

Universidad de las Ciencias Informáticas
Facultad 3



Título: Sistema de Gestión de Seminarios
Científicos

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autora: Yennis González Cádiz

Tutores:

MSc. Henry Raúl González Brito

MSc. Naryana Linares Pons

La Habana, Cuba

Junio del 2013

Pensamiento



“El objetivo principal de la educación es crear personas capaces de hacer cosas nuevas, y no simplemente repetir lo que otras generaciones hicieron”

Jean Piaget

Declaración de Autoría

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yennis González Cadiz

MSc. Henry Raúl González Brito

Firma del autor

Firma del tutor

MSc. Naryana Linares Pons

Firma del tutor

Datos de Contacto

MSc. Naryana Linares Pons

Graduada de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI), 2007, Ciudad de La Habana, Cuba; con Título de Oro. Profesora Asistente. Graduada de Máster en Gestión de Proyectos Informáticos, 2012.

Ha asistido a una decena de congresos y eventos nacionales e internacionales como delegado, ponente o autor. Tiene en su haber 20 publicaciones electrónicas en revistas y eventos. Ha cursado y aprobado veinticinco cursos de postgrado. Ha impartido docencia de pregrado desde el año 2007. Su actividad académico-investigativa está respaldada por el impacto socio-político y cultural y generalizada a otros centros del país. Por sus resultados ha llegado a obtener el “Premio del Rector” de la Universidad de las Ciencias Informáticas 2011.

MSc. Henry Raúl González Brito

Graduado de Ingeniero Informático en el año 2005, profesor asistente de la Universidad de las Ciencias Informáticas. Ha trabajado durante 4 años en proyectos de Soluciones de Gestión. Tiene varios artículos publicados en eventos nacionales e internacionales. Fue líder del proyecto ERP- Cuba entre los años 2006 y 2008. Actualmente es el subdirector de la subdirección de Investigación y Postgrado del Centro de Informatización de la Gestión de Entidades de la Facultad 3.

Agradecimientos

A mis padres Adailsa Cádiz Marzo y Humberto González Hernández por su apoyo incondicional, consejos, respeto y amor. Gracias por existir, los quiero mucho.

A mi hermanito Dani y mi primito Caleb por alegrarme la vida con sus travesuras.

A mi novio Javier de León Álvarez, por estar cada día a mi lado ofreciéndome su amor, por ayudarme y comprenderme tanto en los cinco años de la carrera, sobre todo en la realización de esta tesis. Te amo bebé.

A toda mi familia, abuelos, tíos, tías, en especial a mi tía Deisy y mi prima Beni.

A mis tutores por ser protagonistas de esta obra, ya que sin su ayuda no hubiera sido posible la realización de este trabajo.

A los integrantes del tribunal y el oponente por sus recomendaciones y señalamientos oportunos.

A mis suegros Miguel de León y Deisy Álvarez por sus buenos consejos y apoyo durante los 5 años de la carrera.

A mi querido amigo Josbel Rodríguez, por haberme permitido ocupar un lugarcito en su corazón, pasar momentos agradables y prestarme su laptop cada vez que la necesité. Diana Tapia gracias por traer al mundo un ser tan especial.

A mi amiga Mailer Navarro y su hermano Maikel por su ayuda. Muchas gracias.

A mis compañeras de apartamento Dayana (La negrita), Yessika, Leiza (La trola), Kirenia (La marmota), Meylin y Wendy Ávila.

A mis compañeros de aula, en especial al Rikisimo, Anchel, Baboberto, Hernan, Juan Carlos, Aroldo, Gustavo.

Dedicatoria

A mis padres y hermano, las personas que más amo en mi vida.

A bebé por estar siempre a mi lado ofreciéndome su amor incondicional.

A mis suegros por apoyarme.

Al chiquitico por la confianza que siempre ha tenido en mí.

A la Revolución y Fidel.

Resumen

La educación superior cubana enfrenta el reto de mantener su organización y desarrollo sobre bases científicas. El perfeccionamiento continuo pone de manifiesto la pertinencia de la Universidad cubana que se renueva continuamente dada la necesidad creciente de actualizar y ampliar el conocimiento. El presente Trabajo de Diploma tiene como objetivo el desarrollo de un sistema informático para la gestión de los seminarios científicos que satisfaga las necesidades de la dirección del Centro de Informatización de la Gestión de Entidades (CEIGE) de la facultad 3. La utilización del sistema informatizado significaría una mejora considerable en cuanto a tiempo, planificación, organización, recuperación de información, permitiendo establecer indicadores y controlar los seminarios científicos de los profesionales que se encuentran el proceso de maestría.

Para el desarrollo del sistema se utilizó el marco de trabajo Sauxe y como metodología de desarrollo el Proceso Unificado de Desarrollo (RUP). Apoyándose de las herramientas PostgreSQL como gestor de Base de Datos, Apache como servidor web y el Netbeans como entorno de desarrollo integrado.

Palabras claves: Seminarios Científicos.

Tabla de contenido

Tabla de contenido

| | |
|--|----|
| Introducción | 1 |
| Capítulo 1: Fundamentación Teórica..... | 4 |
| 1.1 Introducción..... | 4 |
| 1.2 Conceptos básicos relacionados con el dominio de la investigación | 4 |
| 1.3 Estado del arte | 5 |
| 1.3.1 Sistemas de información científica..... | 5 |
| 1.3.2 Sistemas de gestión de información de seminarios científicos..... | 5 |
| 1.4 La producción científica de equipos de proyectos informáticos | 6 |
| 1.4.1 Impacto de la Gestión del Conocimiento en los resultados científicos de los equipos de proyectos informáticos..... | 7 |
| 1.5 Marco de trabajo Sauxe | 8 |
| 1.6 Descripción de los lenguajes y herramientas utilizadas..... | 9 |
| 1.7 Arquitectura de software..... | 14 |
| 1.7.1 Patrón Arquitectónico Modelo Vista Controlador (MVC)..... | 14 |
| 1.8 Metodologías de desarrollo de software | 15 |
| 1.8.1 Metodología RUP..... | 16 |
| 1.8.2 Extreme Programing (XP)..... | 17 |
| Conclusión | 19 |
| Capítulo 2: Análisis y Diseño de la solución | 20 |
| 2.1 Introducción..... | 20 |
| 2.2 Descripción de las características generales del sistema..... | 20 |
| 2.2.1 Artefactos generados | 20 |
| 2.3 Captura y especificación de requisitos funcionales..... | 22 |
| RF 1: Gestionar grupos de profesionales..... | 23 |
| RF 2: Gestionar perfil del profesional..... | 23 |

| | |
|---|----|
| RF 3: Gestionar la planificación de las presentaciones a los seminarios científicos.. | 23 |
| RF 4: Gestionar las tareas generadas en una presentación..... | 23 |
| RF 5: Gestionar cursos de los profesionales..... | 24 |
| RF 6: Gestionar los créditos no lectivos de los profesionales..... | 24 |
| RF 7: Gestionar grado científico | 24 |
| RF 8 Gestionar línea temática | 25 |
| RF 9 Gestionar programa de maestría..... | 25 |
| RF 10 Gestionar tipo de tarea..... | 25 |
| RF 11 Gestionar estado de tarea..... | 25 |
| RF 12 Gestionar modo de participación..... | 26 |
| RF 13 Mostrar los cursos de un profesional..... | 26 |
| RF 14 Mostrar las planificaciones de un profesional | 26 |
| RF 15 Mostrar las tareas de un profesional | 26 |
| 2.4 Descripción de los requisitos no funcionales | 26 |
| 2.5 Definición de los actores del sistema..... | 27 |
| 2.5.1 Definición de los casos de uso (CU) del sistema..... | 28 |
| 2.6 Diagramas de clases del análisis | 36 |
| 2.7 Diagramas de clases del diseño..... | 38 |
| 2.7.1 Patrones del diseño | 39 |
| 2.8 Diagramas de secuencia. | 42 |
| 2.9 Diagrama Entidad Relación | 44 |
| 2.10 Validación del diseño propuesto | 46 |
| Conclusiones parciales | 48 |
| Capítulo 3: Implementación y prueba | 49 |
| 3.1 Introducción..... | 49 |
| 3.2 Diagrama de despliegue..... | 49 |
| 3.3 Diagrama de componentes..... | 50 |

| | |
|---|----|
| 3.4 Nomenclatura de las clases..... | 52 |
| 3.4.1 Nomenclatura de las funciones | 54 |
| 3.5 Pruebas de caja negra | 54 |
| 3.5.1 Casos de prueba..... | 55 |
| Conclusiones parciales | 57 |
| Conclusiones Generales | 58 |
| Recomendaciones | 59 |
| Referencias Bibliográficas | 60 |
| Anexos | 62 |
| Glosario de Términos | 68 |

Índice de tablas

| | |
|---|----|
| Tabla 1: Artefactos generados en esta investigación | 21 |
| Tabla 2: Actores del sistema..... | 28 |
| Tabla 3: Descripción del caso de uso Gestionar perfil del profesional..... | 33 |
| Tabla 4: Descripción del caso de uso Gestionar grupos de profesionales..... | 36 |
| Tabla 5: Relación atributo con el tamaño operacional de clases | 47 |
| Tabla 6: Valor de los atributos por clases | 47 |
| Tabla 7: Caso de prueba Adicionar grupo..... | 57 |
| Tabla 8: Descripción CU Ver planificaciones de un profesional | 64 |
| Tabla 9: Descripción del CU Ver cursos..... | 64 |

Índice de Ilustraciones

| | |
|--|----|
| Ilustración 1: Modelo conceptual..... | 22 |
| Ilustración 2 Diagrama de casos de uso del sistema..... | 28 |
| Ilustración 3: Principales casos de uso | 29 |
| Ilustración 4: Diagrama de clases del análisis gestionar profesional | 37 |

| | |
|--|--------------------------------------|
| Ilustración 5: Diagrama de clases del análisis gestionar planificación..... | 38 |
| Ilustración 6: Diagrama de clases del diseño Gestionar cursos de los profesionales | 39 |
| Ilustración 7: Ejemplo patrón experto..... | 40 |
| Ilustración 8: Ejemplo de patrón creador..... | 41 |
| Ilustración 9: Ejemplo de patrón controlador | 41 |
| Ilustración 10: Diagrama de secuencia adicionar grupo | 42 |
| Ilustración 11: Diagrama de secuencia modificar grupo | 43 |
| Ilustración 12: Diagrama de secuencia eliminar grupo | 44 |
| Ilustración 13: Diagrama Entidad Relación | 46 |
| Ilustración 14: Criterio de evaluación de los atributos | 47 |
| Ilustración 15: Gráficos de los resultados de la métrica TOC | 48 |
| Ilustración 16: Diagrama de despliegue | 49 |
| Ilustración 17: Diagrama de componentes CU gestionar grupo | 51 |
| Ilustración 18: Diagrama de componentes CU gestionar cursos de profesionales | 51 |
| Ilustración 19: Ejemplo de estructura de las clases controladoras | 52 |
| Ilustración 20: Ejemplo de estructura de las clases del negocio..... | 52 |
| Ilustración 21: Ejemplo de estructura de las clases de dominio | 53 |
| Ilustración 22: Diagrama de clases del análisis gestionar créditos no lectivos | ¡Error! Marcador no definido. |
| Ilustración 23: Diagrama de clases del diseño Gestionar tareas generadas en una presentación. | ¡Error! Marcador no definido. |
| Ilustración 24: Diagrama de clases del análisis Gestionar perfil del profesional | ¡Error! Marcador no definido. |
| Ilustración 25: Diagrama de clases del diseño Gestionar planificación de seminarios científicos..... | ¡Error! Marcador no definido. |
| Ilustración 26: Diagrama de clases del diseño gestionar tareas..... | 65 |
| Ilustración 27: Diagrama de clases del diseño gestionar planificación de seminarios científicos..... | 66 |

| | |
|---|--------------------------------------|
| Ilustración 28: Diagrama de clases del diseño Gestionar cursos de los profesionales | ¡Error! Marcador no definido. |
| Ilustración 29: Diagrama de secuencia insertar profesional | 66 |
| Ilustración 30: Diagrama de secuencia modificar profesional | ¡Error! Marcador no definido. |
| Ilustración 31: Diagrama de secuencia eliminar profesional | ¡Error! Marcador no definido. |
| Ilustración 32: Diagrama de secuencia Ver cursos | 67 |

Introducción

En la actualidad la ciencia, la tecnología e innovación se han convertido en elementos fundamentales para el desarrollo. La elevación de la profesionalidad de los graduados universitarios constituye un aspecto de gran significación en las profundas transformaciones que se están operando en la educación cubana en los momentos actuales. La necesidad de lograr altos niveles de competencia profesional y avanzadas capacidades para la investigación y la innovación, han determinado un incremento sustancial en la formación académica de postgrado, en especial en las maestrías (Asencio 2011).

La educación de postgrado está considerada como una de las direcciones principales de trabajo de la educación superior en Cuba. Constituye el nivel más alto del sistema de educación superior, dirigido a graduados universitarios y a promover en ellos la educación a lo largo de la vida. En la educación de postgrado concurren uno o más procesos formativos y de desarrollo, no solo de enseñanza aprendizaje, sino también de investigación, articulados armónicamente en una propuesta docente-educativa.

La investigación por parte de los docentes se manifiesta de forma significativa en su perfeccionamiento profesional. Con el seminario investigativo se persigue la superación profesional. La discusión es el método que rige los debates del seminario. Es en este espacio donde se pone a prueba la preparación que tienen los profesionales. Este se realiza en forma de diálogo abierto, cuyo propósito es esclarecer distintos puntos de vista o aportar rutas o soluciones a problemas planteados.

La actividad de postgrado en la Universidad de las Ciencias Informáticas (UCI) atiende las necesidades de capacitación, además potencia el intercambio y brinda amplias posibilidades para estrechar vínculos entre organismos e instituciones de las Ciencias Informáticas. La formación académica de graduados universitarios se realiza a través de estudios de especialidades, maestrías y doctorados. En la superación profesional se organizan diplomados, entrenamientos, cursos cortos y actividades como conferencias, seminarios, talleres, que posibilitan el estudio, actualización y divulgación de los avances científicos y tecnológicos en la rama de la Informática (UCI 2011).

En entrevistas realizadas al subdirector de Investigación y Postgrado del Centro de Informatización de la Gestión de Entidades (CEIGE) se detectó que actualmente en la Universidad de las Ciencias Informáticas

el seguimiento y control de los profesionales que están en el proceso de maestría ha tenido deficiencias. Como ejemplo de ello se tiene que el seguimiento se hace mediante reuniones, afectando la toma de decisiones del proceso pues se hace muy engorroso y demorado saber las tareas y señalamientos pendientes de la reunión anterior. La revisión de los documentos para conocer el estado en cuanto a la superación profesional de los graduados y la actualización de los cursos que van concluyendo los maestrantes se realiza de forma manual. La actualización del cumplimiento de las tareas y señalamientos por parte de los maestrantes así como la planificación de las exposiciones se hace vía correo electrónico, lo que provoca que los profesionales a veces vuelvan a pedir la información por haberlos borrado o perdido. Las recomendaciones y señalamientos se hacen de manera personal y sobre todo en el espacio de los seminarios, ya que esta información no está disponible para que pueda ser revisada y se emitan criterios por personas calificadas. Por ese motivo en el presente curso escolar, la dirección del CEIGE se ha dado a la tarea de organizar un sistema de seminarios donde se pueda ir evaluando el nivel de avance del maestrante, tanto a nivel cualitativo como cuantitativo. Teniendo en cuenta esta problemática se puede identificar el siguiente **problema a resolver**: ¿Cómo gestionar la información referente a los seminarios científicos para que se satisfagan las necesidades de la dirección del CEIGE?

La presente investigación tiene como **objetivo general**: Desarrollar un sistema informático para la gestión de los seminarios científicos que satisfaga las necesidades de la dirección del CEIGE. Para alcanzar el objetivo general se propone como **objeto de estudio**: El proceso de investigación-superación en entidades de educación superior, tendiendo como **campo de acción**: Sistemas informáticos para la gestión de la investigación-superación. Se derivan del objetivo general los siguientes **objetivos específicos**:

1. Realizar un estudio de los procesos de gestión de los seminarios científicos en el CEIGE y formalizar el marco teórico-conceptual de la investigación.
2. Realizar el análisis y diseño del Sistema de Gestión de Seminarios Científicos para el CEIGE con el objetivo de lograr el desarrollo de un sistema estable.
3. Implementar la solución utilizando la tecnología propuesta para mejorar el proceso de gestión de seminarios científicos en el CEIGE.
4. Validar la solución propuesta y realizar un análisis de los resultados para corroborar el cumplimiento de los objetivos propuestos en la investigación.

Se define como **idea a defender**: El desarrollo de un sistema informático para la gestión de los seminarios científicos contribuirá a satisfacer las necesidades de la dirección del CEIGE.

El trabajo está estructurado en 3 capítulos de los que se realiza una breve descripción a continuación:

Capítulo 1: Contiene la fundamentación teórica del trabajo, donde se describen aspectos importantes sobre el proceso de gestión de superación profesional. Se realiza un estudio del estado del arte relativo a los sistemas informáticos que gestionan información referente a investigación-superación. Además se efectúa un breve análisis de los lenguajes de programación, herramientas, tecnologías y metodologías a usar en el desarrollo del sistema.

Capítulo 2: Se identifican y describen los requisitos funcionales, no funcionales y los casos de uso del sistema. Se expone el análisis y diseño de la solución propuesta lo cual incluye la definición del modelo de clases del análisis, modelo de clases del diseño al igual que los diagramas de colaboración y el modelo entidad relación definidos por la metodología RUP.

Capítulo 3: Muestra cómo va estar estructurada la implementación del sistema. Además se explica cuáles son las pruebas que se realizan al sistema, contiene los diagramas de componentes, de despliegue y se valida la solución.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se realiza un estudio sobre los elementos que fundamentan la base teórica conceptual de la gestión de seminarios científicos para el seguimiento y control de las maestrías del CEIGE. Se incluye una breve descripción de los lenguajes, las herramientas, tecnologías y metodología, que son utilizadas para todo el proceso de modelado de la aplicación.

1.2 Conceptos básicos relacionados con el dominio de la investigación

Se define postgrado como el ciclo de estudios especializados que se cursa tras la graduación o licenciatura e incluye maestrías y doctorado. El postgrado suele fomentar las actividades de investigación y la actualización de los contenidos universitarios, además de la especialización del conocimiento (U.net 2013). La maestría proporciona a los graduados universitarios dominio profundo de los métodos de investigación, amplia cultura científica y conocimientos avanzados para desarrollar habilidades y modo de actuación en el trabajo que ejercerá, docente o de investigación (MES 2007). En la formación postgraduada la investigación tiene un papel fundamental, es la búsqueda o inquisición de un hecho desconocido o algo que se quiere investigar. Se caracteriza por la creatividad del acto, la innovación de ideas, los métodos rigurosos utilizados, la validación o juicio crítico. Una de las formas de dar a conocer sus resultados es a través de seminarios científicos. Estos son presentaciones de índole científica que se realiza con el fin de difundir conocimientos o desarrollar investigaciones, en la que participan alrededor de 50 personas, donde los principales protagonistas son los maestrantes, los cuales discuten sus trabajos frente a un tribunal y un grupo de participantes (Chamorra 2008). Para que los seminarios tengan la calidad requerida, se hace necesario planificarlos de forma tal que se organicen en grupos, informando hora, fecha y lugar donde se realizará. Además de registrar de cada uno de los participantes el nombre con sus apellidos y su condición (Tribunal, Ponente o Participante).

1.3 Estado del arte

1.3.1 Sistemas de información científica

Los Sistemas de Información Científica (SIC) también conocidos por su denominación inglesa de Current Research Information Systems (CRIS), son la solución que se está adoptando en la mayor parte de universidades para tener controlada la actividad de sus investigadores (Cabezas 2011).

El desarrollo de las Tecnologías de la Información y las Comunicaciones ha resultado fundamental para analizar y tratar el volumen de conocimiento científico que se genera en la actualidad. Además ha permitido que se desarrollen sistemas de información científica que sirvan de soporte a la gestión de la investigación en materia de política científica y tecnológica. Entre los sistemas de información científica visitados se pueden citar el Sistema de Información Científica de Andalucía, SICA, este permite recoger, registrar, procesar, almacenar, recuperar y visualizar información relacionada con las actividades y resultados producidos por los investigadores en sus centros de desarrollo o en colaboración con otras instituciones nacionales o internacionales. Además utiliza la información que se sistematiza para promover e incrementar el intercambio y comunicación entre científicos e investigadores en el ámbito regional, nacional o internacional. Otros de los sistemas fueron el de la Vicerrectoría de Política Científica e Investigación-Universidad de Granada y el Sistema sobre Investigación de la Universidad de Sevilla (SISIUS), donde se visualiza la información de diferentes actividades relacionadas con los grupos de investigación y resultados alcanzados. Además se constató que las conexiones que se realizan son de forma segura a través del protocolo "https".

El estudio realizado de estos sistemas permitió ver la forma en que se registra, almacena, recuperan los datos relacionados a la gestión de información científica, sirviendo como referencia y brindando elementos importantes a la presente investigación.

1.3.2 Sistemas de gestión de información de seminarios científicos

Luego de un estudio realizado para encontrar un sistema que tuviera funcionalidades relacionadas con la gestión de la información de los seminarios científicos y aportara elementos a esta investigación, no se encontró en la literatura consultada un sistema que cumpliera con estas características. Al no encontrar un

sistema se estudiaron diferentes sitios web de algunas universidades tales como: El Centro de Estudios de Migraciones Internacionales (CEMI) (<http://www.uh.cu/centros/cemi/eventos-cientificos/>) perteneciente a la Universidad de la Habana. En la sección dedicada a eventos científicos se observó que sólo se brinda información acerca de las convocatorias de los eventos de este centro y su programa. El programa consiste en la información sobre la hora, el lugar donde se realizará, la fecha en que se efectuará y las ponencias a presentarse. El principal problema que presenta este sitio es que es completamente de carácter informativo. Su objetivo se basa fundamentalmente en la promoción de los eventos que se realizan, así como brindar información para la participación de los personas en los mismos, por tanto no aporta elementos de importancia a esta investigación.

Otros de los sitios visitados fueron el del Instituto de Salud Global de Barcelona y el de la Ciudad Universitaria José Antonio Echeverría. En estos se pudo constatar que al igual que el anterior, tienen un carácter puramente informativo, por lo que no ofrece elementos a la investigación. Después de haber estudiado los sitios de las universidades antes mencionadas, no se quiso dejar pasar por alto a la UCI, donde se encontró en el sitio de la facultad 3 una sección dedicada a brindar información sobre los eventos de investigación y postgrado. Este tampoco aporta elementos a la investigación, ya que tiene un carácter informativo.

1.4 La producción científica de equipos de proyectos informáticos

La producción científica de un equipo es un indicador que permite medir la madurez científica y el desarrollo que este ha ido adquiriendo. Son los artículos científicos y las publicaciones que logren los equipos de proyectos informáticos, reflejo del equilibrio entre resultados en Investigación + Desarrollo y producción científica. Por tal razón estimular el aprendizaje organizacional, la comunicación, socialización y la capacidad de investigación deben ser constantes que promuevan el trabajo en los proyectos y encaminen su desempeño hacia el desarrollo de todos los miembros del equipo (Vásquez 2009). Llevar una adecuada gestión de los seminarios científicos e investigaciones que realizan los profesionales que se encuentran en el proceso de maestría, ayudaría a la dirección del CEIGE llevar el seguimiento y control de la superación de los mismos. Además permite obtener indicadores como cantidad de cursos cursados, cantidad de créditos no lectivos por presentaciones en eventos, una mayor organización y observar el avance de los profesionales.

1.4.1 Impacto de la Gestión del Conocimiento en los resultados científicos de los equipos de proyectos informáticos

Cuando en una empresa se habla de hacer Gestión del Conocimiento (GC), se está haciendo referencia a implementar procesos, técnicas y herramientas. Combinados ofrecen una cultura organizacional que ayuda a minimizar los actuales problemas de comunicación y gestión informacional existentes. El análisis que se propone a continuación, tiene como objetivo estudiar la relación que existe entre la GC y procesos tan medulares para una organización como lo es el desarrollo de equipos de proyecto (Pons 2012).

La GC integra en un proceso único las áreas de creatividad e innovación, el conocimiento intelectual, la investigación y desarrollo, la contabilidad del capital con tecnologías radicalmente nuevas. De igual forma se añade que entre sus principales objetivos se encuentran: contribuir a comprender cómo conseguir organizaciones más competitivas y adaptables. Además de permitir crear procesos y mecanismos de gestión que aceleren los procesos de *aprendizaje, la creación, adaptación y difusión del conocimiento*, tanto en la organización como entre la organización y su entorno (Alvarado 2012). La GC fomenta la creación y difusión de una cultura organizacional y un entorno de colaboración que favorezca dichas acciones. Así mismo crea las condiciones necesarias para que la información fluya en forma idónea sobre la base de un soporte tecnológico que facilite y agilice el flujo de la información y el conocimiento (Pons 2012). En toda organización o proyecto se vuelve necesario crear un ambiente de comunicación basado en la motivación. Ello permitirá actuar sobre el comportamiento de los miembros del equipo, a partir de poder comprometer a todas las partes con la solución. En la medida en que lo anterior se logre, se comenzará a percibir mayor responsabilidad con el trabajo, aumentará el nivel de desempeño, pero sobre todo, se logrará un mayor desarrollo del equipo.

Tener un lugar donde guardar y consultar toda la información referente a los seminarios científicos es de gran importancia para una institución donde se realicen este tipo de eventos. En ocasiones estas empresas desarrollan aplicaciones que ayudan a dicha gestión. Una de las vías más factibles lo constituyen los marcos de trabajo, que entre sus ventajas tienen que promueven una estructura estándar para el uso de las buenas prácticas de la programación.

1.5 Marco de trabajo Sauxe

Un marco de trabajo proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Este facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. El desarrollo de la solución a petición del cliente se realizará utilizando el marco de trabajo Sauxe. Este contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo (Gómez Baryolo 2011). Sauxe es la unión de Zend, la librería Extjs que muestra al usuario una interfaz amigable y en la capa de acceso a datos el Lenguaje de Consulta de Datos (DQL) que implementa Doctrine.

Zend Framework 1.7

Es un marco de trabajo para el desarrollo de aplicaciones y servicios web con PHP. Brinda soluciones para construir sitios web modernos, robustos y seguros. Está formado por una serie de métodos estáticos y componentes que usarán estos métodos. Como características fundamentales de Zend se tienen:

- Trabaja con Modelo Vista Controlador (MVC).
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Completa documentación y pruebas de alta calidad (Pupo 2010).

ExtJS 2.2

Es una librería Javascript utilizada para la creación y el desarrollo de aplicaciones web dinámicas. Su potencia radica en la rica colección de componentes para el diseño de interfaces de usuarios del lado del cliente haciendo uso extensivo de Ajax. ExtJS permite que con pocas líneas de código sea posible realizar interfaces amigables para los usuarios. La comunicación asíncrona permite el intercambio de información con el servidor sin necesidad de pedirle una acción al usuario. De alto rendimiento, es compatible con la mayoría de los navegadores (Internet Explorer, Netscape, Opera, Mozilla Firefox) que permiten crear páginas e interfaces web dinámicas.

Doctrine 1.0

Doctrine es un potente y completo sistema Mapeador de Objeto Relacional (ORM) para PHP 5.2.3 + que incorpora una Capa de Abstracción a Base de Datos (DBL). Una de sus principales características es la opción de escribir las consultas de base de datos en un objeto con una propiedad orientada al dialecto SQL llamada Doctrine Query Language (DQL) (Wage 2009). Entre otros elementos se tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

1.6 Descripción de los lenguajes y herramientas utilizadas

PHP 5.4

Es un lenguaje script de propósito general, adecuado para desarrollos web que puede ser embebido en HTML. Contienen en gran parte sintaxis de lenguajes conocidos como Java, Perl, C. Para su funcionamiento necesita tener un servidor web con el intérprete de PHP. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación de páginas dinámicas (php.net 2012). Con PHP se puede realizar el procesamiento de información en formularios, foros de discusión entre otras. Dentro de las ventajas que posee se encuentran:

- Posee una comunidad muy grande de desarrolladores y existe abundante documentación, tutoriales, ejemplos de código, foros etc.
- PHP es fácil de aprender comparado con otros lenguajes de programación.
- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos, se puede descargar desde www.php.net.
- Es un lenguaje que incluye soporte para la programación orientada a objeto.
- La versión 5 de PHP está diseñada para el soporte de características de programación como herencia, métodos y atributos públicos o privados, clases y métodos abstractos, constructores e interfaces.

CSS

Las Hojas de Estilo en Cascada (Cascading Style Sheets conocido como CSS, por sus siglas en inglés) son un lenguaje formal usado para definir la presentación de un documento estructurado, escrito en HTML o XML (Lenguaje de Marcas Extensible). Este estilo propone las ventajas:

- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea) (López 2010).

Gestor de Bases de Datos

Un sistema gestor de base de datos (SGBD) se puede definir como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Las características de un Sistema Gestor de Base de Datos SGBD son:

- Abstracción de la información. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- Independencia. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Redundancia mínima. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- Consistencia. En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

- Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- Integridad. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- Respaldo y recuperación. Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- Control de la concurrencia. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias (Hechavarría 2009).

PostgreSQL 8.4

El gestor de base de datos PostgreSQL es un sistema muy potente de código abierto. Soporta gran parte del estándar SQL y en algunos aspectos, está diseñado para que sea extensible por los usuarios. Debido a la licencia liberal BSD (Berkeley Software Distribution por sus siglas en inglés). PostgreSQL puede ser usado, modificado y distribuido por cualquiera de forma gratuita para cualquier propósito, ya sea privado, comercial o académico (PostgreSQL 1996-2013). Maneja grandes cantidades de información, soporta alta concurrencia, o sea, que a la misma vez que un usuario escribe en una tabla otros pueden estar leyendo de ella sin dificultad. Además se pueden definir consultas anidadas, vistas, crear funciones, no sólo en el lenguaje natural SQL. Es multiplataforma, presenta una estabilidad muy alta, gran seguridad de los datos, soporta la réplica y procedimientos almacenados.

Algunas de las ventajas que posee este gestor son:

- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Incluye herencia entre tablas, por lo que se incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, siendo capaz de soportar gran cantidad de peticiones simultáneas de manera correcta.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel.

Servidor Web Apache 2.2

Apache es un programa de servidor web de código libre, robusto, cuya implementación se ha realizado y se sigue realizando de forma colaborativa, con prestaciones, características y funcionalidades equivalentes a las de cualquier servidor comercial (cibernetia 2010). Algunas de sus características:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierta.
- Trabaja con PHP y otros lenguajes de script. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor (Reyes 2009).

NetBeans IDE 7.2

El Entorno de Desarrollo Integrado Netbeans es una herramienta libre y gratuita sin restricciones de uso, licenciada bajo la CDDL (Common Development and Distribution License, por sus siglas en inglés), que en español sería Desarrollo Común y Licencia de Distribución. La herramienta es utilizada por los programadores para escribir, compilar, depurar y ejecutar programas. Permite el desarrollo de aplicaciones web. Está escrito en Java pero puede soportar cualquier otro lenguaje de programación. NetBeans está apoyado por una comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación, así como una gran cantidad de plugins (NetBeans 2012).

Características:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de datos).
- Administración de ventanas (Glick 2009).

Visual Paradigm 8.0

Visual Paradigm para UML es una herramienta CASE (por sus siglas en inglés Computer Aided Software Engineering) en español Ingeniería de Software Asistida por Ordenador que soporta el ciclo de vida completo del desarrollo de software. Permite representar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Alvarez 2011). Entre sus principales características están:

- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Entorno de creación de diagramas para UML 2.0.
- Disponibilidad de integrarse en los principales IDEs.
- Dispone de múltiples versiones dependiendo de la necesidad.
- Posee un modelo y un código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad en múltiples plataformas.

Las herramientas CASE tienen como objetivos:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo, costo de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto (Verdecia 2010).

Mozilla Firefox 17.0

Se ha concebido para que sea simple y funcional. Agrupa todas las funciones necesarias para navegar, con la posibilidad de añadir extensiones, es decir, programas que le dan nuevas funcionalidades a Mozilla Firefox. Se puede navegar, realizar búsquedas en internet, enviar archivos y leer el contenido de las páginas web de manera simple y eficaz (Villeneuve 2005).

1.7 Arquitectura de software

En esta tesis se asume el concepto de arquitectura propuesto por Verdecia, el cuál plantea que la arquitectura de software, denominada además como arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. Establece los fundamentos para que analistas, diseñadores, programadores, entre otros, trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel (Verdecia 2010). Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requisitos de un sistema. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto.

1.7.1 Patrón Arquitectónico Modelo Vista Controlador (MVC)

El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.

- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (Potencier 2007).

Al sistema estar desarrollado con el marco de trabajo Sauxe, la arquitectura que implementa es Modelo Vista Controlador. En la capa de presentación usa ExtJS en su versión 2.2, en el controlador Zend en la versión 1.7 y en el modelo Doctrine versión 1.0.

1.8 Metodologías de desarrollo de software

Todo desarrollo de software es riesgoso y difícil de controlar, de ahí la importancia de contar con una metodología que indique paso a paso las actividades que deben desarrollarse, las personas involucradas en la realización de las mismas, así como el papel que desempeñan estas en aras de lograr un proceso de software exitoso. Las metodologías de desarrollo de software se pueden clasificar en dos grandes grupos:

Metodologías Ligeras/Ágiles: metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando.

Metodologías Pesadas/Tradicionales: metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Requieren de una amplia documentación.

A continuación se describen dos de las metodologías más utilizadas en el desarrollo de software.

1.8.1 Metodología RUP

La metodología RUP sus siglas vienen de Rational Unified Process (Proceso Unificado de Desarrollo) propone un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Esta metodología divide en 4 fases el desarrollo del software:

- Inicio: El objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: Se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
- Construcción: La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.
- Transición: El objetivo es poner el producto en manos de los usuarios finales.

El Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental (Larman 2001).

1. Iterativo e Incremental.

RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

2. Dirigido por los casos de uso.

Los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba. Basándose en los casos de usos los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de usos. De este modo los casos de uso no solo inician el proceso de desarrollo sino que le proporcionan un hilo conductor.

3. Centrado en la arquitectura.

La arquitectura determina la línea base y los elementos de software estructurales a partir de los elementos de la arquitectura de negocio. Interviene en la gestión de cambios y diseña la evolución e integración del producto. La arquitectura orienta las prioridades en la producción y resuelve las necesidades tecnológicas y de soporte para el desarrollo. Asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. Constituye además, una propuesta de proceso para el desarrollo de software orientado a objeto, utilizando UML (del inglés Unified Model Language), para describir todo el proceso, basándose en componentes. Este lenguaje es estándar, con él se puede visualizar, especificar, construir y documentar los artefactos de un sistema (Rumbaugh 2000).

1.8.2 Extreme Programming (XP)

La Programación Extrema (XP por sus siglas en inglés) se ha convertido en la metodología ágil más popular. XP consiste en una programación rápida o extrema, cuyo rasgo característico es tener como parte del equipo de desarrollo al cliente que es el usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Esta se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. El ciclo de vida ideal de XP está compuesto por seis fases que permiten todo el proceso de desarrollo en un tiempo record, estas fases son las siguientes:

- Exploración.
- Planificación de la entrega.

- Iteraciones.
- Producción.
- Mantenimiento.
- Cierre del Proyecto (Acuña 2009).

Luego de haber realizado un estudio entre estas dos metodologías XP y RUP, se hace necesario analizar el modelo de desarrollo del CEIGE debido a que esta es la metodología que el centro usa para el desarrollo de sus productos.

El modelo de desarrollo de Software cuenta de dos fases:

Inicio o Estudio preliminar: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto a un alto nivel, la evaluación de la factibilidad del proyecto y el registro de este. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo, y decidir si se ejecuta o no el proyecto.

Desarrollo: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se refinan los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. En esta fase se ejecutan las disciplinas Modelado de negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas internas y Pruebas de liberación (Obregón 2012).

Este modelo está basado en procesos y no genera ningún artefacto relacionado con los diagramas de casos de uso. Para esta investigación se hace necesario que queden plasmados los actores que intervienen en el sistema, así como los casos de usos y la descripción detallada de estos.

Metodología seleccionada para el desarrollo del trabajo

Para el desarrollo de este trabajo se decidió utilizar una de las metodologías pesadas debido a que están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo. El proceso es mucho más controlado, con diferentes políticas y normas, mientras que las ágiles están basadas en

heurísticas que provienen de prácticas de producción de código y procesos menos controlados y con poca documentación. El modelo de desarrollo de software que propone el CEIGE a pesar de estar bien documentado en cada una de sus fases, al estar orientado a procesos no genera ningún artefacto que tenga que ver con casos de uso.

La metodología seleccionada fue RUP, donde el ciclo de vida está dirigido por casos de uso, describiendo lo que se espera del software y reflejando lo que los usuarios requieren y desean, representándose a través de los requisitos. Los casos de uso guían todo el proceso de desarrollo, donde la arquitectura del software es fundamental y muestra una visión común del sistema. Además permite que esté bien documentado todo el proceso.

Conclusión

En este capítulo se realizó un profundo estudio sobre las tendencias actuales del desarrollo profesional; a través de conceptos se fundamenta los principios de este trabajo. También se visitaron diferentes sitios web para analizar la forma en que gestionan la información de los seminarios científicos, pero no se encontró ninguno que aportara elementos concretos para la presente investigación. Además se realizó un análisis completo de la tecnología a utilizar a lo largo del desarrollo del sistema propuesto, en este caso Sauxe. Las herramientas seleccionadas para el desarrollo del sistema fueron: como entorno de desarrollo integrado el Netbeans, como gestor de bases de datos PostgreSQL, como servidor web Apache y el Visual Paradigm para el modelado, debido a que soporta el ciclo de vida completo del desarrollo de software.

Capítulo 2: Análisis y Diseño de la solución

2.1 Introducción

En este capítulo se presenta la propuesta del software a desarrollar, especificando los requisitos funcionales y no funcionales. También se muestran las interfaces de usuarios y diagramas clases del sistema propuesto. Además de los artefactos que se generan en esta etapa, aplicando la metodología RUP y haciendo uso de la herramienta Visual Paradigm.

2.2 Descripción de las características generales del sistema

Se pretende desarrollar un sistema que permita la gestión de los seminarios científicos del CEIGE para lograr llevar un seguimiento de los profesionales que están en el proceso de maestría. Este sistema permitirá la gestión de los profesionales por grupos, la planificación para su presentación en el seminario, así como las tareas que se le orientan, conocer de los cursos que van concluyendo pertenecientes a un programa de maestría determinado, gestionar los créditos no lectivos, entre otras funcionalidades. El sistema contará con dos tipos de roles principales, los cuales tendrán acceso solo a las funcionalidades que les correspondan.

2.2.1 Artefactos generados

Los artefactos son documentos que se generan durante las diferentes fases del desarrollo de un proyecto de software. En esta investigación se consideran de suma importancia que se generen los que a continuación se relacionan para cada una de las fases que propone la metodología RUP:

| RUP | | | UML |
|--------|--------------------------|-----------------------|-------------------------|
| Fase | Actividad | Entregable | |
| Inicio | Modelamiento del negocio | Modelo conceptual | |
| U – | Requisitos | Modelo de caso de uso | Diagrama de caso de uso |

| | | | |
|--------------|----------------------|----------------------------|---------------------------|
| | Análisis y diseño | Modelo del análisis | Diagrama de secuencia |
| | | Diseño de clases | Diagrama de clases |
| | | Diseño de la base de datos | Diagrama Entidad Relación |
| Construcción | Implementación | Modelo de despliegue | Diagrama de despliegue |
| | | Modelo de componentes | Diagrama de componentes |
| | Prueba | Diseños de Casos de prueba | |
| Transición | Prueba de aceptación | Acta de aceptación | |

Tabla 1: Artefactos generados en esta investigación

Con el objetivo de entender el funcionamiento del proceso de gestión de los seminarios científicos, a continuación se establecen relaciones entre los conceptos más importantes.

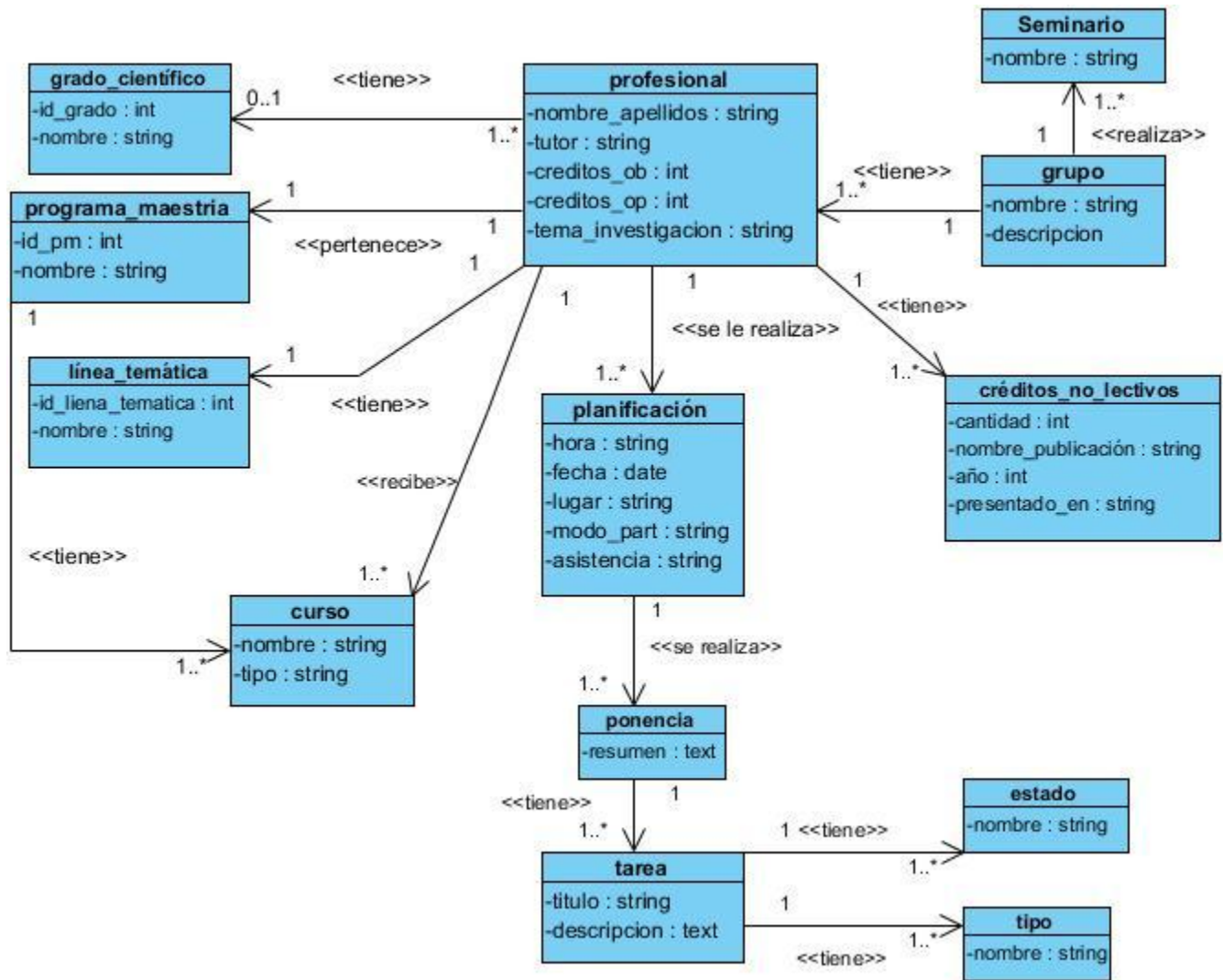


Ilustración 1: Modelo conceptual

2.3 Captura y especificación de requisitos funcionales

El proceso de captura de requisitos suele ser uno de los pasos más difíciles en la ingeniería de software, debido a que una mala captura de ellos conllevaría al desarrollo de un sistema que no responde a las necesidades del cliente. La técnica utilizada en esta investigación fue la entrevista, luego de varias reuniones con el cliente se comprendió y se obtuvieron las funcionalidades que el sistema debía cumplir. En cuanto a la definición de los requisitos se usó la técnica de los escenarios que

consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos. Se realizó de forma textual encaminada hacia una representación gráfica en forma de diagramas de flujo. El análisis de los escenarios de una forma u otra ofrecieron información importante sobre las necesidades funcionales del sistema.

RF 1: Gestionar grupos de profesionales

- RF1.1 Adicionar los datos de un nuevo grupo.
- RF 1.2 Modificar los datos de un grupo existente.
- RF 1.3 Eliminar los datos de un grupo existente.
- RF 1.4 Listar los nombres de los grupos existentes.

RF 2: Gestionar perfil del profesional

- RF 2.1 Adicionar los datos del perfil de un nuevo profesional.
- RF 2.2 Modificar los datos del perfil de un profesional existente.
- RF 2.3 Eliminar los datos del perfil de un profesional existente.
- RF 2.4 Mostrar los datos del perfil de todos los profesionales existentes.

RF 3: Gestionar la planificación de las presentaciones a los seminarios científicos

- RF 3.1 Adicionar los datos de una nueva planificación.
- RF 3.2 Modificar los datos de una planificación existente.
- RF 3.3 Eliminar los datos de una planificación existente.
- RF 3.4 Mostrar los datos de las planificaciones existentes.

RF 4: Gestionar las tareas generadas en una presentación

- RF 4.1 Adicionar los datos de una nueva tarea.
- RF 4.2 Modificar los datos de una tarea existente.

RF 4.3 Brindar un reporte de las tareas cumplidas, incumplidas, pendientes y rechazadas.

RF 4.4 Mostrar los datos de todas las tareas existentes.

RF 4.5 Eliminar los datos de una tarea existente.

RF 5: Gestionar cursos de los profesionales

RF 5.1 Adicionar los datos de un nuevo curso.

RF 5.2 Modificar los datos de un curso existente.

RF 5.3 Brindar un reporte de la cantidad total de cursos, la cantidad de cursos optativos y la cantidad de cursos obligatorios cursados por un profesional.

RF 5.4 Mostrar los datos de todos los curso existentes.

RF 5.5 Eliminar los datos de un curso existente.

RF 6: Gestionar los créditos no lectivos de los profesionales

RF 6.1 Adicionar los datos de un nuevo crédito no lectivo.

RF 6.2 Modificar los datos de los crédito no lectivo existentes.

RF 6.3 Eliminar los datos de crédito no lectivo existentes.

RF 6.4 Mostrar los datos de créditos no lectivos existentes.

RF 7: Gestionar grado científico

RF 7.1. Adicionar un grado científico.

RF 7.2 Modificar un grado científico existente.

RF 7.3 Mostrar los grados científicos existentes.

RF 7.4 Eliminar un grado científico existente.

RF 8 Gestionar línea temática

RF 8.1 Adicionar una línea temática.

RF 8.2 Modificar una línea temática existente.

RF 8.3 Mostrar las líneas temáticas existentes.

RF 8.4 Eliminar una línea temática existente.

RF 9 Gestionar programa de maestría

RF 9.1 Adicionar un programa de maestría.

RF 9.2 Modificar un programa de maestría.

RF 9.3 Mostrar los programas de maestrías existentes.

RF 9.4 Eliminar un programa de maestría existente.

RF 10 Gestionar tipo de tarea

RF 10.1 Adicionar un tipo de tarea.

RF 10.2 Modificar un tipo de tarea.

RF 10.3 Mostrar los tipos de tareas existentes.

RF 10.4 Eliminar un tipo de tarea existente.

RF 11 Gestionar estado de tarea

RF 11.1 Adicionar un estado de tarea.

RF 11.2 Modificar un estado de tarea.

RF 11.3 Mostrar los estados de tareas existentes.

RF 11.4 Eliminar un estado de tarea existente.

RF 12 Gestionar modo de participación

RF 12.1 Adicionar un modo de participación.

RF 12.2 Modificar un modo de participación.

RF 12.3 Mostrar los modos de participación existentes.

RF 12.4 Eliminar un modo de participación existente.

RF 13 Mostrar los cursos de un profesional

RF 14 Mostrar las planificaciones de un profesional

RF 15 Mostrar las tareas de un profesional

2.4 Descripción de los requisitos no funcionales

Requisitos de Software

Para el cliente:

- Navegador Mozilla Firefox 2.2 – 21.0.
- Sistema operativo Windows o Linux.

Para el servidor:

- Sistema operativo Linux en la distribución Ubuntu.
- Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible. Este debe estar configurado con la extensión pgsql, pdo, pdo_pgsql, soap incluida.

- Un servidor de base de datos PostgreSQL 8.3 - 9.4.

Requisitos de Hardware

Para el servidor:

- Requerimientos mínimos: Procesador Pentium IV a 2.50GHz de velocidad de procesamiento y 1Gb de memoria RAM.
- Al menos 10Gb de espacio libre en disco duro.
- Tarjeta de red.

Para el cliente:

- Requisitos mínimos: Procesador Pentium III a 1GHz con 256Mb de memoria RAM.
- Tarjeta de red.

Requisitos de Seguridad

- Se establecen diferentes niveles de acceso (roles) para los diferentes usuarios garantizando que la información se muestre las funcionalidades para las cuales está autorizado.

Requisitos de apariencia o interfaz externa

- Se realizará un diseño amigable, fácil y simple de usar por el usuario.
- Empleo de colores claros e interfaces sencillas.

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. En esta investigación se usó la técnica de prototipos de interfaz para validar los requisitos. [Ver anexo 1](#): Prototipos de interfaz de usuario.

2.5 Definición de los actores del sistema

| Actores del sistema | Descripción |
|---------------------|-------------|
|---------------------|-------------|

| | |
|--|--|
| Administrador | Usuario con privilegios para gestionar toda la información referente a los seminarios científicos, la gestión de usuarios y la asignación de permisos. |
| Subdirector de Investigación y Postgrado | Usuario que gestiona la planificación de los seminarios científicos, los cursos optativos y obligatorios de los profesionales, la creación de los grupos de seminarios y los créditos no lectivos. |
| Profesional | Usuario que accede al sistema para ver la información de las planificaciones, cursos, y tareas que les han sido asignadas. |

Tabla 2: Actores del sistema

2.5.1 Definición de los casos de uso (CU) del sistema

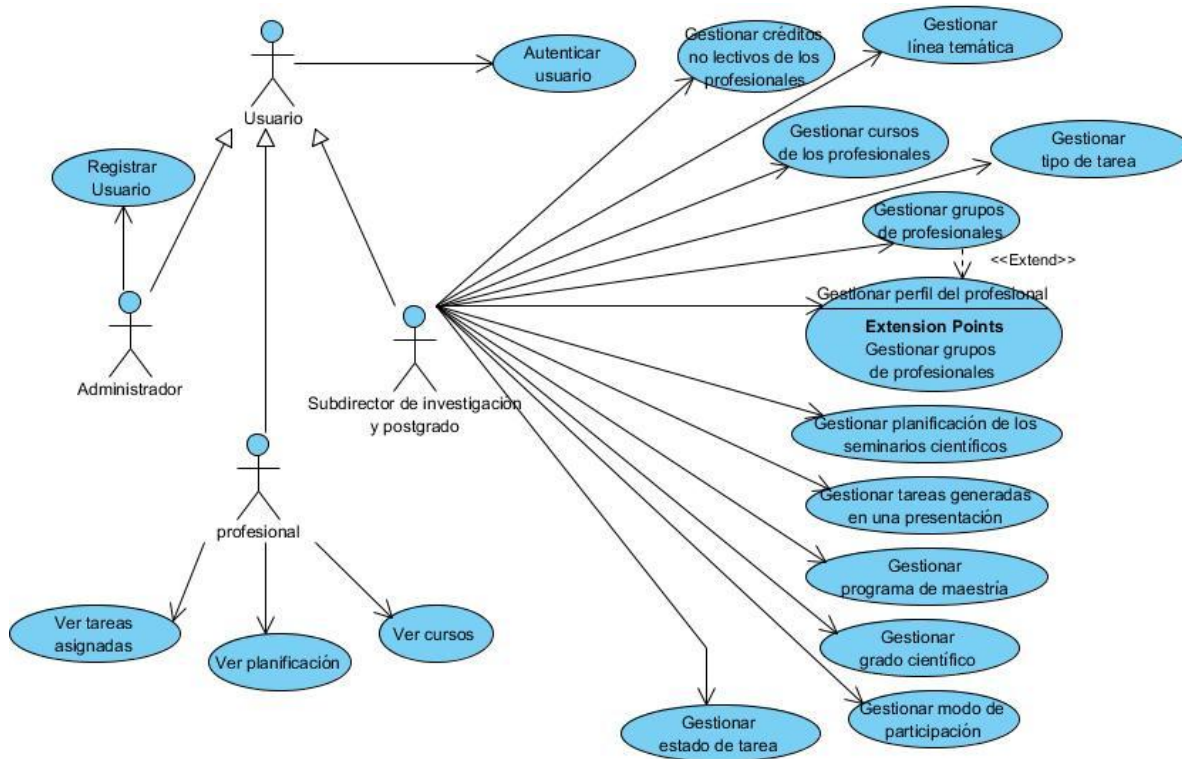


Ilustración 2 Diagrama de casos de uso del sistema

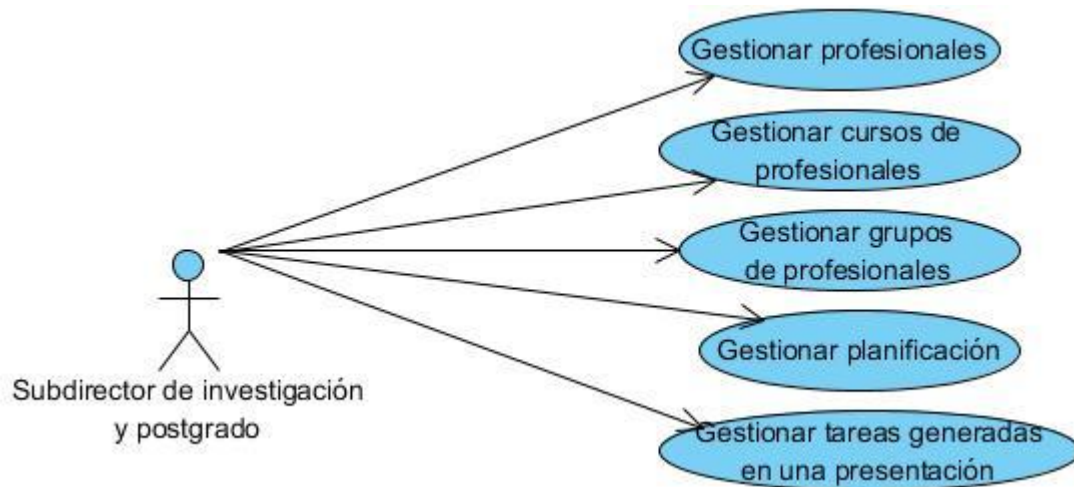


Ilustración 3: Principales casos de uso

Los casos de uso del sistema son procesos que responden a las funcionalidades definidas en los requisitos funcionales. En las tablas siguientes se realiza una breve descripción de los casos de uso del sistema.

| | | |
|---|--|--|
| Caso de uso | Gestionar perfil del profesional | |
| Actor: | Subdirector de investigación y postgrado | |
| Resumen: | El caso de uso inicia cuando el actor selecciona un grupo. Se cargan los profesionales pertenecientes a ese grupo. Después el actor selecciona la opción que desee, ya sea adicionar, modificar o eliminar un profesional. | |
| Precondiciones: | Debe existir un grupo creado | |
| Flujo Normal de Eventos | | |
| Sección “Flujo Normal de Eventos” | | |
| Acción del Actor | Respuesta del Sistema | |
| 1- Accede al sistema y selecciona la opción Grupo ->Gestionar grupo | 2- Muestra la funcionalidad gestionar grupo de profesionales | |
| 3- Selecciona un grupo | 4- Muestra un listado de los profesionales pertenecientes al grupo seleccionado y las | |

| | |
|---|---|
| | <p>funcionalidades:</p> <ul style="list-style-type: none"> - Adicionar Profesional - Modificar Profesional - Eliminar Profesional |
| 5- Selecciona una de las funcionalidades | <p>6- Si el actor selecciona:</p> <ul style="list-style-type: none"> - Adicionar Profesional ver sección "Adicionar profesional" - Modificar Profesional sección alterno "Modificar profesional" - Eliminar Profesional sección alterno "Eliminar profesional" |
| Sección: "Adicionar profesional" | |
| Flujo normal de eventos | |
| Acción del actor | Respuesta del sistema |
| <p>7- Introduce los siguientes datos para el profesional:</p> <ul style="list-style-type: none"> - Nombre y apellidos - Tutor(es) - Créditos OP - Créditos OB - Grado Científico - Tema de investigación - Resumen del tema de investigación - Línea temática - Programa de maestría - Usuario <p>8- Presiona el botón Aceptar</p> | <p>9- Verifica que al adicionar los datos del profesional no existan campos vacíos y datos incorrectos.</p> <p>10- Adiciona el profesional y muestra el mensaje: "Se ha adicionado el profesional correctamente".</p> |

Prototipo de interfaz

Flujos Alternos

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 8.1- Presiona el botón Cancelar . Ir al flujo normal 4 | 10.1 Cierra el formulario. Ir al flujo normal 5 |

Sección: "Modificar profesional"

Flujo normal de eventos

| Acción del actor | Respuesta del sistema |
|---|---|
| | 7- Muestra los profesionales que han sido creados |
| 8- Selecciona el profesional al que desea modificar los datos y escoge la opción Modificar Profesional. | 9- Muestra los siguientes datos para modificar el profesional: <ul style="list-style-type: none"> - Nombre y apellidos - Tutor(es) - Créditos OP - Créditos OB - Grado Científico - Tema de investigación - Resumen del tema de investigación - Línea temática - Programa de maestría - Grupo |

| | |
|--|--|
| | |
| 10- Modifica los datos que desea y escoge la opción aceptar. | 11- El sistema verifica que al modificar los datos de del profesional no existan campos vacíos o datos incorrectos. Muestra el mensaje: "Se ha modificado el profesional correctamente". |

Prototipo de interfaz

Flujos Alternos

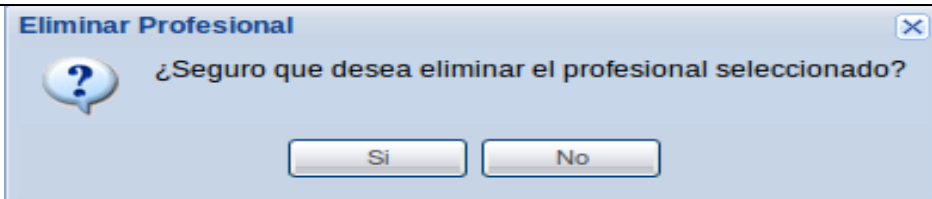
| Acción del Actor | Respuesta del Sistema |
|--|----------------------------|
| 10.1- Presiona el botón Cancelar . Ir al flujo normal 5 | 11-1 Cierra el formulario. |

Sección: "Eliminar profesional"

Flujo normal de eventos

| Acción del actor | Respuesta del sistema |
|--|---|
| | 7- Muestra los profesionales que han sido creados. |
| 8- Selecciona el profesional que desea eliminar. | 9- El sistema muestra el siguiente mensaje de confirmación: "¿Seguro que desea eliminar el profesional seleccionado"? |

Prototipo de interfaz



| Flujos Alternos | |
|---|--|
| Acción del Actor | Respuesta del Sistema |
| 8.1- Presiona el botón Sí. | 9.1 Elimina el profesional y muestra el mensaje "Se ha eliminado el profesional seleccionado" Cierra el mensaje. Ir al flujo normal 5. |
| 8.2 Presiona el botón No. Ir al flujo normal 5. | |

Tabla 3: Descripción del caso de uso Gestionar perfil del profesional

| Caso de uso | Gestionar grupos de profesionales |
|---|---|
| Actor: | Subdirector de investigación y postgrado |
| Resumen: | Este caso de uso consiste en organizar los profesionales por grupos, para que a la hora de planificarle la asistencia a un seminario exista una mejor organización. |
| Precondiciones: | Debe existir un grupo creado |
| Flujo Normal de Eventos | |
| Sección "Flujo Normal de Eventos" | |
| Acción del Actor | Respuesta del Sistema |
| 1 Accede al sistema y selecciona la opción Grupo ->Gestionar grupo. | 2. Muestra la funcionalidad gestionar grupo de profesionales. |
| 3. Selecciona un grupo. | 4. Muestra un listado de los profesionales pertenecientes al grupo seleccionado y las funcionalidades: <ul style="list-style-type: none"> • Adicionar Grupo • Modificar Grupo • Eliminar Grupo |

| | |
|--|---|
| <p>5. Selecciona una de las funcionalidades.</p> | <p>6. Si el actor selecciona:</p> <ul style="list-style-type: none"> - Adicionar Grupo ver sección "Adicionar grupo" - Modificar Grupo sección alterno "Modificar grupo" - Eliminar Grupo sección alterno "Eliminar grupo" |
|--|---|

Sección: "Adicionar grupo"

Flujo normal de eventos

| Acción del actor | Respuesta del sistema |
|--|---|
| <p>7. Introduce los siguientes datos para el grupo:</p> <ul style="list-style-type: none"> • Nombre • Descripción <p>8. Presiona el botón Aceptar</p> | <p>9. Verifica que al adicionar los datos del grupo el campo nombre no se encuentre vacío , ya que la descripción es opcional</p> <p>10. Adiciona el grupo y muestra el mensaje: "Se ha adicionado el grupo correctamente".</p> |

Prototipos de interfaz



Flujos Alternos

| Acción del Actor | Respuesta del Sistema |
|--|---|
| <p>8.1- Presiona el botón Cancelar. Ir al flujo normal 4.</p> | <p>10.1 Cierra el formulario. Ir al flujo normal 5.</p> |

Sección: "Modificar profesional"

| Flujo normal de eventos | |
|---|---|
| Acción del actor | Respuesta del sistema |
| | 7- Muestra los grupos que han sido creados |
| 8- Selecciona el grupo al que desea modificar los datos y escoge la opción Modificar Grupo. | 9- Muestra los siguientes datos para modificar el grupo: - Nombre - Descripción |
| 12- Modifica los datos que desea y oprime el botón aceptar. | 13- Verifica que al modificar los datos del grupo no existan campos vacíos o datos incorrectos. |
| 14- Presiona el botón Aceptar . | 15- Muestra el mensaje: "Se ha modificado el grupo correctamente". |

Prototipo de interfaz

Flujos Alternos

| Acción del Actor | Respuesta del Sistema |
|---|----------------------------|
| 14.1- Presiona el botón Cancelar . Ir al flujo normal 5. | 15-1 Cierra el formulario. |

Sección: "Eliminar grupo"

Flujo normal de eventos

| Acción del actor | Respuesta del sistema |
|------------------|---|
| | 7- Muestra los grupos que han sido creados. |

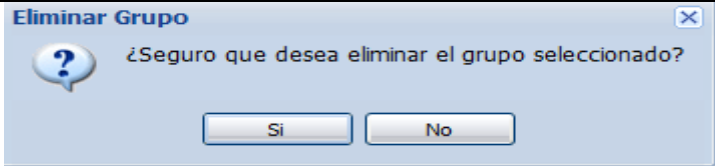
| | |
|--|--|
| 8- Selecciona el grupo que desea eliminar. | 9- El sistema muestra el siguiente mensaje de confirmación: “¿Seguro que desea eliminar el grupo seleccionado”? |
| Prototipo de interfaz | |
|  | |
| Flujos Alternos | |
| Acción del Actor | Respuesta del Sistema |
| 8.1- Presiona el botón Sí. | 9.1 Elimina el grupo con los profesionales que tiene y muestra el mensaje “Se ha eliminado el grupo seleccionado”. |
| 8.2 Presiona el botón No. Ir al flujo normal 5. | |

Tabla 4: Descripción del caso de uso Gestionar grupos de profesionales

[Ver anexo 2:](#) Donde se muestran otras de las descripciones de los casos de usos del sistema.

2.6 Diagramas de clases del análisis

El modelo de análisis ofrece una especificación más precisa de los requisitos, puede ser considerado como una primera aproximación al modelo de diseño que se describe utilizando el lenguaje de los desarrolladores y estructura los requisitos de modo que facilita su comprensión, preparación, modificación y mantenimiento.

La realización de casos de uso del análisis describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos en interacción. Para ello RUP propone que para cada caso de uso:

- Se construya un diagrama de clases que muestre las clases de análisis participantes.
- Se construya alguno de los tipos de diagrama de interacción para representar el flujo en términos de objetos del análisis.

El diagrama de clase del análisis es un artefacto donde se adjuntan los diagramas de clases a las realizaciones de casos de uso, mostrando sus clases participantes y sus relaciones.

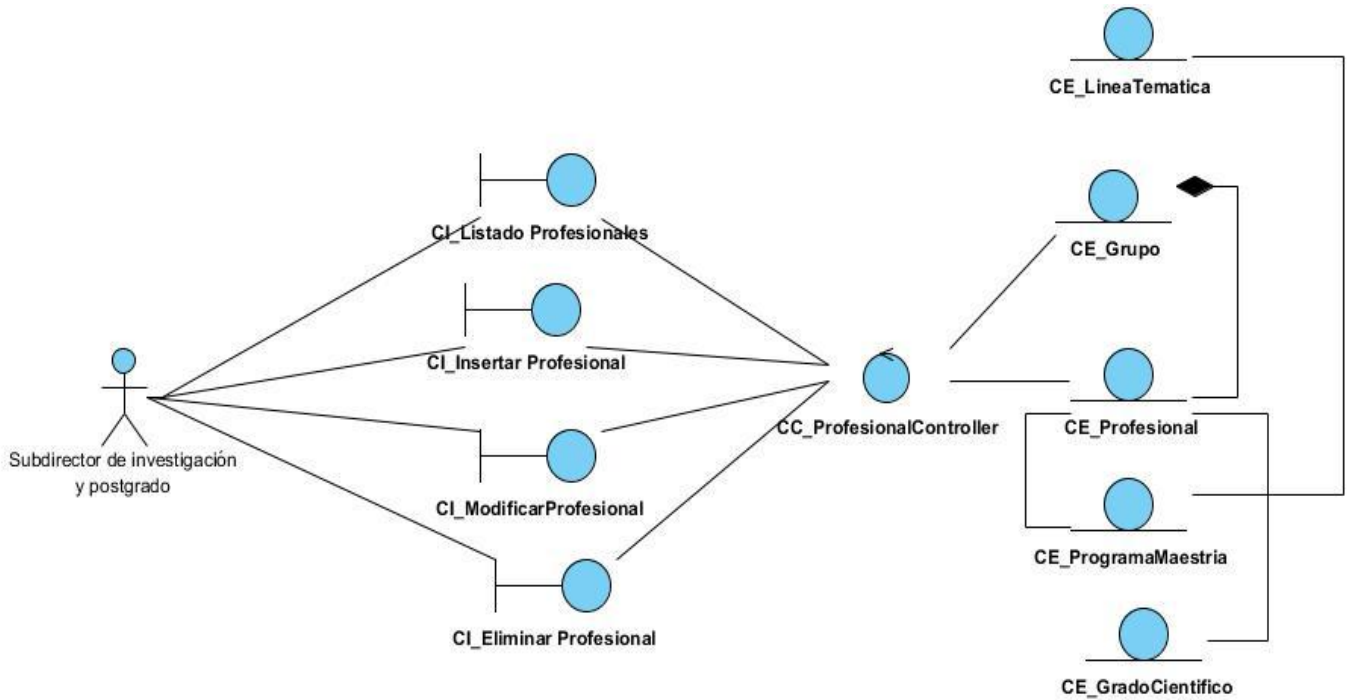


Ilustración 4: Diagrama de clases del análisis gestionar profesional

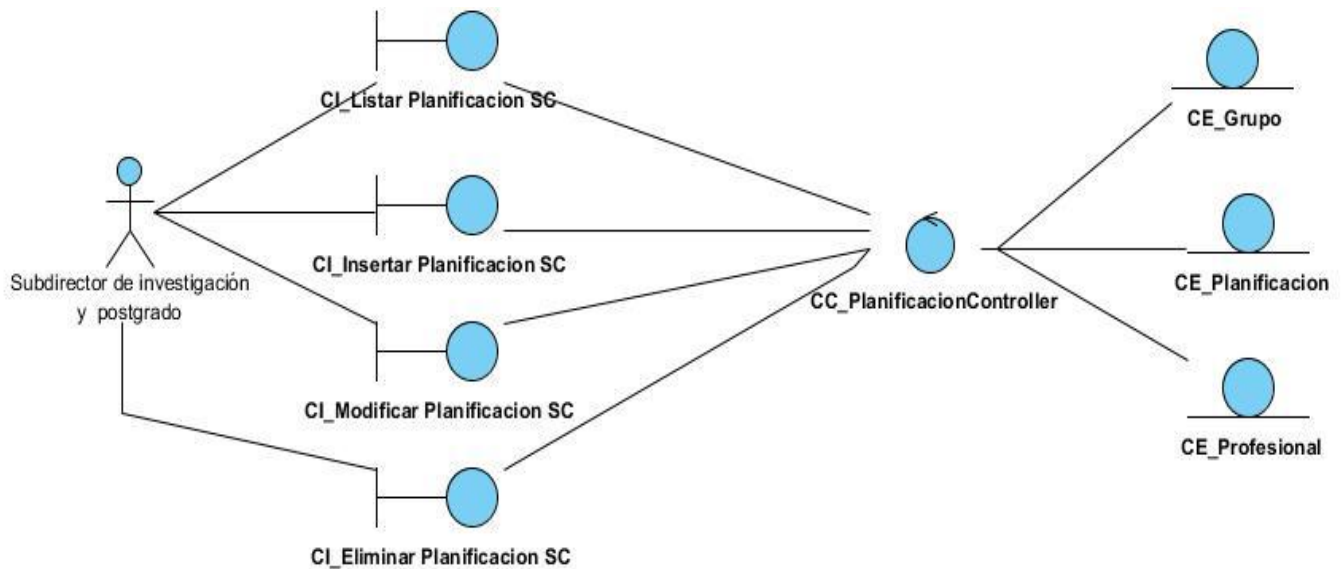


Ilustración 5: Diagrama de clases del análisis gestionar planificación

[Ver anexo 3](#): Donde se muestran otros diagramas de clases del análisis del sistema.

2.7 Diagramas de clases del diseño

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones del sistema con sus relaciones estructurales dando así una vista del diseño estático. En el caso de las aplicaciones web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. Los diagramas de clases del diseño son utilizados con el objetivo de representar las relaciones que existen entre los distintos tipos de clases.

Contiene la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.
- Dependencias.

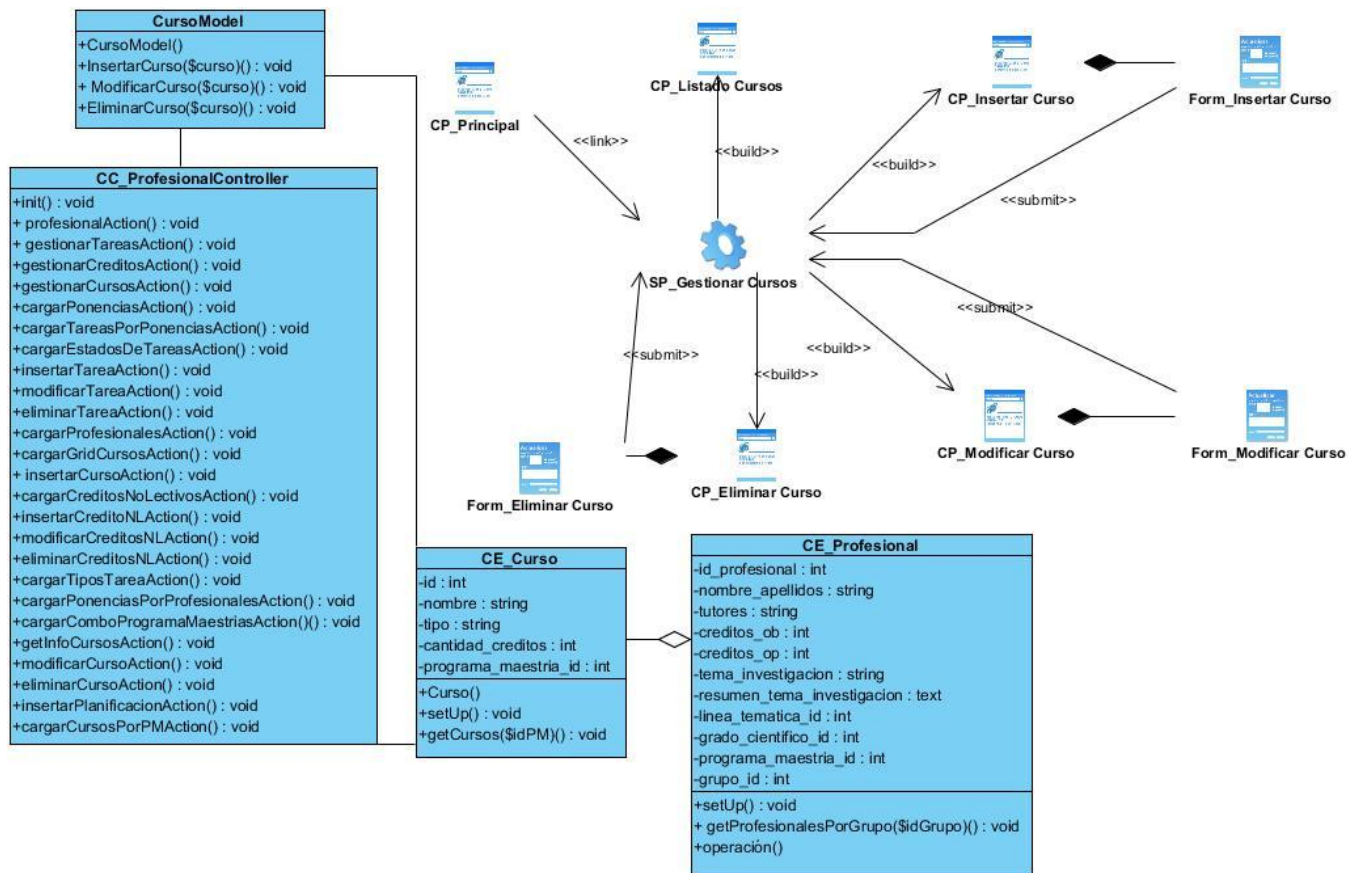


Ilustración 6: Diagrama de clases del diseño Gestionar cursos de los profesionales

[Ver anexo 4:](#) Donde se muestran otros de los diagramas de clases del diseño del sistema.

El modelo de diseño describe la realización física de los casos de uso centrándose en los requisitos funcionales, no funcionales y otras restricciones que tienen impacto en el sistema. Además sirve de abstracción de la implementación y como entrada fundamental de sus actividades.

2.7.1 Patrones del diseño

Un patrón es una solución probada a un problema común, el cual es muy recomendado usar en el diseño de aplicaciones de software. Describen la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Los mismos se aplican durante la fase de diseño detallado para solucionar problemas de claridad del

diseño. Con su uso, se pretende establecer un lenguaje común entre los programadores, contribuir a la reutilización, ahorrar tiempo en la implementación y obtener un producto con calidad (Tedeschi 2012).

Los patrones GRASP (General Responsibility Assignment Software Patterns) es sus siglas llevadas al español corresponde a los Patrones Generales de Software de Asignación de Responsabilidades, describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman 2001). A continuación se describen los patrones básicos de asignación de responsabilidades utilizados en el desarrollo del sistema:

Experto: Un marco de trabajo como SAUXE está organizado de forma tal que cada clase realice las funcionalidades que les toca. Ejemplo de ello son las clases modelos **ProfesionalModel**, **GrupoModel**, **TareaModel** y **CursosModel**, que son las encargadas de insertar, modificar y eliminar la información correspondiente a las sus respectivas clases entidades.

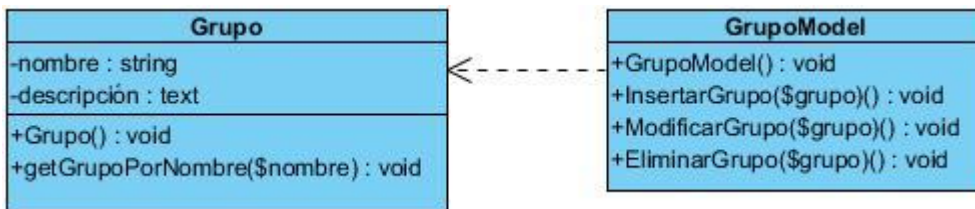


Ilustración 7: Ejemplo patrón experto

Creador: La clase controladora es la encargada de instanciar en cada uno de los objetos del modelo que será utilizado para almacenar la información para presentarla a la vista posteriormente. Ejemplo de ello es **insertarTareaAction**, **insertarCursoAction**, **insertarCreditoNLAction**, todos pertenecientes a la clase **ProfesionalController**.

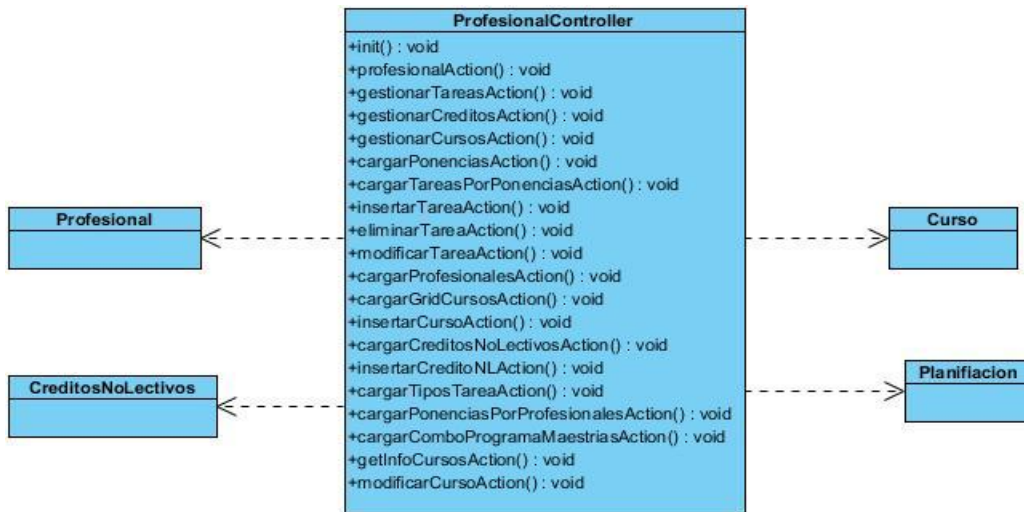


Ilustración 8: Ejemplo de patrón creador

Controlador: La clase **GrupoController** es un ejemplo de la aplicación de este patrón, la misma tiene a cargo la responsabilidad de manejar las funcionalidades dentro de un grupo.

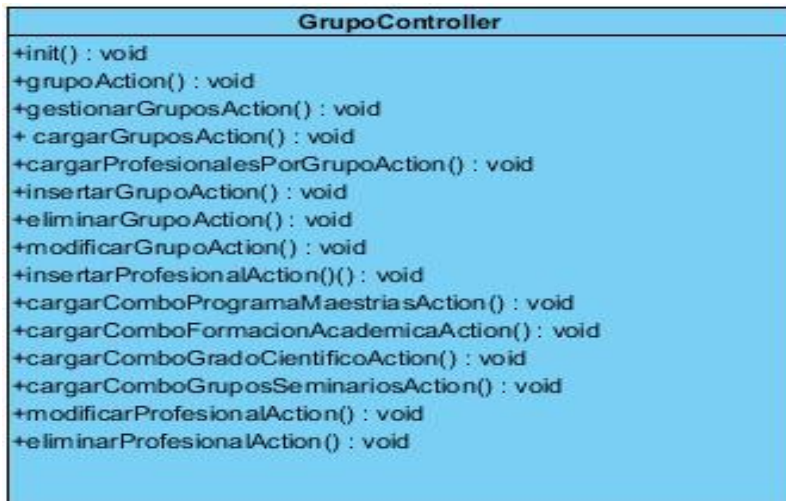


Ilustración 9: Ejemplo de patrón controlador

2.8 Diagramas de secuencia.

Los diagramas de secuencia se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el escenario que ilustra un comportamiento determinado. En el contexto de las clases se describe la forma en que grupos de objetos colaboran para proveer un comportamiento. Mientras que un diagrama de casos de uso presenta una visión externa del sistema, la funcionalidad de dichos casos de uso se recoge como un flujo de eventos utilizando para ello interacciones entre sociedades de objetos (Alvarez 2011).

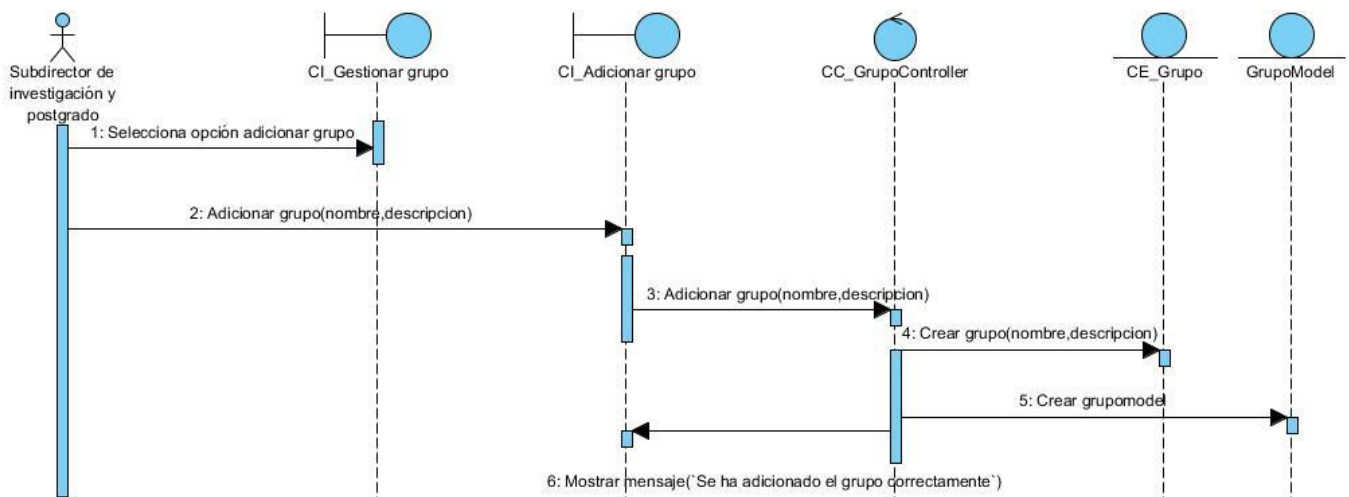


Ilustración 10: Diagrama de secuencia adicionar grupo

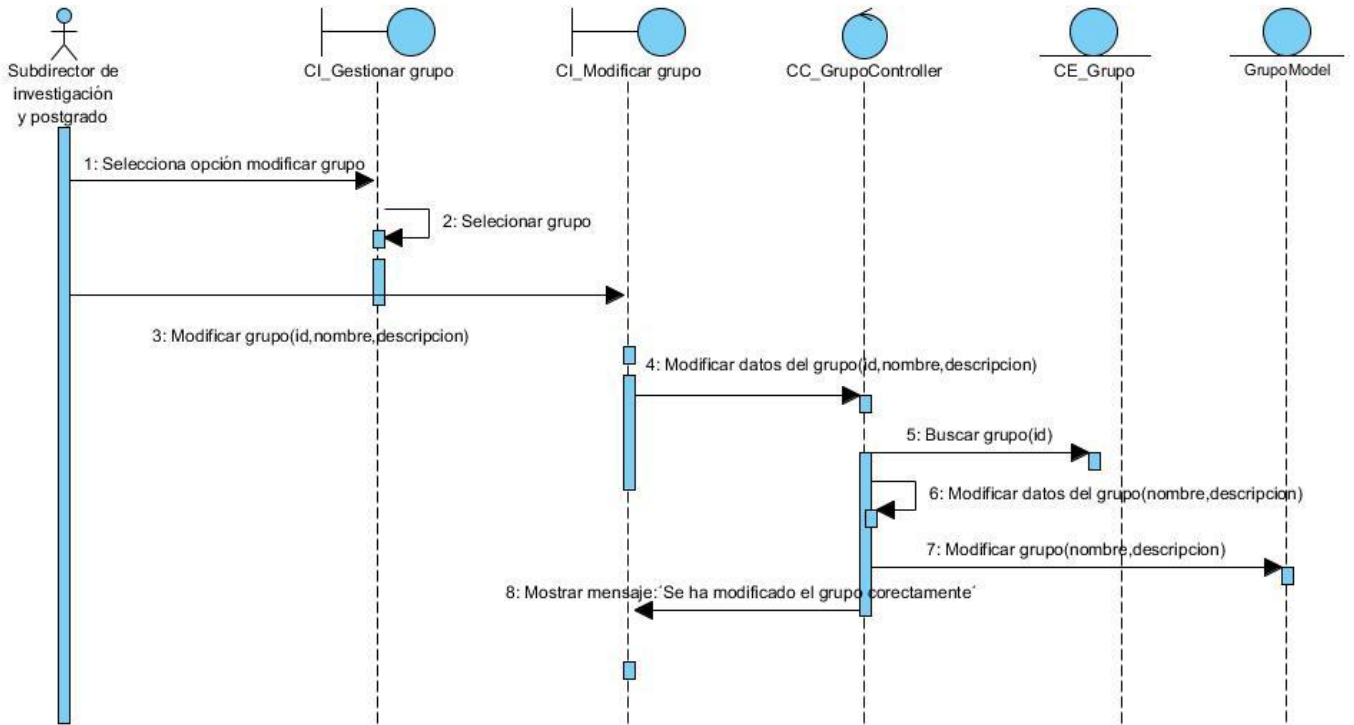


Ilustración 11: Diagrama de secuencia modificar grupo

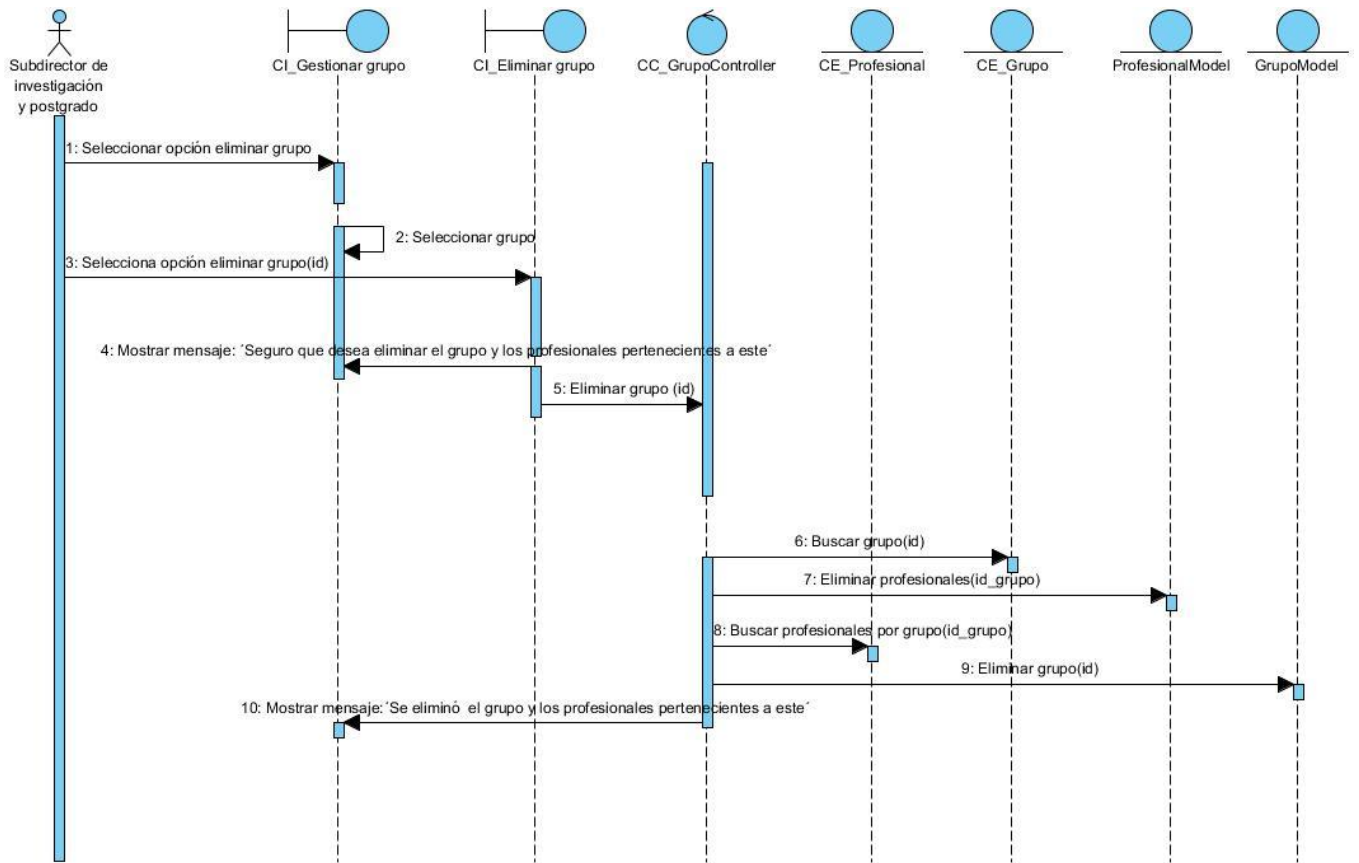


Ilustración 12: Diagrama de secuencia eliminar grupo

[Ver anexo 5](#): Donde se muestran otros de los diagramas de secuencia del sistema.

2.9 Diagrama Entidad Relación

El modelo entidad – relación representa la estructura que debe tener la base de datos para almacenar la información del sistema. El modelo propuesto cuenta con 14 tablas, de ellas 8 son de datos y 6 son nomencladores.

A continuación se muestra la descripción de las clases:

- Profesional: Es la encargada de guardar los datos de los profesionales, la misma se relaciona con las tablas, programa_maestria, línea_tematica, grado_cientifico y grupo, es la tabla de mayor importancia del sistema ya que es el centro del mismo.
- Planificación: En ella se guardan los datos referentes a las planificaciones de los profesionales organizados por grupos, la misma se relaciona con las tablas tarea y grupo.
- Planificacion_Profesional: Surge de una relación de muchos a muchos entre las tablas profesional y planificación, en ella se guardan los datos referentes a la asociación de un profesional a una planificación.
- Tarea: En esta tabla se guarda la información referente a las tareas que se generan en una presentación, se relaciona con la tabla planificación.
- Grupo: En esta tabla se guarda la información referente a los grupos, se relaciona con la tabla profesional.
- Creditos_No_Lectivos: En esta tabla se guarda la información referente a los créditos no lectivos, se relaciona con la tabla profesional.
- Curso: En esta tabla se guarda la información referente a los cursos, se relaciona con la tabla programa_maestria y profesional_curso.
- Profesional_Curso: Surge de una relación de muchos a muchos entre las tablas profesional y curso, en ella se guardan los datos referentes a la asociación de un profesional a un curso.
- Programa_Maestria: Es un nomenclador estático que guarda el nombre de los programas de maestrías de un profesional, la misma se relaciona con la tabla profesional.
- Linea_Tematica: Es un nomenclador estático que guarda el nombre de las líneas temáticas de un profesional, la misma se relaciona con la tabla profesional.
- Grado_Cientifico: Es un nomenclador estático que guarda los grados científicos de un profesional, la misma se relaciona con la tabla profesional.
- Tipo_Tarea: Es un nomenclador estático que guarda los tipos de tareas, la misma se relaciona con la tabla tarea.
- Estado_Tarea: Es un nomenclador estático que guarda los estados de las tareas, la misma se relaciona con la tabla tarea.
- Condición: Es un nomenclador estático que guarda la condición con la que un profesional puede ir a un seminarios científico, la misma se relaciona con la tabla planificacion_profesional.

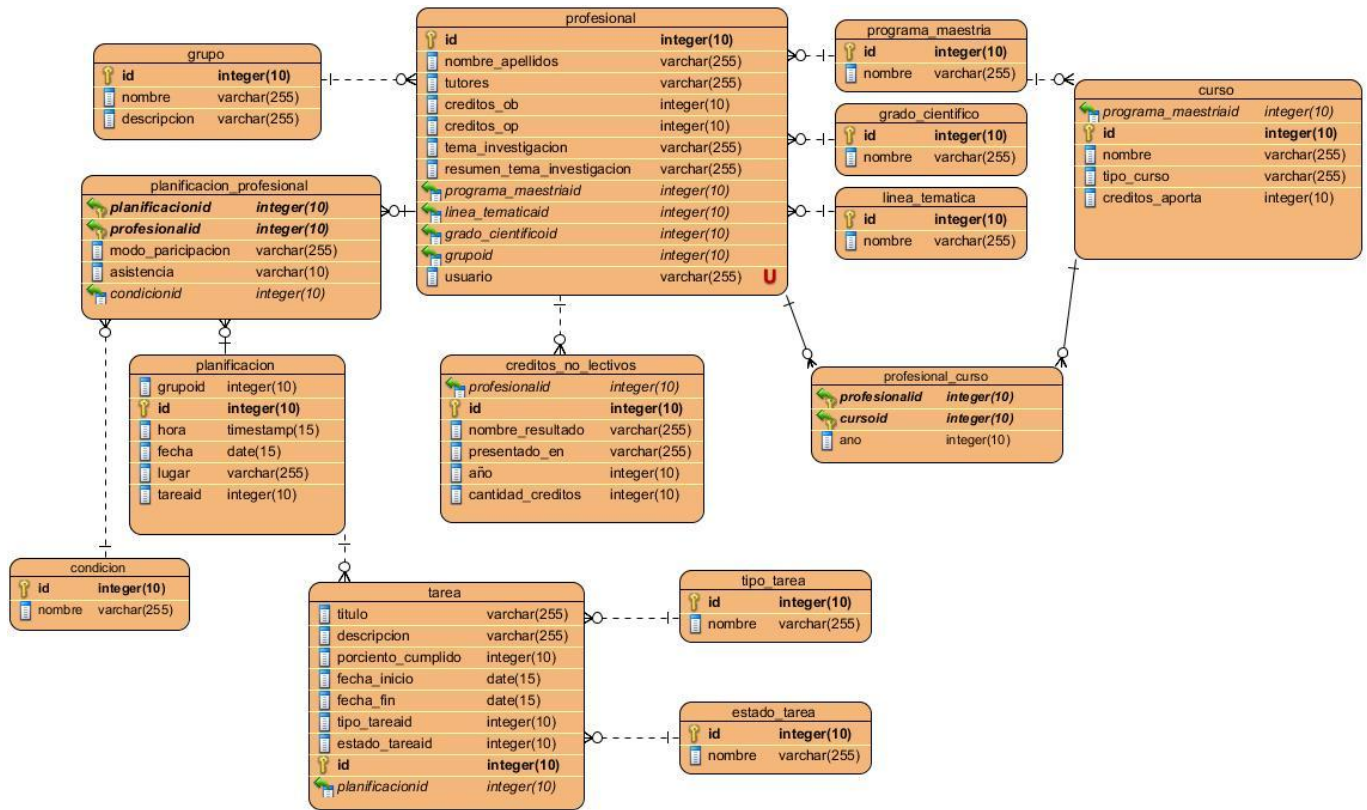


Ilustración 13: Diagrama Entidad Relación

2.10 Validación del diseño propuesto

La métrica concebida como instrumento para evaluar la calidad del diseño y su relación con los atributos de calidad fue Tamaño Operacional de Clase (TOC). Los atributos que evalúa son: Responsabilidad, Complejidad de implementación y la Reutilización, existiendo una relación directa con los dos primeros e inversa con el último.

| Atributo que afecta | Modo en que lo afecta |
|---------------------|--|
| Responsabilidad | Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase. |

| | |
|-------------------------------|--|
| Complejidad de implementación | Un aumento del TOC implica un aumento de la complejidad de implementación de la clase. |
| Reutilización | Un aumento del TOC implica una disminución en el grado de reutilización de la clase. |

Tabla 5: Relación atributo con el tamaño operacional de clases

Se aplicó a un total de 33 clases, obteniendo un promedio de procedimientos de 8.26. Los procedimientos son la cantidad de métodos que tenga implementado una clase.

| No | Clase | Cantidad de Procedimientos | Responsabilidad | Complejidad | Reutilización |
|----|-------------------------------|----------------------------|-----------------|-------------|---------------|
| 1 | GrupoController | 16 | Media | Media | Media |
| 2 | ProfesionalController | 25 | Alta | Alta | Baja |
| 3 | PlanificacionController | 14 | Media | Media | Media |
| 4 | InformacionController | 9 | Media | Media | Media |
| 5 | NomencladorController | 37 | Alta | Alta | Baja |
| 6 | ProfesionalModel | 4 | Baja | Baja | Alta |
| 7 | PlanificacionModel | 4 | Baja | Baja | Alta |
| 8 | GrupoModel | 4 | Baja | Baja | Alta |
| 9 | CreditosNoLectivosModel | 4 | Baja | Baja | Alta |
| 10 | CursoModelModel | 4 | Baja | Baja | Alta |
| 11 | EstadoTareaModel | 4 | Baja | Baja | Alta |
| 12 | GradoCientificoModel | 4 | Baja | Baja | Alta |
| 14 | ProfesionalCursoModel | 4 | Baja | Baja | Alta |
| 15 | ProfesionalPlanificacionModel | 4 | Baja | Baja | Alta |
| 16 | ProgramaMaestriaModel | 4 | Baja | Baja | Alta |

Tabla 6: Valor de los atributos por clases

Para determinar las categorías de los atributos se tuvo en cuenta los siguientes criterios:

| Atributos | Categoría | Criterio |
|----------------------------|-----------|-----------------------|
| Responsabilidad | Baja | < =Prom. |
| | Media | Entre Prom. y 2* Pom. |
| | Alta | > 2* Prom. |
| Complejidad implementación | Baja | < =Prom. |
| | Media | Entre Prom. y 2* Pom. |
| | Alta | > 2* Prom. |
| Reutilización | Baja | > 2*Prom. |
| | Media | Entre Prom. y 2* Pom. |
| | Alta | <= Prom. |

Ilustración 14: Criterio de evaluación de los atributos



Ilustración 15: Gráficos de los resultados de la métrica TOC

Mediante la aplicación de esta métrica se puede observar de manera general que las clases del diseño propuesto presentan un 85% de reponsabilidad y complejidad baja y un 82% de reutilización alta.

Conclusiones parciales

Los prototipos de interfaz de usuario ayudaron a entender y especificar las relaciones entre actores y el sistema durante la captura de requisitos. Además de permitir desarrollar una interfaz gráfica mejor y comprender los casos de uso. La aplicación de la métrica tamaño operacional de la clase arrojó valores bajos de complejidad y responsabilidad, y altos de reutilización, lo cual evidencia la calidad del diseño propuesto. Este capítulo permitió sentar las bases para poder realizar la implementación de la propuesta de solución.

Capítulo 3: Implementación y prueba

3.1 Introducción

En este capítulo quedan contenidos todos los detalles referentes a la implementación del sistema. Se detallan los diagramas de componentes y de despliegue, permitiendo obtener una visión más clara de la implementación. Además se muestran ejemplos de la nomenclatura usada para las clases y las funciones haciendo uso de las notaciones Pascal Casing y Camel Casing.

3.2 Diagrama de despliegue

Los diagramas de despliegues describen la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software. Describen la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución y los componentes que residen en ellos. El elemento principal del diagrama son los nodos (Scribd 2008).

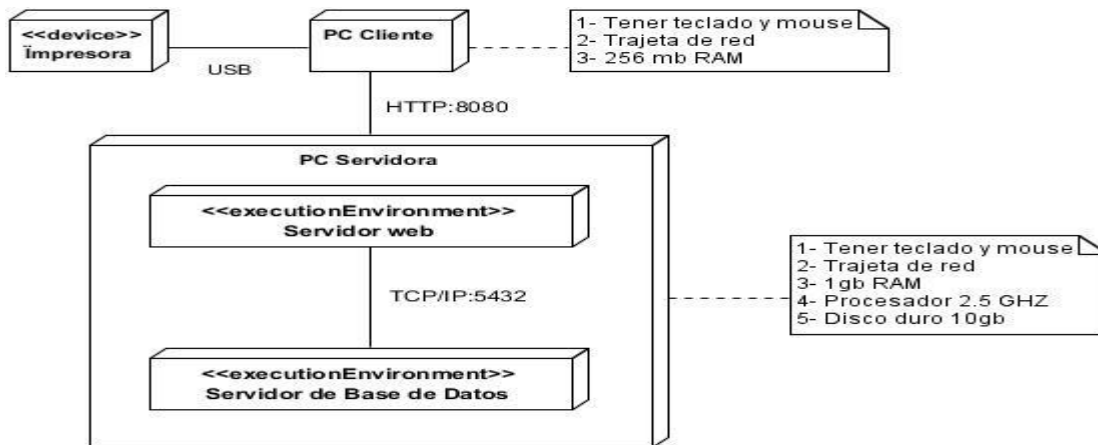


Ilustración 16: Diagrama de despliegue

PC Cliente

Este nodo representa a las máquinas desde donde podrá conectarse cualquier usuario del sitio con los permisos correspondientes a su rol. Requiere de la conexión de dispositivos externos para la impresión de reportes.

Impresora

Dispositivo externo que permitirá la impresión de los reportes desde las máquinas clientes. La forma de conexión de estos dispositivos con la máquina cliente estará acorde a la disponibilidad que posea la dirección de CEIGE.

PC Servidora

Es la computadora servidora que contendrá la base de datos y donde se mantendrá corriendo el servidor web. El nodo será accesible para las máquinas clientes que requerirán del sistema, y este, para responder a los pedidos, accederá a la base de datos.

3.3 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes y bibliotecas cargadas dinámicamente. Muestra como el sistema está dividido en componentes y las dependencias entre ellos. Provee una vista arquitectónica de alto nivel del sistema y ayuda a los desarrolladores a visualizar el camino de la implementación (Scribd 2008).

Más adelante se muestran algunos de los componentes del sistema. Para mayor comprensión estos se agruparon por paquetes. El flujo de datos entre los elementos que componen el patrón arquitectónico MVC es el siguiente:

- Vista – Controlador: Los datos recogidos en un formulario son enviados al Controlador haciendo uso del protocolo de comunicación HTTP a través del método —post para ser procesados y los resultados son enviados por el controlador a la vista en un JSON a través del método —echo.
- Controlador – Modelo: El Controlador toma los datos recibidos desde la vista, instancia una determinada clase del modelo y llama a uno de sus métodos, pasándole como parámetros los datos recibidos.
- Modelo: El Modelo es el encargado de insertar, recuperar, modificar o eliminar los datos almacenados en la base de datos

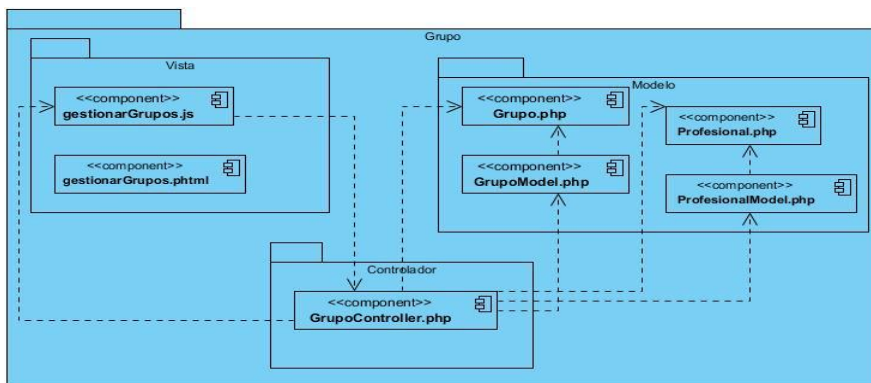


Ilustración 17: Diagrama de componentes CU gestionar grupo

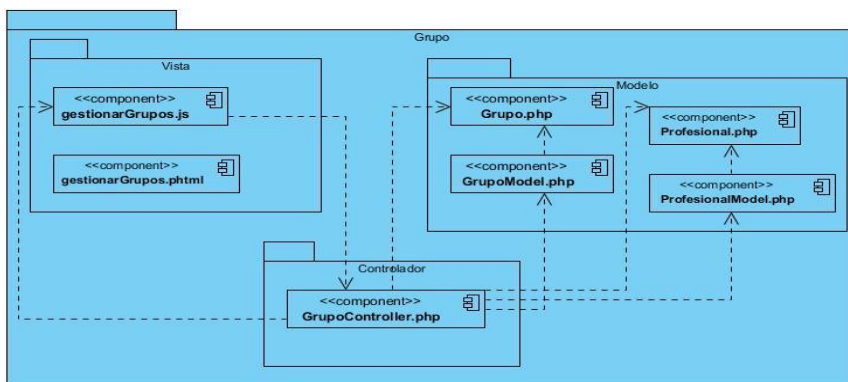


Ilustración 18: Diagrama de componentes CU gestionar cursos de profesionales

3.4 Nomenclatura de las clases

Para la nomenclatura de las clases se utilizó la notación Pascal Casing. En este caso, los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula.

Clases controladoras: Las clases controladoras después del nombre llevan la palabra: “Controller”.

- Controllers (Controladoras)

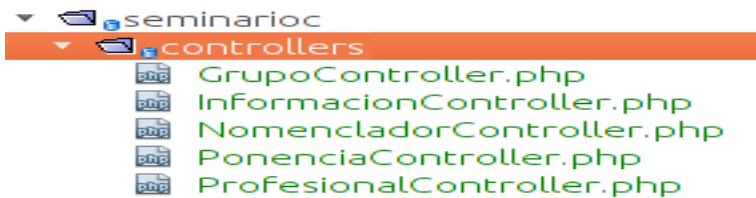


Ilustración 19: Ejemplo de estructura de las clases controladoras

Clases de los modelos:

- Business (Negocio)

Las clases que se encuentran dentro de Business después del nombre llevan la palabra: “Model”.

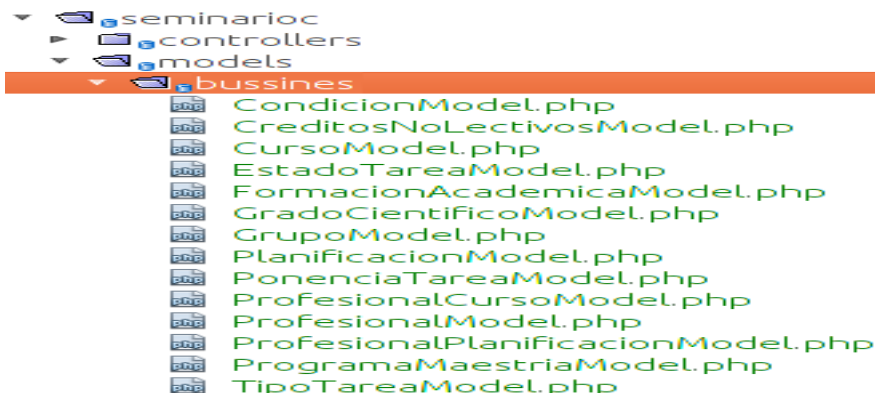


Ilustración 20: Ejemplo de estructura de las clases del negocio

- Domain (Dominio)

Las clases que se encuentran dentro de Domain el nombre que reciben es el de la tabla en la Base de Datos.

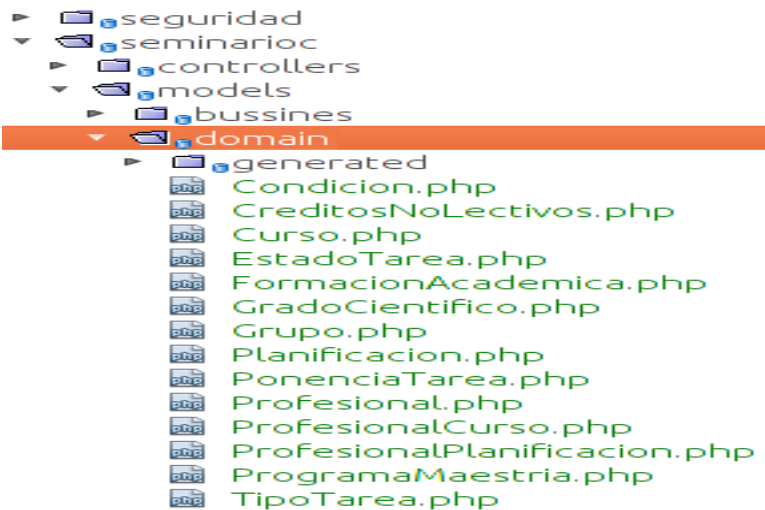
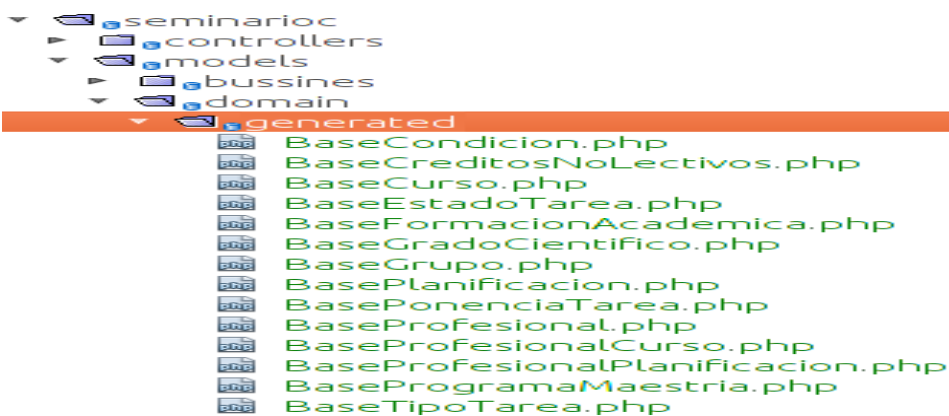


Ilustración 21: Ejemplo de estructura de las clases de dominio

- Generated (Dominio bases)

Las clases que se encuentran dentro de Generated el nombre comienza con la palabra: “Base” y seguido el nombre de la clase.



3.4.1 Nomenclatura de las funciones

Se empleó la notación Camel Casing, la cual es parecida a la notación Pascal Casing con la excepción de que la letra inicial del identificador no debe estar en mayúscula. Ejemplo: **cargarProfesionalesPorGrupo**. En caso de ser una acción de la clase controladora se le pone el nombre y seguido la palabra: "Action".

Ejemplo: **cargarProfesionalesPorGrupoAction**

```
function cargarProfesionalesPorGrupoAction() {
    $idGrupo = $this->_request->getPost('idGrupo');
    $infoProfesionales = Profesional::getProfesionalesPorGrupo($idGrupo);

    $res = array();
    foreach ($infoProfesionales as $valor) {
        $arrAux = array();

        // $arrAux['id_semc'] = $valor['id'];
        $arrAux['id'] = $valor['id_profesional'];
        $arrAux['nombre'] = $valor['nombre_apellidos'];
        $arrAux['creditos_op'] = $valor['creditos_op'];
        $arrAux['creditos_ob'] = $valor['creditos_ob'];
        $arrAux['tutores'] = $valor['tutores'];
        $arrAux['temaI'] = $valor['tema_investigacion'];
        $arrAux['resumen'] = $valor['resumen_tema_investigacion'];
        $arrAux['grado'] = $valor['GradoCientifico']['nombre'];
        $arrAux['FA'] = $valor['FormacionAcademica']['nombre'];
        $arrAux['PM'] = $valor['ProgramaMaestria']['nombre'];
        $arrAux['GR'] = $valor['Grupo']['nombre'];
        $res[] = $arrAux;
    }
    echo json_encode(array('datos' => $res));
}
```

El uso de la nomenclatura en clases y funciones permite una mayor legibilidad en el código.

3.5 Pruebas de caja negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores (Pressman 2002). Con la aplicación de este tipo de prueba se pudo comprobar que el sistema cumplía con cada uno de los requisitos identificados durante el análisis.

3.5.1 Casos de prueba.

Los casos de pruebas definen un conjunto específico de entradas de pruebas, ejecución de condiciones y resultados esperados. Un diseño de los casos de prueba es fundamental para el éxito de las pruebas. Un caso de prueba especifica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. El caso de prueba simula las interacciones del actor con el sistema para verificar que el sistema hace lo que se espera de él. Así, el principal artefacto para obtener pruebas del sistema son los requisitos funcionales (Cuesta 2007).

A continuación se muestra el diseño de caso de prueba perteneciente al caso de uso gestionar grupo de profesionales para el escenario Adicionar grupo.

Condiciones de Ejecución:

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar el sistema **Seminarios Científicos/Grupos/Gestionar grupos**
- Debe existir al menos una persona registrada.

| Nombre del requisito | Descripción general | Escenarios de prueba | Flujo del escenario |
|-----------------------|--|---|---|
| SC 1: Adicionar grupo | El sistema muestra un formulario que te permite adicionar los datos de un grupo. | EC 1.1: Adicionar grupo introduciendo letras. | - Se presiona el botón Adicionar Grupo . - Se introducen letras. - Se presiona el botón Aceptar . - Se muestra un mensaje: Se ha adicionado el grupo correctamente. - Se presiona el botón Aceptar . |

| | | |
|--|---|---|
| | <p>EC 1.2: Adicionar grupo introduciendo datos válidos.</p> | <ul style="list-style-type: none"> - Se presiona el botón Adicionar Grupo. - Se introducen números. - Se presiona el botón Aceptar. - Se muestra un mensaje: Se ha adicionado el grupo correctamente. - Se presiona el botón Aceptar. |
| | <p>EC 1.3: Adicionar grupo introduciendo datos válidos.</p> | <ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen combinación de números y letras. - Se presiona el botón Aceptar. - Se muestra un mensaje: Se ha adicionado el grupo correctamente. - Se presiona el botón Aceptar. |
| | <p>EC 1.4: Adicionar grupo introduciendo datos inválidos.</p> | <ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen caracteres especiales. - El sistema muestra una excepción: Este campo no puede contener caracteres especiales. |
| | <p>EC 1.5: Adicionar grupo</p> | <ul style="list-style-type: none"> - Se presiona el botón Adicionar. |

| | | | |
|--|--|-------------------------|---|
| | | dejando campos vacíos. | <ul style="list-style-type: none"> - Se muestra un mensaje: Por favor verifique, existen campos con valores incorrectos. - Se presiona el botón Aceptar. |
| | | EC 1.6: Cancelar | <ul style="list-style-type: none"> - Se presiona el botón Cancelar. - Se cierra el formulario. |

Tabla 7: Caso de prueba Adicionar grupo

Luego de la realización de los dueños de casos de prueba, el sistema fue presentado en el departamento de calidad del centro CEIGE. Las pruebas de calidad fueron realizadas por un especialista, se realizaron 2 iteraciones, en la primera se detectaron 16 no conformidades y ya en la segunda estos fueron corregidos. Además la aplicación fue presentada ante el cliente, mostrándose satisfecho con la solución informática y emitiendo su acta de aceptación en la cual consta que el sistema realizado cumple con las condiciones de calidad necesarias y satisface las necesidades de la dirección del CEIGE. [Ver anexo 6](#).

Conclusiones parciales

En este capítulo queda plasmando cómo está implementado el sistema a través de los diagramas de componentes del sistema. También se muestra el diagrama de despliegue, el cuál ilustra los nodos que serán usados para la implantación del sistema. Además de un ejemplo de caso de prueba, mostrando resultados satisfactorios, demostrándose la calidad del sistema realizado. Se explica la nomenclatura para las clases y las funciones para lograr un mejor entendimiento del código y una mayor organización.

Conclusiones Generales

- El estudio de los sistemas de información científica y gestión de seminarios científicos existentes hoy en el mundo y en Cuba en la Educación Superior, sirvió para formalizar el marco teórico-conceptual de la investigación.
- El análisis y diseño del Sistema de Gestión de Seminarios Científicos del CEIGE permitió que se implementara correctamente cada una de las funcionalidades identificadas.
- La implementación del Sistema de Gestión de Seminarios Científicos utilizando la tecnología propuesta posibilitó:
 - Organizar los profesionales por grupos.
 - Planificar la participación de los profesionales en los seminarios científicos.
 - Llevar un control del completamiento de cursos optativos y obligatorios cursados por los profesionales que se encuentran en un programa académicos de postgrado.
 - Gestionar la información referente a los créditos no lectivos de los profesionales.
 - Por cada presentación que un profesional realizó, asignarle tareas y llevar un seguimiento de las mismas.
 - Brindar reportes a instancias superiores acerca de las tareas y cursos por cada profesional.
 - Permite que los profesionales vean las planificaciones, cursos y tareas que le han sido asignadas.
- Mediante las pruebas realizadas al sistema se comprobó que el mismo cumple todos los objetivos para los que fue diseñado e implementado.

Recomendaciones

A pesar de haber cumplidos los objetivos trazados en la investigación, se recomienda:

- Darle continuidad al sistema desarrollando nuevas funcionalidades.
- Valorar la integración de esta aplicación propuesta como un subsistema del sistema de Postgrado.
- Validar el sistema para la gestión de seminarios científicos en otros centros de desarrollo de software en la universidad.

Referencias Bibliográficas

- Acuña, K. B. (2009) "Selección de Metodologías de Desarrollo para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos.", 240.
- Alvarado, M. (2012) "Gestión del conocimiento en tecnología de la información: Análisis de la experiencia en el Sistema de Bibliotecas de la UC."
- Alvarez, A. (2011). Análisis y diseño de una aplicación web para la gestión de información de eventos científicos. Habana, UCI: 88.
- Alvarez, A. (2011). Análisis y diseño de una aplicación web para la gestión de información de eventos científicos. La Habana, UCI: 88.
- Asencio, C. (2011). ""La Tutoría de la Maestría en Ciencias de la Educación en la Provincia de Villa Clara."" Revista Varela **1-Nro28-2011**: 16.
- Cabezas, D. T. (2011). "Herramientas para la evaluación de la ciencia en universidades y centros I+D: descripción y usos." from <http://www.thinkepi.net/tag/sistemas-de-informacion-cientifica>.
- cibernetia. (2010). "Apache." from http://www.cibernetia.com/manuales/instalacion_servidor_web/2_0_apache.php.
- Cuesta, A. (2007). Sistema Asistente para Gestionar Pruebas de Caja Negra. Habana, UCI: 118.
- Chamorra, M. A. (2008) "Seminario, Ciencia, el Método Científico, la Investigación y sus Fases." **3**, 51.
- Glick, T. B. J. (2009). NetBeans. NetBeans: The Definitive Guide.: 674.
- Gómez Baryolo, O. (2011) "Arquitectura tecnológica para el desarrollo de software." 20.
- Hechavarría, N. R. Y. (2009). Aplicación Web para la gestión de eventos científicos en la Universidad de las Ciencias Informáticas. Habana, UCI: 141.
- Larman, C. (2001). UML y patrones. Introducción al análisis y diseño orientado a objetos.
- López, J. (2010) "CSS Hojas de estilo en cascada.", 1.
- MES. (2007). "REGLAMENTO DE LA EDUCACIÓN DE POSGRADO DE LA REPÚBLICA DE CUBA.", from <http://www.uvs.sld.cu/archivos/reglamento-de-posgrado-mes-cuba.pdf>.
- NetBeans. (2012). "NetBeans IDE 7.2.1 Release Information." from <https://netbeans.org/community/releases/72/>.

- Obregón, W. G. (2012) "Modelo de desarrollo de software." **1**, 111.
- php.net. (2012). " PHP: Hypertext Preprocessor." from <http://php.net/>.
- Pons, N. L. (2012). Modelo de Gestión del conocimiento ISECO para el desarrollo de equipos de proyectos informáticos. Habana, UCI: 86.
- PostgreSQL. (1996-2013). "PostgreSQL 8.4.17 Documentation." from <http://www.postgresql.org/docs/8.4/static/intro-what-is.html>.
- Potencier, F. (2007). Symphony La Guia Definitiva. Symphony La Guia Definitiva. España: 435.
- Pressman, R. (2002). Ingeniería de software.Un enfoque Practico. Ingeniería de software.Un enfoque Practico. McGraw-Hill: 640.
- Pupo, Y. C. (2010). Libro de Ayuda del Marco de Trabajo Sauxe, En su versión 2.0. La Habana, UCI: 72.
- Reyes, Y. (2009). Gestión del Evento Científico NeuroRehabana 2010. La Habana, UCI: 94.
- Rumbaugh, I. J. G. B. J. (2000). El proceso unificado del desarrollo de software. El proceso unificado del desarrollo de software, Addison Wesley: 464.
- Scribd. (2008). "Arquitectura de software II.Digrama de componentes y de despliegue." Retrieved 2 de diciembre, 2012, from <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
- Tedeschi, N. (2012). "¿Qué es un Patrón de Diseño?", from <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
- U.net. (2013). "Qué es un posgrado." Retrieved 10 de noviembre, 2012, from <http://www.la-u.net/es/que-es-un-posgrado/>.
- UCI. (2011). "Postgrado." from www.uci.cu.
- Vásquez, F. E. (2009). "Gestionando el conocimiento.Acercamiento conceptual en entornos organizacionales y proyección en el contexto académico." Revista Iberoamericana de Ciencia, Tecnología y Sociedad-CTS **8**(23): 21.
- Verdecia, D. G. A. (2010). Aplicación web para la gestión de eventos científicos en la Universidad. La Habana, UCI: 85.
- Villeneuve, L. (2005). Mozilla Firefox Mozilla Firefox (Windows y Linux):navegue de otra manera: 202.
- Wage, J. (2009). Doctrine ORM for PHP. Doctrine ORM fot PHP.Guide to Doctrine 1.0: 407.

Anexos

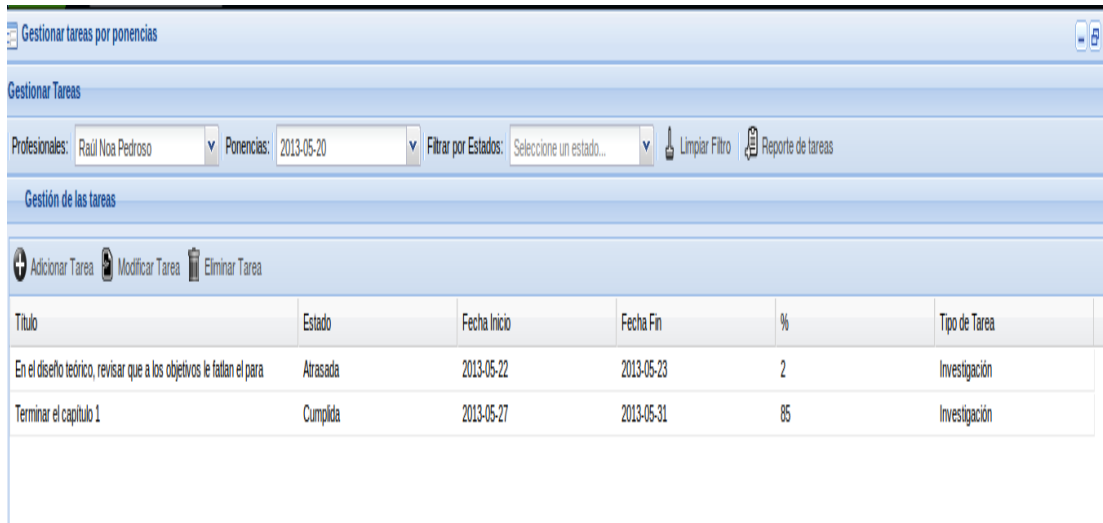
Anexo 1 Prototipos de interfaz de usuario



Interfaz 1: Gestionar grupos

| Gestión de Profesionales | | | | | |
|--|------------------|-------------|-------------|---|---|
| Adicionar Profesional Modificar Profesional Eliminar Profesional | | | | | |
| Nombre | Grado Científico | Créditos OP | Créditos OB | Tema Investigación | Programa de Maestría |
| Listley Castell Gomez | Ninguno | 2 | 2 | Monitorización y control en una línea de productos de software | Maestría en Gestión de Proyectos Informáticos |
| Dayne Gutierrez Gonzalez | Master | 2 | 1 | Aplicación informática para caracterizar ontologías | Maestría en Calidad de Software |
| Vimavis La Rosa Sordo | Master | 1 | 3 | EPF composer: complemento del Proceso de Mejoras en la Universidad de Ciencias Informáticas | Maestría en Educación a Distancia |
| Carlos Abel Capeans Hurtado | Ninguno | 1 | 2 | Análisis del sistema para graficar columnas litológicas de pozos de petróleo | Maestría en Informática Aplicada |
| Raúl Noa Pedroso | Ninguno | 5 | 2 | Nuevo paradigma en la tecnología: computación en la nube | Maestría en Gestión de Proyectos Informáticos |
| Daimara Mustelier Sanchidrian | Ninguno | 0 | 1 | Vías para mejorar el rendimiento de un laboratorio de pruebas de software | Maestría en Educación a Distancia |

Interfaz 2: Gestionar profesional



Interfaz 3: Gestionar tareas por ponencias



Interfaz 4: Gestionar créditos no lectivos

Anexo 2 Descripción de los casos de uso del sistema

| | | |
|---|---|--|
| Caso de uso | Ver planificaciones | |
| Actor: | Profesional | |
| Resumen: | El profesional ve todas las planificaciones que tiene registradas | |
| Precondiciones: | El profesional tiene que estar autenticado en el sistema | |
| Flujo Normal de Eventos | | |
| Sección "Flujo Normal de Eventos" | | |
| Acción del Actor | Respuesta del Sistema | |
| 1- Selecciona opción ver mis planificaciones. | 2- Muestra un listado un listado de las | |

| | |
|-------------------------|---|
| | planificaciones de ese profesional. En caso de que no exista ninguna planificación ver flujo alternativo 1.a. |
| Flujos Alternos | |
| Acción del Actor | Respuesta del Sistema |
| | 1.a Muestra el mensaje : Usted no tiene ninguna planificación. |

Tabla 8: Descripción CU Ver planificaciones de un profesional

| | |
|--|---|
| Caso de uso | Ver cursos |
| Actor: | Profesional |
| Resumen: | El profesional ve la información de todos los cursos que ha cursado |
| Precondiciones: | Deben existir profesionales con planificaciones registradas |
| Flujo Normal de Eventos | |
| Sección "Flujo Normal de Eventos" | |
| Acción del Actor | Respuesta del Sistema |
| 1- Selecciona opción ver mis cursos. | 2- Muestra un grid con el nombre del curso, el año en que lo cursó y el tipo. |

Tabla 9: Descripción del CU Ver cursos

| | |
|--|--|
| Caso de uso | Gestionar créditos no lectivos |
| Actor: | Subdirector de Investigación y Postgrado |
| Resumen: | |
| Precondiciones: | Deben existir profesionales registrados |
| Flujo Normal de Eventos | |
| Sección "Flujo Normal de Eventos" | |
| Acción del Actor | Respuesta del Sistema |
| 1- Selecciona opción ver mis cursos. | 2- |

Anexo 4 Diagramas de clases del diseño

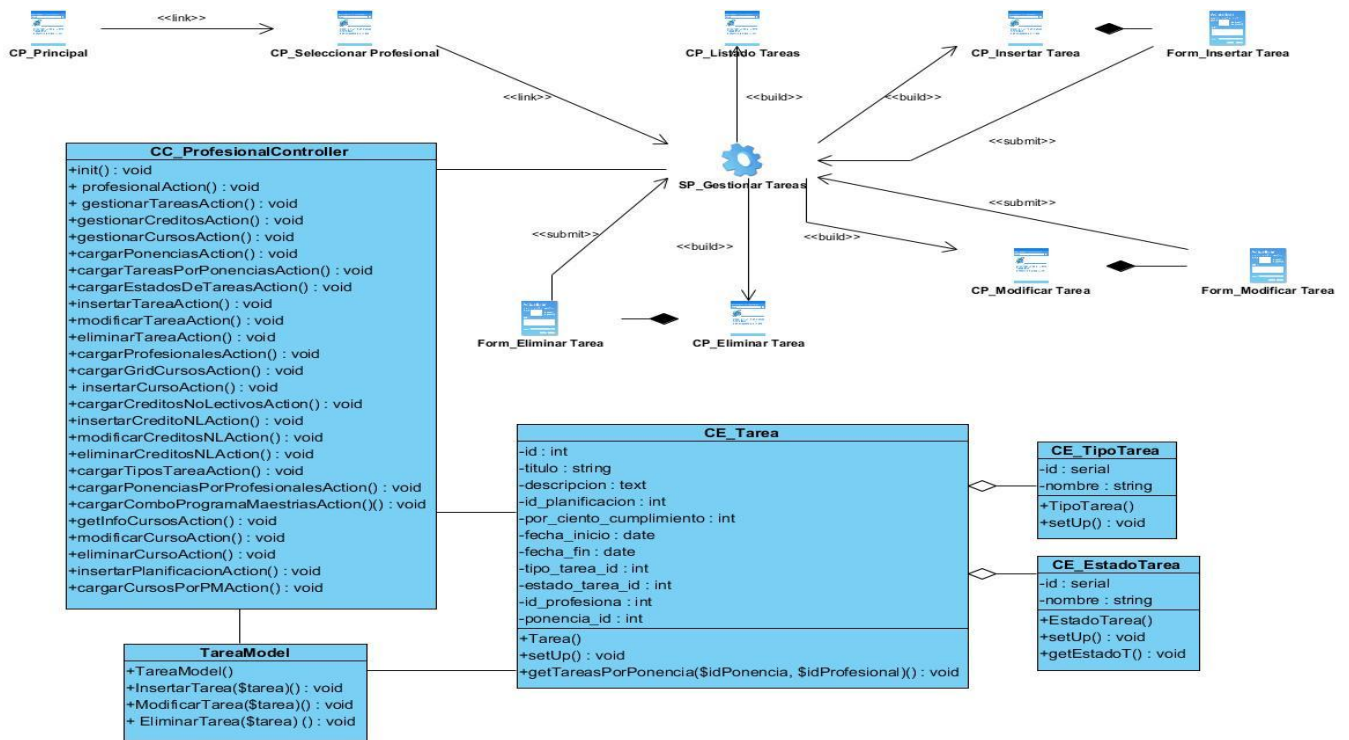


Ilustración 22: Diagrama de clases del diseño gestionar tareas

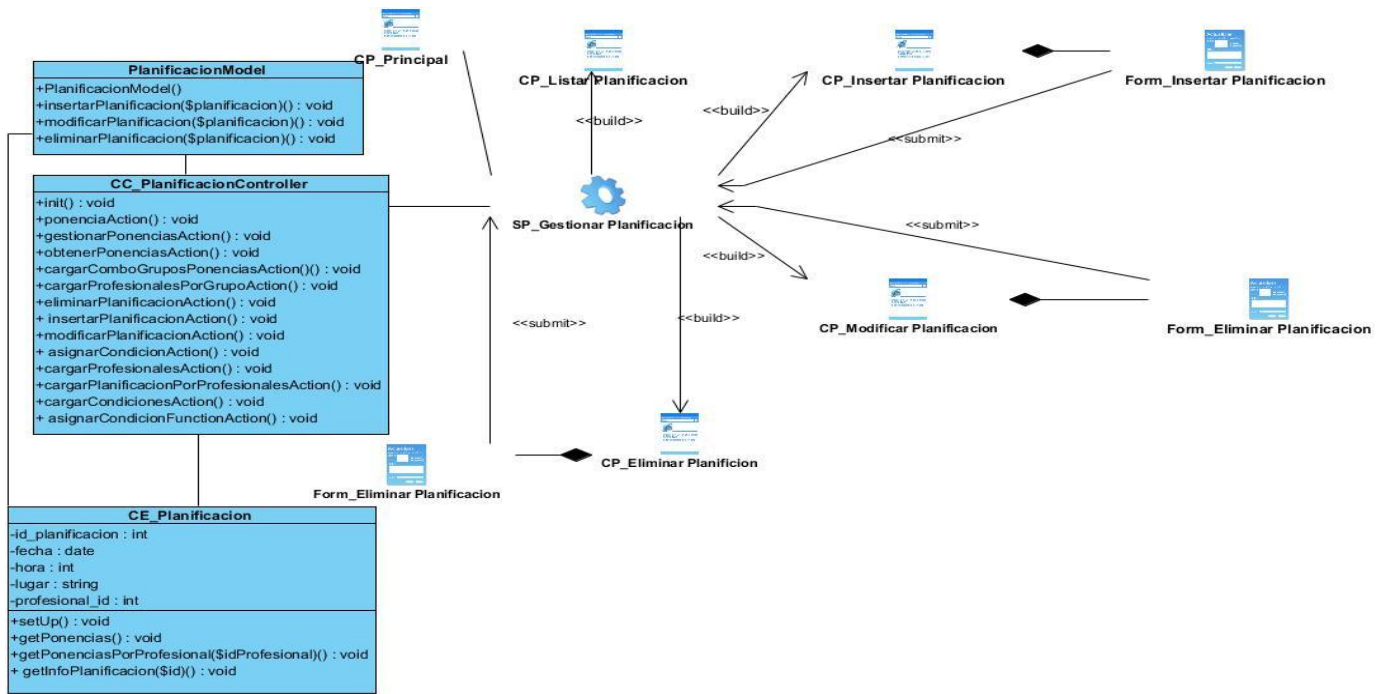


Ilustración 23: Diagrama de clases del diseño gestionar planificación de seminarios científicos

Anexo 5 Diagramas de secuencia

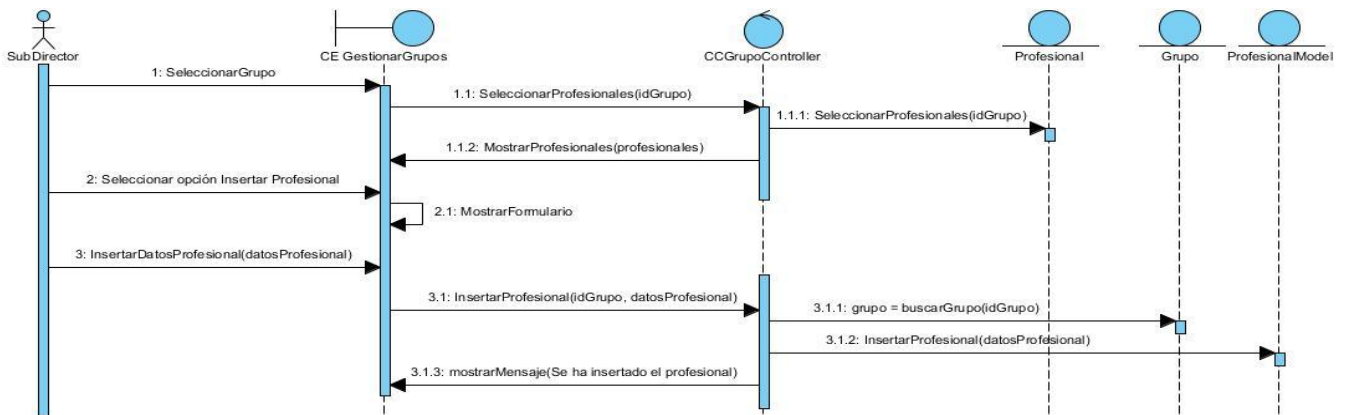


Ilustración 24: Diagrama de secuencia insertar profesional

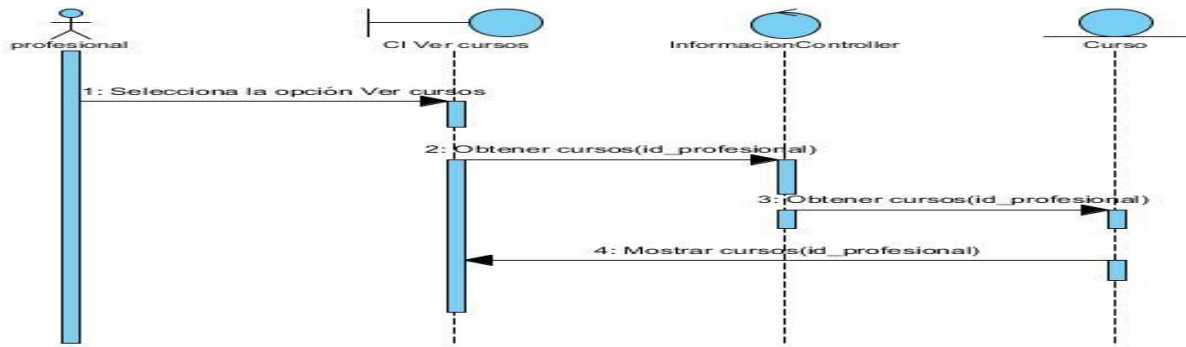


Ilustración 25: Diagrama de secuencia Ver cursos

Glosario de Términos

Seminario científico

Presentación de índole científica que se realiza con el fin de difundir conocimientos o desarrollar investigaciones, en la que participan alrededor de 50 o 60 personas, donde los principales protagonistas son los maestrantes, los cuales discuten sus trabajos frente a un tribunal y un grupo de participantes.

Créditos lectivos

Se obtienen a partir de los cursos obligatorios y optativos que exige el programa de maestría o doctorado.

Créditos no lectivos

Se obtienen en cuanto a participación en eventos y publicaciones en revistas referenciadas de acuerdo a los niveles que establece la comisión nacional de grado científico en relación con los principios del Ministerio de Educación Superior (MES).

Curso

Organización de un conjunto de contenidos que aborden resultados de investigación relevantes o asuntos trascendentes de actualización con el propósito de complementar o actualizar los conocimientos.

Curso optativo

Que se puede elegir según una preferencia y por tanto no es obligatorio.

Entrenamientos

Posibilita la formación básica y especializada de los graduados universitarios, particularmente en la adquisición de habilidades y destrezas y en la asimilación e introducción de nuevos procedimientos y tecnologías con el propósito de complementar, actualizar, perfeccionar y consolidar conocimientos y habilidades prácticas.

Diplomados:

El diplomado tiene como objetivo la especialización en un área particular del desempeño y propicia la adquisición de conocimientos y habilidades académicas, científicas y/o profesionales en cualquier etapa del desarrollo de un graduado universitario de acuerdo a las necesidades de su formación profesional o cultural.