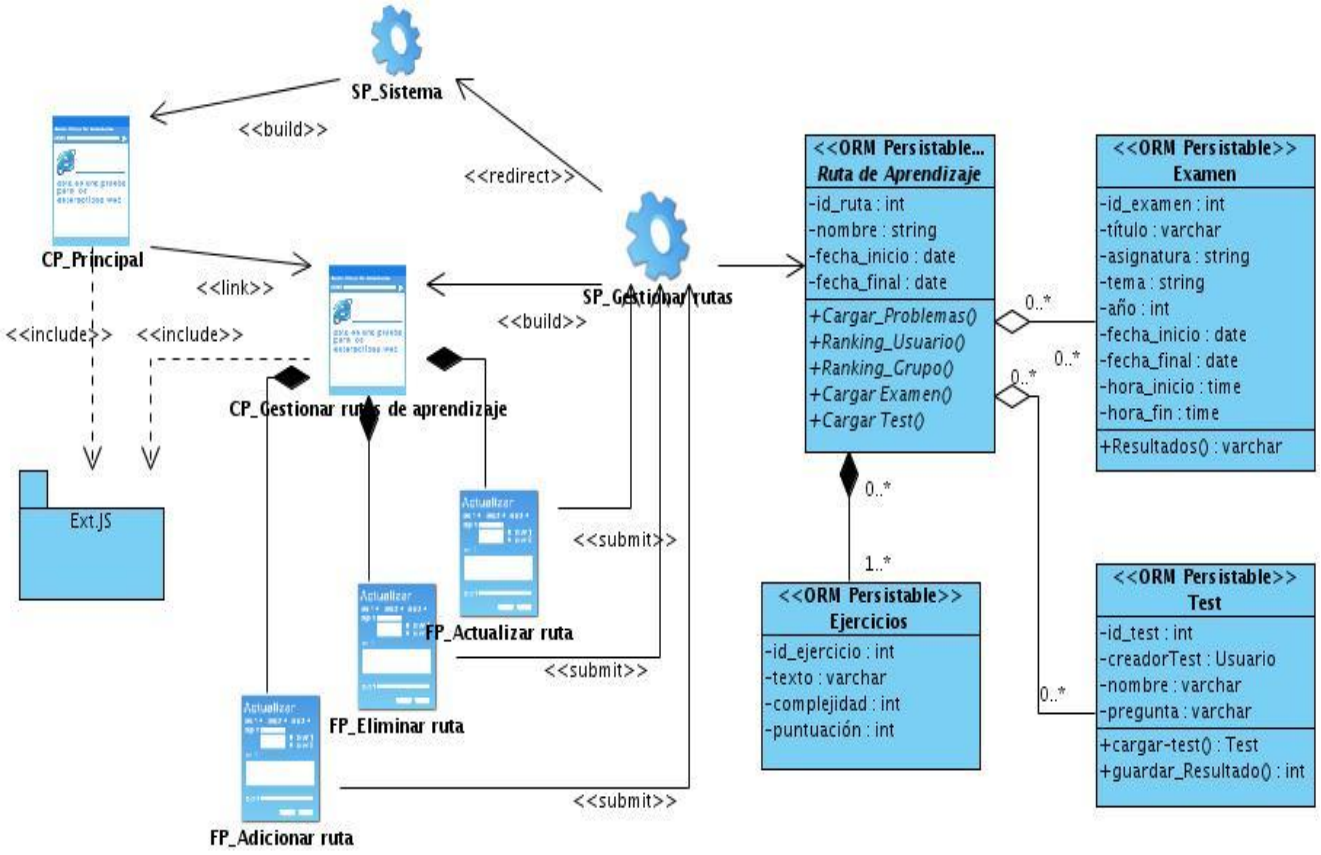
1. Los requisitos son muy simples y/o bien conocidos.

2. Si es fácil identificar la forma del sistema (incluyendo su arquitectura)

3. Si los desarrolladores cuentan con una cierta comprensión, intuitiva pero correcta, de los requisitos, y son capaces de construir un sistema que los encarne de una manera bastante fácil (Jacobson, y otros, 2000). Debido a las características del sistema y de sus desarrolladores, se decide no realizar el flujo de análisis y realizar directamente el diseño de la aplicación.

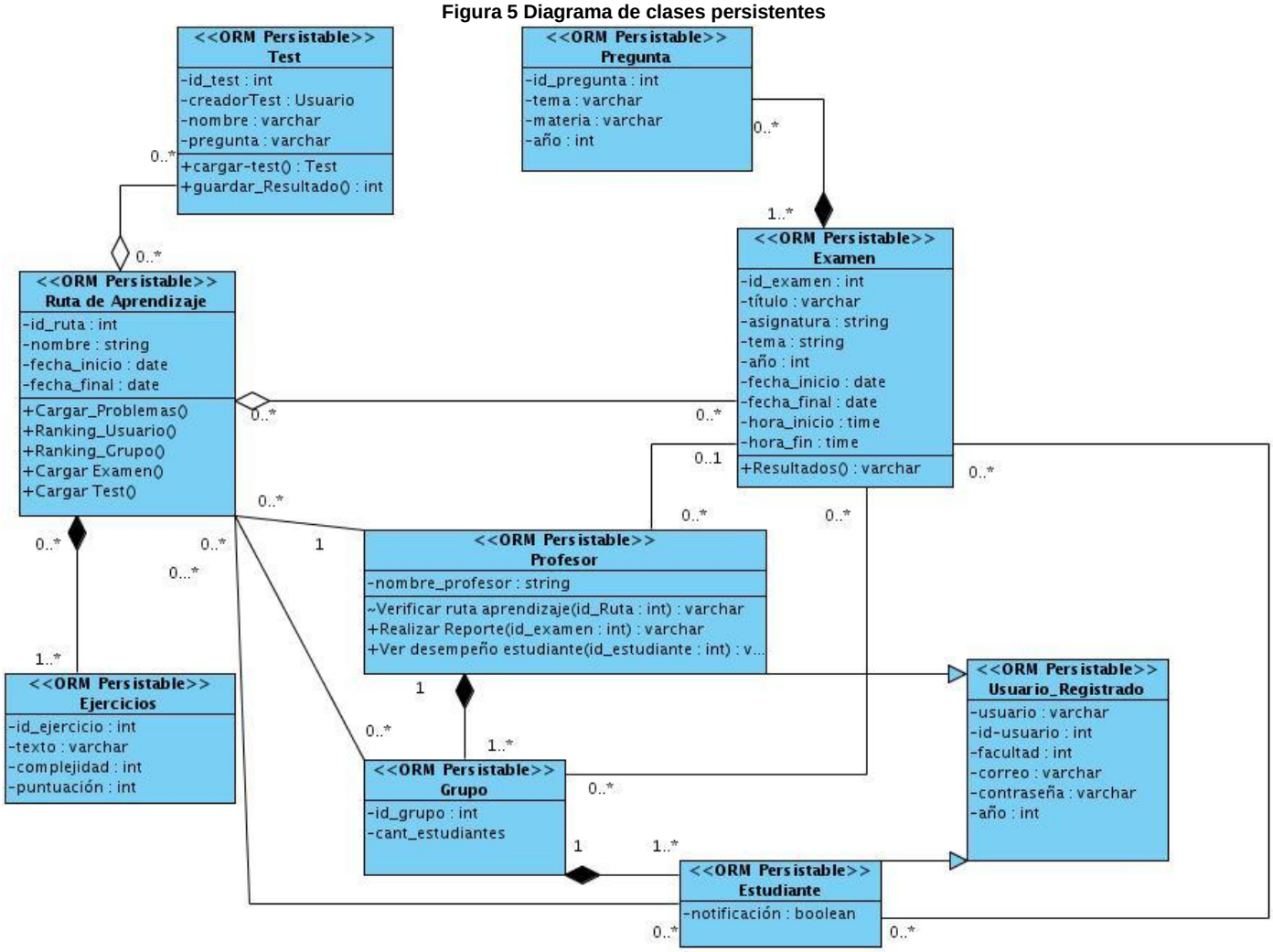
1.7.1 DIAGRAMA DE CLASES DEL DISEÑO

Figura 4 Diagrama de clases del diseño del CU Gestionar Rutas de Aprendizaje



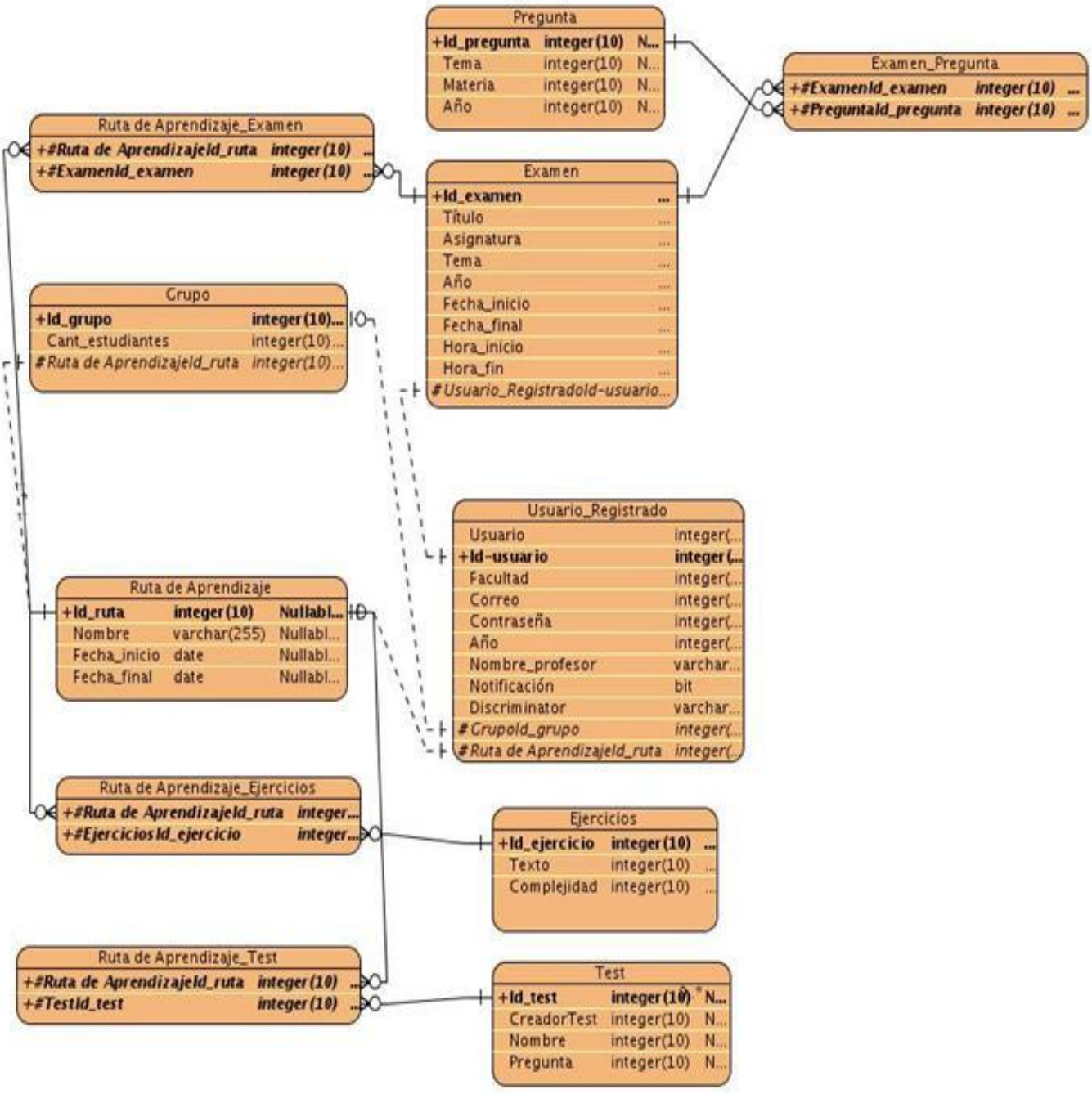
1.7.2 DIAGRAMA DE CLASES PERSISTENTES

Figura 5 Diagrama de clases persistentes



1.7.3 MODELO DE DATOS

Figura 6 Modelo de datos



CONCLUSIONES

- La realización de un análisis crítico de la aplicación existente permitió determinar el estado del cumplimiento de los objetivos iniciales definidos en iteraciones anteriores.
- La obtención de la especificación de los requisitos funcionales y no funcionales a cumplir por el sistema posibilitó realizar la descripción de los casos de uso que darán solución a los problemas identificados.
- La utilización de la estructura de clases básicas del CMS Joomla permitió la consolidación de la arquitectura Modelo-Vista-Controlador definida para la aplicación.
- Se generaron los diferentes artefactos que servirán de entrada para la realización del Modelo de Implementación del sistema.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

INTRODUCCIÓN

En el presente capítulo se realiza una descripción de la implementación del sistema y de las pruebas aplicadas al mismo. Se incluyen el modelo de implementación, los diagramas de componentes y de despliegue seguidos de las pruebas aplicadas con el uso del método de caja negra y se ilustran los diferentes casos de pruebas seleccionados.

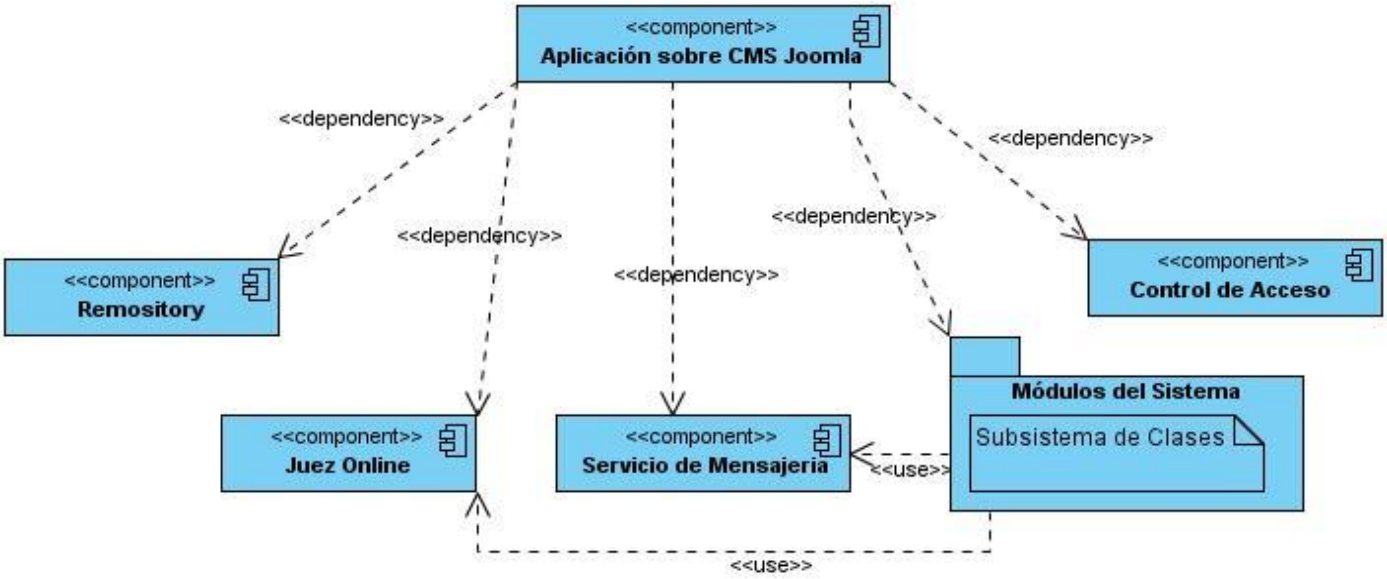
3.1 MODELO DE IMPLEMENTACIÓN

En El Proceso Unificado de Desarrollo de Software se especifica que se comienza a trabajar en la implementación de un sistema con el resultado obtenido del modelo de diseño y que su objetivo principal es desarrollar la arquitectura y el sistema como un todo. Uno de los artefactos que se genera en este flujo de trabajo es el modelo de implementación donde se *describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes*…. (Jacobson, y otros, 2000).

3.1.1 DIAGRAMAS DE COMPONENTES

En una etapa previa de desarrollo del sistema sobre el cual se basa este trabajo de tesis, se estructuró la organización en paquetes del modelo de implementación de la aplicación Coliseo Virtual, haciendo uso del patrón Modelo-Vista-Controlador que fue definido en la descripción de la arquitectura del sistema. A continuación se muestra la interacción entre los componentes que conforman el sistema.

Figura 7 Diagrama de componentes del sistema



1. El gestor Remository permite facilitar la gestión de recursos en el sistema.

2. El Juez en Línea es una solución de software que permite el buen funcionamiento de los diferentes estilos de competencia que requieren de un proceso de calificación.

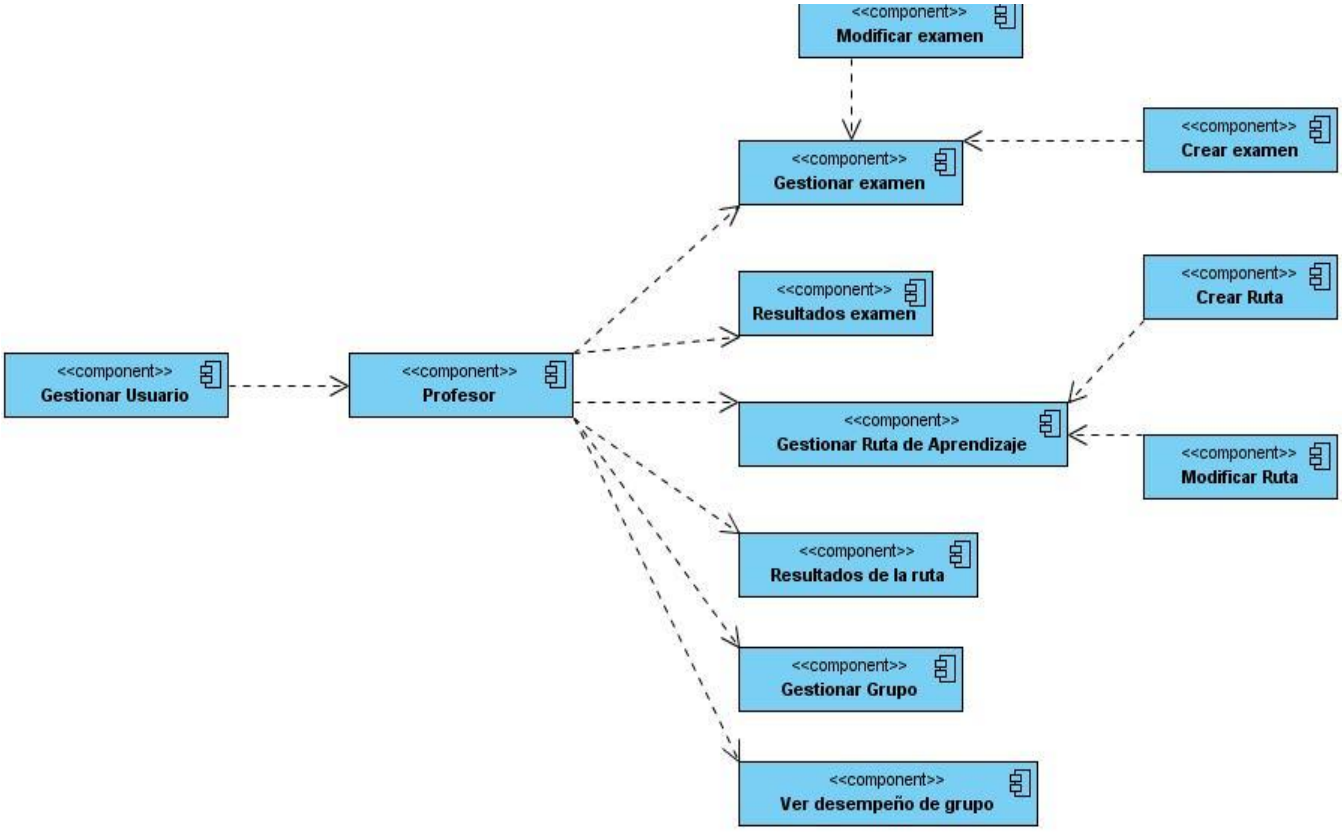
3. El Servicio de Mensajería utiliza un componente denominado UddeIM que posibilita el envío de mensajes entre usuarios y administradores en caso de ser requerido.

4. El Control de Acceso es un componente propuesto por el CMS Joomla para dar solución a este propósito.

5. En el subsistema de implementación Módulos del Sistema se muestra la distribución de los mismos en la aplicación.

En la actual etapa de desarrollo del sistema Coliseo Virtual se definieron nuevos diagramas de componentes que formarán parte del subsistema de implementación Módulos del Sistema al cual se hizo referencia con anterioridad. En el siguiente diagrama se muestra el módulo de Profesores, en los Anexos se pueden encontrar el resto de los diagramas.

Figura 8 Diagrama de Componentes: Profesor



3.1.2 DIAGRAMA DE DESPLIEGUE

Con el objetivo de describir la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo fue elaborado el modelo de despliegue de la aplicación informática Coliseo Virtual que se presenta a continuación.

Figura 9 Diagrama de Despliegue

Para el trabajo con la aplicación se necesita una computadora cliente con conexión intranet o a internet, que se conectara mediante el protocolo HTTP al servidor de la aplicación web. Dicho servidor debe poseer como mínimo 512 Mb de memoria RAM y para un funcionamiento óptimo se recomienda el uso de 1Gb de memoria. El servidor de la aplicación estará conectado mediante el protocolo TCP/IP al gestor de base datos MySQL.

3.1.3 ASPECTOS A SEGUIR PARA LA CODIFICACIÓN

Durante la primera fase de desarrollo del sistema Coliseo Virtual no fue definido ningún estándar de codificación para la implementación del sistema, sin embargo se establecieron una serie de aspectos para realizar la programación del mismo. Los desarrolladores del presente módulo continuaron haciendo uso de esta forma de organización del código la cual será descrita a continuación.

Los nombres establecidos durante la etapa de implementación, ya sean nombres de rutina, de variables, así como las denominaciones de tablas de la base de datos y componentes del sistema; serán escritos con letra minúscula y en caso de requerir dos o más palabras éstas estarán separadas por un guión bajo.

```
function asignar_grupo_examen($id_grupo, $id_examen)

var contador_reloj,
```

Para que la organización del código sea más clara se establece el siguiente formato:

- Se estableció una sangría de 3 espacios para el alinear las secciones de código.
- Las llaves de apertura y cierre estarán alineadas verticalmente.

```
function adicionar_examen ($nombre_examen)
{
   ...
}
```

- Se aplicará una sangría de dos espacios a las construcciones en las líneas de construcción lógica.

```
if($valor_asignado == NULL)
  {
    ...
  }
else

 if($valor_asignado == 2)
   {
     ...
   }

else
 {
   ...
 }
```

- Se dejará un espacio antes y después de cada operador siempre que no se altere la sangría aplicada al código.

```
for($i = 0; $i < count($temporal); $i++)
  {
   ...
  }
```

3.2 MODELO DE PRUEBAS

Según Jacobson, Booch y Rumbaugh plantean en el libro El Proceso Unificado del Desarrollo del Software, en el flujo de trabajo de Pruebas se verifica el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregados. En el presente epígrafe se muestran las pruebas que se aplicaron al sistema informático Coliseo Virtual en virtud de mostrar los resultados que se obtuvieron con las mismas, las cuales proveerán un elemento crítico para evaluar la calidad del software.

3.2.1 MÉTODO DE CAJA NEGRA

En el método de caja negra las pruebas se llevan a cabo sobre la interfaz del software. La idea es demostrar que las funciones del software son operativas y que la integridad de la información externa se mantiene. Este método ha sido utilizado con el objetivo de realizar pruebas que se centren principalmente en los requisitos funcionales permitiendo encontrar (Aguilar, 2007):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de rendimiento.
- Errores de inicialización y terminación.

A continuación se presenta la descripción de los casos de prueba de caja de negra.

Casos de Prueba del CU Gestionar ruta de aprendizaje:

Casos de prueba: Adicionar Ruta de Aprendizaje

Tabla 3 Caso de prueba gestionar ruta de aprendizaje

Condición de entrada	Casos válidos	Casos no válidos
Escribir el nombre de la ruta de aprendizaje.	Ruta_12$	Dejar el campo vacio
Seleccionar una fecha de inicio	2010-04-14	Dejar el campo vacio
Seleccionar una fecha de fin	2010-05-01	Dejar el campo vacio
Seleccionar los problemas de la ruta	Seleccionar uno o más problemas	No seleccionar ningún problema

Tabla 4 Escenario 1 del Caso de Prueba Adicionar Rutas de Aprendizaje

Entrada	
El profesor introduce los datos de manera correcta para crear la ruta: **Nombre:** Ruta 1 **Fecha de Inicio:** 2010-04-14 **Fecha de Fin:** 2010-04-20 **Problemas:** el profesor selecciona los problemas que desea agregar a la ruta de aprendizaje	
Resultado esperado:	Se crea la ruta de aprendizaje, que estará disponible a partir de la fecha seleccionada y estará compuesta por los ejercicios seleccionados.
Resultado:	Satisfactorio

Tabla 5 Escenario 2 del Caso de Prueba Adicionar Rutas de Aprendizaje

Entrada	
El profesor introduce incorrectamente los datos para crear la ruta de aprendizaje: **Nombre:** Ruta 1 **Fecha de Inicio:** 2010-04-14 **Fecha de Fin:** ´Campo vacío´ **Problemas:** el profesor selecciona los problemas que desea agregar a la ruta de aprendizaje	
Resultado esperado:	El sistema muestra el siguiente mensaje de alerta: “Faltan datos necesarios para crear la ruta”
Resultado:	Satisfactorio

Tabla 6 Caso de Prueba: Eliminar Ruta de Aprendizaje

Condición de entrada	Casos válidos	Casos no válidos
Selección de una ruta de aprendizaje	Seleccionar una ruta de aprendizaje	No seleccionar ninguna ruta de aprendizaje
Selección de la opción aceptar	Selecciona la opción aceptar	
Selección de la opción cancelar	Selecciona la opción cancelar	

Tabla 7 Escenario 1 del Caso de Prueba Eliminar Ruta de Aprendizaje

Entrada	
El profesor selecciona la ruta que desea eliminar.	
Resultado esperado:	El sistema muestra una ventana que permite aceptar la eliminación o cancelarla.
Resultado:	Satisfactorio

Tabla 8 Escenario 2 del Caso de Prueba Eliminar Ruta de Aprendizaje

Entrada	
El profesor selecciona la opción de aceptar la eliminación	
Resultado esperado:	El sistema elimina la ruta y muestra el siguiente mensaje de notificación: “La ruta ha sido eliminada con éxito”
Resultado:	Satisfactorio

Tabla 9 Escenario 3 del Caso de Prueba Eliminar Ruta de Aprendizaje

Entrada	
El profesor no selecciona una ruta para eliminarla.	
Resultado esperado:	El sistema muestra el siguiente mensaje de notificación: “Debe seleccionar una ruta de aprendizaje”
Resultado:	Satisfactorio

Tabla 10 Caso de prueba: Modificar Ruta de Aprendizaje

Condición de entrada	Casos válidos	Casos no válidos
Selección de una ruta de aprendizaje	Seleccionar una ruta de aprendizaje para ser modificada	No seleccionar ninguna ruta de aprendizaje
Escribir el nombre de la ruta de aprendizaje	Escribir el nombre de la ruta de aprendizaje	Dejar el campo vacio
Seleccionar una fecha de inicio	2010-04-28	Dejar el campo vacio
Seleccionar una fecha de fin	2010-05-06	Dejar el campo vacio

Tabla 11 Escenario 1 del Caso de Prueba: Modificar Rutas de Aprendizaje

Entrada	
El profesor introduce los datos de manera correcta para modificar la ruta: **Nombre:** Cambio de ruta **Fecha de Inicio:** 2010-04-20 **Fecha de Fin:** 2010-04-30 **Problemas:** el profesor selecciona los problemas que desea cambiar en la ruta de aprendizaje	
Resultado esperado:	Se modifica la ruta de aprendizaje, que estará disponible a partir de la fecha seleccionada y estará compuesta por los ejercicios seleccionados.
Resultado:	Satisfactorio

Tabla 12 Escenario 2 del Caso de Prueba: Modificar Rutas de Aprendizaje

Entrada	
El profesor introduce incorrectamente los datos necesarios para modificar la ruta de aprendizaje: **Nombre:** 'Campo vacío' **Fecha de Inicio:** 2010-04-14 **Fecha de Fin:** 'Campo vacío' **Problemas:** el profesor selecciona los problemas que desea agregar a la ruta de aprendizaje.	
Resultado esperado:	El sistema muestra el siguiente mensaje de alerta: “Faltan datos necesarios”.
Resultado:	Satisfactorio

Al finalizar la fase de pruebas se habían realizado 15 casos de prueba en dos iteraciones, donde se detectaron una serie de no conformidades que para la culminación de este período de desarrollo fueron solucionadas por el equipo de trabajo demostrando así que el módulo desarrollado respondía a los requisitos funcionales identificados en etapas previas. Seguidamente se muestra un resumen de las pruebas de Caja Negra realizadas a la aplicación.

Tabla 13 Resumen de las pruebas de caja negra

Casos de prueba	Escenarios del Caso de Prueba	Resultado
1. Adicionar examen.	1.1 Adicionar los datos de un examen introduciendo los mismos correctamente.	Satisfactorio.
	1.2 Adicionar los datos de un examen dejando campos requeridos en blanco.	Satisfactorio.
	1.3 Cancelar ¨ Adicionar examen ¨.	Satisfactorio.
2. Eliminar examen.	2.1 Eliminar examen correctamente.	Satisfactorio.
	2.2 Cancelar eliminación de examen.	Satisfactorio.
3. Modificar examen.	3.1 Modificar los datos del examen introduciendo los mismos correctamente.	Satisfactorio.
	3.2 Modificar los datos del examen dejando campos requeridos en blanco.	Satisfactorio.
	3.3 Cancelar ¨ Modificar examen ¨.	Satisfactorio.
4. Asignar examen.	4.1 Asignar examen a estudiantes correctamente.	Satisfactorio.
	4.2 Asignar examen a grupos correctamente.	Satisfactorio.
	4.3 Cancelar asignación de examen.	Satisfactorio.
5. Realizar examen.	5.1 Realizar el examen hasta el final.	Satisfactorio.
	5.2 Abandonar el examen guardándolo previamente.	Satisfactorio.
	5.3 Abandonar el examen sin guardarlo.	Satisfactorio.
6. Ver reporte específico de examen.	6.1 Ver reporte específico del examen.	Satisfactorio.
	6.2 Salir del reporte específico.	Satisfactorio.
7. Ver reporte general del examen.	7.1 Ver reporte general del examen.	Satisfactorio.
	7.2 Salir del reporte general del examen.	Satisfactorio.
8. Adicionar ruta de aprendizaje.	8.1 Adicionar los datos de una ruta de aprendizaje introduciendo los mismos de manera correcta.	Satisfactorio.
	8.2 Adicionar los datos de una ruta de aprendizaje dejando campos requeridos en blanco.	Satisfactorio.
	8.3 Cancelar ¨ Adicionar ruta de aprendizaje ¨.	Satisfactorio.
9. Eliminar ruta de aprendizaje.	9.1 Eliminar una ruta de aprendizaje correctamente.	Satisfactorio.
	9.2 Cancelar ¨ Eliminar ruta de aprendizaje ¨.	Satisfactorio.
10. Modificar ruta de aprendizaje.	10.1 Modificar una ruta de aprendizaje introduciendo los datos correctamente.	Satisfactorio.
	10.2 Modificar los datos de la ruta de aprendizaje dejando campos requeridos en blanco.	Satisfactorio.
	10.3 Cancelar ¨ Modificar ruta de aprendizaje ¨.	Satisfactorio.

11. Asignar ruta de aprendizaje.	11.1 Asignar una ruta de aprendizaje a grupos de manera correcta.	Satisfactorio.
	11.2 Asignar una ruta de aprendizaje a estudiantes de manera correcta.	Satisfactorio.
	11.3 Cancelar ¨ Asignar ruta de aprendizaje ¨.	Satisfactorio.
12. Realizar ruta de aprendizaje.	12.1 Realizar problemas.	Satisfactorio.
	12.2 Realizar test.	Satisfactorio.
	12.3 Abandonar test.	Satisfactorio.
	12.4 Realizar examen.	Satisfactorio.
	12.5 Abandonar examen.	Satisfactorio.
13. Ver reporte específico de rutas de aprendizaje.	13.1 Ver reporte específico de test.	Insatisfactorio.
	13.2 Ver reporte específico del examen.	Satisfactorio.
	13.3 Ver reporte específico de los problemas.	Satisfactorio.
	13.4 Salir del reporte específico de la ruta de aprendizaje.	Satisfactorio.
14. Ver reporte general de la ruta de aprendizaje.	14.1 Ver reporte general de la ruta de aprendizaje.	Satisfactorio.
	14.2 Salir del reporte general de la ruta de aprendizaje.	Satisfactorio.
15. Ver ranking de rutas.	15.1 Ver ranking de rutas.	Insatisfactorio.
	15.2 Salir del ranking	Satisfactorio.

Tabla 14 No conformidades detectadas

Caso de prueba	Escenario	No Conformidad	Significativo	No Significativo	Estado de la No Conformidad
13. Ver reporte específico de rutas de aprendizaje	13.1 Ver reporte específico de test.	El sistema no carga los test realizados en el combobox.	X		Pendiente:28/05/2010
					Resuelta: 29/05/2010
15. Ver ranking de rutas	15.1 Ver ranking.	Sistema asigna de manera desordenada las puntuaciones del ranking	X		Pendiente: 30/05/2010
					Resuelta: 01/06/2010
15. Ver ranking de rutas	15.1 Ver ranking.	Está escrita la palabra ¨Anno" en			Pendiente:09/06/2010

		la tabla que muestra el ranking		X	Resuelta: 10/06/2010

CONCLUSIONES

- La utilización del modelo de diseño especificado en el capítulo anterior permitió obtener el modelo de implementación, con el cual se muestra la estructura final de los diferentes componentes del sistema.
- Se elaboró un Diagrama de Despliegue que permite conocer la distribución física del sistema.
- Se comprobó, mediante la utilización de pruebas de caja negra, el correcto funcionamiento de la aplicación, dando así cumplimiento a los requisitos funcionales definidos.

CONCLUSIONES GENERALES

Al culminar el desarrollo del presente trabajo se considera cumplido el objetivo propuesto sobre la base de las siguientes conclusiones:

- El estudio realizado de los jueces en línea existentes dentro y fuera del país, permitió comprobar que no contaban con características similares a las que se deseaban en la aplicación informática Coliseo Virtual.
- El análisis de las diferentes herramientas y metodologías asociadas a la informática que pudiesen ser utilizadas en el proceso de desarrollo posibilitó la identificación de las que resultaban adecuadas según las características del sistema a implementar.
- La realización del proceso de levantamiento de requisitos permitió identificar las funcionalidades requeridas en la aplicación.
- La inclusión de las nuevas funcionalidades dentro del modelamiento del sistema, el Modelo de Diseño, el Modelo de Implementación y la base de datos, posibilitó la actualización de la documentación y la generación de los artefactos correspondientes.
- Se desarrolló la implementación de las funcionalidades propuestas obteniendo un módulo para profesores y estudiantes que permitirá un mayor intercambio entre los mismos y un sistema de atención diferenciada al estudiante.
- Las pruebas realizadas sobre la aplicación permitieron mejorar su funcionamiento e identificar y solucionar problemas no detectados en el proceso de implementación.

RECOMENDACIONES

- Perfeccionar la interfaz de la portada de la aplicación con la asistencia de diseñadores y especialistas en el tema.
- Agregar opciones que permitan a los profesores personalizar la forma en la que los estudiantes resuelven las rutas de aprendizaje para que el proceso funcione de manera más eficiente.
- Incluir nuevas categorías de ejercicios que permitan que los profesores creen rutas de aprendizaje de mayor variedad, potenciando una mejor formación de habilidades en los estudiantes.

BIBLIOGRAFÍA

Aguilar, Maria Elena. 2007. *Flujo de trabajo Pruebas.* 2007.

Álvarez, Miguel Ángel. 2003. Desarrollo Web. [En línea] 2003. http://www.desarrolloweb.com/articulos/1178.php.

Asteasuain, Fernando. 2002. Lafhis. [En línea] 2002. [Citado el: 19 de Marzo de 2009.] www.lafhis.dc.uba.ar.

Aumaille, Benjamin. 2000. *JavaScript y VBScript.* 2000.

Belando, José Ramón. 2007. Diseño e implementación de una herramienta para la gestión telemática de guías docentes. 2007.

Bird, Richard. 2000. *Introducción a la Programación Funcional con Haskell.* 2000.

Booch, Grady. 1999. *Software architecture and the UML.* 1999.

Canós, José H., Letelier, Patricio y Penadés, María Carmen. *Metodologías Ágiles en el Desarrollo de Software.* Valencia : s.n.

Capuz, H. Alonso. 2008. *Genereitor 2.* 2008.

Casares, Claudio. 2008. Maestros del Web. [En línea] 2008. http://www.maestrosdelweb.com/editorial/tutsql1/.

Catalani, Exequiel. 2007. wordpress. [En línea] agosto de 2007. http://exequielc.wordpress.com.

Craig, Larman. 1999. *UML y Patrones.* 1999.

Del Castillo Morán, Pedro. 2008. Modelo de evolución de las TIC en la pequeña y mediana empresa. 2008.

Díaz, Aidelyn, Reyes, Raudel y Armas, Dayron. 2009. El desarrollo humano local en los entornos virtuales: aplicación tecnológica Universitas Cuba. 2009.

Dique, Raúl Gonzalez. 2010. Mundo week. [En línea] 12 de 02 de 2010. http://mundogeek.net/archivos/2010/02/12/.

Drupal.org. 2009. Drupal. [En línea] 2009. http://drupal.org/.

Ext JS Inc. 2010. Ext Js. [En línea] 2010. http://www.extjs.com/products/js/.

Facultad Regional de Buenos Aires. 2005. *Fundamentos teóricos de los Paradigmas de Programación.* Buenos Aires : s.n., 2005.

Fernández, Carlos Alberto. 2000. *El Proceso Unificado Rrational para el desarrollo de software.* 2000.

Flores, Xavier. 2009. *Computación II.* 2009.

García, Joaquín. 2005. Prácticas y métodos para mejorar el desarrollo de proyectos de software. *IngenieroSoftware.* [En línea] 2005. http://www.ingenierosoftware.com/analisisydiseno/uml.php.

Gómez de Silva García, Andrés, de Jesús, Ignacio y Briceño, Ania. 2008. *Introducción a la Computación.* 2008.

Guevara, Roberto Carlos. 2008. *Sentencias básicas usadas en la programación de computadoras.* 2008.

IBM. 2006. FTP Directory: ftp://ftp.software.ibm.com/. [En línea] 2006. ftp://ftp.software.ibm.com/software/rational/web/datasheets/rose_ds.pdf.

Jacobson, Booch y Rumbaugh. 2000. *El proceso unificado de desarrollo de software.* 2000.

Kent, Beck. 1999. *¨Extreme Programming Explained¨ First Edition.* 1999.

Kruchten, Philippe. 2004. *The Rational Unified Process (Third Edition).* 2004.

Lara, Pablo y Duart, Josep María. 2005. Gestión de contenidos en el e-learning: acceso y uso de objetos de información como recurso estratégico. 2005.

Letelier, Patricio. 2009. [En línea] 2009. https://pid.dsic.upv.es.

Martínez, Lucía. 2006. monografías.com. [En línea] 2006. http://www.monografias.com/trabajos37/arquitectura-de-sistemas/arquitectura-de-sistemas2.shtml.

Microsoft Corporation. 2010. Imagine Cup. [En línea] 2010. http://imaginecup.com.

Oracle Corporation. 2010. Oracle Corporation and/or its affiliates. [En línea] 2010. http://www.mysql.com/about/.

PC Magazine. 2010. PC Magazine. [En línea] 2010. http://www.pcmag.com.

PostgreeSQL Global Development Group. 2010. postgreSQL. [En línea] 2010. http://www.postgresql.org/.

Rossel, Gerardo. 2004. Amzi! Prolog. *sitio Web Amzi!* [En línea] 2004. http://www.amzi.com/.

Rubio, Manuel y Cayetano, Manuel. 2002. PDFMoo.com. [En línea] 2002. http://www.pdfmoo.com/database-files/1029/server-databases-clash.html.

Ruíz, Francisco. 2000. UCLM-ESI Bases de Datos . [En línea] 2000. http://alarcos.inf-cr.uclm.es/doc/bda/index.htm.

Sánchez, María A. Mendoza. 2004. Informatizate. [En línea] 2004. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Sánchez, Miguel. 2001. *Javascript.* 2001.

Sierra, María. 2008. Universidad de Cantabria. [En línea] 2008. http://www.unican.es/.

Tianjin University. 2005. Online Judge. [En línea] 2005. http://acm.tju.edu.cn/toj/.

Toledo, Alma, y otros. 2002. Universidad Autónoma del Estado de Morelos . [En línea] 2002. http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf.

Trevejo Alonso, Juan Antonio. 2008. Joomla para principiantes: Aprendiendo a crear y mantener sitios web. 2008.

UML. **Becerra, Carlos. 2008.** Chile : s.n., 2008. Jornadas chilenas de compuitación.

Visual Paradigm International. 2007. Sitio Oficial de Visual Paradigm. [En línea] 2007. http://www.visual-paradigm.com/.

Vivas White, Pedro y López, Emilio. 2009. *Cuerpo de Profesores de Enseñanza Secundaria. Informática. Temario Vol. III.* 2009.

Weitzenfeld, Alfredo. 2004. *Ingeniería de software orientada a objetos con UML, Java e Internet.* 2004.

Wells, Don. 2009. Agile Software Development: A gentle introduction. [En línea] 2009. http://www.agile-process.org/.

Winblad, Ann L. 1993. *Software Orientado a Objetos.* 1993.

TRABAJOS CITADOS

Aguilar, Maria Elena. 2007. *Flujo de trabajo Pruebas.* 2007.

Álvarez, Miguel Ángel. 2003. Desarrollo Web. [En línea] 2003. http://www.desarrolloweb.com/articulos/1178.php.

Asteasuain, Fernando. 2002. Lafhis. [En línea] 2002. [Citado el: 19 de Marzo de 2009.] www.lafhis.dc.uba.ar.

Aumaille, Benjamin. 2000. *JavaScript y VBScript.* 2000.

Belando, José Ramón. 2007. Diseño e implementación de una herramienta para la gestión telemática de guías docentes. 2007.

Bird, Richard. 2000. *Introducción a la Programación Funcional con Haskell.* 2000.

Booch, Grady. 1999. *Software architecture and the UML.* 1999.

Canós, José H., Letelier, Patricio y Penadés, María Carmen. *Metodologías Ágiles en el Desarrollo de Software.* Valencia : s.n.

Capuz, H. Alonso. 2008. *Genereitor 2.* 2008.

Casares, Claudio. 2008. Maestros del Web. [En línea] 2008. http://www.maestrosdelweb.com/editorial/tutsql1/.

Catalani, Exequiel. 2007. wordpress. [En línea] agosto de 2007. http://exequielc.wordpress.com.

Craig, Larman. 1999. *UML y Patrones.* 1999.

Del Castillo Morán, Pedro. 2008. Modelo de evolución de las TIC en la pequeña y mediana empresa. 2008.

Díaz, Aidelyn, Reyes, Raudel y Armas, Dayron. 2009. El desarrollo humano local en los entornos virtuales: aplicación tecnológica Universitas Cuba. 2009.

Dique, Raúl Gonzalez. 2010. Mundo week. [En línea] 12 de 02 de 2010. http://mundogeek.net/archivos/2010/02/12/.

Drupal.org. 2009. Drupal. [En línea] 2009. http://drupal.org/.

Ext JS Inc. 2010. Ext Js. [En línea] 2010. http://www.extjs.com/products/js/.

Facultad Regional de Buenos Aires. 2005. *Fundamentos teóricos de los Paradigmas de Programación.* Buenos Aires : s.n., 2005.

Fernández, Carlos Alberto. 2000. *El Proceso Unificado Rrational para el desarrollo de software.* 2000.

Flores, Xavier. 2009. *Computación II.* 2009.

García, Joaquín. 2005. Prácticas y métodos para mejorar el desarrollo de proyectos de software. *IngenieroSoftware.* [En línea] 2005. http://www.ingenierosoftware.com/analisisydiseno/uml.php.

Gómez de Silva García, Andrés, de Jesús, Ignacio y Briceño, Ania. 2008. *Introducción a la Computación.* 2008.

Guevara, Roberto Carlos. 2008. *Sentencias básicas usadas en la programación de computadoras.* 2008.

IBM. 2006. FTP Directory: ftp://ftp.software.ibm.com/. [En línea] 2006. ftp://ftp.software.ibm.com/software/rational/web/datasheets/rose_ds.pdf.

Jacobson, Booch y Rumbaugh. 2000. *El proceso unificado de desarrollo de software.* 2000.

Kent, Beck. 1999. *¨Extreme Programming Explained¨ First Edition.* 1999.

Kruchten, Philippe. 2004. *The Rational Unified Process (Third Edition).* 2004.

Lara, Pablo y Duart, Josep María. 2005. Gestión de contenidos en el e-learning: acceso y uso de objetos de información como recurso estratégico. 2005.

Letelier, Patricio. 2009. [En línea] 2009. https://pid.dsic.upv.es.

Martínez, Lucía. 2006. monografías.com. [En línea] 2006. http://www.monografias.com/trabajos37/arquitectura-de-sistemas/arquitectura-de-sistemas2.shtml.

Microsoft Corporation. 2010. Imagine Cup. [En línea] 2010. http://imaginecup.com.

Oracle Corporation. 2010. Oracle Corporation and/or its affiliates. [En línea] 2010. http://www.mysql.com/about/.

PC Magazine. 2010. PC Magazine. [En línea] 2010. http://www.pcmag.com.

PostgreeSQL Global Development Group. 2010. postgreSQL. [En línea] 2010. http://www.postgresql.org/.

Rossel, Gerardo. 2004. Amzi! Prolog. *sitio Web Amzi!* [En línea] 2004. http://www.amzi.com/.

Rubio, Manuel y Cayetano, Manuel. 2002. PDFMoo.com. [En línea] 2002. http://www.pdfmoo.com/database-files/1029/server-databases-clash.html.

Ruíz, Francisco. 2000. UCLM-ESI Bases de Datos . [En línea] 2000. http://alarcos.inf-cr.uclm.es/doc/bda/index.htm.

Sánchez, María A. Mendoza. 2004. Informatizate. [En línea] 2004. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Sánchez, Miguel. 2001. *Javascript.* 2001.

Sierra, María. 2008. Universidad de Cantabria. [En línea] 2008. http://www.unican.es/.

Tianjin University. 2005. Online Judge. [En línea] 2005. http://acm.tju.edu.cn/toj/.

Toledo, Alma, y otros. 2002. Universidad Autónoma del Estado de Morelos . [En línea] 2002. http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf.

Trevejo Alonso, Juan Antonio. 2008. Joomla para principiantes: Aprendiendo a crear y mantener sitios web. 2008.

UML. **Becerra, Carlos. 2008.** Chile : s.n., 2008. Jornadas chilenas de compuitación.

Visual Paradigm International. 2007. Sitio Oficial de Visual Paradigm. [En línea] 2007. http://www.visual-paradigm.com/.

Vivas White, Pedro y López, Emilio. 2009. *Cuerpo de Profesores de Enseñanza Secundaria. Informática. Temario Vol. III.* 2009.

Weitzenfeld, Alfredo. 2004. *Ingeniería de software orientada a objetos con UML, Java e Internet.* 2004.

Wells, Don. 2009. Agile Software Development: A gentle introduction. [En línea] 2009. http://www.agile-process.org/.

Winblad, Ann L. 1993. *Software Orientado a Objetos.* 1993.

ANEXOS

Diagrama de casos de uso para el actor estudiante

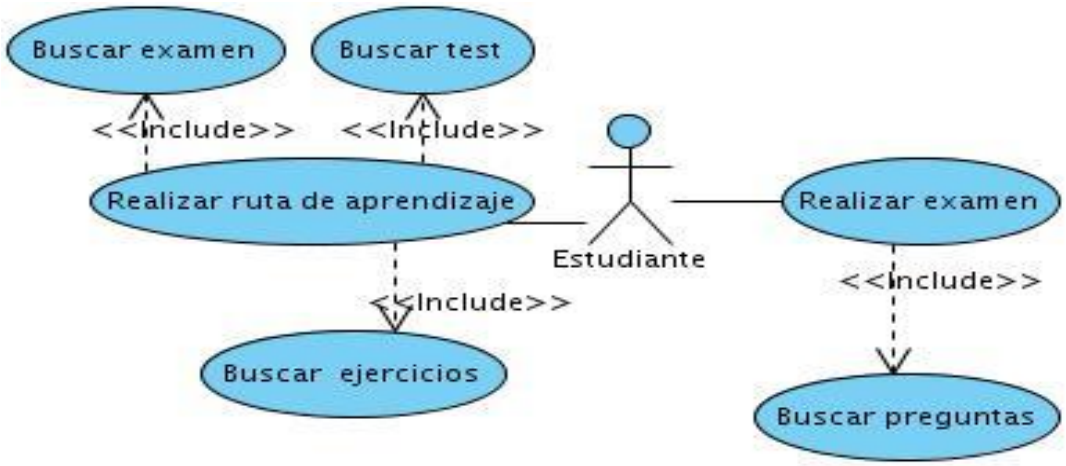


Diagrama de casos de uso para el actor profesor

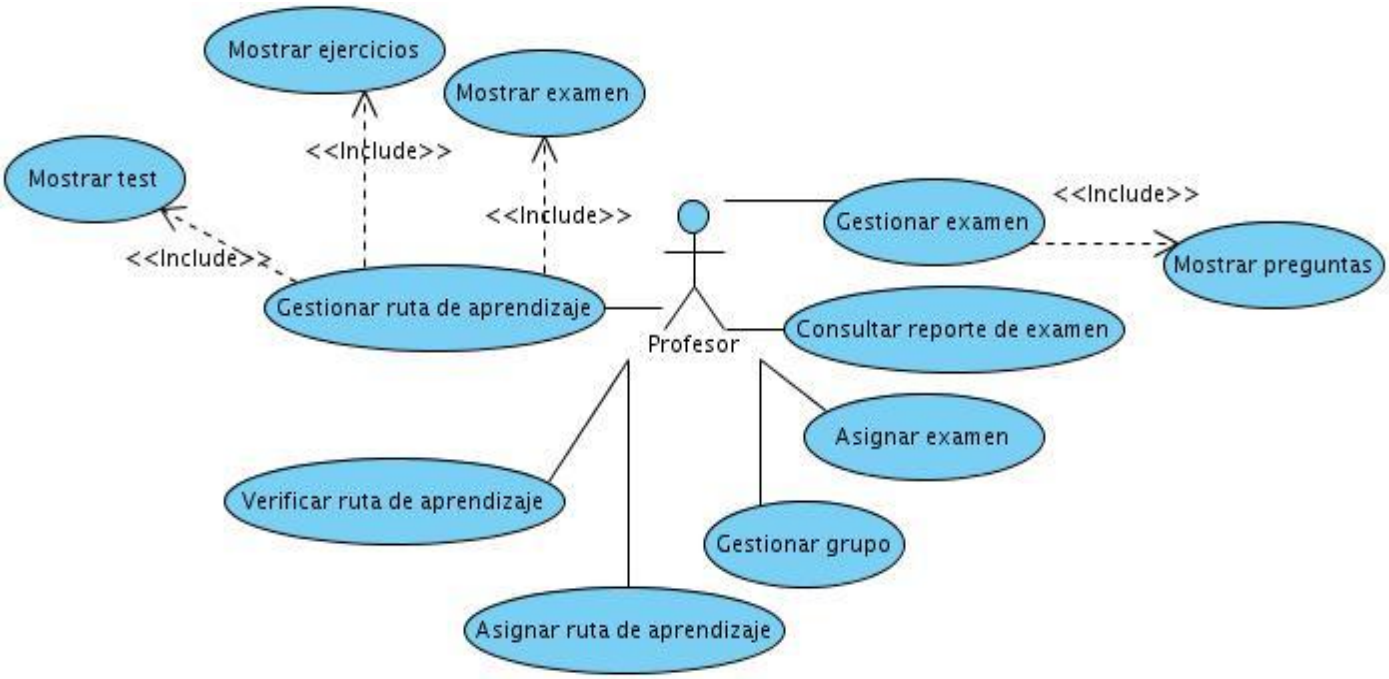


Diagrama de clases del diseño: Asignar ruta de aprendizaje

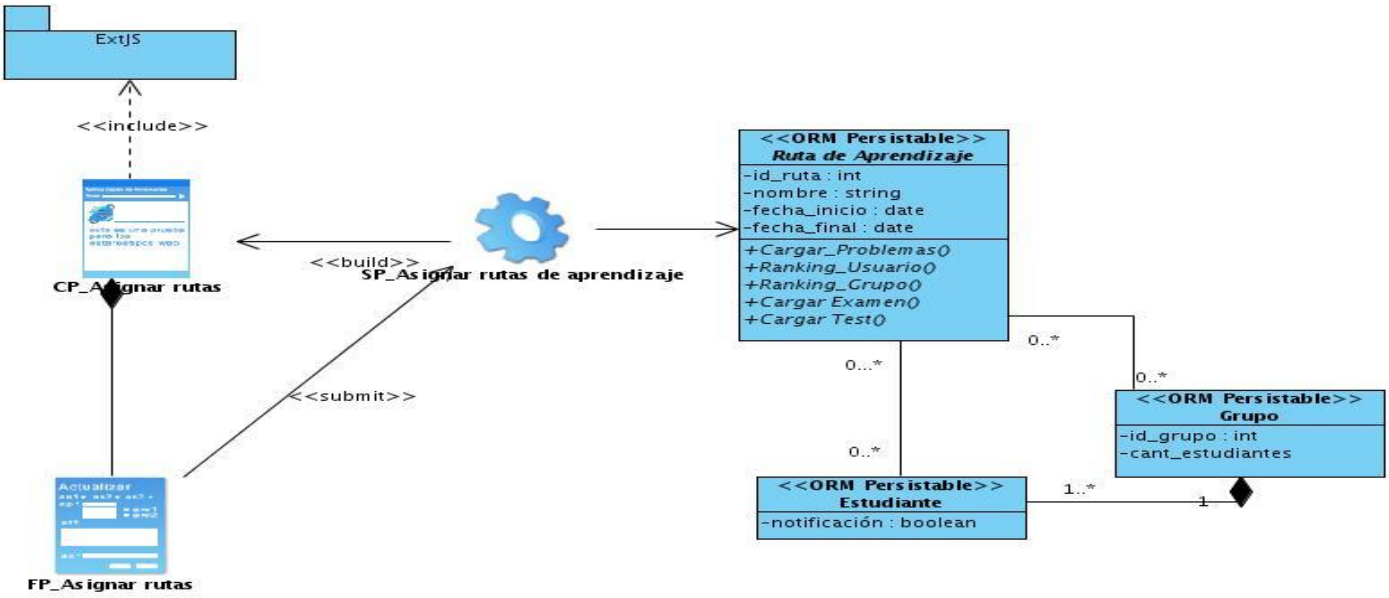


Diagrama de clases de diseño: Verificar rutas

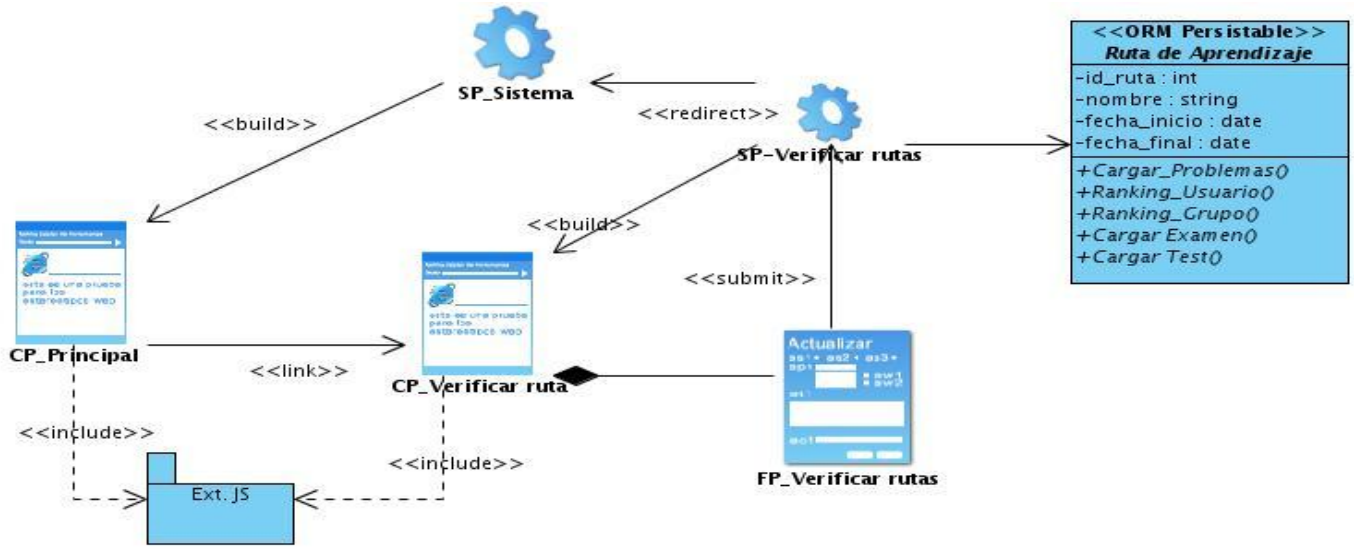


Diagrama de clases de diseño: Realizar ruta de aprendizaje

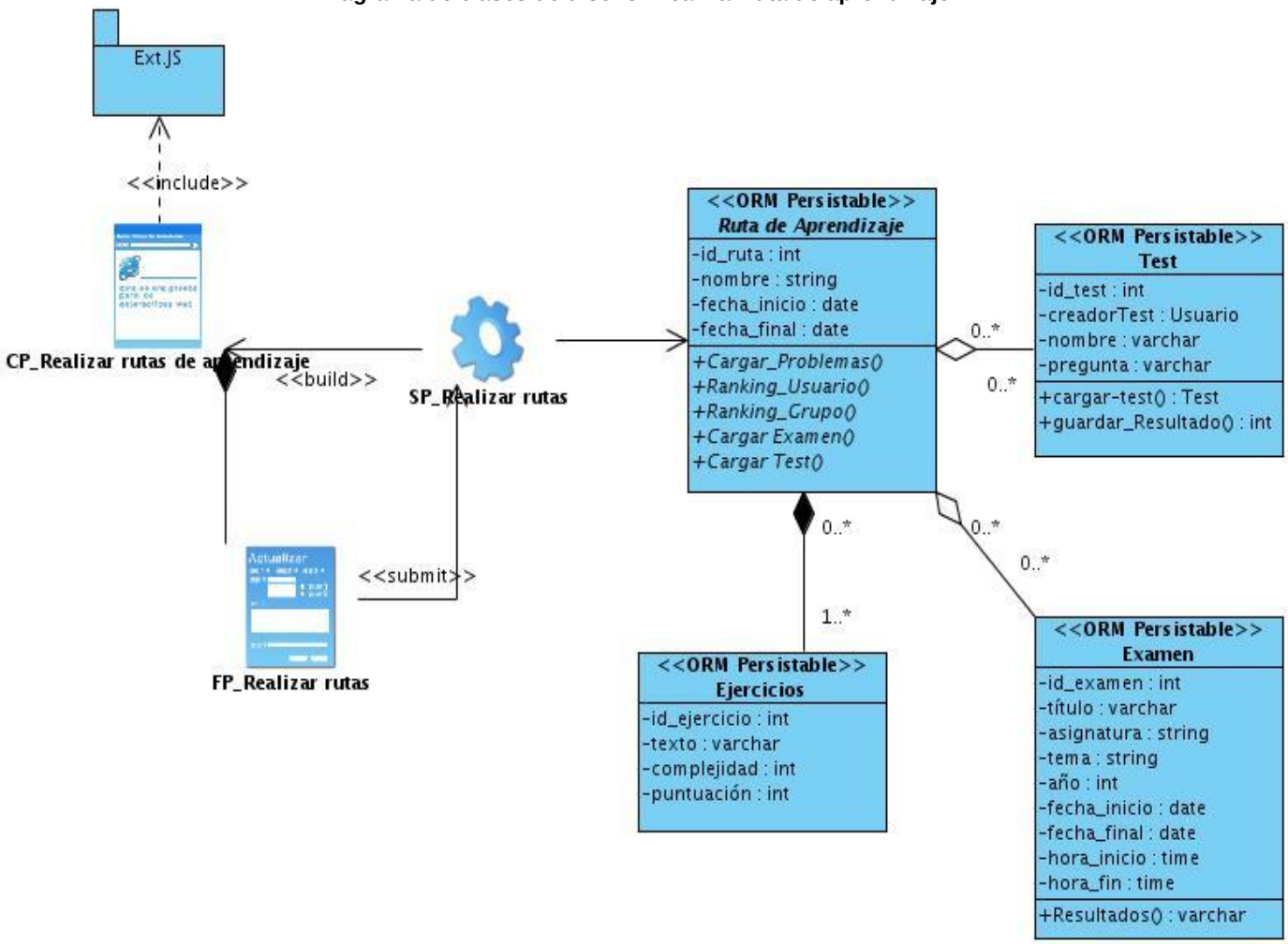


Diagrama de secuencia: Adicionar ruta de aprendizaje

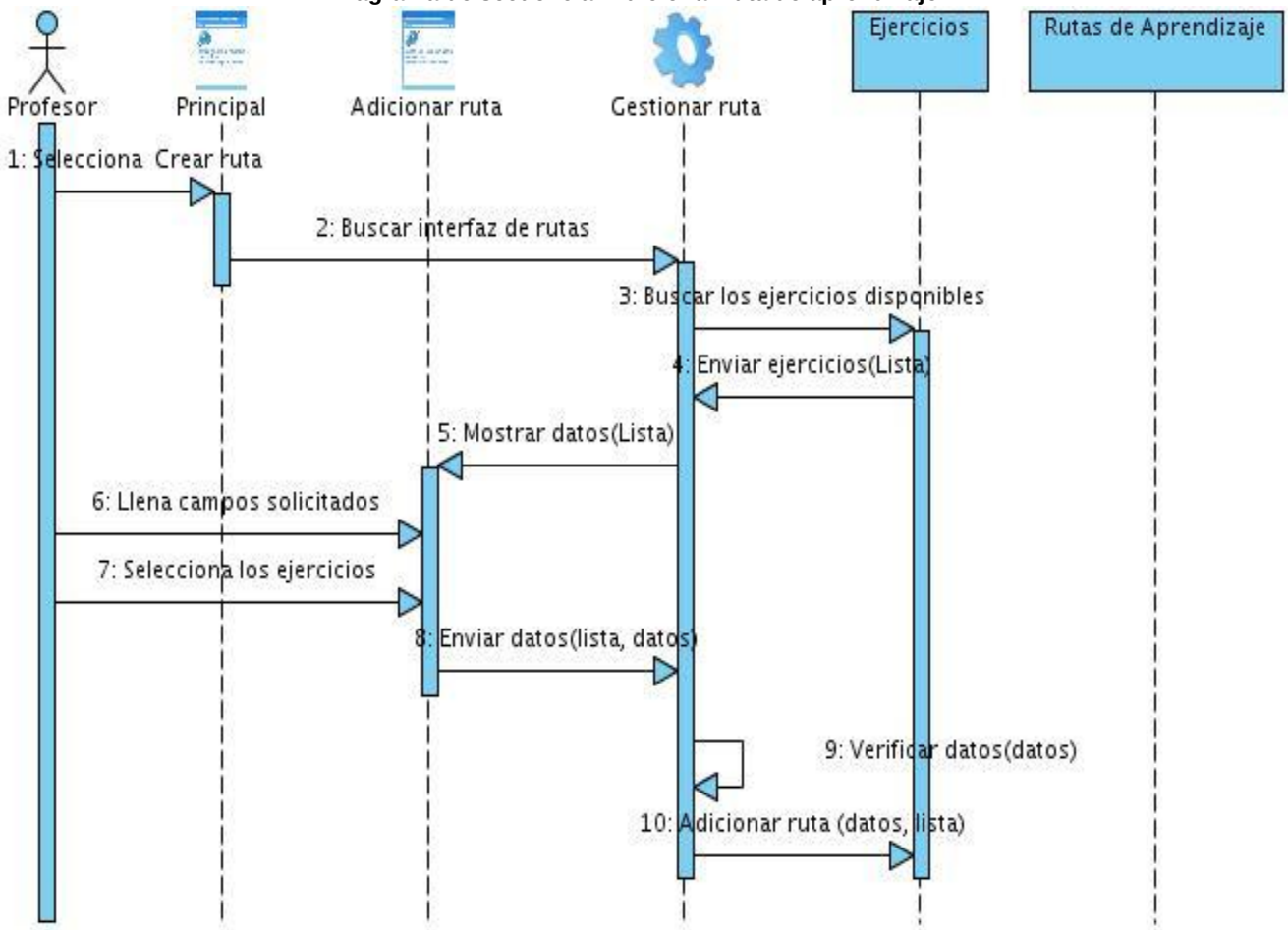


Diagrama de secuencia: Modificar ruta de aprendizaje