

Universidad de las Ciencias Informáticas

Facultad 2



Título: Sistema automatizado para la gestión de información de los profesores de la facultad

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

Autores: Alejandro Obregón Rodríguez.
Yosdany Perdigón Obregón.

Tutor: Ing. Biasmey Morgado Guirola

Ciudad de La Habana, junio del 2007

Haz primero lo que es necesario, luego lo que es posible y de pronto te encontrarás haciendo lo imposible.

San Francisco de Asís.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

Alejandro Obregón Rodríguez.

Yosdany Perdigón Obregón.

Ing. Biasmey Morgado Guirola

Firma del Autor

Firma del Autor

Firma del Tutor

AGRADECIMIENTOS

Ante todo a mi familia, en especial a mis padres por creer en mí, por apoyarme en todo momento, por brindarme todo el amor del mundo.

A la revolución, a Fidel.

A todas las personas que de una forma u otra apoyaron nuestra idea, en especial a mi novia Aylen, a todos los profesores, a todos los amigos.

Alejandro Obregón Rodríguez

RESUMEN

La gestión de la información se ha convertido en un proceso importante en las instituciones y empresas, debido a los beneficios que reporta una buena gestión de la información.

Este trabajo consiste en un sistema para la Gestión de la información de los profesores de la facultad 2. En nuestra universidad ha habido intentos para desarrollar una aplicación y se han presentado en jornadas científicas, pero por alguna razón no se han logrado desarrollar completamente. Es por ello que la Facultad 2 tiene la necesidad de implementar un sistema para gestionar la información de sus profesores y actividades, pues en ella se desarrollan procesos que no están informatizados. Hasta el momento gestionar la información emanada de estos procesos se hace lenta y engorrosa para los Jefes de Departamento que son los encargados de llevarla a cabo.

El objetivo fundamental de este trabajo es desarrollar una aplicación Web, segura y de interfaz amigable, que permita gestionar la información de forma rápida así como hacer más cómodo el trabajo para los Jefes de Departamentos.

Este documento recoge los resultados de todo el trabajo realizado. Se estudia como se llevan a cabo los procesos en la facultad, se realiza un análisis de las tendencias y tecnologías existentes y que más se adaptan al desarrollo de la aplicación. Se modela el negocio para un mejor entendimiento, se muestran los resultados del análisis y diseño de la propuesta del sistema así como la implementación final del sistema automatizado.

INDICE

| | |
|---|-----|
| AGRADECIMIENTOS | II |
| RESUMEN | III |
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA | 5 |
| 1.1. Introducción al Capítulo. | 5 |
| 1.2. Sistemas de Gestión de Información existentes. | 5 |
| 1.3. Tecnologías utilizadas en el desarrollo del sistema. | 6 |
| 1.3.1. Dreamweaver. | 6 |
| 1.3.2. Servidor HTTP (Apache). | 7 |
| 1.3.3. PostgreSQL como gestor de Base de Datos. | 7 |
| 1.3.4. Rational Rose. | 8 |
| 1.3.5. Lenguaje de programación. PHP. | 8 |
| 1.3.6. JavaScript. | 10 |
| 1.3.7. Language HTML. | 10 |
| 1.3.8. Patrones de diseño Modelo-Vista-Controlador. | 10 |
| 1.4. Metodología de desarrollo RUP. | 11 |
| 1.4.1. RUP. El Proceso Unificado de Desarrollo. | 11 |
| 1.4.2. El Lenguaje Unificado de Modelación UML. | 12 |
| 1.5. Conclusiones del Capítulo. | 14 |
| CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA | 15 |
| 2.1. Introducción al Capítulo. | 15 |
| 2.2 Objeto de estudio. | 15 |
| 2.2.1. Problema y situación problemática. | 15 |
| 2.2.2. Objetivos estratégicos de la organización y procesos de negocio que los soportan. ... | 16 |
| 2.2.3. Objeto de automatización. | 16 |
| 2.2.4. Información que se maneja. | 16 |
| 2.3. Propuesta del Sistema. | 17 |
| 2.4. Modelo del negocio. | 17 |
| 2.4.1. Actores del Negocio. | 17 |
| 2.4.2. Tabla 1. Definición de los Actores del Negocio. | 17 |
| 2.4.3. Definición de los trabajadores del negocio. | 18 |
| 2.4.4. Tabla 2. Definición de los Trabajadores del Negocio. | 18 |
| 2.4.5. Diagrama 1. Diagrama de casos de uso del negocio. | 19 |
| 2.4.6. Descripción de los casos de uso en formato expandido y mediante los diagramas de actividades. | 19 |

| | |
|--|-----------|
| 2.4.7. Diagramas de Actividad..... | 20 |
| 2.4.8. Diagrama de clases del modelo de objetos. CU Gestionar Actividad..... | 20 |
| 2.5. Relación de los Requerimientos..... | 20 |
| 2.5.1. Listado de los requerimientos funcionales enumerados..... | 20 |
| 2.5.2. Definición de los requerimientos no funcionales..... | 22 |
| 2.6. Modelo de Casos de Uso del Sistema..... | 22 |
| 2.6.1. Definición de los Actores del Sistema..... | 23 |
| 2.6.2. Tabla 6. Actores del Sistema..... | 23 |
| 2.6.3. Diagrama de casos de uso del sistema a automatizar..... | 24 |
| 2.6.4. Descripción del caso Casos de uso expandidos..... | 24 |
| 2.7. Conclusiones del Capítulo..... | 25 |
| CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA..... | 26 |
| 3.1. Introducción..... | 26 |
| 3.2. Análisis..... | 26 |
| 3.2.1. Modelo de Análisis..... | 26 |
| 3.2.2. Diagrama 7. Diagrama de clases de análisis. CU Autenticar..... | 26 |
| 3.2.3. Diagrama 8. Diagrama de clases de análisis. CU Gestionar Profesor..... | 27 |
| 3.2.4. Diagrama 9. Diagrama de clases de análisis. CU Gestionar Tarea..... | 28 |
| 3.2.5. Diagrama 10. Diagrama de clases de análisis. CU Solicitar Reporte..... | 28 |
| 3.2.6. Diagrama 12. Diagrama de clases de análisis. CU Cambiar Contraseña..... | 29 |
| 3.2.7. Diagrama 13. Diagrama de clases de análisis. CU Gestionar Departamento..... | 29 |
| 3.2.8. Diagrama 14. Diagrama de clases de análisis. CU Gestionar Asignatura..... | 30 |
| 3.2.9. Diagrama 15. Diagrama de clases de análisis. CU Gestionar Grupo..... | 30 |
| 3.3. Modelos del Diseño..... | 31 |
| 3.3.1 Diagramas de interacción..... | 31 |
| 3.3.2. Diagramas de clases del diseño..... | 31 |
| 3.3.3. Diagrama 16. Diagrama de clases del diseño. CU Autenticar..... | 32 |
| 3.3.4. Diagrama 17. Diagrama de clases del diseño. CU Cambiar Contraseña..... | 32 |
| 3.3.5. Diagrama 18. Diagrama de clases del diseño. CU Gestionar Profesor..... | 33 |
| 3.3.6. Diagrama 19. Diagrama de clases del diseño. CU Gestionar Tarea..... | 34 |
| 3.3.7. Diagrama 20. Diagrama de clases del diseño. CU Solicitar Reporte..... | 35 |
| 3.3.8. Diagrama 21. Diagrama de clases del diseño. CU Gestionar Departamento..... | 36 |
| 3.3.9. Diagrama 22. Diagrama de clases del diseño. CU Gestionar Asignatura..... | 37 |
| 3.3.10. Diagrama 23. Diagrama de clases del diseño. CU Gestionar Grupo..... | 38 |
| 3.4. Diseño de la Base de Datos..... | 39 |
| 3.4.1. Modelo lógico de datos..... | 39 |
| 3.4.2. Diagrama de clases persistentes..... | 39 |
| 3.4.3. Modelo físico de datos..... | 40 |
| 3.5. Conclusiones del Capítulo..... | 41 |
| CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA..... | 42 |
| 4.1. Introducción al capítulo..... | 42 |

| | |
|--|----|
| 4.2. Diagrama de despliegue. | 42 |
| 4.3. Diagrama de componentes. | 43 |
| 4.3.1. Subsistema de implementación Autenticarse. | 43 |
| 4.3.2. Subsistema de implementación Cambiar contraseña. | 44 |
| 4.3.3. Subsistema de implementación Gestiona Departamento. | 44 |
| 4.3.4. Subsistema de implementación Gestionar Asignatura. | 45 |
| 4.3.5. Subsistema de implementación Gestionar grupo. | 45 |
| 4.3.6. Subsistema de implementación Gestionar profesor. | 46 |
| 4.3.7. Subsistema de implementación Gestionar tarea. | 46 |
| 4.3.8. Subsistema de implementación Solicitar reporte. | 47 |
| 4.4. Conclusiones del Capítulo. | 48 |
| CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD. | 49 |
| 5.1. Introducción. | 49 |
| 5.2. Planificación basada en casos de usos. | 49 |
| 5.2.1. Tabla 5.1 Factor de peso de los actores sin ajustar. | 49 |
| 5.2.2. Tabla 5.2 Factor de peso de los casos de uso sin ajustar. | 50 |
| 5.2.3. Cálculo de los Puntos de casos de uso ajustados. | 50 |
| 5.2.4. Tabla 5.3 Factor de complejidad técnica. | 51 |
| 5.2.5. Tabla 5.4 Factor de ambiente. | 52 |
| 5.2.6. Estimación de esfuerzo a través de los puntos de casos de uso. | 53 |
| 5.2.7. Calcular esfuerzo de todo el proyecto. | 54 |
| 5.2.8. Tabla 5.5 Esfuerzo del proyecto. | 54 |
| 5.3. Beneficios tangibles e intangibles. | 55 |
| 5.4. Análisis de costos y beneficios. | 56 |
| 5.5. Conclusiones del Capítulo. | 57 |
| CONCLUSIONES GENERALES | 58 |
| RECOMENDACIONES | 59 |
| REFERENCIAS BIBLIOGRÁFICAS | 60 |
| BIBLIOGRAFÍA | 62 |
| ANEXOS | 63 |
| Anexo 1. | 63 |
| Tabla 3. Descripción textual del caso de uso: Gestionar Actividad. | 63 |
| Tabla 4. Descripción textual del caso de uso: Gestionar persona. | 65 |
| Tabla 5. Descripción textual del caso de uso: Solicitar Información. | 66 |
| Anexo 2. | 68 |
| Diagrama 2. Diagrama de actividades para el Caso de uso Gestionar Actividad. | 68 |
| Diagrama 3. Diagrama de Actividades para el caso de uso Gestionar Profesor. | 69 |

| | |
|--|----|
| Diagrama 4. Diagrama de Actividades para el caso de uso Solicitar Información..... | 70 |
| Anexo 3. | 71 |
| Diagrama de clases del modelo de objetos. CU Gestionar Actividad..... | 71 |
| Diagrama de clases del modelo de objetos. CU Gestionar Persona..... | 71 |
| Diagrama de clases del modelo de objetos. CU Solicitar Información..... | 72 |
| Anexo 4. | 73 |
| Tabla 7. Caso de uso Autenticar..... | 73 |
| Tabla 8. Descripción del caso de uso Gestionar Tarea..... | 74 |
| Tabla 9. Descripción del caso de uso Gestionar Profesor..... | 76 |
| Tabla 10. Descripción del caso de uso Solicitar Reporte..... | 78 |
| Tabla 11. Descripción del caso de uso Cambiar Contraseña..... | 79 |
| Tabla 12. Descripción del caso de uso Gestionar Grupo..... | 80 |
| Tabla 13. Descripción del caso de uso Gestionar Asignatura..... | 81 |
| Tabla 14. Descripción del caso de uso Gestionar Departamento..... | 83 |
| Anexo 5. | 86 |
| Diagramas de secuencia. CU Autenticar..... | 86 |
| Diagrama de secuencia. CU Gestionar Profesor..... | 87 |
| Diagrama de colaboración. CU Gestionar Tarea..... | 88 |
| Diagrama de colaboración. CU Solicitar Reporte..... | 89 |
| Diagrama de colaboración. CU Cambiar Contraseña..... | 89 |
| Diagrama de secuencia. CU gestionar Asignatura..... | 90 |
| Diagrama de secuencia. CU Gestionar Departamento..... | 91 |
| Diagrama de colaboración. CU Gestionar Grupo..... | 92 |
| GLOSARIO DE TÉRMINOS | 93 |

INTRODUCCIÓN

Actualmente las organizaciones tienen la necesidad imperiosa de gestionar eficazmente un recurso muy importante al cual no se le prestaba la debida atención y tratamiento: la información. Esta produce siempre un efecto sobre el conocimiento del receptor, permite generar puntos de vistas, discusiones y análisis de la realidad; y su grado de utilidad se deriva de la importancia concedida, la comprensión, aprehensión y uso que se haga de ella.

En virtud de que la información es un recurso cuya génesis y desaparición está en función de ser o no aprehendida por el hombre, en su esencia misma subyace una acción comunicativa, es decir, que su existencia demanda vías o mecanismos que viabilicen su conocimiento por parte de los sujetos. Desde esta perspectiva, la información requiere de soportes que le permitan perpetuarse, diseminarse y por ende ser utilizada; es por esto que el hombre ha ideado en diversas épocas medios para garantizar la permanencia de este recurso en el tiempo; el más tradicional de ellos es el documento escrito, que, aunque los avances tecnológicos han dado paso nuevas formas para conservar la información, sigue prevaleciendo.

Las computadoras son usadas por el hombre desde hace varias décadas con intereses propios, desde entonces se han venido creando aplicaciones enfocadas a esos intereses. La necesidad de sistemas que trabajen con gran volumen de datos ha hecho que se desarrolle el software y surjan nuevas aplicaciones donde el objetivo primario es la gestión y presentación de información. Además, en todas las empresas se ha visto como una necesidad, tener una documentación de los resultados de sus procesos del negocio. En la actualidad la información que se genera tiene diferentes formatos que muchas veces no cumple con funcionalidades que se requieren, tal es el caso de las facturas, informes de marketing, etc., se debe, por tanto dar respuesta a necesidades complejas y diversas pero de forma y planteamientos no tradicionales, con el uso de la nuevas tecnologías, ellas nos ofrecen disponibilidad, mayor funcionalidad, potencialidad, versatilidad y posibilita obtener nuestros más variados objetivos, resultando un apoyo importante en los procesos del negocio.

Las organizaciones exitosas ya no son aquellas que poseen más capital sino las que más y mejor uso hacen de la información. [1]

La gestión de la información universitaria

A través de la gestión de la información, articulada en servicios bien administrados y orientados, las universidades pueden optimizar su funcionamiento y su productividad, ya que reciben el soporte efectivo a los objetivos planteados.

“La función esencial de la gestión de información universitaria es la de supervisar, controlar y garantizar el flujo correcto de la información hacia todas las instancias que conforman la estructura de la institución, con el fin de minimizar, en lo posible, los problemas típicos que suelen presentarse en estas organizaciones, como lo es: la sobrecarga de papeles con la consecuente escasez de información, la resistencia al cambio, los celos por la información, la poca claridad en la propiedad de la información, los retrasos o extemporaneidad en el suministro de la información y la coexistencia de diferentes estilos para manejar este valioso recurso”.[1]

Los departamentos de la facultad 2 de la Universidad de las Ciencias Informáticas (UCI) tienen la urgente necesidad de un sistema que gestione su información correctamente y que asegure la buena marcha de la institución.

La Facultad 2 (F-2) de la UCI se divide en tres departamentos docentes que son: Departamento de Ciencias Básicas, Departamento de Humanidades, Departamento de Programación y dentro de estos las asignaturas relacionadas con cada departamento, aunque pueden existir cambios en esta estructura. Los jefes de departamentos son los encargados de llevar el control de la información que se genera producto del trabajo y es almacenada en documentos Word y tablas en Excel. Esto trae consigo que no esté ordenada y precisa, además del tiempo que se invierte en consultarla. Estas herramientas no les brindan en muchos de los casos posibilidades que les facilite el proceso de gestión de la información y no se sienten identificados con un entorno que sea más específico para su trabajo.

Con el objetivo de mejorar la calidad de este proceso surgió la idea de desarrollar un sistema automatizado que de solución a estos problemas, proporcione más control y pueda manejar más datos.

La intención es desarrollar un sistema automatizado que permita a los jefes de departamento de la F-2 de la UCI controlar la información de sus profesores y actividades. Por supuesto que todo esto trae consigo un gran uso de las tecnologías software o hardware que existen hoy en día en la universidad.

El **aporte práctico esperado del trabajo** es darle un enfoque de sistema a todo el proceso de gestión de información, integrando todos los servicios requeridos en la misma aplicación.

Se definió como **objeto de estudio y campo de acción** la gestión de la información de los profesores en la facultad 2.

El **Objetivo general** es desarrollar un sistema automatizado que permita almacenar y consultar la información del personal docente de la facultad 2.

Se plantearon como **Objetivos específicos** hacer un estudio y documentar los procesos del negocio de la entidad y el sistema, analizar, diseñar e implementar este sistema.

La **Situación Problemática** presente es que actualmente en la F-2 se hace difícil gestionar la información de los profesores de manera eficiente. Los Jefes de Departamentos (JD) deben tener entre otras responsabilidades la de llevar un adecuado control de toda esta información con el objetivo de mejorar el desempeño de la facultad. La información se lleva en documentos Office y se almacena en diferentes formatos que no cumplen con funcionalidades que se necesitan.

De todo lo anterior se deriva el **Problema Científico**, ¿Como solucionar las dificultades que existen en la facultad 2 con la información de los profesores?

El **objeto de investigación** es el proceso de gestión de la información de los profesores en los departamentos de la UCI.

Para ello se plantearon las **tareas de la investigación**:

- 1.- Entrevistar a los jefes de departamentos para conocer su funcionamiento.
- 2.- Estudiar las herramientas necesarias para la implementación del sistema.

3.- Implementar una aplicación Web que cumpla con las funcionalidades que se requieren.

Este Trabajo está estructurado en 5 capítulos que recogen la documentación generada durante el proceso de desarrollo del sistema a realizar.

El **Capítulo 1** incluye un estado del arte del tema tratado a nivel nacional, de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad y en las que se apoya para la solución del problema que se enfrenta, sobre los cuales es necesario profundizar.

El **Capítulo 2** incluye Objeto de estudio, Propuesta de sistema, Modelo de negocio, sus actores y trabajadores, además del diagrama de casos de uso del negocio y los diagramas de actividades, se identifican los requerimientos y los casos de usos del sistema a automatizar.

En el **Capítulo 3** se recoge todo lo referente al análisis y diseño y se muestra toda la documentación que se generó en este flujo de trabajo

En el **Capítulo 4** recoge del flujo de trabajo Implementación los diagramas de despliegue y componentes.

Y por último en el **Capítulo 5** se hace el estudio de factibilidad, el tiempo de duración del proyecto, costo y esfuerzo.

Se espera que este trabajo sea un aporte más a la facultad en la tarea de informatizar todos sus procesos, beneficie en el trabajo a los Jefes de Departamentos y que sirva como antecedente a otros sistemas con objetivos semejantes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción al Capítulo.

En este capítulo se citará la existencia de otros sistemas homólogos al que se piensa desarrollar. Se abordarán las diferentes tecnologías y herramientas que se utilizaron en este proyecto para llevar a cabo el desarrollo del sistema que se propone y se explicará cada una de ellas enfatizando en las ventajas que trae su uso.

1.2. Sistemas de Gestión de Información existentes.

Ámbito Nacional

En la actualidad no existe un sistema en las universidades de nuestro país que gestione la información de sus profesores, pero sí existen sistemas que gestionan la información que se genera en los procesos docentes, mas bien referente a los estudiantes, estas se asemejan en algo a lo que se requiere pero no llegan a brindarnos la solución que se necesita, por lo que se hace necesario llevar a cabo la implementación de un sistema con características propias que se adapte y de solución a los problemas que existen en nuestra facultad en la gestión de la información de sus profesores.

GESTACAD (Sistema para la Gestión Académica).

Este sistema fue desarrollado en la Universidad de Matanzas “Camilo Cienfuegos” por la Facultad de Informática. El sistema tiene dos versiones. La primera fue la creación de módulos Stand Alone para la actualización de los datos (lo que garantizaba un primer nivel de seguridad al no estar disponible las vías de actualización de la información de forma pública) y un sitio Web destinado a mostrar reportes de la información almacenada. [2]

La segunda versión, ya en explotación en la Universidad de Matanzas, está concebida para desprenderse de aquellas dependencias de los módulos programados en un lenguaje no Open Source y la migración total hacia plataformas de Software Libre. Esta versión está completamente basada en plataforma Web y

está totalmente programada en PHP. Incluye los módulos correspondientes de administración, de secretarías y de matrícula. [2]

En la actualidad el sistema es capaz de realizar diversas acciones y brindar numerosos reportes los cuales son fruto de los requisitos funcionales del sistema recogidos en la fase de análisis como son:

1. Reporte de notas por asignatura y grupo (examen final, extraordinario, especial, premio).
2. Tabla con los resultados docentes de un grupo en un semestre.
3. Reporte de los resultados académicos de un estudiante en toda su carrera (Hoja de Rendimiento).
4. Dar baja a un estudiante.
5. Un Módulo Web para las Secretarías Docentes encaminado a la Gestión de Estudiantes que permite la realización de acciones generales comunes en una Secretaría Docente así como la obtención de reportes oficiales

El sistema tiene otros subsistemas los cuales están controlados por un sistema de sesiones que garantiza el acceso seguro y restringido a la información relevante. El acceso está controlado por un sistema de usuario y contraseña que es independiente para cada subsistema. [2]

1.3. Tecnologías utilizadas en el desarrollo del sistema.

1.3.1. Dreamweaver.

Es un editor WYSIWYG de páginas Web creado por Adobe (anteriormente por Macromedia). Es el programa de este tipo más utilizado en el sector del diseño y la programación Web por sus funcionalidades, su integración con otras herramientas como Adobe Flash y por su soporte a los estándares del World Wide Web Consortium. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas. Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional y soporta gran cantidad de tecnologías. Esta herramienta es muy fácil de usar y permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para visualizar las páginas Web, filtrar y mostrar el contenido utilizando tecnología de script como por ejemplo, ASP (Active Server Pages), ASP.NET, ColdFusion, JSP (JavaServer Pages) y PHP, sin necesidad de tener experiencia previa en programación. [3]

Todas estas características de Dreamweaver hacen que se ajuste adecuadamente a las necesidades de los programadores y se puede usar como entorno de desarrollo.

1.3.2. Servidor HTTP (Apache).

El servidor HTTP Apache es un software (libre) además de ser servidor HTML de código abierto para plataformas Unix, Windows, Macintosh. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero carece de una interfaz gráfica que ayude en su configuración. Es el servidor HTTP más usado, alrededor del 70% de los sitios Web en el mundo lo usan. La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y mucha de la funcionalidad que podría considerarse básica para un servidor Web es provista por módulos. [4]

1.3.3. PostgreSQL como gestor de Base de Datos.

Los Sistemas de gestión de base de datos, están orientados a servir de interfaz entre los usuarios, base de datos y aplicaciones que lo utilizan, su principal objetivo es manejar un conjunto de información de forma clara y sencilla.

Como este sistema que se propone se integra como un modulo de un sistema más amplio, y a petición de los clientes se ha decidido utilizar PostgreSQL como gestor de base de datos, software que ha sido diseñado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento. Además de esto, el código fuente está disponible para todos sin costo. Esto posibilita que, si se necesita extender o personalizar PostgreSQL de alguna manera, se puede hacer con un mínimo esfuerzo, sin costos adicionales.

Este motor de búsqueda tiene bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes con la potencia que cada uno de ellos aporta. Una de las grandes ventajas de PostgreSQL es que mediante un sistema denominado MVCC (Acceso concurrente multiversión)

PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Este gestor de base de datos es multiplataforma y está diseñado para ambientes de alto volumen. PostgreSQL usa una estrategia de almacenamiento de filas para conseguir una mejor respuesta en ambientes de grandes volúmenes.

Este buscador tiene una gran estabilidad y confiabilidad legendarias. Es muy común que muchas entidades reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad, ni una sola vez; simplemente funciona. [5]

1.3.4. Rational Rose.

Rational Rose es una herramienta para modelado visual que en conjunto con otras cubre todo el ciclo de desarrollo de un software.

Se caracteriza principalmente por permitir generar código a partir de modelos y en sentido inverso, chequeo semántico de los modelos, admite notaciones como UML y Booch, Integración con modelado de datos, etc.

Rational Rose permite completar una gran parte de de los flujos fundamentales del proceso unificado de Rational (RUP) como el modelado del negocio, la captura de requisitos en forma parcial, el análisis y diseño y parte de la implementación, así como el control de cambios. [6]

1.3.5. Lenguaje de programación. PHP.

A solicitud del cliente y decisión de los programadores, se optó por utilizar el lenguaje de programación PHP por las características que posee, además de ser este un módulo de un sistema más amplio y permitirá estandarizar.

Además, *"PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. Es un acrónimo recurrente que significa "PHP Hypertext Pre-processor", y se trata de un*

lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web “. [7]

Su interpretación y ejecución se da en el servidor, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. [7]

PHP permite la conexión a diferentes tipos de servidores de base de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas. PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Windows y MacOS X. [7]

Las principales ventajas por las que se decidió utilizar PHP son las siguientes:

- Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Puede leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Posee una amplia documentación en su página oficial, en la que se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- El código PHP es mucho más legible que el de PERL. Es similar en sintaxis a C y a PERL.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, XML, creación de PDF, etc.).
- Está siendo utilizado con éxito en millones de sitios Web.
- Es multiplataforma, funciona en todas las plataformas que soporten apache.
- Se puede incrustar código PHP con etiquetas HTML. [7]

1.3.6. JavaScript.

Es un lenguaje de programación interpretado, es decir, que no requiere compilación. Es utilizado especialmente en páginas Web embebido en el código HTML o similares. La mayoría de los navegadores pueden interpretar los códigos JavaScript incluidos en las páginas Web. Este es un lenguaje basado en prototipos, pues las nuevas clases se generan clonando las clases base (prototipos) y extendiendo sus funcionalidades. [8]

Este lenguaje se utilizará para las validaciones en la parte del cliente del sistema.

1.3.7. Language HTML.

HTML (**H**ypertext **M**arkup **L**anguage). Es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, es el formato estándar de las páginas Web, gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. Para crear una página Web se pueden utilizar varios programas especializados en esto, como por ejemplo, el Microsoft FrontPage, Macromedia Dreamweaver o cualquier editor de textos básico, [9]

1.3.8. Patrones de diseño Modelo-Vista-Controlador.

Este patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Maneja la visualización de la información.

Controlador: Controla el flujo entre la vista y el modelo (los datos).

Tanto la vista como el controlador dependen del modelo, y este no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes. [10]

Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo. [10]

1.4. Metodología de desarrollo RUP.

1.4.1. RUP. El Proceso Unificado de Desarrollo.

En un proceso de desarrollo de software existen un conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto a modo de plantilla que explica los pasos necesarios para terminar el proyecto; dando como resultado final un producto terminado. [11]

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo. [11]

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales de los cuales los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo:

- Modelamiento del negocio
- Requerimientos
- Análisis y diseño
- Implementación
- Prueba (Testeo)
- Instalación
- Administración del proyecto
- Administración de configuración y cambios
- Ambiente

RUP tiene 4 fases por las que obligatoriamente tiene que transitar todo proyecto: Conceptualización o Inicio, esta es la fase donde se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema. Elaboración es la fase en la que se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. En la fase de Construcción se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario y la última fase es la de Transición, en la que el release ya está listo para su instalación en las condiciones reales. [11]

1.4.2. El Lenguaje Unificado de Modelación UML.

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Además, intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

El modelo gráfico de UML tiene un vocabulario en el que se identifican:

- Elementos (abstracciones que constituyen los bloques básicos de construcción).
- Relaciones: Ligan los elementos.
- Diagramas: Es la representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas.[12]

1.5. Conclusiones del Capítulo.

En este capítulo se profundizó en las herramientas, lenguaje de programación y metodología utilizada en el desarrollo del sistema, fundamentándose en el lenguaje de programación y gestor de base de datos, además se mostró la existencias de otro sistema semejantes al que se pretende realizar y por qué es necesario la implementación de uno nuevo.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción al Capítulo.

Cada organización tiene su propia orientación de procesos. Algunas empresas dirigen el negocio siguiendo procesos bien definidos, mientras que otras ni siquiera piensan en esos términos. Aunque la empresa ya sea consciente de sus procesos, el primer paso hacia una mejor comprensión de su negocio es la descripción y documentación de los procesos que ya están en vigor. [13]

En este capítulo se abordarán temas relacionados con las características del sistema, sus requerimientos, se describen los procesos del negocio de la organización y se muestra la descripción de los casos de uso mediante diagramas UML.

2.2 Objeto de estudio.

2.2.1. Problema y situación problémica.

En nuestra facultad la gestión de la información de los profesores es llevada de forma semiautomatizada, esto se ve en el uso de herramientas Microsoft Office principalmente Word y Excel, pero esto no soluciona una serie de dificultades que aún existen y trae en muchos casos la aparición de otras. La Facultad 2 se ha enfocado a automatizar todos sus procesos, con software que identifiquen y se relacionen más específicamente con el trabajo de la facultad, en cursos anteriores ha habido intentos por desarrollar un sistema con este fin por otros estudiantes, que por diversas razones no han sido concluidos.

En la Facultad 2 se plantea una situación problémica y es que actualmente se hace difícil gestionar la información de los profesores de manera eficiente, la falta de control sobre las actividades de los profesores, la gran variedad de formatos en que es guardada la información, la dificultad de encontrar con rapidez datos estadísticos de los profesores, entre otras cosas, atenta contra el buen desempeño de la Facultad 2. Los Jefes de Departamentos son los responsables de llevar el control de toda esta información. Mediante un estudio y análisis de los procesos del negocio de la organización se determinaron los actores, trabajadores, casos de usos del negocio y requerimientos que se involucran.

2.2.2. Objetivos estratégicos de la organización y procesos de negocio que los soportan.

Dentro de los objetivos estratégicos de la organización se encuentra la formación de un personal altamente calificado en Ingeniería Informática, que contribuya en gran medida al desarrollo en ese campo y a la economía del país, la cual es su principal función. Para llevar a cabo esta tarea, la facultad 2 ha seguido una serie de estrategias con el fin de lograr estos objetivos, donde las nuevas tecnologías de la informática y las comunicaciones (TIC) juegan un papel importante. La facultad cuenta con aulas y laboratorios dotados de equipamiento tecnológico que ayudan en las tareas docentes y estudio independiente, además de laboratorios vinculados a proyectos con los que se apoya el proceso de formación de los estudiantes.

2.2.3. Objeto de automatización.

Se desea automatizar todo lo referente a la información de los profesores, que cubre datos personales como son: nombre, apellidos, número de carné de identidad, provincia, integración política; además de registrar el nivel científico de estos, departamento al que pertenecen, asignatura que imparte, grupos a los que imparte clases, además de todos los procesos del negocio en los que están involucrados y que serán descritos más adelante.

El sistema servirá de gran ayuda en la toma de decisiones de los jefes de departamentos a la hora de asignar tareas que pueden o no estar programadas, apoyándose en el sistema, analiza los datos de los profesores y decide a quien(es) se le asigna(n), de esta forma se controla de manera más eficiente su cumplimiento y todas estas actividades quedan registradas en el sistema.

Ocurre también que un profesor puede presentar problemas de cualquier índole y a través de una búsqueda avanzada se puede ver cuales actividades se verán afectadas.

2.2.4. Información que se maneja.

La información que va a ser procesada no se define como una resolución ministerial pero sí se ha querido tener un orden que sirva de mejor organización en el trabajo de los jefes de departamentos, es por eso que se idearon en documentos Word y Excel, tablas que recogieran los datos más relevantes para ser luego consultados, estas tablas varían según el departamento y los intereses del jefe del departamento y se presenta como uno de los principales problemas que afecta el control de esta información.

2.3. Propuesta del Sistema.

Sistema automatizado para la gestión de información de los profesores de la facultad.

Este sistema está diseñado con el objetivo de que los departamentos de la facultad 2 posean una herramienta rápida, de fácil uso y cómoda que les permita gestionar la información de sus profesores.

2.4. Modelo del negocio.

En los departamentos de la facultad 2 se maneja constantemente cierto volumen de información relacionada con los procesos en los que están involucrados sus profesores, se definirán estos procesos, logrando un entendimiento que sirva para llevar a cabo la implementación del sistema que se pretende, se definirán los actores, trabajadores y los procesos del negocio para lograr identificar las actividades a automatizar.

2.4.1. Actores del Negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [14]

2.4.2. Tabla 1. Definición de los Actores del Negocio.

| Actores del Negocio | Descripción |
|--------------------------------|---|
| Directivos | Este actor incluye el “Decano” y “Dirección de la asignatura_UCI” |
| Decano | Es el principal directivo de la facultad, el orienta nuevas tareas (tanto docente como administrativas), entre otras cosas y es uno de los beneficiados en el negocio |
| Dirección de la asignatura_UCI | Es el principal directivo de las asignaturas a nivel de universidad, el trabajo de los departamentos está orientado a las tareas que él orienta y se beneficia de ello. |

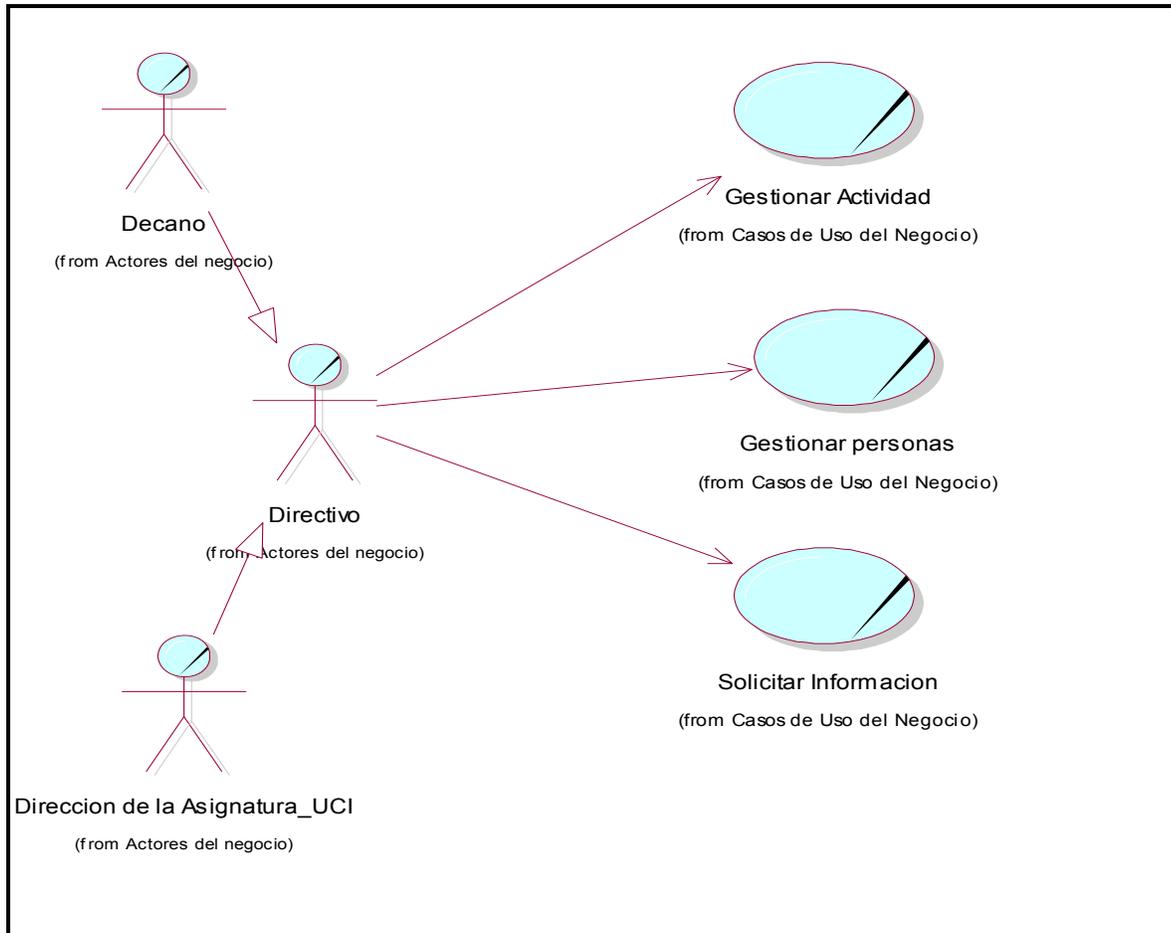
2.4.3. Definición de los trabajadores del negocio.

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. [14]

2.4.4. Tabla 2. Definición de los Trabajadores del Negocio.

| Trabajadores del Negocio | Descripción |
|--------------------------|--|
| Jefe de Departamento | Realiza la mayoría de las tareas del negocio |
| Profesor | Se subordina al Jefe de Departamento, realizando las actividades que este le asigna. |

2.4.5. Diagrama 1. Diagrama de casos de uso del negocio.



2.4.6. Descripción de los casos de uso en formato expandido y mediante los diagramas de actividades.

La descripción de los casos de uso del negocio muestra cómo colaboran los trabajadores y entidades del negocio para ejecutar el proceso. Son reseñas textuales del caso de uso y explica los procesos o actividades que tienen lugar en el mismo. [14] (Ver Anexo 1)

2.4.7. Diagramas de Actividad.

Los diagramas de actividades son provechosos para entender el comportamiento a un alto nivel de la ejecución de un sistema, sin profundizar en los detalles, describen la secuencia de las actividades. (Ver Anexo 2)

2.4.8. Diagrama de clases del modelo de objetos. CU Gestionar Actividad.

Representan la relación entre los trabajadores del negocio y las diferentes entidades del mismo. (Ver Anexo 3)

2.5. Relación de los Requerimientos.

¿Qué se entiende por requisito?

-Rational: Una condición o capacidad que el sistema en construcción debe cumplir.

-IEEE(Institute of Electrical and Electronic Engineers):Una funcionalidad del software necesaria para que el usuario resuelva su problema o cumpla sus objetivos. [15]

2.5.1. Listado de los requerimientos funcionales enumerados.

Los requerimientos funcionales son una tarea simple enunciada con un solo verbo. [15]

- RF1 Definir niveles de acceso.
 - RF1.1 Controlar que cada usuario acceda a la información que le corresponda
- RF2 Gestionar Profesor
 - RF2.1 Adicionar profesor.
 - RF2.2 Eliminar Profesor.
 - RF2.3 Modificar Profesor.
- RF3 Gestionar Tareas
 - RF3.1 Adicionar Tareas.
 - RF3.2 Modificar Tareas.

RF3.3 Eliminar Tareas.

- RF4 Solicitar Reporte.

RF4.1 Buscar y mostrar cualquier dato de los profesores a través de búsquedas generales o específicas.

RF4.2 Buscar y mostrar los datos de las actividades a través de búsquedas generales o específicas.

- RF5 Gestionar Departamentos.

RF5.1 Adicionar Departamentos.

RF5.2 Modificar Departamentos.

RF5.3 Eliminar Departamentos.

- RF6 Gestionar Grupos

RF6.1 Adicionar Grupo.

RF6.2 Eliminar Grupo.

- RF7 Gestionar Asignaturas

RF7.1 Adicionar Asignaturas.

RF7.2 Eliminar Asignaturas.

- RF8 Gestionar Asignatura por Departamentos.

RF8.1 Asignar a los departamentos asignaturas.

RF8.2 Modificar las asignaturas de los departamentos.

- RF9 Gestionar Contraseña.

RF9.1 Cambiar contraseña.

- RF10 Gestionar Usuario.

RF10.1 Adicionar usuario.

RF10.2 Modificar usuario.

RF10.3 Eliminar usuario.

- RF11 Gestionar Profesores por Tareas.

RF11.1 Asignar profesores participantes a las tareas.

RF11.2 Eliminar profesores participantes a las tareas.

RF11.3 Asignar profesor responsable a una tarea.

RF11.4 Modificar profesor responsable a una tarea.

RF11.5 Asignar profesores incumplidores por tareas.

RF11.6 Modificar profesores incumplidores por tareas.

2.5.2. Definición de los requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto de software debe tener. [15]

Apariencia o interfaz

Debe tener un entorno amigable que se identifique con la facultad.

El sistema debe tener una interfaz sencilla y fácil de usar por usuarios con pocos conocimientos informáticos.

Usabilidad

Fácil de aprender, el sistema tendrá una ayuda para una mejor utilización.

Seguridad

El sistema debe garantizar que todos sus servicios sean usados por el personal autorizado y la información sea modificada y vista únicamente por quien tenga permiso para esto.

Disponibilidad.

El sistema debe permitir a los usuarios conectarse desde cualquier PC de la red de la UCI.

Portabilidad

Multiplataforma.

Rendimiento.

Debe estar disponible las 24 horas.

Hardware.

Para el servidor Web y de Base de Datos, PC Pentium 4 ,1.60Ghz de velocidad, 256 MB de RAM y como mínimo 5G de espacio en disco duro.

Restricciones en el diseño y la implementación

El sistema será implementado en el lenguaje de programación Web PHP.

El gestor de base de datos a utilizar será Postgre.

2.6. Modelo de Casos de Uso del Sistema.

Los CU son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para los actores.

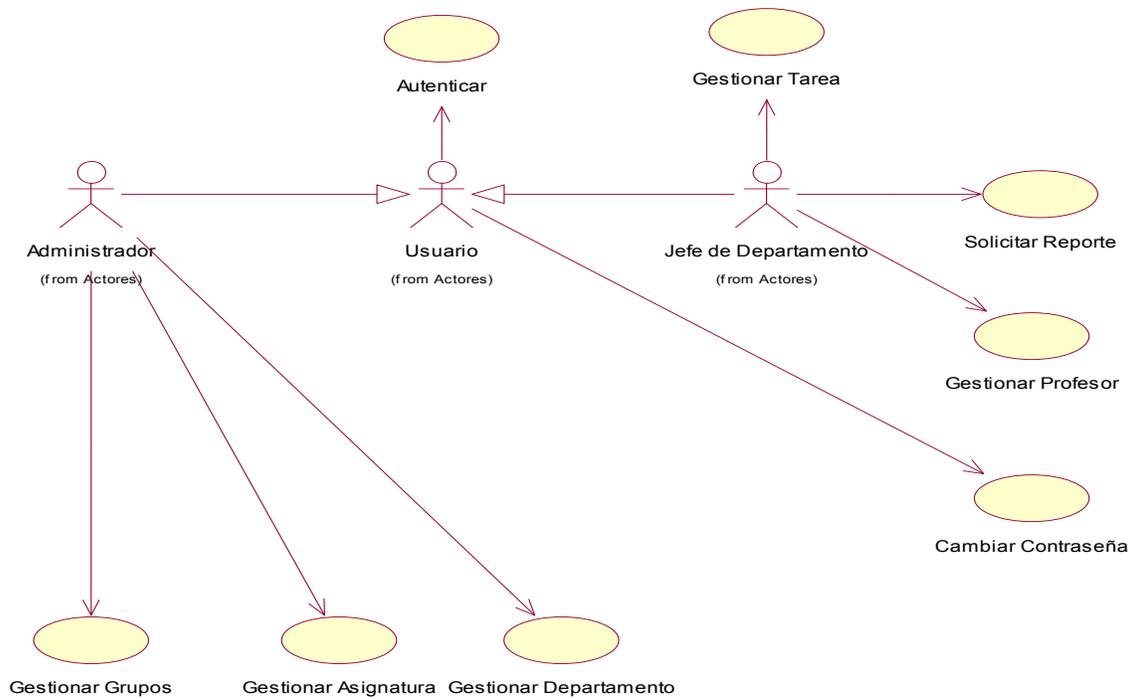
2.6.1. Definición de los Actores del Sistema.

Un actor representa cualquier cosa que interactúe con el sistema.

2.6.2. Tabla 6. Actores del Sistema.

| Actor | Justificación |
|----------------------|---|
| Jefe de departamento | Es el profesor Jefe de Departamento el gestiona las tarea (crea, modifica, elimina), los profesores, visualiza la información a través de los reportes que es de gran ayuda para la toma de decisiones, estos lo convierte en el principal beneficiado. |
| Administrador | Es la persona que se encarga de la administración del sistema, inicia los casos de uso: Gestionar Permisos, Gestionar Departamento, Gestionar Asignatura, Gestionar Grupo, se recomienda que sea un profesor de mayor rango que los jefes de departamentos. |
| Usuario | Es cualquiera de los dos actores antes mencionados, su principal objetivo es el de iniciar el caso de uso Autenticar y Cambiar Contraseña condición indispensable para acceder al sistema. |

2.6.3. Diagrama de casos de uso del sistema a automatizar.



2.6.4. Descripción del caso Casos de uso expandidos.

(Ver Anexo 4)

2.7. Conclusiones del Capítulo.

En este capítulo se analizó el negocio de la entidad y a partir de ello se propuso las posibles actividades a automatizar, además se definieron los requisitos funcionales y no funcionales que el sistema debe cumplir.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción.

En este capítulo se profundizará en los casos de usos del sistema, detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos y encontrar la forma para que se soporten todos los requerimientos y crear una entrada apropiada y punto de partida para la implementación.

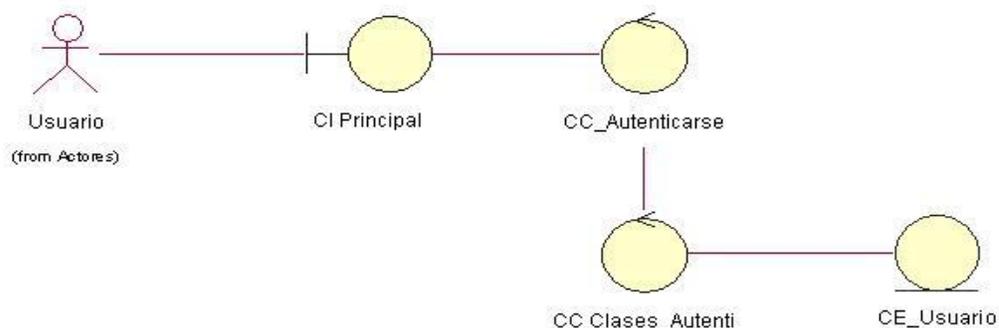
3.2. Análisis.

3.2.1. Modelo de Análisis.

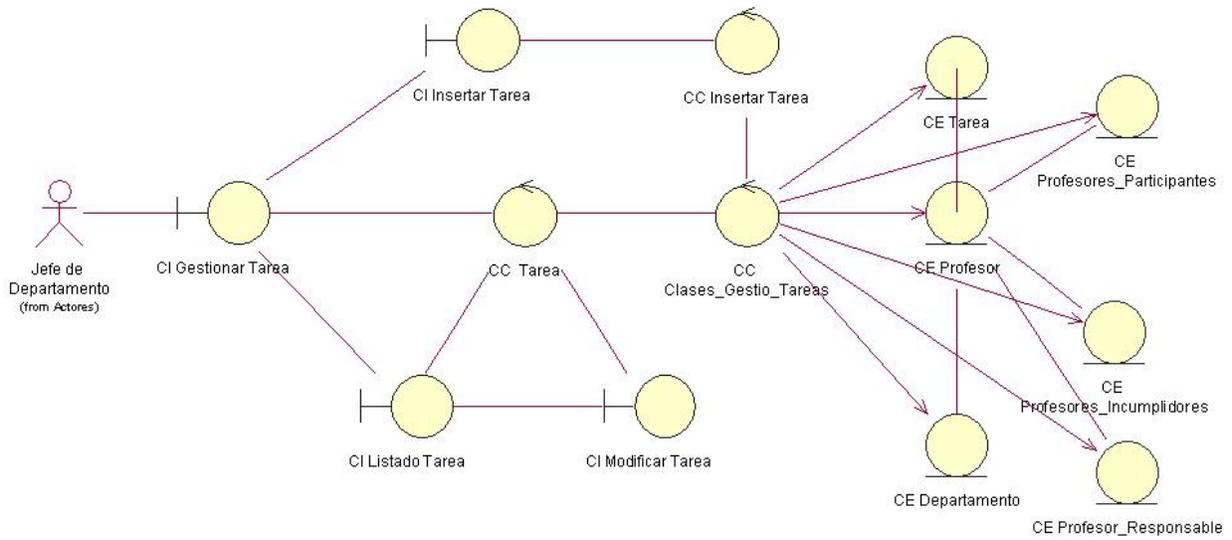
En el modelo del análisis hay un refinamiento de los requisitos del software, teniendo como objetivo comprender perfectamente los mismos. Esta estructura (basada en clases de análisis y paquetes) es independiente de la estructura que se dio a los requisitos (basada en casos de uso). [16]

El modelo de análisis debe ser una representación mínima del sistema que se está modelando, suficiente para capturar la lógica esencial del sistema, sin entrar en temas de rendimiento o construcción. [16]

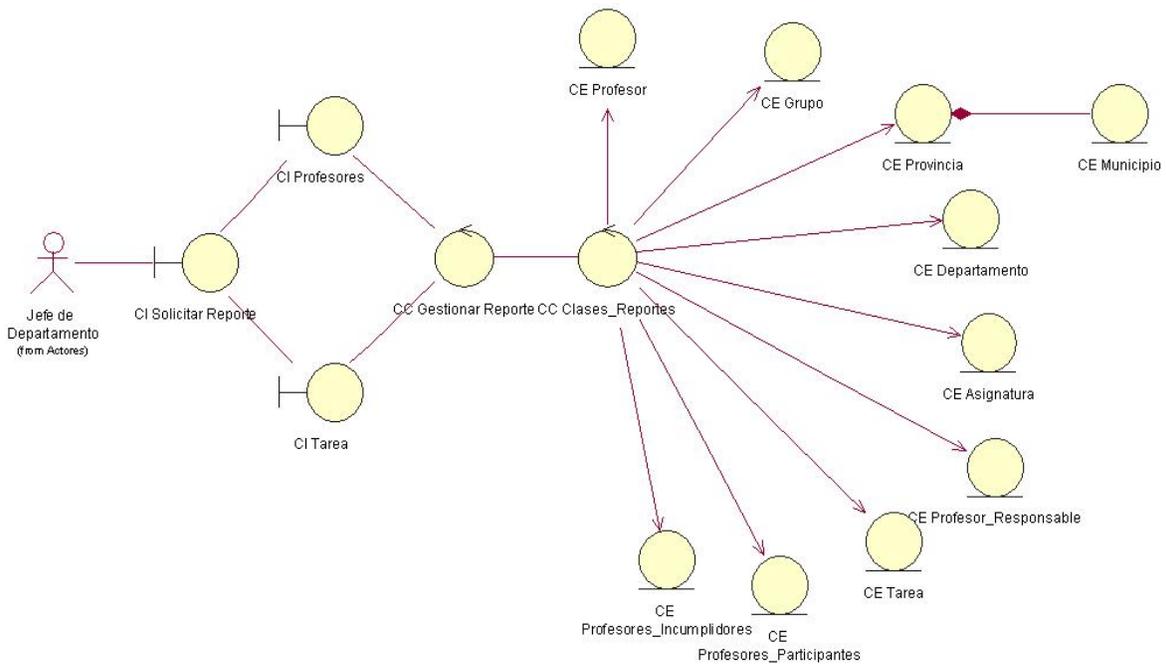
3.2.2. Diagrama 7. Diagrama de clases de análisis. CU Autenticar.



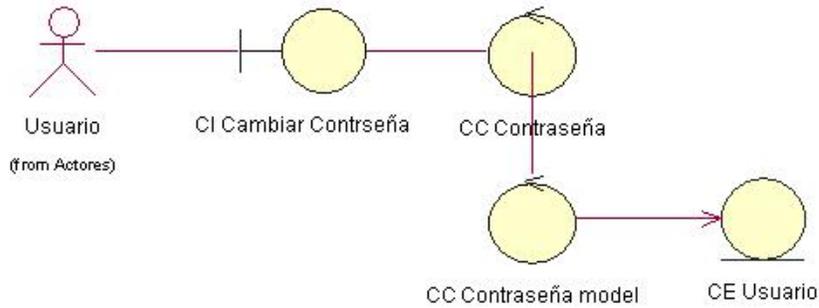
3.2.4. Diagrama 9. Diagrama de clases de análisis. CU Gestionar Tarea.



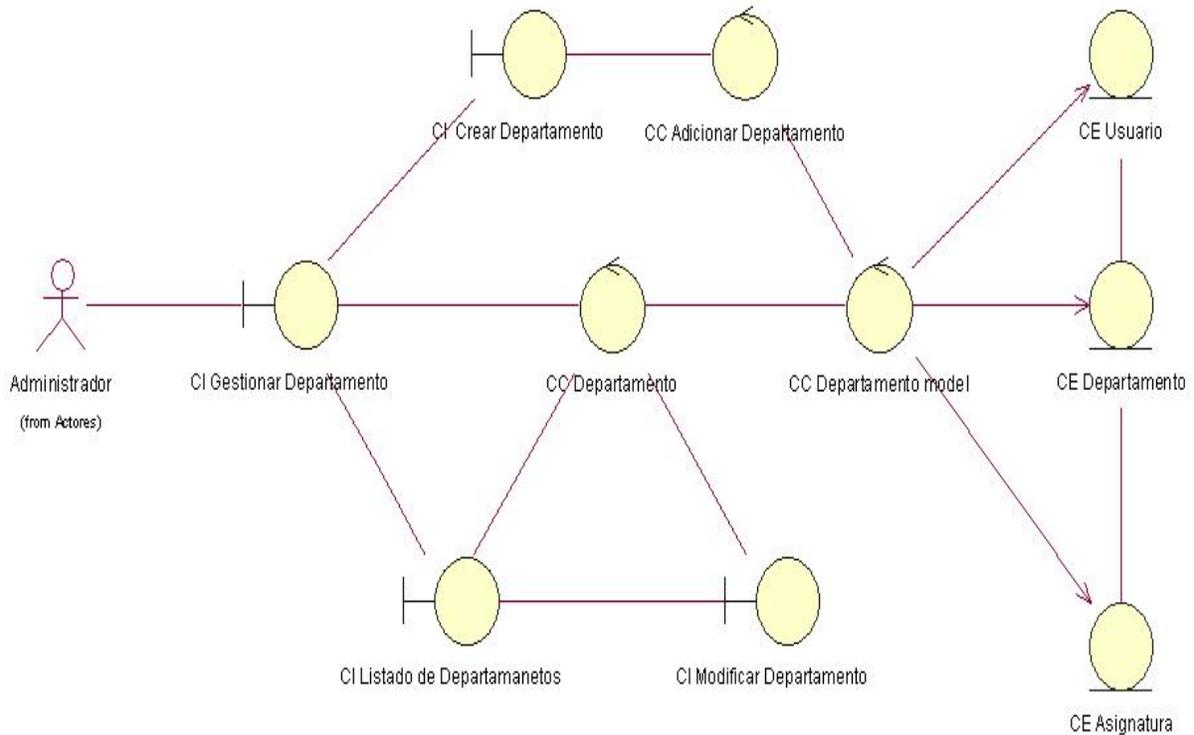
3.2.5. Diagrama 10. Diagrama de clases de análisis. CU Solicitar Reporte.



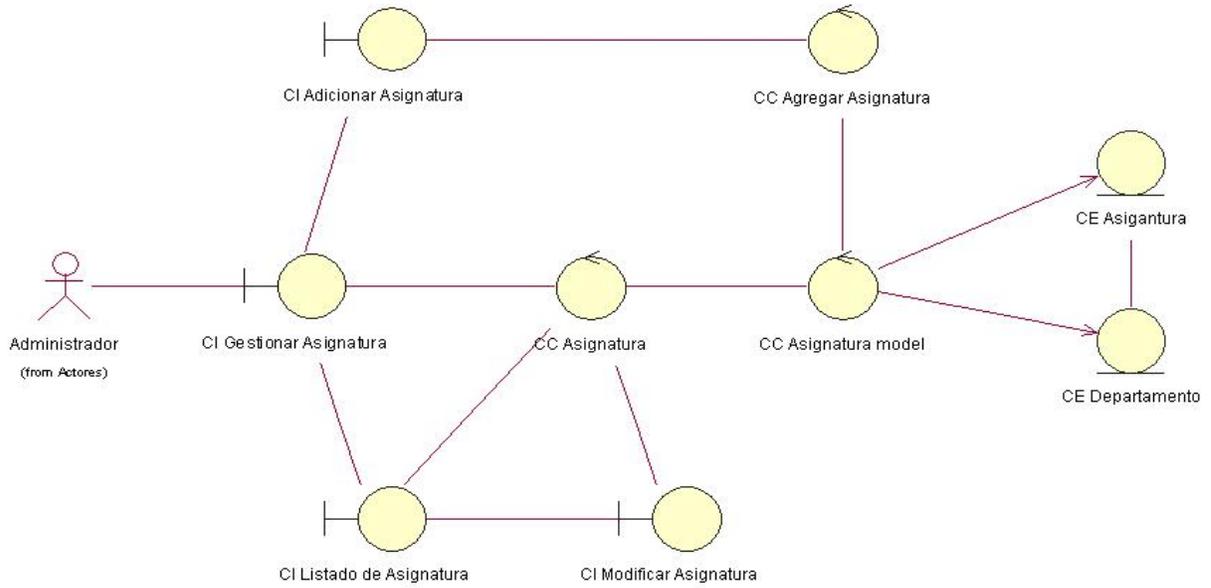
3.2.6. Diagrama 12. Diagrama de clases de análisis. CU Cambiar Contraseña.



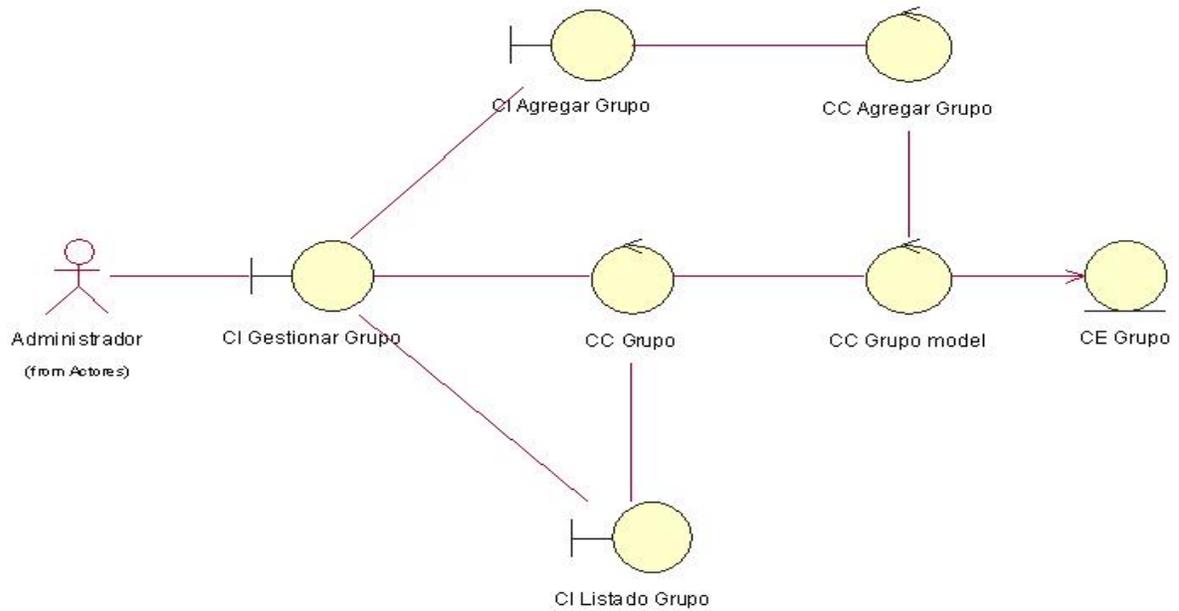
3.2.7. Diagrama 13. Diagrama de clases de análisis. CU Gestionar Departamento



3.2.8. Diagrama 14. Diagrama de clases de análisis. CU Gestionar Asignatura.



3.2.9. Diagrama 15. Diagrama de clases de análisis. CU Gestionar Grupo.



3.3. Modelos del Diseño.

El modelo de diseño es un modelo de objetos que:

- Describe la realización física de los casos de uso.
- Se centra en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a considerar.
- Descompone los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. [17]

3.3.1 Diagramas de interacción.

Los diagramas de interacción modelan aspectos dinámicos del sistema, muestra una interacción, muestran las interacciones entre objetos mediante transferencia de mensajes entre objetos o subsistemas. (Ver Anexo 5)

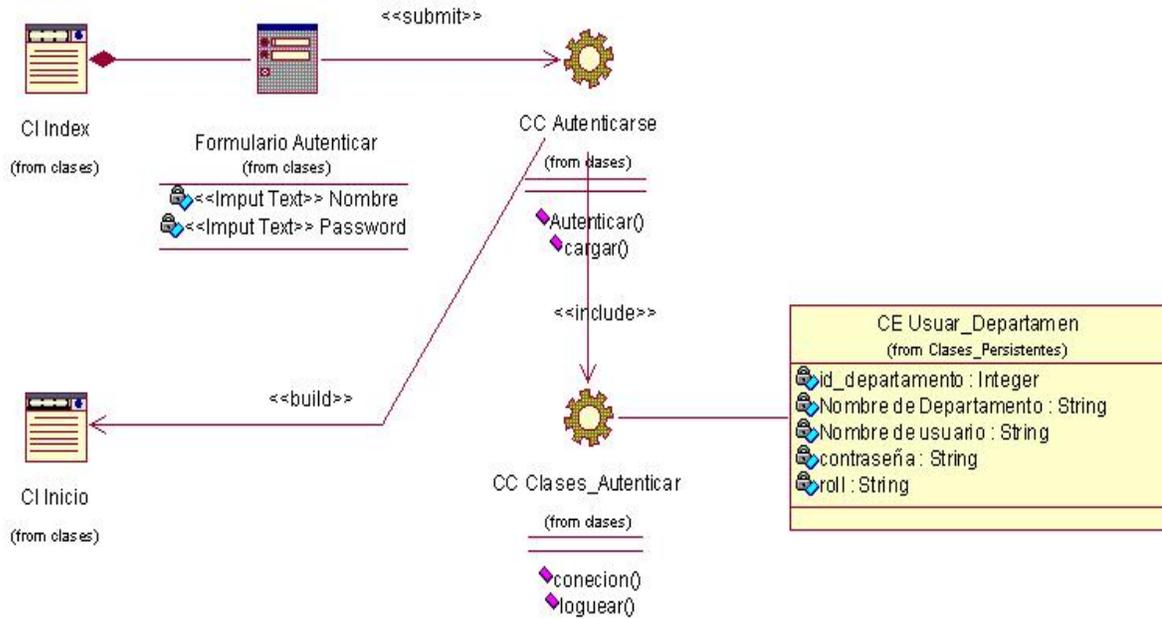
- Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes.
- Un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

Para modelar los aspectos dinámicos de este sistema se utilizaron diagramas de secuencia por cada caso de uso. [17]

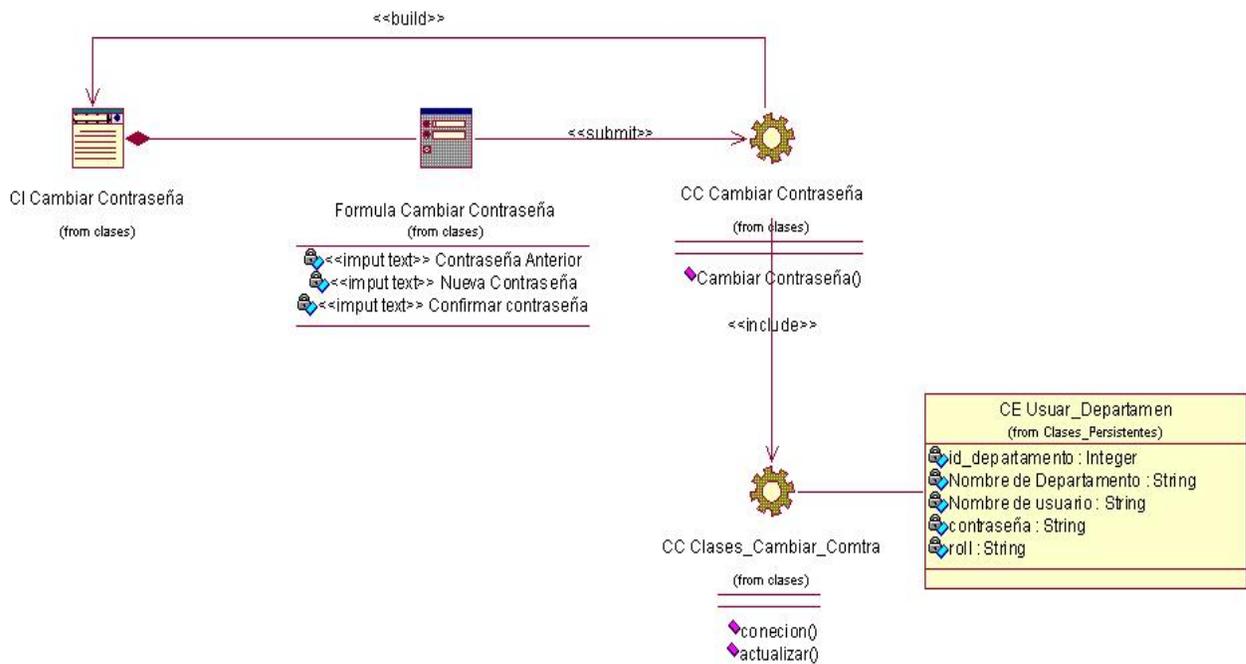
3.3.2. Diagramas de clases del diseño.

En este tipo de diagramas se especifica la estructura de las clases del sistema, incluyendo las relaciones entre clases. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal, durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de la implementación. [18]

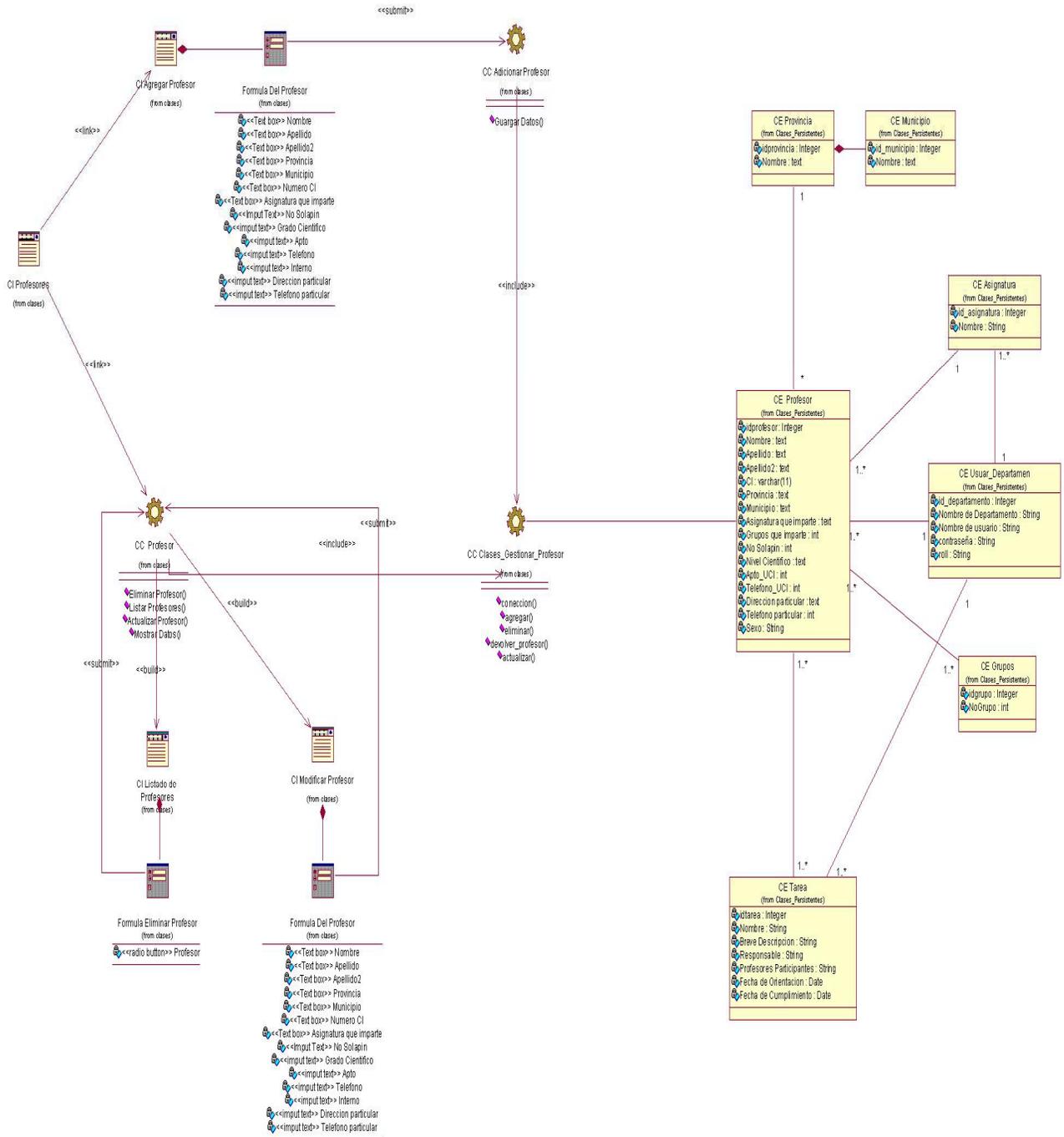
3.3.3. Diagrama 16. Diagrama de clases del diseño. CU Autenticar



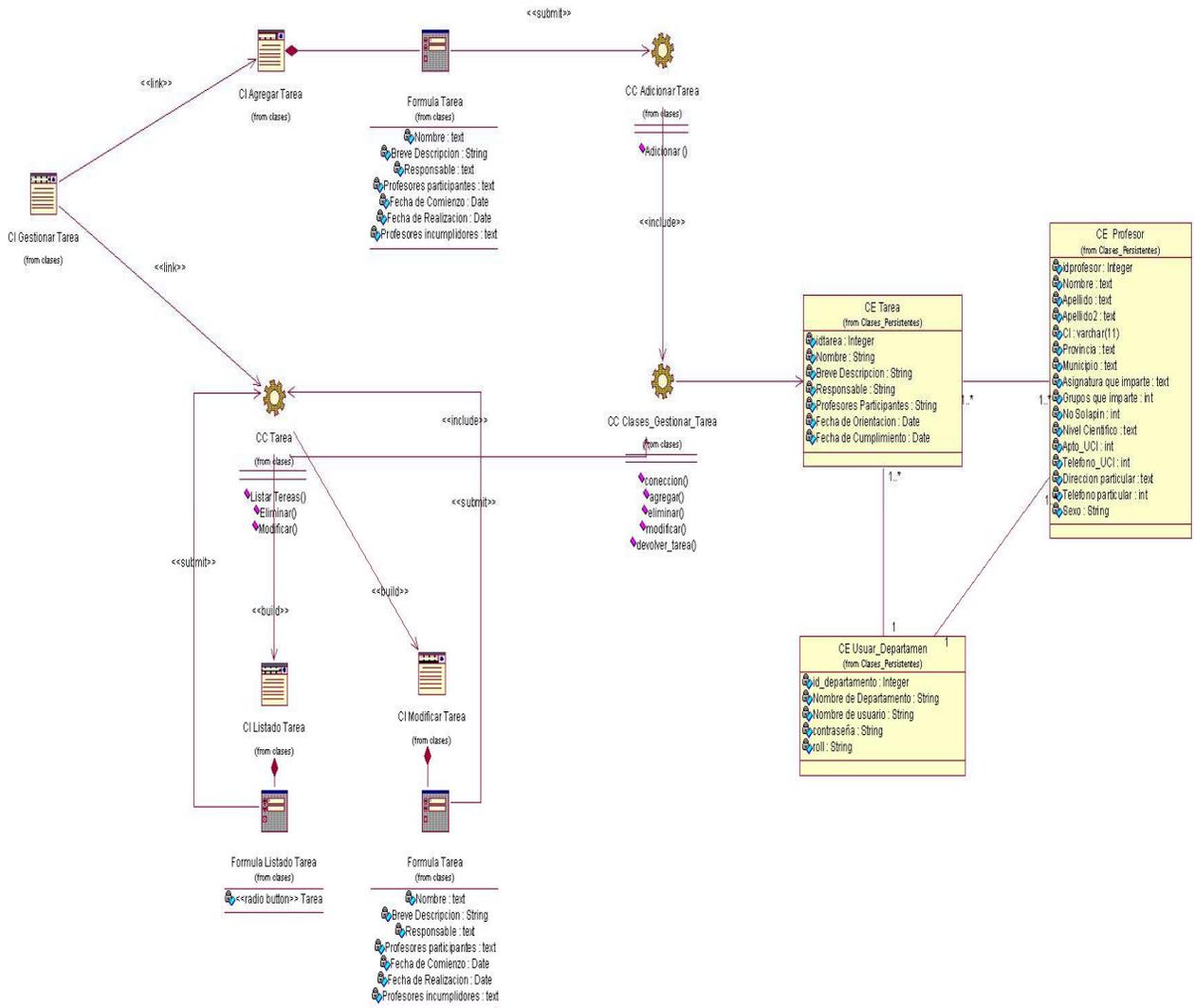
3.3.4. Diagrama 17. Diagrama de clases del diseño. CU Cambiar Contraseña.



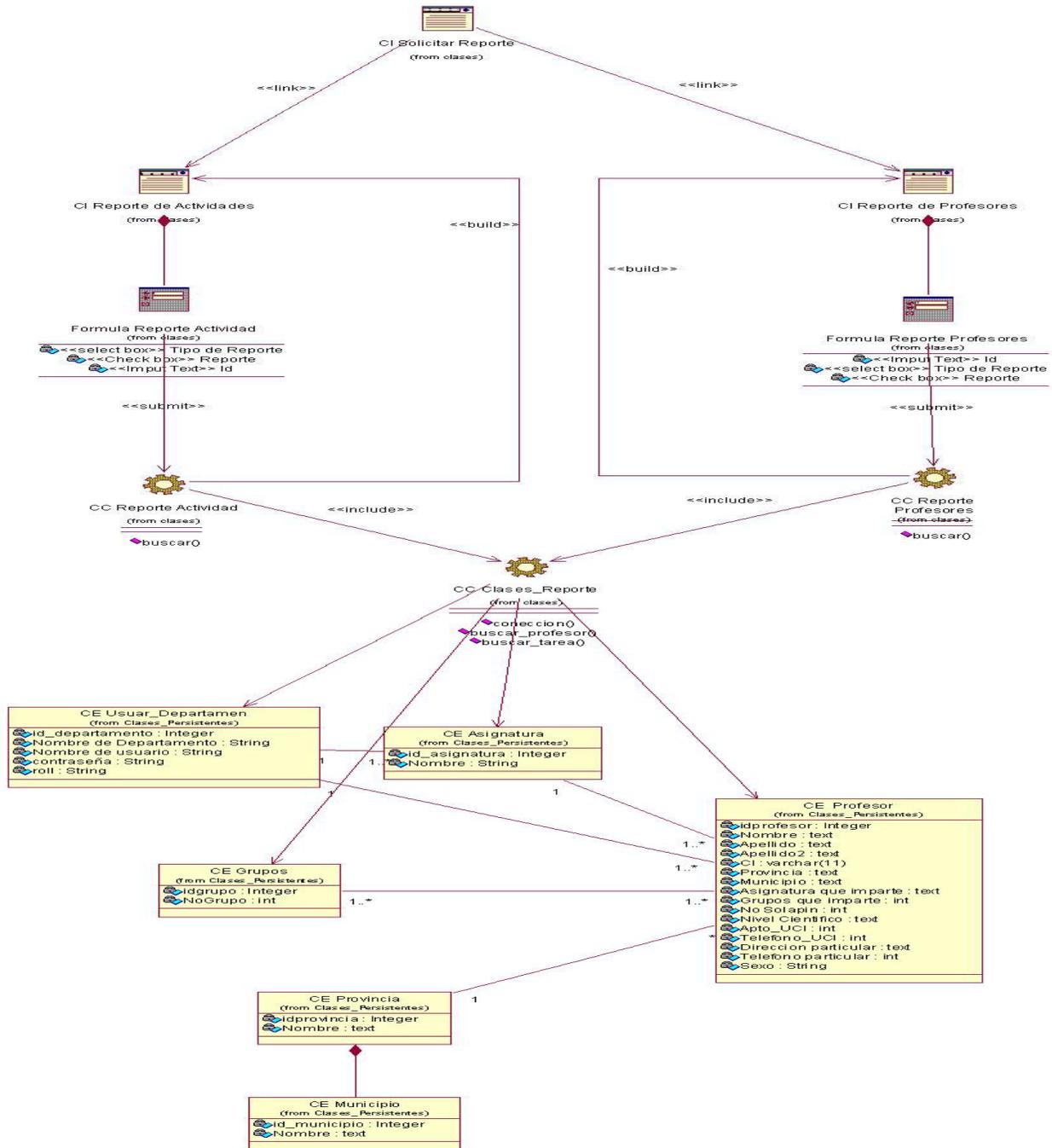
3.3.5. Diagrama 18. Diagrama de clases del diseño. CU Gestionar Profesor.



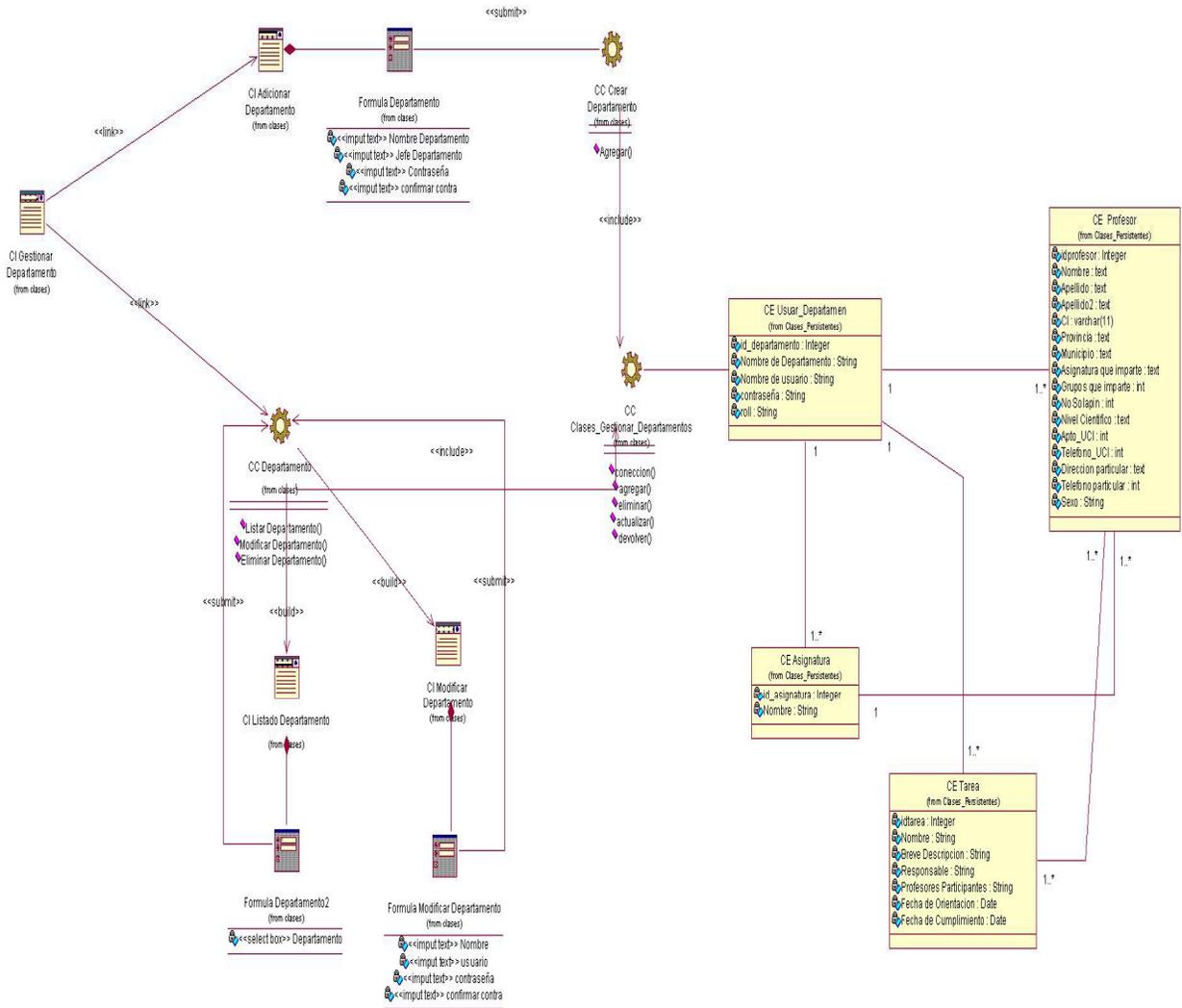
3.3.6. Diagrama 19. Diagrama de clases del diseño. CU Gestionar Tarea.



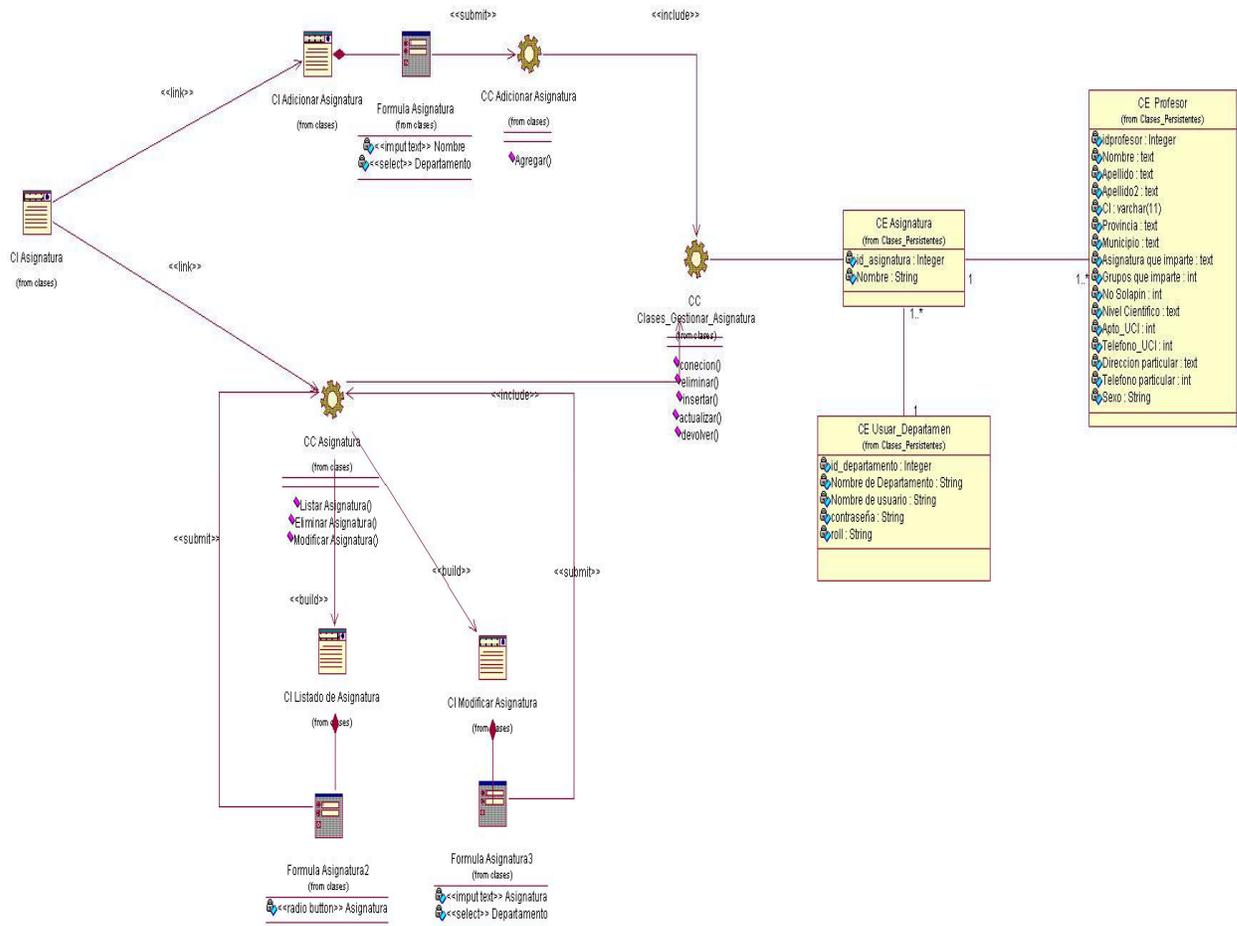
3.3.7. Diagrama 20. Diagrama de clases del diseño. CU Solicitar Reporte.



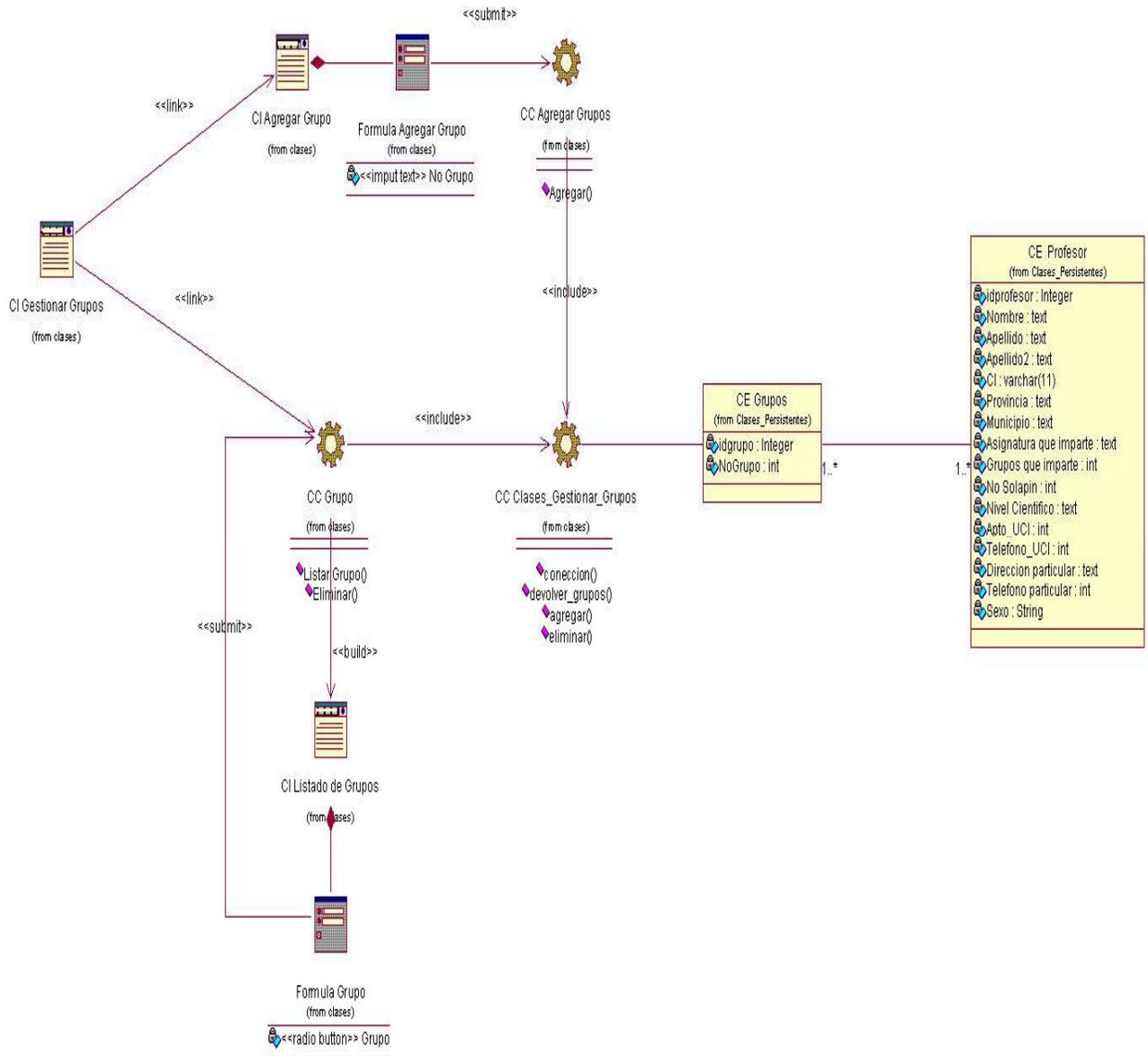
3.3.8. Diagrama 21. Diagrama de clases del diseño. CU Gestionar Departamento.



3.3.9. Diagrama 22. Diagrama de clases del diseño. CU Gestionar Asignatura.



3.3.10. Diagrama 23. Diagrama de clases del diseño. CU Gestionar Grupo.

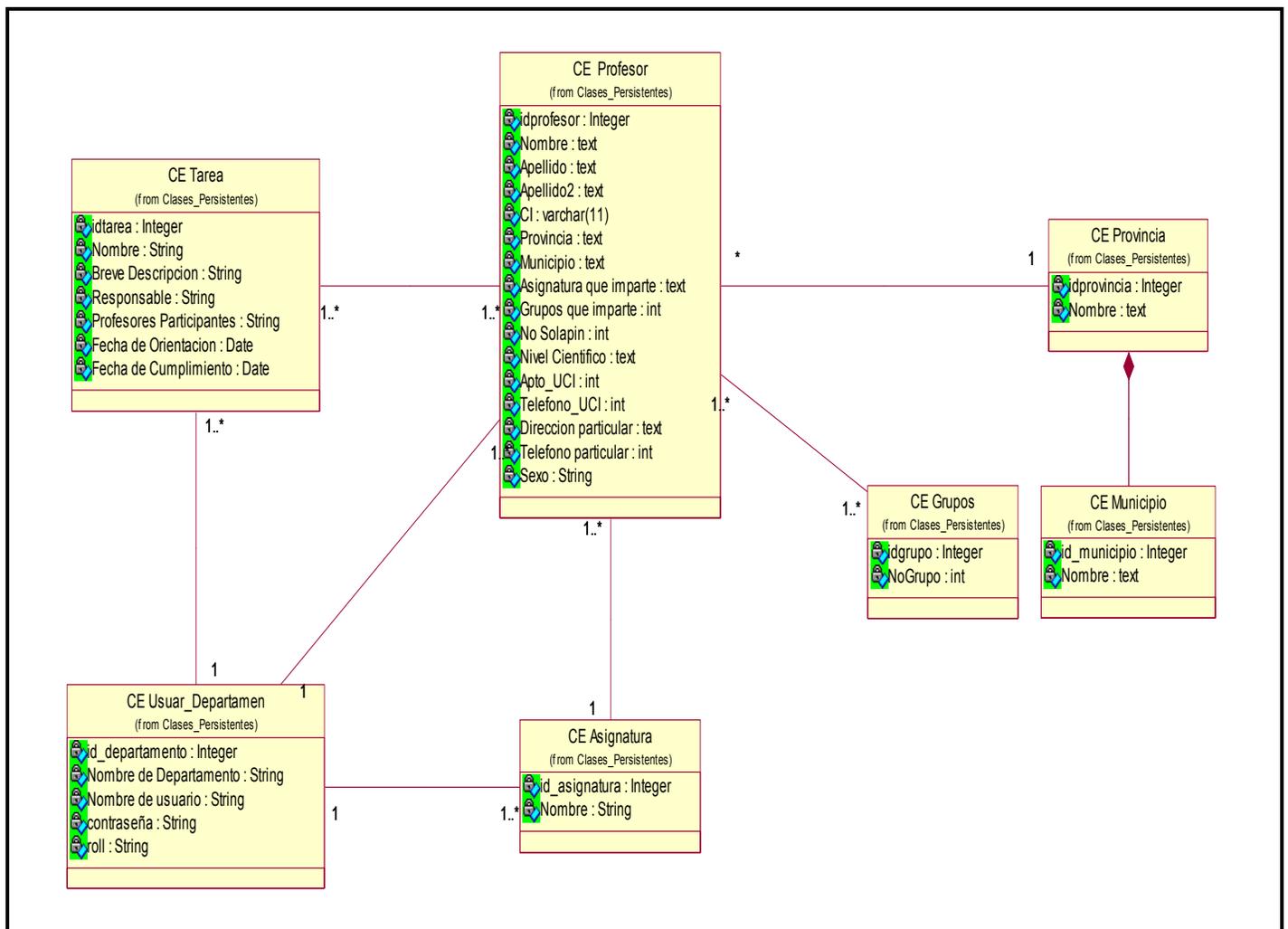


3.4. Diseño de la Base de Datos.

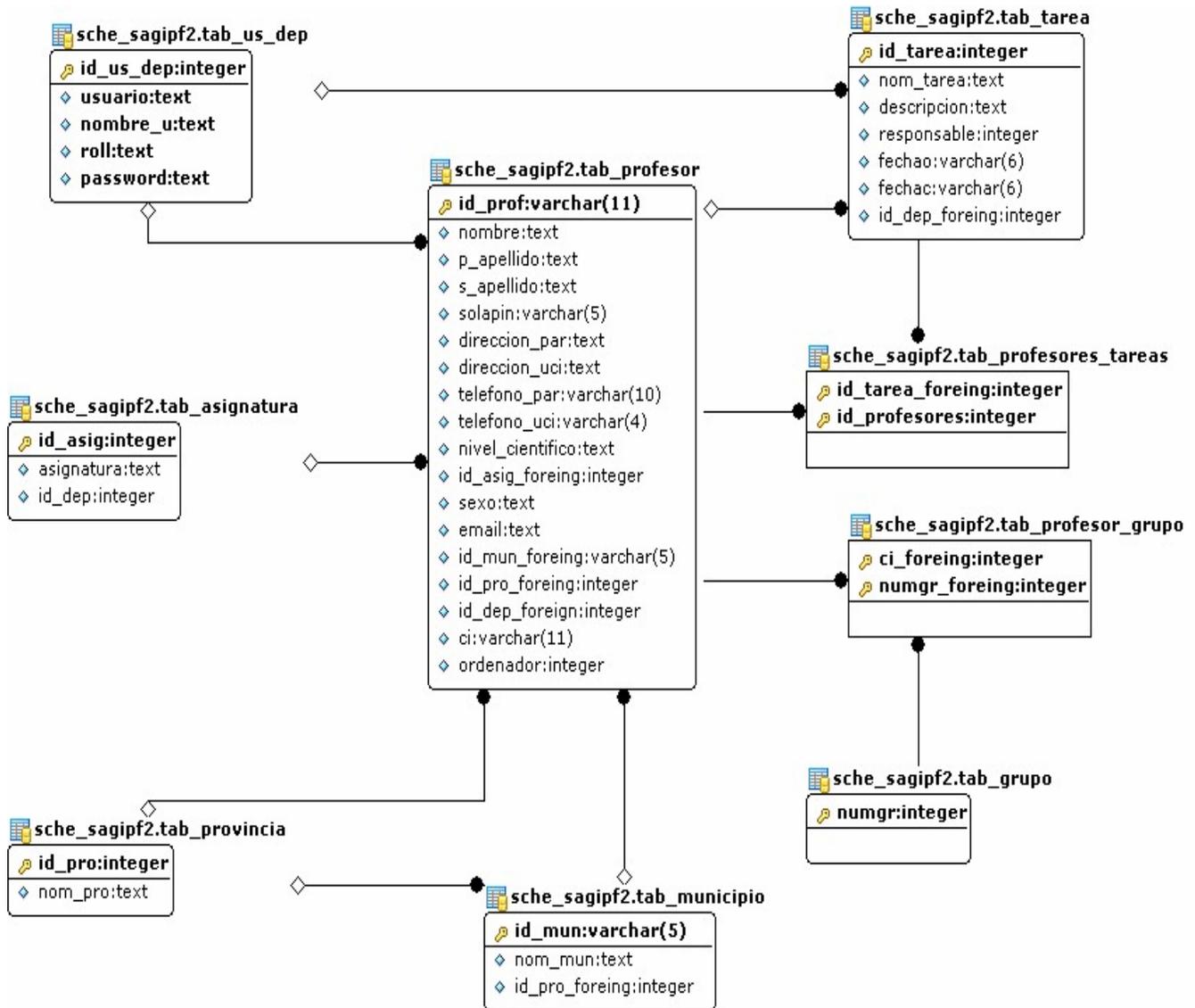
3.4.1. Modelo lógico de datos.

El modelo lógico de datos provee de una vista de las entidades lógicas de datos y sus relaciones con independencia de la plataforma de base de datos a utilizar. [19]

3.4.2. Diagrama de clases persistentes.



3.4.3. Modelo físico de datos.



3.5. Conclusiones del Capítulo.

En este capítulo se ha profundizado en las características del futuro sistema y se modeló detalladamente la solución propuesta, a través del modelo de diseño, se representó el modelo lógico de los datos así como el modelo físico.

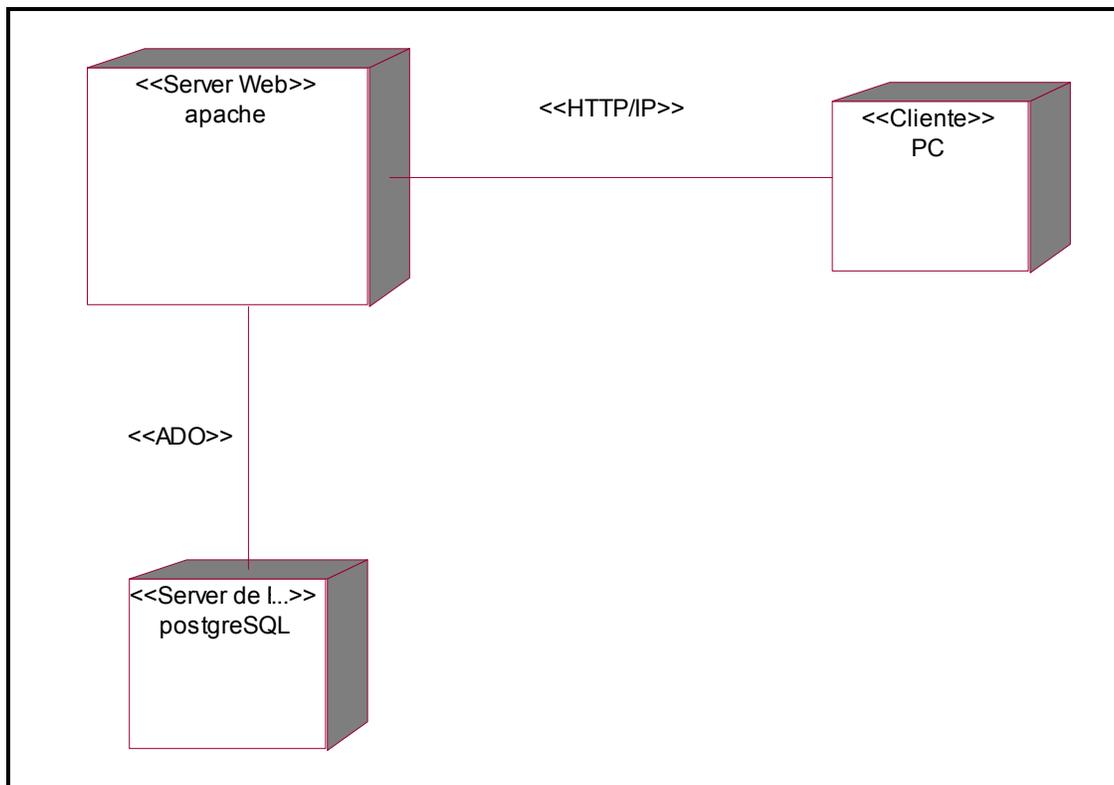
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.

4.1. Introducción al capítulo.

En este capítulo se describe la distribución física del sistema y como se distribuyen las funcionalidades entre los nodos de cómputo, así como los elementos físicos (componentes) que pueden hallarse dentro de estos, como son: ejecutables, librerías, ficheros de código fuentes y poder ver así la correspondencia de los componentes sobre los nodos, todo esto a través de los diagramas de despliegue y componentes.

4.2. Diagrama de despliegue.

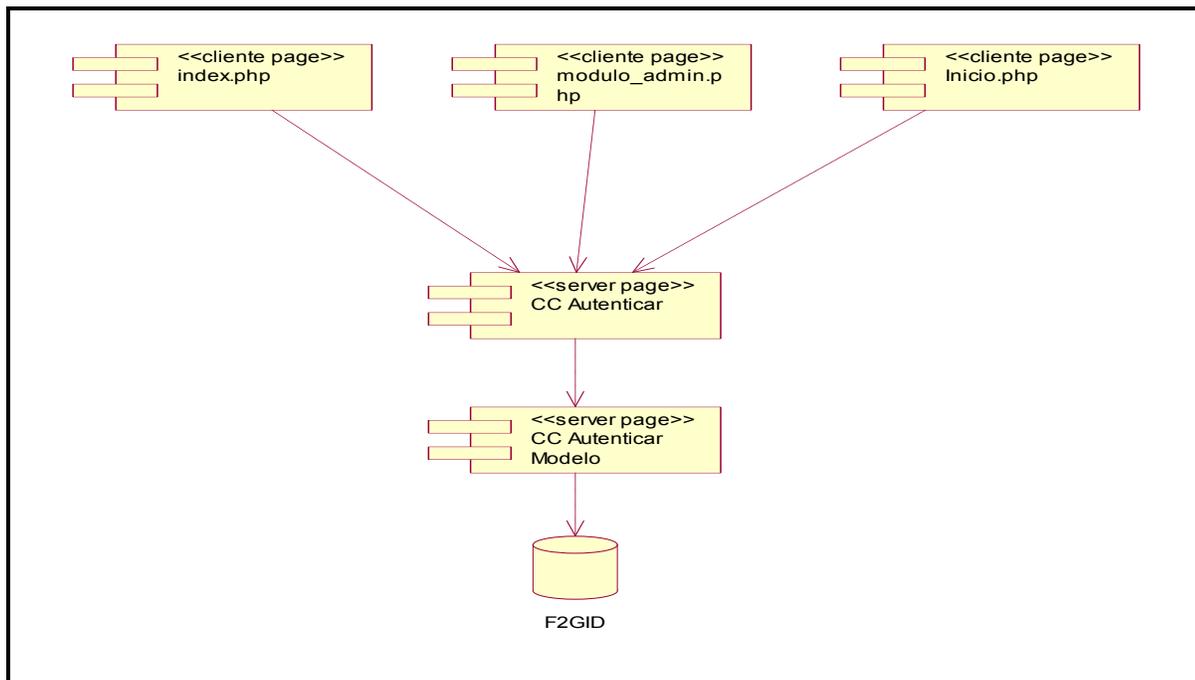
El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputos. Los diagramas de despliegue representan la topología del hardware, indica la situación física de los componentes lógicos desarrollados, se sitúa el software en el hardware que lo contiene. [20]



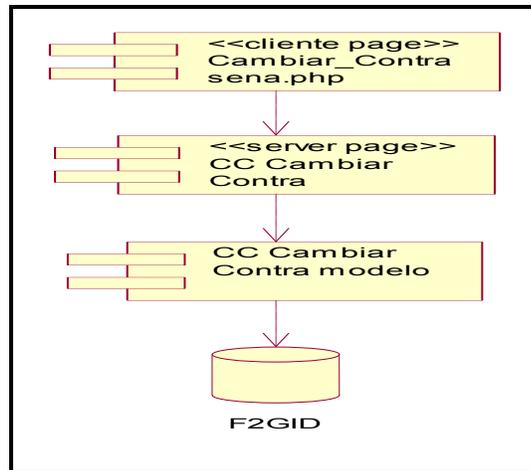
4.3. Diagrama de componentes.

Los componentes se utilizan para modelar los elementos físicos que pueden hallarse en un nodo, tales como ejecutables, bibliotecas, tablas, archivos y documentos, un componente es una parte física y reemplazable de un sistema. [20]

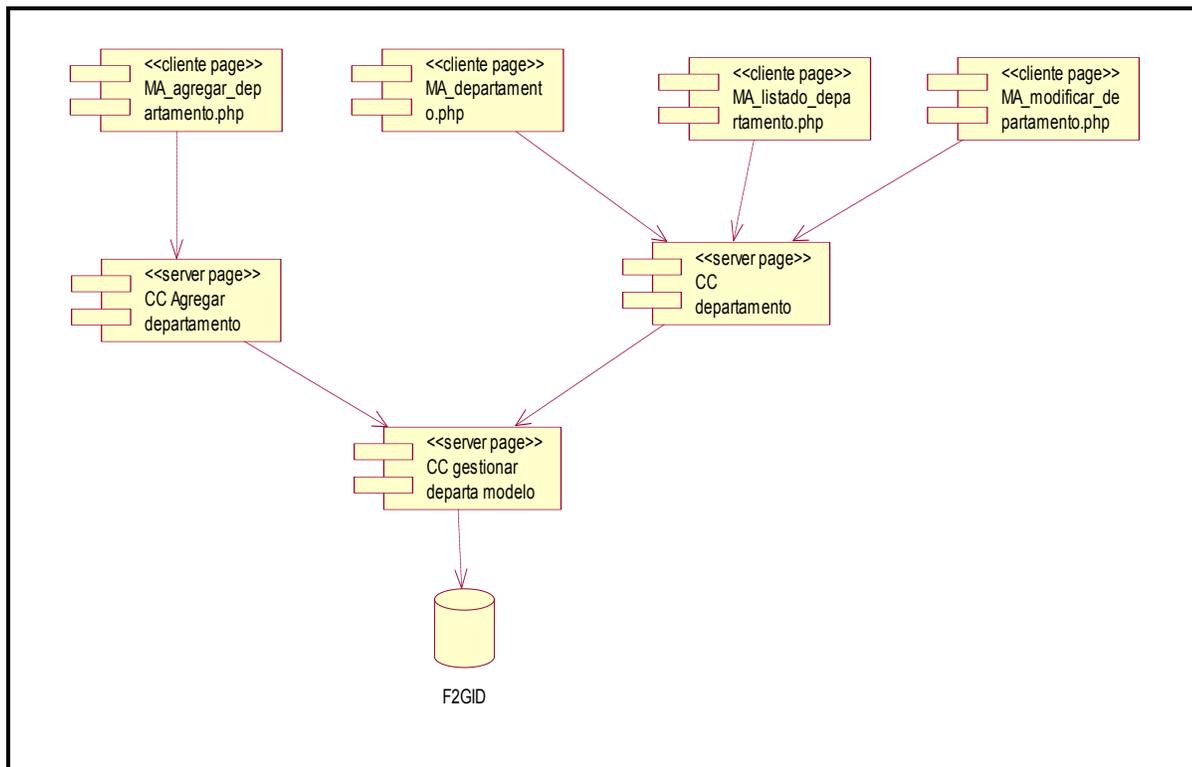
4.3.1. Subsistema de implementación Autenticarse.



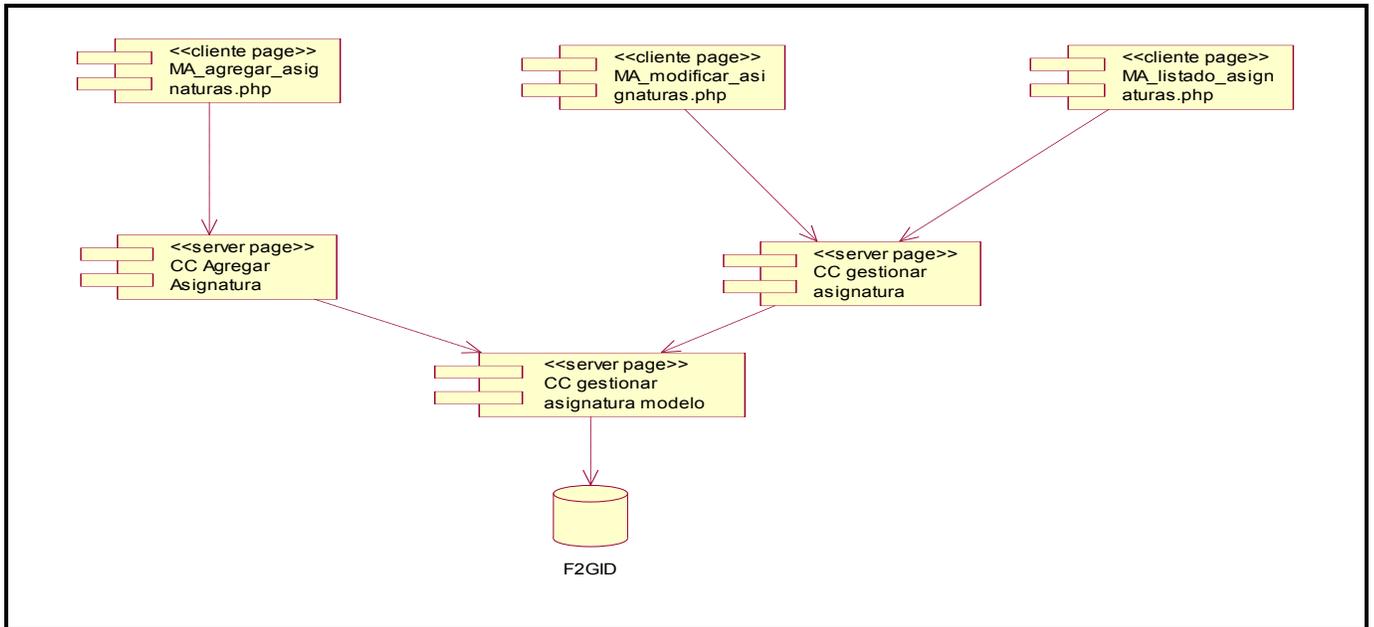
4.3.2. Subsistema de implementación Cambiar contraseña.



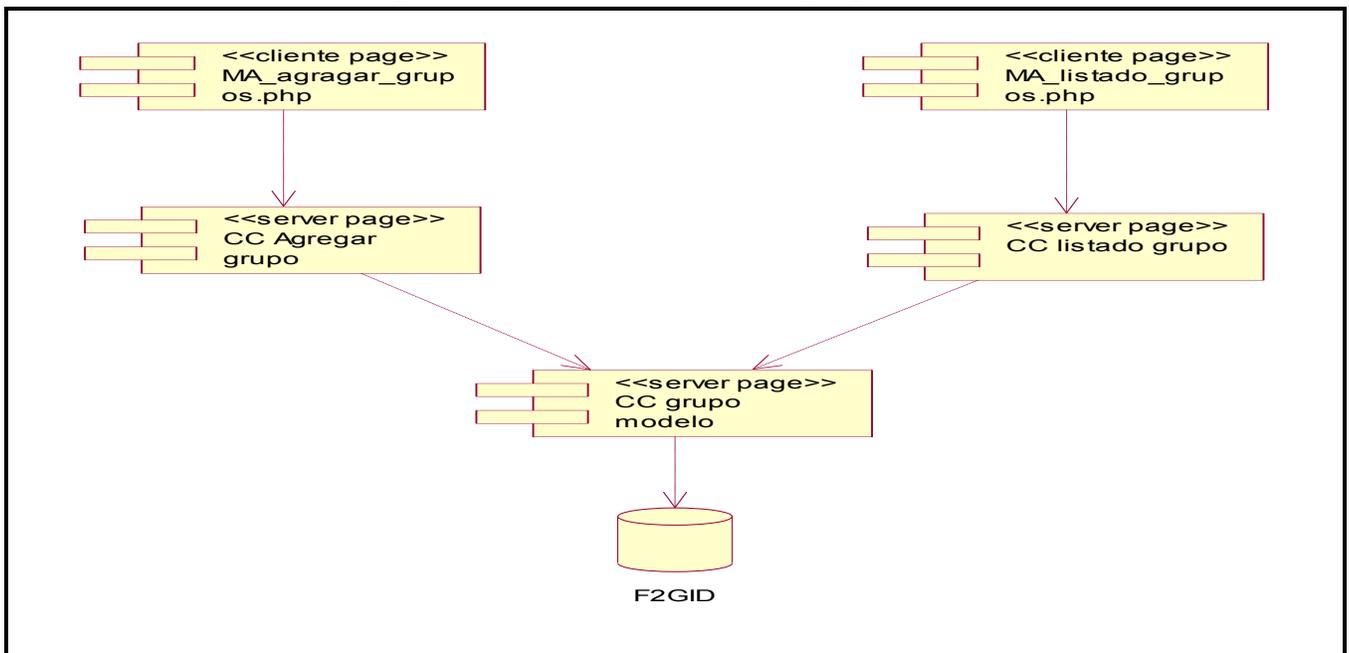
4.3.3. Subsistema de implementación Gestiona Departamento.



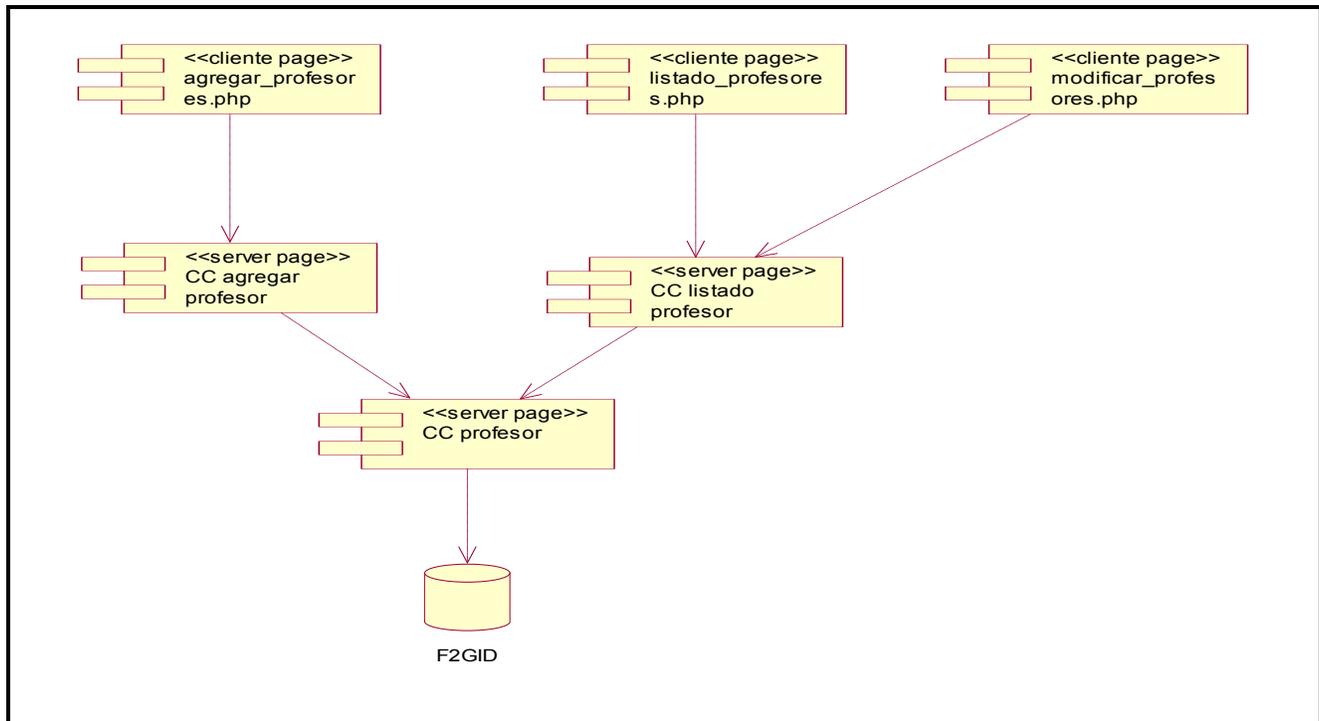
4.3.4. Subsistema de implementación Gestionar Asignatura.



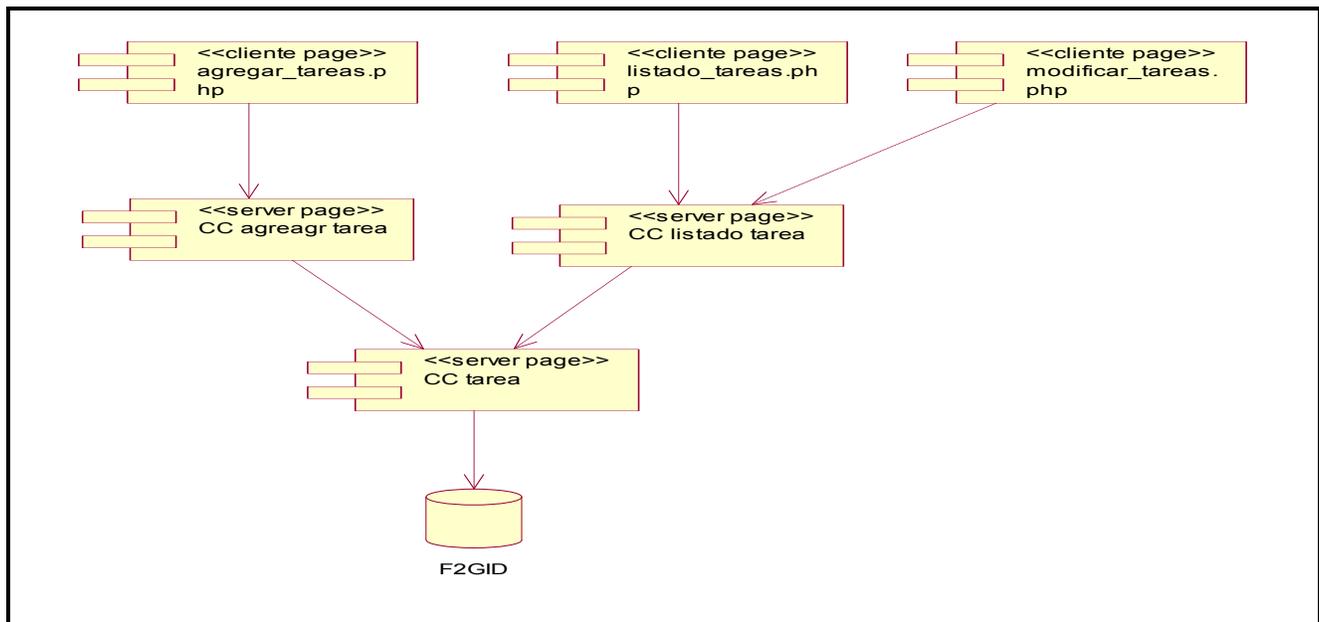
4.3.5. Subsistema de implementación Gestionar grupo.



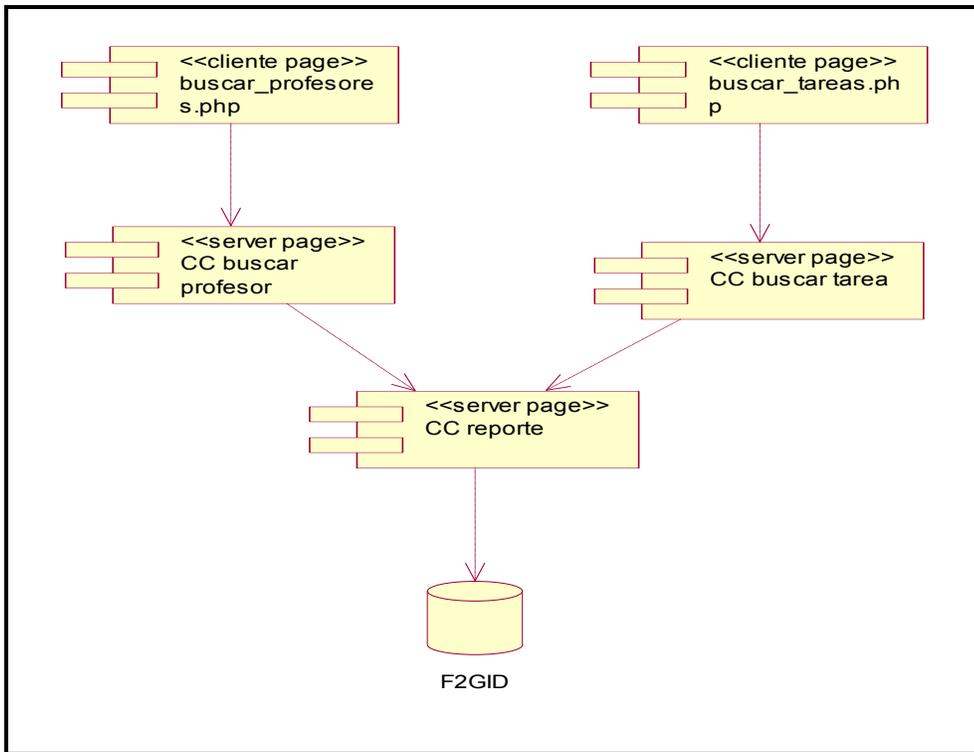
4.3.6. Subsistema de implementación Gestionar profesor.



4.3.7. Subsistema de implementación Gestionar tarea.



4.3.8. Subsistema de implementación Solicitar reporte.



4.4. Conclusiones del Capítulo.

En este capítulo se realizó un refinamiento de la vista de la arquitectura a través de los diagramas de despliegue y componente y así ver la dependencia entre los diferentes componentes y la situación física de los mismos.

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.

5.1. Introducción.

Es de gran importancia para asumir la realización de un proyecto conocer la factibilidad y viabilidad del mismo, es primordial conocer los beneficios que reportará. En este capítulo se hace un estudio que permite conocer y estimar el tiempo de desarrollo en meses, el costo y la cantidad de personas que se necesitan para desarrollar el proyecto.

5.2. Planificación basada en casos de usos.

Cálculo de los Puntos de casos de uso Desajustados.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

5.2.1. Tabla 5.1 Factor de peso de los actores sin ajustar.

| Tipo de actor | Descripción | Factor de peso | Actores | Total |
|---------------|---|----------------|---------|-------|
| Simple | Sistema con sistema a través de interfaz de programación. | 1 | 0 | 0 |
| Medio | Sistema con sistema mediante protocolo de interfaz basada en texto. | 2 | 0 | 0 |
| Complejo | Persona que interactúa con el sistema mediante interfaz gráfica. | 3 | 3 | 9 |
| Total | | | 3 | 9 |

UAW= Σ cant actores*peso

UAW=3*3

UAW=9

5.2.2. Tabla 5.2 Factor de peso de los casos de uso sin ajustar.

| Tipo de CU | Descripción | Peso | Cantidad de CU | Total |
|------------|--|------|----------------|-------|
| Simple | El caso de uso tiene de 1 a 3 transacciones. | 5 | 1 | 5 |
| Medio | El caso de uso tiene de 4 a 7 transacciones. | 10 | 2 | 20 |
| Complejo | El caso de uso tiene más de 8 transacciones. | 15 | 5 | 75 |
| Total | | | | 75 |

UUCW= Σ cantCU*Peso

UUCW=75

UUCP=UAW+UUCW

UUCP=9+75

UUCP=84

5.2.3. Cálculo de los Puntos de casos de uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

5.2.4. Tabla 5.3 Factor de complejidad técnica.

| Factor | Descripción | Peso | Valor asignado | Comentario | Total |
|--------|---------------------------------|------|----------------|---|-------|
| T1 | Sistema distribuido | 2 | 0 | El sistema es centralizado. | 0 |
| T2 | Tiempo de respuesta | 1 | 3 | La velocidad es relativamente Alta. | 3 |
| T3 | Eficiencia del usuario final | 1 | 3 | Abundantes restricciones de eficiencia. | 3 |
| T4 | Funcionamiento Interno complejo | 1 | 3 | La complejidad es normal. | 3 |
| T5 | El código debe ser reutilizable | 1 | 3 | Se requiere que el código sea reutilizable. | 3 |
| T6 | Facilidad de instalación | 0,5 | 5 | Algunos requerimientos de facilidad de instalación. | 2,5 |
| T7 | Facilidad de uso | 0,5 | 5 | Elevada. | 2,5 |
| T8 | Portabilidad | 2 | 5 | Se requiere que el sistema sea portable. | 5 |
| T9 | Facilidad de cambio | 1 | 4 | Se requiere un costo relativamente alto de mantenimiento. | 4 |
| T10 | Concurrencia | 1 | 5 | Hay concurrencia. | 5 |

| | | | | | |
|-------|--|---|---|---|----|
| T11 | Incluye objetivos especiales de seguridad | 1 | 2 | Seguridad Normal | 2 |
| T12 | Provee acceso directo a terceras partes | 1 | 0 | No todos los usuarios web tienen acceso directo. | 0 |
| T13 | Se requieren facilidades especiales de entrenamiento de usuarios | 1 | 0 | Pocos usuarios internos, el sistema es relativamente fácil de usar. | 0 |
| Total | | | | | 38 |

$$TCF=0.6+0.01* \Sigma (\text{peso}*\text{valor asignado})$$

$$TCF=0.6+0.01*38$$

$$TCF=0.6+0.22$$

$$TCF=0.98$$

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

5.2.5. Tabla 5.4 Factor de ambiente.

| Factor | Descripción | Peso | Valor asignado | Comentario | Total |
|--------|--|------|----------------|---|-------|
| E1 | Familiaridad con el modelo de proyecto utilizado | 1,5 | 3 | El grupo está bastante familiarizado con el Modelo. | 4,5 |
| E2 | Experiencia en la aplicación | 0,5 | 3 | La mayoría del grupo ha trabajado mucho tiempo en | 1,5 |

| | | | | | |
|-------|--|-----|---|---|------|
| | | | | ésta aplicación. | |
| E3 | Experiencia en la orientación a objetivos. | 1 | 3 | La mayoría del grupo programa en Objetos. | 3 |
| E4 | Capacidad del analista líder. | 0,5 | 3 | Tiene bastante Experiencia. | 1,5 |
| E5 | Motivación. | 1 | 5 | El grupo está altamente motivado. | 5 |
| E6 | Estabilidad de requerimientos | 2 | 4 | No se esperan muchos cambios. | 8 |
| E7 | Personal Part–Time | -1 | 0 | Todo el grupo no es full-time. | 0 |
| E8 | Dificultad del lenguaje de programación | -1 | 3 | Se usará PHP5. | -3 |
| Total | | | | | 20,5 |

$$EF=1.4 - 0.03* \Sigma(\text{peso}*\text{valor asignado})$$

$$EF=1.4 - 0.03* \Sigma(\text{peso}*\text{valor asignado})$$

$$EF=1.4 - 0.03*20,5$$

$$EF=1.4 - 0.6$$

$$EF=0.785$$

$$UCP=UUCP*TCF*EF$$

$$UCP=89*0.82*0.8$$

$$UCP=58,384$$

5.2.6. Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

CF = 20 Horas-Hombre / Punto de Casos de uso.

E= UCP*CF

E=83,8537*20

E=1677,074 Horas-Hombre

5.2.7. Calcular esfuerzo de todo el proyecto.

5.2.8. Tabla 5.5 Esfuerzo del proyecto.

| Actividad | Porcentaje % | Horas-Hombres |
|--------------------------------|--------------|---------------|
| Análisis | 8 | 191,6656 |
| Diseño | 15 | 359,373 |
| Implementación | 70 | 1677,074 |
| Pruebas | 5 | 119,791 |
| Sobrecarga (otras actividades) | 2 | 47,9164 |
| Total | 100 | 2395,82 |

Si $E_T = 2395,82$ horas-hombre y se estima que cada mes tiene como promedio 192 horas laborables, eso daría un $E_T = 11,0917593$ mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en 11 meses

Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = 2 * Salario Promedio

CHM = 200.00 \$/mes

Costo = CHM * E_T / CH

Costo = 200.00 * 11,0917593 / 2

Costo = \$ 1109,17593

Tiempo = E_T / CH

Tiempo = 11,0917593 / 2

Tiempo ≈ 5 meses

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en 8 meses y su costo total se estima que sea \$1109,17593.

5.3. Beneficios tangibles e intangibles.

El Sistema para la automatización de la gestión de la información de los profesores de la facultad 2 no es un producto con fines comerciales, su principal objetivo es resolver los problemas que existen en la entidad en la gestión de la información.

El beneficio fundamental del sistema es contar con una aplicación Web flexible, dinámica y de interfaz agradable que le permita agregar, modificar y conocer de una forma más precisa y en el menor tiempo posible la información referente a los profesores de la facultad 2.

Por tanto, los beneficios inmediatos son generalmente intangibles:

- Disminución del tiempo y esfuerzo que se invierte en almacenar y consultar esta información.
- Fácil y rápido acceso a la información actualizada.

- Fácil procesamiento y mejor organización de la información.

5.4. Análisis de costos y beneficios.

Desarrollar un producto informático cuesta. Justificar entonces su desarrollo depende de los beneficios que reportarían su implantación y utilización. Los beneficios pueden ser económicos y de orden social, estos últimos son de tanta importancia como los primeros. El sistema que se propone está dirigido específicamente a los departamentos de las facultades de la Universidad de las Ciencias Informáticas.

Una vez implantado el sistema, éste contribuirá a aumentar la eficiencia en el trabajo de los jefes de departamentos y servirá de ayuda en la toma de. La tecnología utilizada para el desarrollo del sistema es en su mayoría libre y no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable, por tanto un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios; debido a la estructuración en capas de los procesos del negocio que se diseñaron. Analizando el costo del proyecto y los numerosos beneficios que con él se obtienen, se puede concluir que su implementación es realmente factible.

5.5. Conclusiones del Capítulo.

En este capítulo se detalló el estudio de factibilidad correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado. La herramienta propuesta reportará beneficios significativos e importantes para los departamentos de la facultad 2 de la Universidad de las Ciencias Informáticas, que contribuirá a mejorar la actividad de gestión de su información.

CONCLUSIONES GENERALES

Con la realización de este trabajo se pretende responder a la necesidad de contar con una alternativa en los departamentos de la Facultad 2 de la Universidad de las Ciencias Informáticas para la gestión de la información que se genera de su trabajo, beneficiando a los jefes de departamentos que llevan el control de esta información de forma manual en un proceso lento y complejo. Se expone además un estudio de los procesos del negocio de la entidad con el objetivo de conocer las necesidades de los clientes, definir los requerimientos y restricciones.

Con el objetivo de lograr una solución se analizaron las herramientas, técnicas, tecnologías, metodologías de desarrollo más viables para el desarrollo, llegándose a la conclusión de que el sistema sería implementado en el lenguaje de programación PHP, gestor de bd PostgreSQL, Servidor Web Apache, el diseño de interfaz en Dreamweaver y la metodología de desarrollo RUP. Se modeló el negocio propuesto, se definieron los actores y trabajadores, y se identificaron las actividades a automatizar. Se refinaron los requerimientos funcionales y no funcionales. Usando la metodología RUP, se transitó por los flujos de trabajo y sus cuatros fases, apoyándonos en el lenguaje de modelado UML para llegar a obtener como resultado un sistema automatizado para la gestión de la información de los profesores de la facultad 2.

Finalmente se analizaron los costos y los beneficios que representaría la construcción del sistema y se concluyó que éste era factible, por tanto el sistema propuesto representa una mejora sin duda para facilitar el trabajo en los departamentos.

RECOMENDACIONES

Se recomienda seguir trabajando en el perfeccionamiento del sistema, adicionándoles nuevas funcionalidades, de acuerdo a las necesidades de la facultad. Dar mantenimiento y actualizaciones periódicas, para que el sistema mantenga la fiabilidad y su funcionamiento sea óptimo y la información que se gestiona a través de él. Mejorar el diseño de las páginas de presentación al usuario con persona especializadas en el tema. Integrar este módulo en un sistema que contenga y gestione todos los procesos de la facultad

REFERENCIAS BIBLIOGRÁFICAS

[1] http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1012-15872003000100006&lng=es&nrm=iso

[2] Dorado Risco, Arismayda. Sistema automatizado para la Gestión Académica. Subsistema “Generador de Reportes”. Universidad de La Habana, Facultad de Matemática Computación. Ciudad de La Habana, 2004.

[3] <http://es.wikipedia.org/wiki/Dreamweaver>

[4] http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

[5] <http://es.wikipedia.org/wiki/PostgreSQL>

[6] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Editorial Félix Varela, La Habana, 2004, Prefacio – pag12

[7] <http://es.wikipedia.org/wiki/Php>

[8] <http://wiki.prod.uci.cu/index.php/JavaScript>

[9] <http://wiki.prod.uci.cu/index.php/Html>

[10] Ingeniería de Software 2, Conferencia 6. Curso 2004-2005.

[11] <http://wiki.prod.uci.cu/index.php/RUP>

[12] <http://wiki.prod.uci.cu/index.php/UML>

[13] http://www.ifsworld.com/es/ifs_applications/enterprise_solutions/business_modeler/default.asp

- [14] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Editorial Félix Varela, La Habana, 2004. Capítulo 6.
- [15] Ingeniería de Software 2, Conferencia 3. Curso 2004-2005.
- [16] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Editorial Félix Varela, La Habana, 2004. Capítulo 8.
- [17] Ingeniería de Software 2, Conferencia Modelo del diseño. Curso 2004-2005.
- [18] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Editorial Félix Varela, La Habana, 2004. Capítulo 9.
- [19] Ingeniería de Software 2, Conferencia diseño de base de datos. Curso 2004-2005.
- [20] <http://teleformacion.uci.cu/course/view.php?id=43> Conferencia 4 FT implementación.

BIBLIOGRAFÍA

Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Editorial Félix Varela, La Habana, 2004.

Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Editorial Félix Varela. La Habana, 2004.

Pressman, Robert. 1998. Ingeniería de Software.

Rational Rose Enterprise Edition. Rational Software Corporation. 2003.

ANEXOS

Anexo 1.

Tabla 3. Descripción textual del caso de uso: Gestionar Actividad.

| | |
|-------------------------------------|--|
| Caso de uso del negocio: | Gestionar Actividad. |
| Actores del negocio: | Directivo |
| Propósito: | Asignar las tareas a los jefes de departamentos y verificar su cumplimiento |
| Resumen: | El caso de uso comienza cuando el directivo le signa una tarea al jefe del departamento, en caso de ser un tarea que ya existe y va ha ser modificada, ya puede ser su fecha de realización o la tarea en si, se actualiza y se le informa a los profesores de los cambios, ocurre lo mismo si es eliminada. Si la tarea es nueva se registra el nombre de la tarea, fecha de creada, fecha de cumplimiento, profesores responsables. El caso de uso termina cuando se verifica el cumplimiento de la tarea. |
| Casos de uso asociados: | |
| Curso normal de los eventos: | |
| Acción del actor | Respuesta del negocio |
| 1. El directivo informa de la tarea | 1.1. El jefe de departamento recibe la tarea. 1.2. El jefe de departamento verifica si ya existe la tarea. 1.3. Si existe la tarea procede a modificarla con los nuevos datos. |

| | |
|------------------------------|---|
| <p>2. Recibe el informe.</p> | <p>1.4. El jefe de departamento informa al profesor de los cambios.</p> <p>1.5. El profesor recibe las orientaciones</p> <p>1.6. El profesor realiza la tarea.</p> <p>1.7. El jefe de departamento verifica la realización de la actividad.</p> <p>1.8. Si es correcta se genera un informe del estado de la actividad.</p> <p>1.9. El jefe de departamento envía el informe al directivo.</p> |
| <p>Flujo alternativo</p> | |
| <p>Acción del actor</p> | <p>Respuesta del negocio</p> |
| <p>2. Recibe el informe.</p> | <p>1.3.1. Si la tarea no existe.</p> <p>1.3.2. El jefe de departamento consulta el registro de profesores.</p> <p>1.3.3. El JD decide que profesores son los encargados de realizarlas.</p> <p>1.3.4. El JD informa a los profesores.</p> <p>1.3.5. El profesor recibe las orientaciones de la nueva tarea.</p> <p>1.3.6. El profesor realiza la actividad.</p> <p>1.3.7. El jefe de departamento verifica la realización de la actividad.</p> <p>1.3.8. Si es correcta se genera un informe del estado de la actividad.</p> <p>1.3.9. El jefe de departamento envía el informe al directivo.</p> |

| | |
|-----------------------------------|--|
| Flujo alterno | |
| Acción del Actor | Respuesta del negocio |
| 2 El directivo recibe el informe. | <p>1.8.1 Si no es correcta el JD informa al profesor donde están los problemas.</p> <p>1.8.2. El profesor Corrige los problemas.</p> <p>1.8.3. El JD envía el informe al el directivo.</p> |

Tabla 4. Descripción textual del caso de uso: Gestionar persona.

| | |
|------------------------------|---|
| Caso de uso del negocio: | Gestionar persona. |
| Actores del negocio: | Directivo |
| Propósito: | Agregar, modificar o eliminar un profesor |
| Resumen: | El caso de uso comienza cuando el directivo le informa al jefe de departamento que el claustro de profesor va a ser cambiado. Si se va ha agregar un nuevo profesor, el jefe de departamento recoge los datos del profesor y los guarda en su archivo, si es un profesor que será dado de baja, el jefe de departamento busca el profesor en su archivo y eliminas sus datos, ocurre lo mismo si es modificado los datos del profesor, el JD buscar el su archivo al profesor y modifica los datos. |
| Casos de uso asociados: | |
| Curso normal de los eventos: | |
| Acción del actor | Respuesta del negocio |

| | |
|--|--|
| 1. El directivo informa de que el claustro de profesores ha sido cambiado. | <p>1.1. El jefe de departamento verifica si es será modificado o será un nuevo profesor.</p> <p>1.2. Si es va a ser modificado el profesor el jefe de departamento busca los datos en su archivo.</p> <p>1.3. Si es una baja el JD procede a eliminar el profesor del archivo.</p> |
| Flujo alternos | |
| Acción del actor | Respuesta del negocio |
| | <p>1.3.1. Si es un nuevo profesor el JD pide los datos al profesor.</p> <p>1.2.2. El profesor envía los datos el JD.</p> <p>1.2.3. El JD registra al nuevo profesor.</p> <p>1.2.4. El JD informa al nuevo profesor de sus responsabilidades</p> <p>1.3.5. El profesor esta de acuerdo.</p> |
| Flujo alterno | |
| Acción del Actor | Respuesta del negocio |
| | <p>1.3.1 Si se modificaran los datos del profesor, JD pide al profesor los datos.</p> <p>1.3.2. El profesor envía los datos.</p> <p>1.8.3. El JD procede a actualizar los datos del profesor.</p> |

Tabla 5. Descripción textual del caso de uso: Solicitar Información.

| | |
|--------------------------|--|
| Caso de uso del negocio: | Solicitar información |
| Actores del negocio: | Directivo |
| Propósito: | Conocer del estado de las tareas y sus |

| | |
|--|---|
| | profesores en cada departamento. |
| Resumen: | El caso de uso comienza cuando el directivo necesita conocer alguna información del departamento, el JD busca esa información y se la envía al directivo. |
| Casos de uso asociados: | |
| Curso normal de los eventos: | |
| Acción del actor | Respuesta del negocio |
| 1. El directivo solicita cierta información al JD. 2. El Directivo recibe el reporte. | 1.1. El JD busca esa información. 1.2. Crea un reporte. 1.3. EL JD envía el reporte al Directivo. |

Anexo 2.

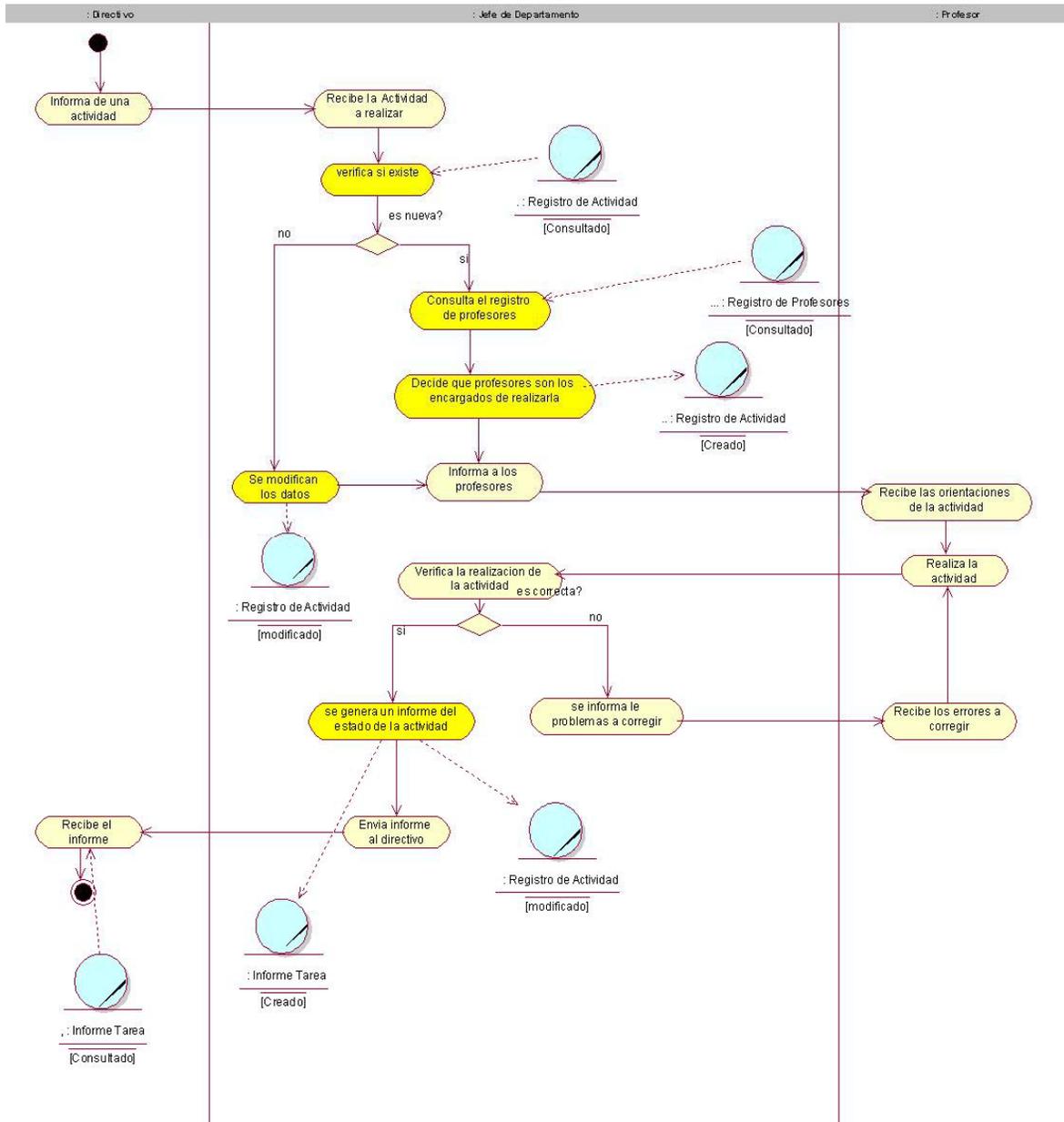


Diagrama 2. Diagrama de actividades para el Caso de uso Gestionar Actividad.

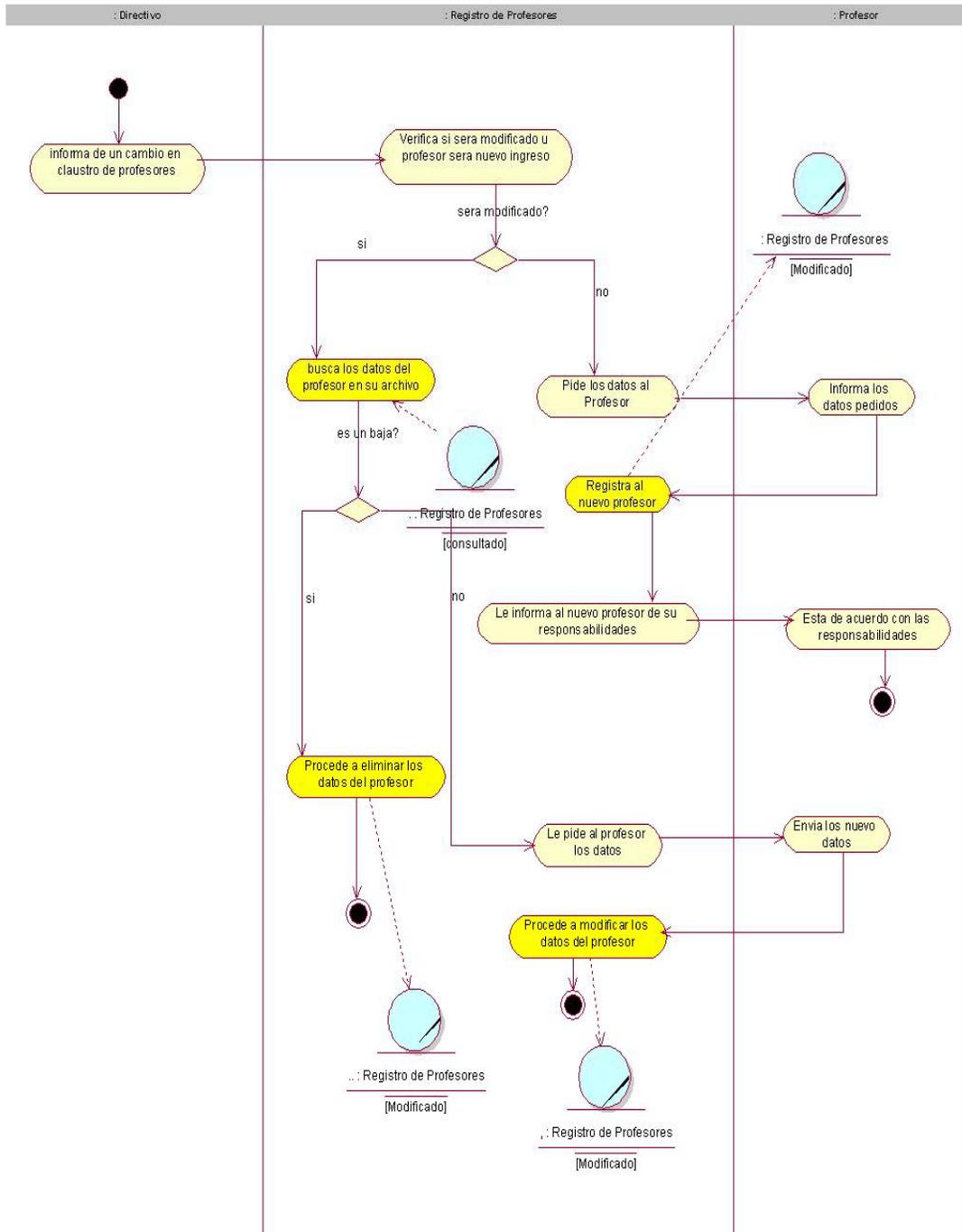


Diagrama 3. Diagrama de Actividades para el caso de uso Gestionar Profesor.

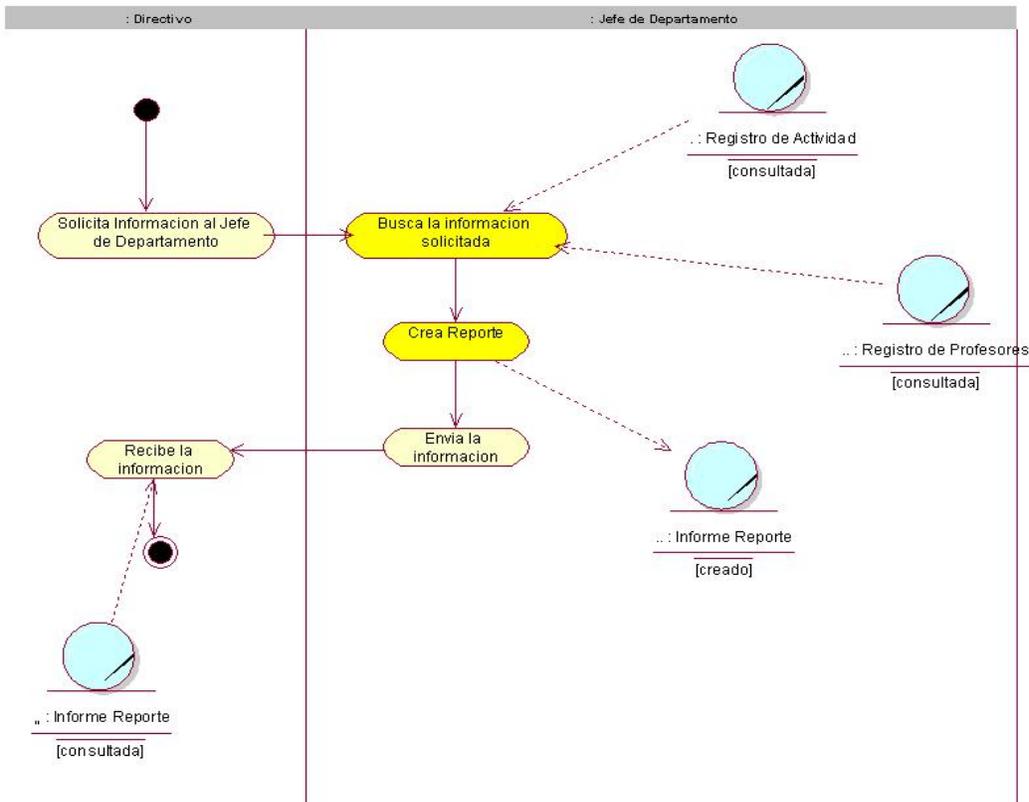


Diagrama 4. Diagrama de Actividades para el caso de uso Solicitar Información.

Anexo 3.

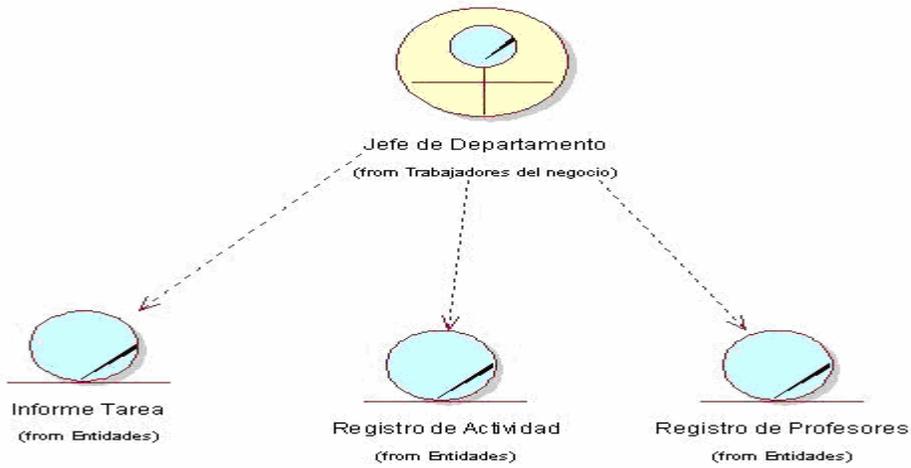


Diagrama de clases del modelo de objetos. CU Gestionar Actividad.

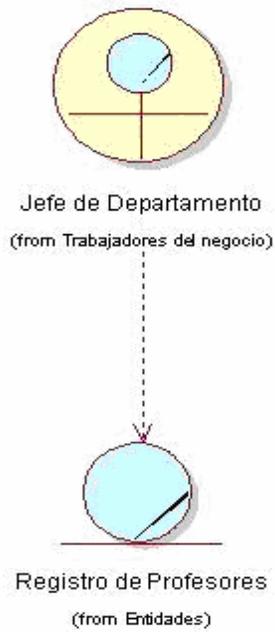


Diagrama de clases del modelo de objetos. CU Gestionar Persona.

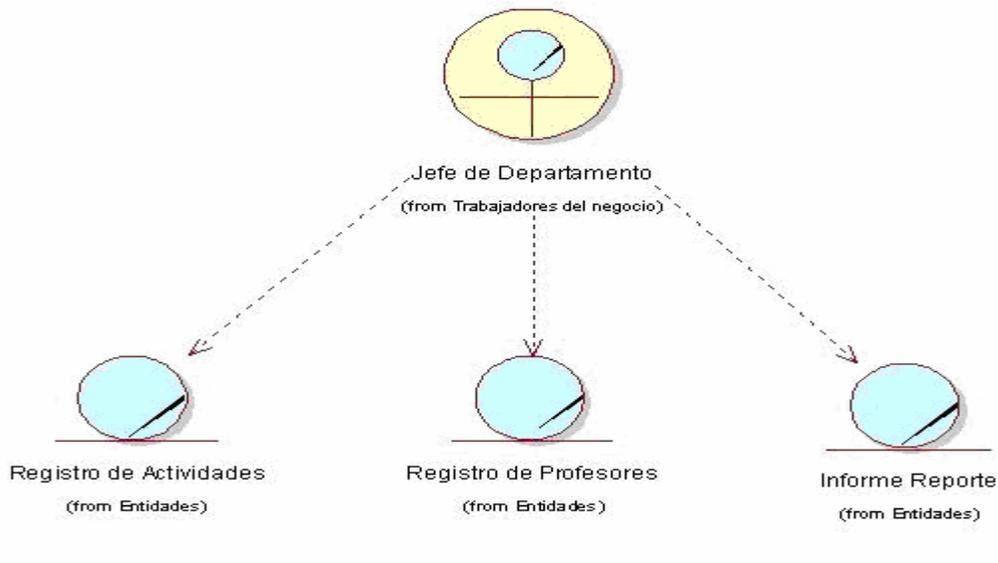


Diagrama de clases del modelo de objetos. CU Solicitar Información.

Anexo 4.

Tabla 7. Caso de uso Autenticar.

| Caso de uso | |
|--|---|
| CU-1 | Autenticar |
| Propósito | Permitir el acceso a usuarios que han sido creados previamente. |
| Actor : Usuario | |
| Resumen: Este caso de uso se inicia cuando un usuario desea acceder al sistema, el sistema verifica los datos y le da acceso según sus privilegios. El caso de uso finaliza cuando se realiza la operación con éxito. | |
| Referencias | RF1 |
| Precondiciones | |
| Poscondiciones | 1. Permitir el acceso |
| Acción del actor | Respuesta del sistema |
| 1- El Usuario introduce nombre de usuario y contraseña. | 1.1. El sistema se conecta a la base de datos para verificar que los datos estén correctos. 1.2. Si los datos son correctos, el sistema le muestra un mensaje de bienvenida y finaliza el caso de uso. |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | 1.2.1. Si los datos no son correctos, el sistema le muestra un mensaje de error y vuelve al paso 1. |

Tabla 8. Descripción del caso de uso Gestionar Tarea.

| Caso de uso | |
|---|---|
| CU-2 | Gestionar Tarea |
| Propósito | Crear, modificar, eliminar tarea |
| Actor: Jefe de Departamento | |
| Resumen: Este caso de uso se inicia cuando el jefe de departamento necesita crear una tarea o modificar o eliminar una ya existente. Si desea crear, registra los datos como son el nombre, breve descripción, fecha de creada, fecha de cumplimiento, responsable(s) y participantes y luego si fue cumplida o no. Para modificar o eliminar una tarea se accede a la lista de tareas y se escoge la opción de modificar o eliminar, en el primer caso se le informa al(los) profesor(es) la tarea modificada. El CU termina cuando el jefe de departamento chequea su cumplimiento; en el caso de que sea eliminada o modificada la tarea, el CU termina cuando se le informa al profesor. | |
| Referencias | RF3, RF11 |
| Precondiciones | <ol style="list-style-type: none"> 1. El Jefe de Departamento debe haberse autenticado. 2. EL Jefe de Departamento debe tener acceso a esta parte del sistema. |
| Poscondiciones | <ol style="list-style-type: none"> 1. Datos de la tarea agregados a la base de datos. 2. Datos de la tarea modificados y actualizados en la base de datos. 3. Datos de la tarea eliminados de la base de datos. |
| Acción del actor | Respuesta del sistema |
| 1. Selecciona la opción gestionar tarea. | 1.1 Muestran dos opciones Agregar y Listado de Tarea. |
| Esenario1: Agregar tarea. | |
| Acción del actor | Respuesta del sistema |
| <ol style="list-style-type: none"> 1. Escoge la opción Agregar. 2 Opera según los datos 3. Selecciona la opción guardar. | <ol style="list-style-type: none"> 1.1 Se muestra los campos donde se introducen los datos y el botón para guardar los datos. 3.1. Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Agregar. |

| Curso alternativo de los eventos: | |
|--|---|
| Acción del actor | Respuesta del sistema |
| | 3.1 Si lo datos son introducidos incorrectamente, el sistema muestra un mensaje de error , muestra el error y retorna al paso 1.1 |
| Escenario 2: Listado de Tareas. | |
| Acción del actor | Respuesta del sistema |
| 1. El usuario escoge la opción listado de tareas. | 1.1 Muestra las tareas y las opciones modificar y eliminar tarea |
| Escenario 3: Modificar Tarea. | |
| 1. Selecciona la tarea. 2. Selecciona la opción modificar tarea 3. Opera según los datos que desee modificar. 4. Elige la opción guardar los cambios. | 2.1. Muestra los campos que pueden ser cambiados y la opción guardar cambios. 4.1. Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Listado de Tarea. |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | 4.1. Si lo datos son introducidos incorrectamente, el sistema muestra un mensaje de error, muestra el error y retorna al paso 2.1 |
| Escenario 4: Eliminar Tarea. | |
| Acción del actor | Respuesta del sistema |
| 1. El usuario selecciona la tarea. 2. El usuario escoge la opción eliminar tarea | 2.1 El sistema elimina la tarea y retorna a la pagina Listado de Tareas |

Tabla 9. Descripción del caso de uso Gestionar Profesor.

| Caso de uso | |
|--|--|
| CU-3 | Gestionar Profesor |
| Propósito | Crear, modificar, eliminar un profesor |
| Actor: Jefe de Departamento | |
| Resumen: Este caso de uso se inicia cuando el jefe de departamento necesita Agregar un profesor o Modificar o Eliminar uno ya existente. Si desea Agregar, registra los datos. Para modificar o eliminar un profesor se accede a la lista de profesores y se escoge la opción de modificar o eliminar. CU termina cuando las operaciones de Agregar, Modificar y Eliminar terminan con éxito. | |
| Referencias | RF2 |
| Precondiciones | <ol style="list-style-type: none"> 1. El Jefe de Departamento debe haberse autenticado. 2. EL Jefe de Departamento debe tener acceso a esta parte del sistema. |
| Poscondiciones | <ol style="list-style-type: none"> 1. Datos del profesor agregados a la base de datos. 2. Datos del profesor modificados y actualizados en la base de datos. 3. Datos del profesor eliminados de la base de datos. |
| Acción del actor | Respuesta del sistema |
| 1. Selecciona la opción gestionar Profesor. | 1.1. Muestran dos opciones Agregar y Listado de Profesores. |
| Escenario 1: Agregar Profesor. | |
| Acción del actor | Respuesta del sistema |
| <ol style="list-style-type: none"> 1. El usuario escoge la opción Agregar. 2. Opera según los datos 3. Selecciona la opción guardar. | <ol style="list-style-type: none"> 1.1. El sistema muestra los campos donde se introducen los datos y el botón para guardar los datos. 3.1. Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Agregar. |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |

| | |
|---|--|
| | 3.1 Si los datos son introducidos incorrectamente, el sistema muestra un mensaje de error, muestra el error y retorna al paso 1.1. |
| Escenario 2: Listado de Profesores. | |
| Acción del actor | Respuesta del sistema |
| 1. El usuario escoge la opción listado de Profesores existentes. | 1.1. Muestra un listado de Profesores y las opciones modificar y eliminar. |
| Escenario 3: Modificar Profesor. | |
| Acción del actor | Respuesta del sistema |
| 1. El usuario selecciona el profesor. 2. Selecciona la opción modificar. 3. Opera según los datos que desee modificar. 4. Elige la opción guardar los cambios. | 2.1. Muestra los campos que pueden ser cambiados y la opción guardar cambios. 4.1. Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Listado de Profesores. |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| . | 4.1. Si los datos son introducidos incorrectamente, el sistema muestra un mensaje de error, muestra el error y retorna al paso 2.1. |
| Escenario 4: Eliminar Profesor. | |
| Acción del actor | Respuesta del sistema |
| 1. El usuario selecciona el profesor. 2. El usuario escoge la opción eliminar profesor. | 2.1. El sistema elimina la tarea y retorna a la página Listado de Profesores. |

Tabla 10. Descripción del caso de uso Solicitar Reporte.

| Caso de uso | |
|--|---|
| CU-4 | Solicitar Reporte |
| Propósito | Conocer a través de los reporte el estado de las actividades y los profesores |
| Actor : Jefe de Departamento | |
| Precondiciones | <ol style="list-style-type: none"> 1. El Jefe de Departamento debe haberse autenticado. 2. EL Jefe de Departamento debe tener acceso a esta parte del sistema. |
| Poscondiciones | 1. El Jefe de Departamento obtiene los reporte según su criterio de búsqueda |
| <p>Resumen: Este caso de uso se inicia cuando el jefe de departamento necesita crear una tarea o modificar o eliminar una ya existente. Si desea crear, registra los datos como son el nombre, breve descripción, fecha de creada, fecha de cumplimiento, responsable(s) y participantes y luego si fue cumplida o no. Para modificar o eliminar una tarea se accede a la lista de tareas y se escoge la opción de modificar o eliminar, en el primer caso se le informa al(los) profesor(es) la tarea modificada. El CU termina cuando el jefe de departamento chequea su cumplimiento; en el caso de que sea eliminada o modificada la tarea, el CU termina cuando se le informa al profesor.</p> | |
| Referencias | RF4 |
| Acción del actor | Respuesta del sistema |
| 1. Selecciona la opción gestionar reporte. | 1.1. Muestran dos opciones Profesores y Tareas. |
| Escenario 1: Reporte Profesores | |
| Acción del actor | Respuesta del sistema |
| <ol style="list-style-type: none"> 1. Escoge la opción Profesores. 2. Opera según los datos 3. Selecciona la opción buscar. | <ol style="list-style-type: none"> 1.1. Se muestra los campos donde se introducen los criterios de búsquedas y el botón buscar. 3.1. El sistema muestra los resultados. |
| Escenario 2: Reporte Tarea. | |
| Acción del actor | Respuesta del sistema |
| <ol style="list-style-type: none"> 1. Escoge la opción Tareas. 2. Opera según los datos. 3 Selecciona la opción buscar | <ol style="list-style-type: none"> 1.1. Se muestra los campos donde se introducen los criterios de búsquedas y el botón buscar. 3.1. El sistema muestra los resultados. |

Tabla 11. Descripción del caso de uso Cambiar Contraseña.

| Caso de uso | |
|--|--|
| CU-5 | Cambiar Contraseña |
| Propósito | Permitir el cambio de contraseña de los Usuarios. |
| Actor: Usuario | |
| Resumen: Este caso de uso se inicia cuando el usuario desea cambiar su contraseña, para ello debe poseer la contraseña anterior, el sistema verifica esa contraseña y le permite cambiarla. El caso de uso finaliza cuando se realiza la operación con éxito. | |
| Referencias | RF9 |
| Precondiciones | 1. Usuario debe haberse autenticado. 2. Poseer la contraseña anterior. |
| Poscondiciones | 1 El Usuario posee nueva contraseña. |
| Acción del actor | Respuesta del sistema |
| 1.- El Usuario elige la opción cambiar contraseña. 2.- El Usuario opera según los datos 3. El usuario ejecuta el botón cambiar. | 1.1. El sistema le muestra una página para que introduzca la contraseña vieja y la nueva. 3.1.El sistema comprueba la contraseña vieja en la base de datos 3.2. Si es correcta contraseña, la cambia por la nueva, le envía un mensaje que la operación fue terminada con éxito. 3.3. Retorna a la página cambiar contraseña. |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | 3.2. Si la contraseña no es correcta, el sistema le muestra un mensaje de error y vuelve al paso 1.1. |

Tabla 12. Descripción del caso de uso Gestionar Grupo.

| Caso de uso | |
|--|---|
| CU-6 | Gestionar Grupo |
| Propósito | Adicionar y eliminar un grupo |
| Actor : Administrador | |
| Resumen: Este caso de uso se inicia cuando el Administrador necesita agregar o eliminar un grupo. Si desea agregar, registra el número del nuevo grupo. Para eliminar se accede a la lista de grupos y se elimina. El CU termina cuando las operaciones de agregar y eliminar terminan con éxito. | |
| Referencias | RF6 |
| Precondiciones | <ol style="list-style-type: none"> 1. El Administrador debe haberse autenticado. 2. El Administrador debe tener acceso a esta parte del sistema. |
| Poscondiciones | <ol style="list-style-type: none"> 1. Nuevo grupo agregado a la base de datos. 2. Eliminado grupo de la base de datos |
| Acción del actor | Respuesta del sistema |
| 1. Selecciona la opción gestionar Grupo. | 1.1. Muestran dos opciones Agregar y Listado de Grupos. |
| Escenario 1: Agregar Grupo | |
| Acción del actor | Respuesta del sistema |
| <ol style="list-style-type: none"> 1. El Administrador escoge la opción Agregar. 2. Opera según los datos 3. Oprime el botón guardar. | <ol style="list-style-type: none"> 1.1. Se muestra el campo donde se introduce el número del grupo y el botón para guardar los datos. 3.1. El sistema guarda los datos y retorna a la página Agregar. |
| Escenario 2: Listado de Grupos. | |
| Acción del actor | Respuesta del sistema |

| | |
|--|--|
| 1. El Administrador escoge la opción listado de grupos. 2. Selecciona un grupo. 3. Oprime el botón eliminar. | 1.1. Muestra un listado de los grupos y la opción eliminar. 3.1. El sistema elimina el grupo y retorna a la página Listado de Grupos. |
|--|--|

Tabla 13. Descripción del caso de uso Gestionar Asignatura.

| Caso de uso | |
|--|--|
| CU-7 | Gestionar Asignatura |
| Propósito | Adicionar , modificar y eliminar una asignatura |
| Actor : Administrador | |
| Resumen: Este caso de uso se inicia cuando el Administrador necesita agregar, modificar o eliminar una asignatura. Si desea agregar, registra el nombre de la nueva asignatura. Para modificar o eliminar se accede a la lista de asignatura y se modifica o elimina. El CU termina cuando las operaciones con éxito. | |
| Referencias | RF7,RF8 |
| Precondiciones | 1. El Administrador debe haberse autenticado. 2. El Administrador debe tener acceso a esta parte del sistema. |
| Poscondiciones | 1. Datos de la asignatura agregados a la base de datos. 2. Datos de la asignatura modificados y actualizados en la base de datos 3. Datos de la asignatura eliminados de la base de datos. |
| Acción del actor | Respuesta del sistema |
| 1. Selecciona la opción gestionar asignatura. | 1.1. Muestran dos opciones Agregar y Listado de Asignaturas. |
| Escenario 1: Agregar Asignatura. | |
| Acción del actor | Respuesta del sistema |

| | |
|--|--|
| <p>1. El Administrador escoge la opción Agregar.</p> <p>2. Opera según los datos</p> <p>3. Oprime el botón guardar.</p> | <p>1.1. El sistema muestra el campo donde se introduce el nombre de la asignatura, el departamento a la que va a pertenecer y el botón para guardar los datos.</p> <p>3.1 Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Agregar.</p> |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | 3.1. Si lo datos son introducidos incorrectamente, el sistema muestra un mensaje de error, muestra el error y retorna al paso 1.1. |
| Escenario 2: Listado de Asignaturas. | |
| Acción del actor | Respuesta del sistema |
| 3.1. Escoge la opción listado de Asignaturas. | 3.2 Muestra un listado de las Asignaturas y la opción Modifica y Eliminar. |
| Escenario 3: Modificar Asignatura. | |
| Acción del actor | Respuesta del sistema |
| <p>1. El Administrador selecciona una Asignatura.</p> <p>2. Elige la opción modificar.</p> <p>3. Opera según los datos</p> <p>4. Elige la opción guardar</p> | <p>2.1. Muestra los campos que pueden ser cambiados y la opción guardar cambios</p> <p>4.1 Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Listado de Asignaturas.</p> |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | 4.1 Si lo datos son introducidos incorrectamente, el sistema muestra un mensaje de error, muestra el error y retorna al paso 2.1 |

| | |
|--|--|
| Escenario 4: Eliminar Asignatura. | |
| Acción del actor | Respuesta del sistema |
| 1. El Administrador elige la opción eliminar | 1.1. El sistema elimina la asignatura y retorna a la página Listado de Asignatura. |

Tabla 14. Descripción del caso de uso Gestionar Departamento.

| | |
|--|---|
| Caso de uso | |
| CU-8 | Gestionar Departamento |
| Propósito | Adicionar, eliminar o modificar un departamento, además de modificar los datos del usuario que son los jefes de departamentos. |
| Actor: Administrador | |
| Resumen: Este caso de uso se inicia cuando el Administrador necesita agregar, modificar o eliminar un departamento. El CU termina cuando las operaciones con éxito. | |
| Referencias | RF5,RF10 |
| Precondiciones | 1. El Administrador debe haberse autenticado. 2. El Administrador debe tener acceso a esta parte del sistema. |
| Poscondiciones | 1. Datos del departamento agregados a la base de datos. 2. Datos del departamento modificados y actualizados en la base de datos 3. Datos del departamento eliminados de la base de datos |
| Acción del actor | Respuesta del sistema |
| 1. Selecciona la opción gestionar departamento. | 1.1. Muestran dos opciones Agregar y Listado de Departamentos. |
| Escenario 1: Agregar Departamento. | |
| Acción del actor | Respuesta del sistema |

| | |
|--|--|
| <ol style="list-style-type: none"> 1. El Administrador escoge la opción Agregar. 2. Opera según los datos. 3. Elige la opción guardar. | <ol style="list-style-type: none"> 1.1. El sistema muestra los campos donde se introducen los datos del nuevo departamento y el botón guardar. 3.1. Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Agregar. |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | 3.1. Si los datos son introducidos incorrectamente, el sistema muestra un mensaje de error, muestra el error y retorna al paso 1.1. |
| Escenario 2: Listado de Departamentos. | |
| Acción del actor | Respuesta del sistema |
| 1. El Administrador escoge la opción listado de Departamentos. | 1.1. El sistema muestra un listado de los departamentos y la opción Modifica y Eliminar. |
| Escenario 2: Modificar Departamento. | |
| Acción del actor | Respuesta del sistema |
| <ol style="list-style-type: none"> 1. El Administrador selecciona un departamento. 2. Elige la opción modificar. 3. Opera según los datos 4. Elige la opción guardar | <ol style="list-style-type: none"> 2.1. Muestra los campos que pueden ser cambiados y la opción guardar cambios. 4.1 Si los datos son introducidos correctamente el sistema guarda los datos y retorna a la página Listado de Departamentos. |
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| | 4.1 Si los datos son introducidos incorrectamente, el sistema muestra un mensaje de error, muestra el error y retorna al paso 2.1. |
| Escenario 4: Eliminar Departamento. | |

| Acción del actor | Respuesta del sistema |
|-----------------------------|--|
| 1. Elige la opción eliminar | 1.1 El sistema elimina la asignatura y retorna a la página Listado de Departamentos. |

Anexo 5.

Diagramas de secuencia. CU Autenticar.

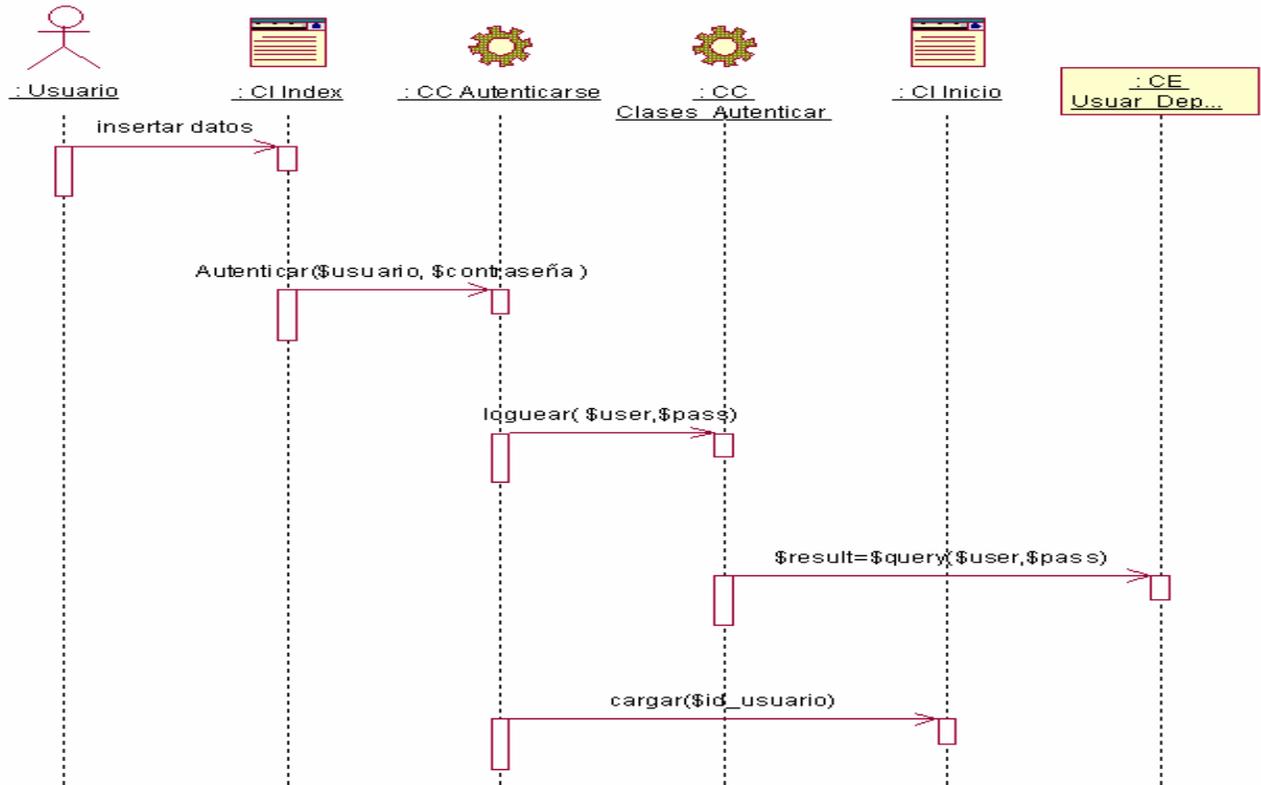


Diagrama de secuencia. CU Gestionar Profesor.

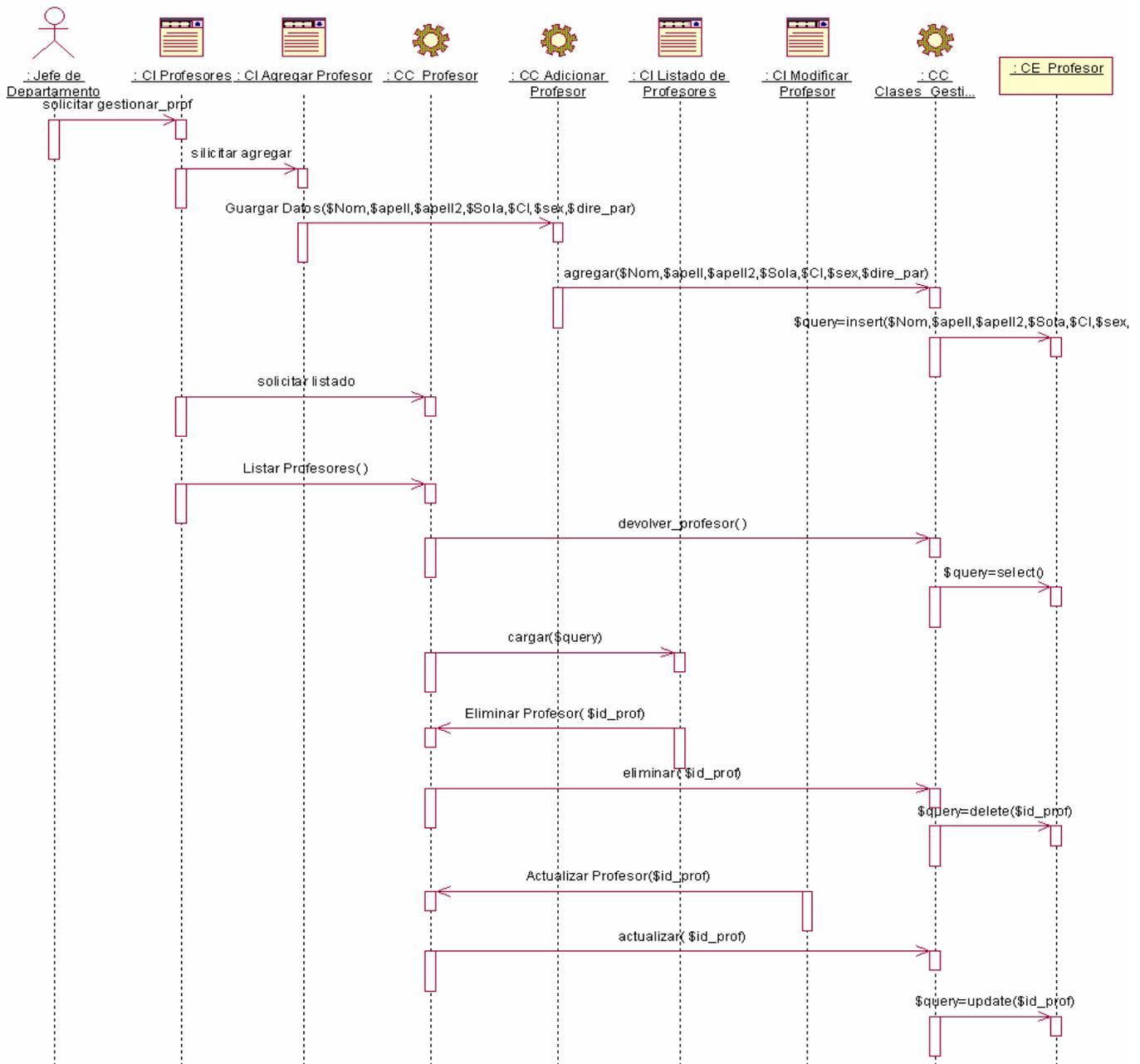


Diagrama de colaboración. CU Gestionar Tarea.

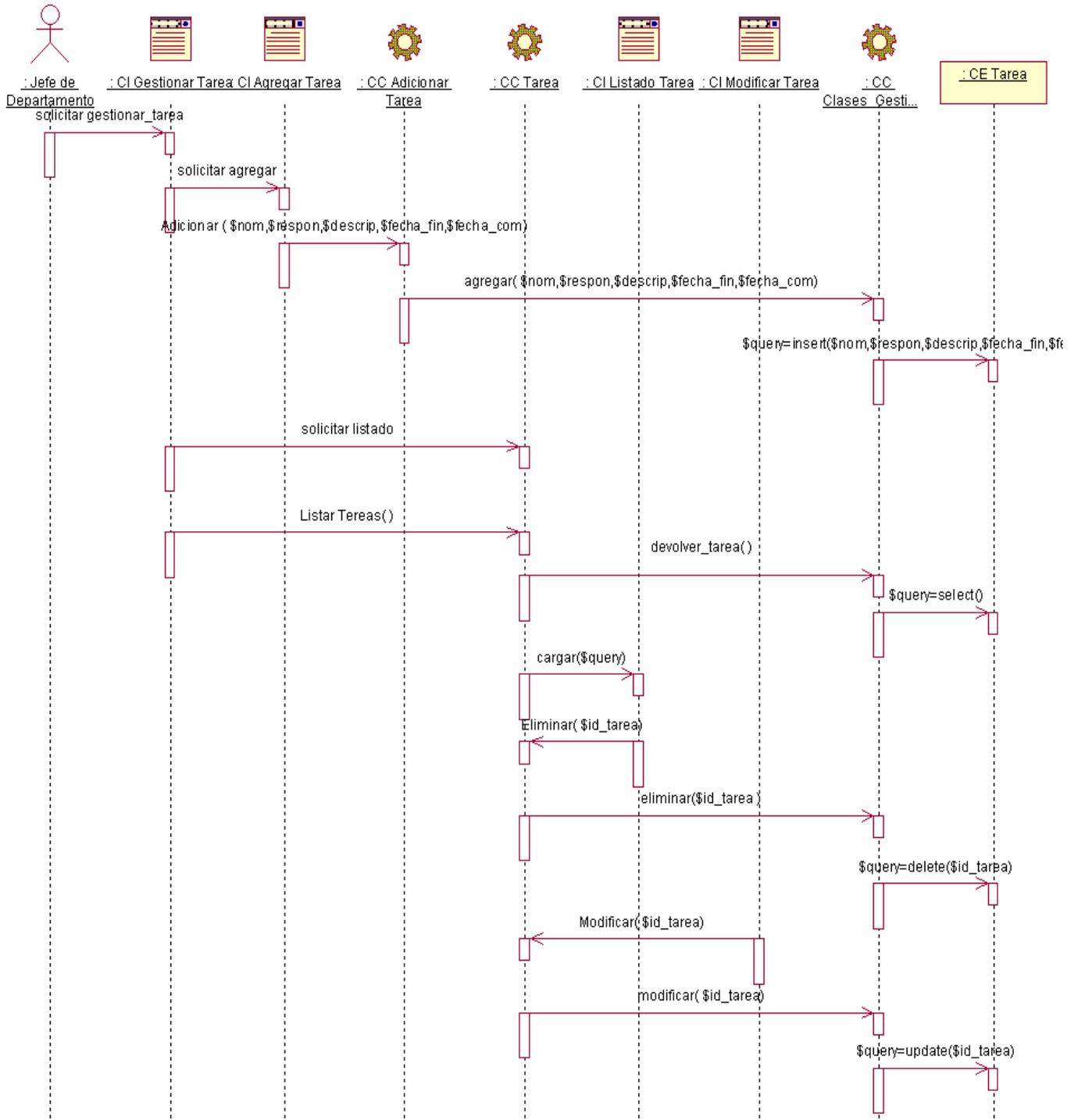


Diagrama de colaboración. CU Solicitar Reporte.

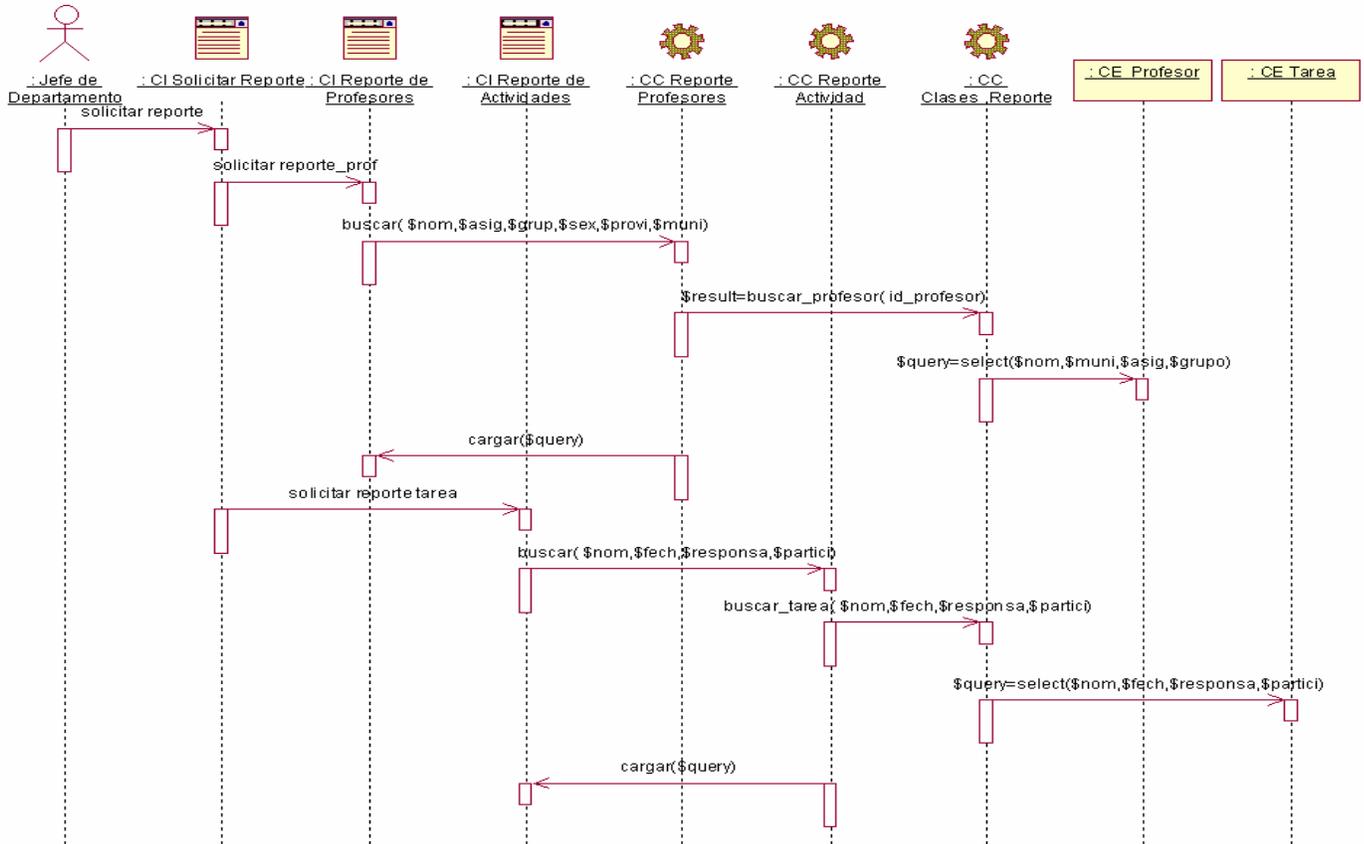


Diagrama de colaboración. CU Cambiar Contraseña.

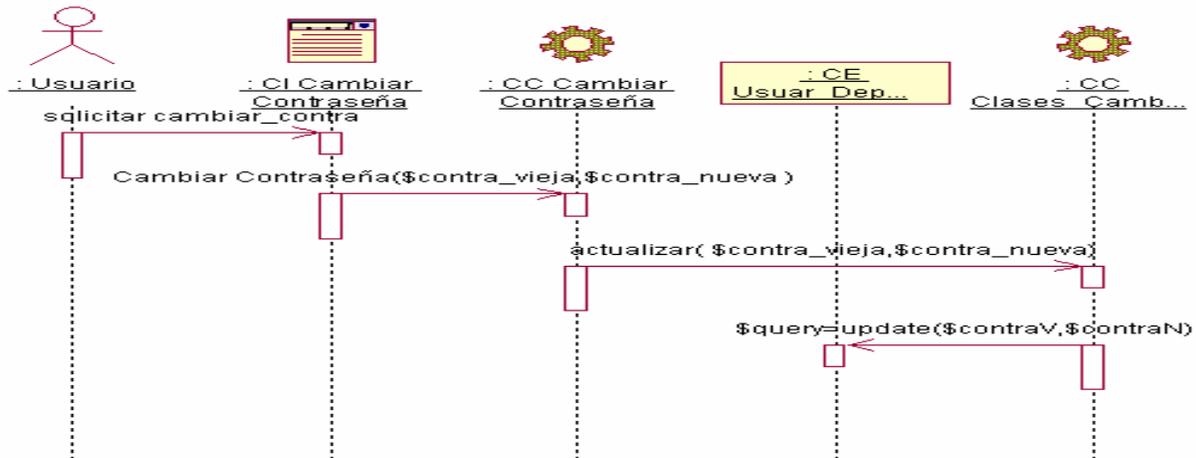


Diagrama de secuencia. CU gestionar Asignatura

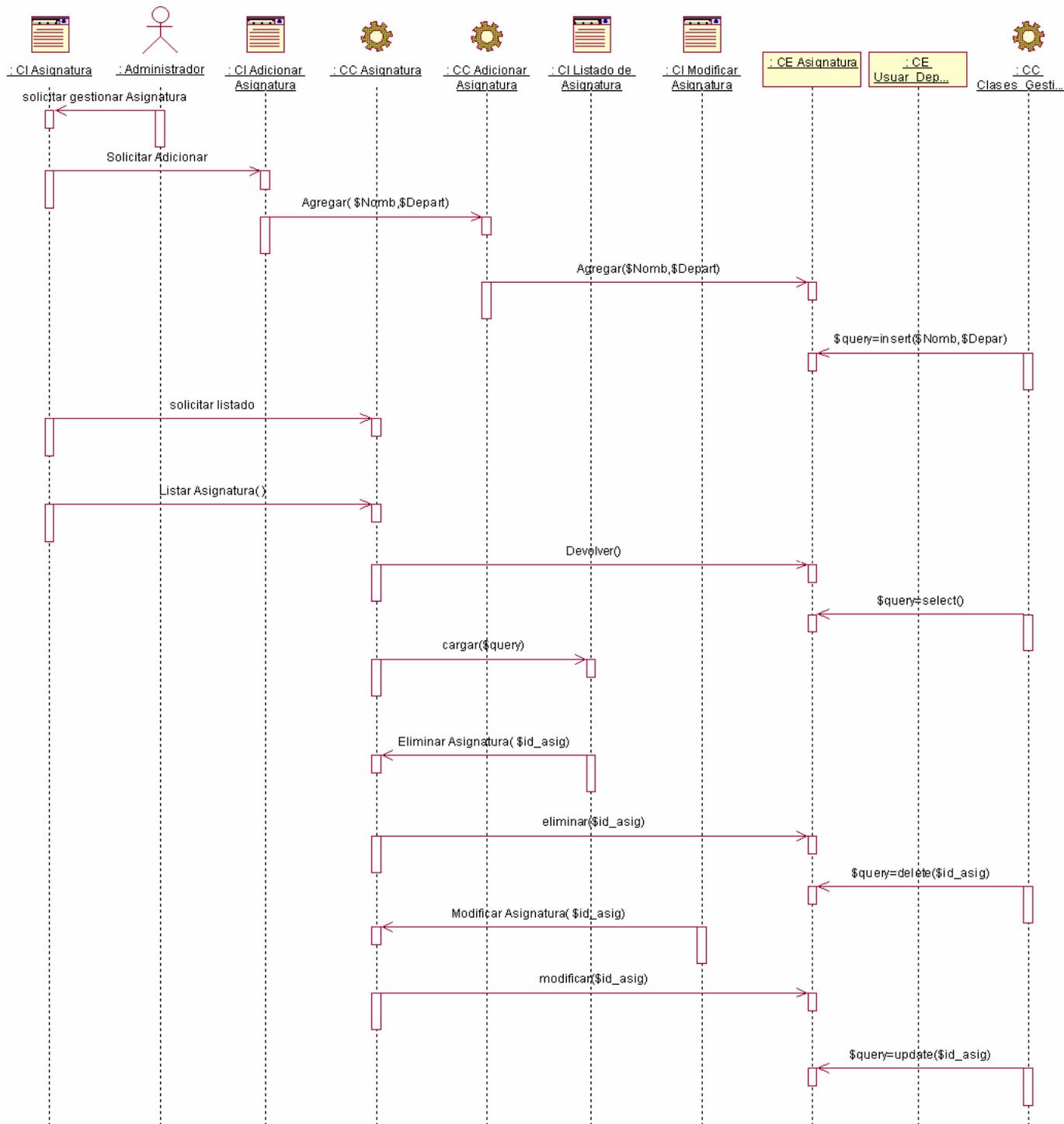


Diagrama de secuencia. CU Gestionar Departamento

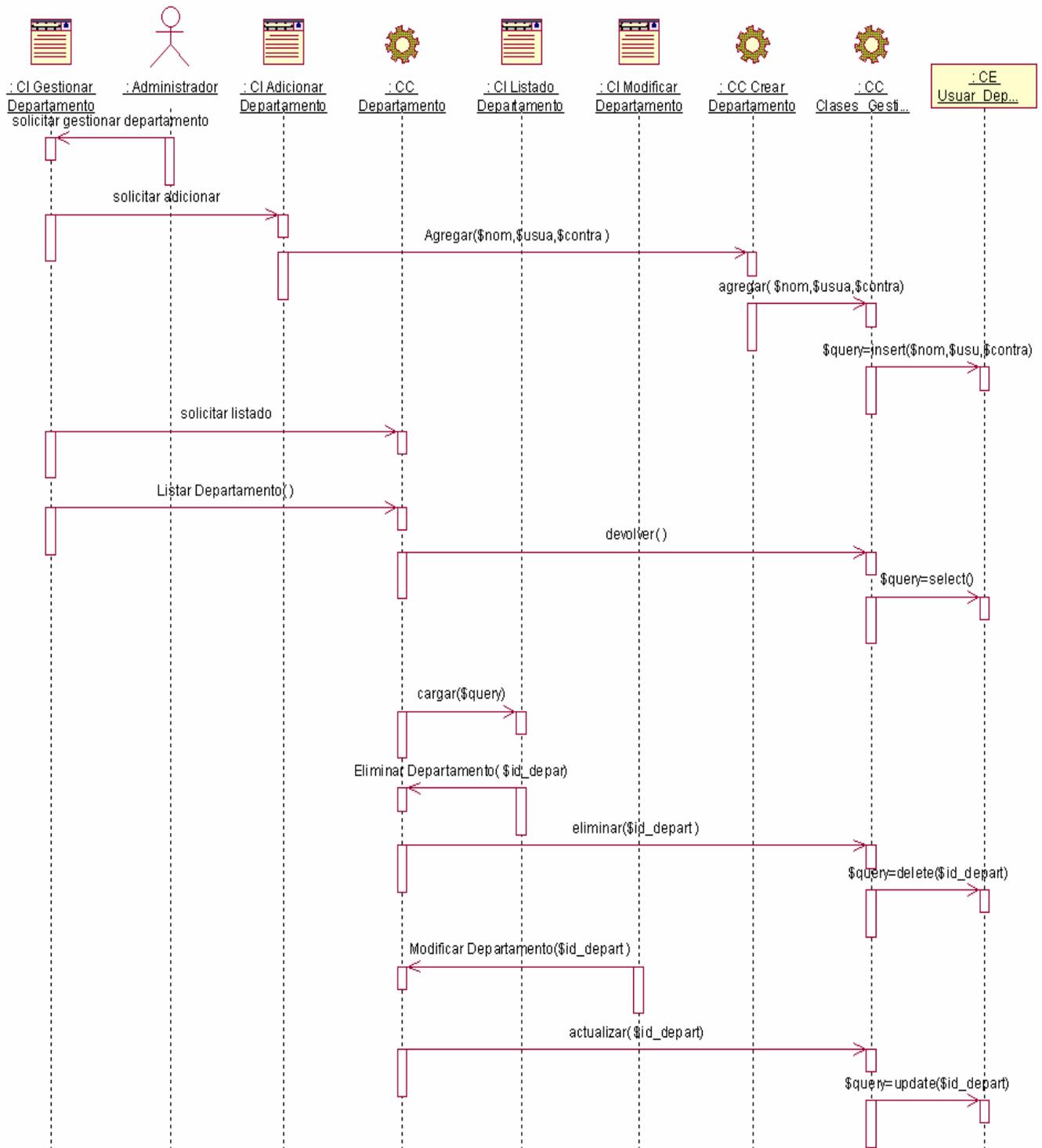
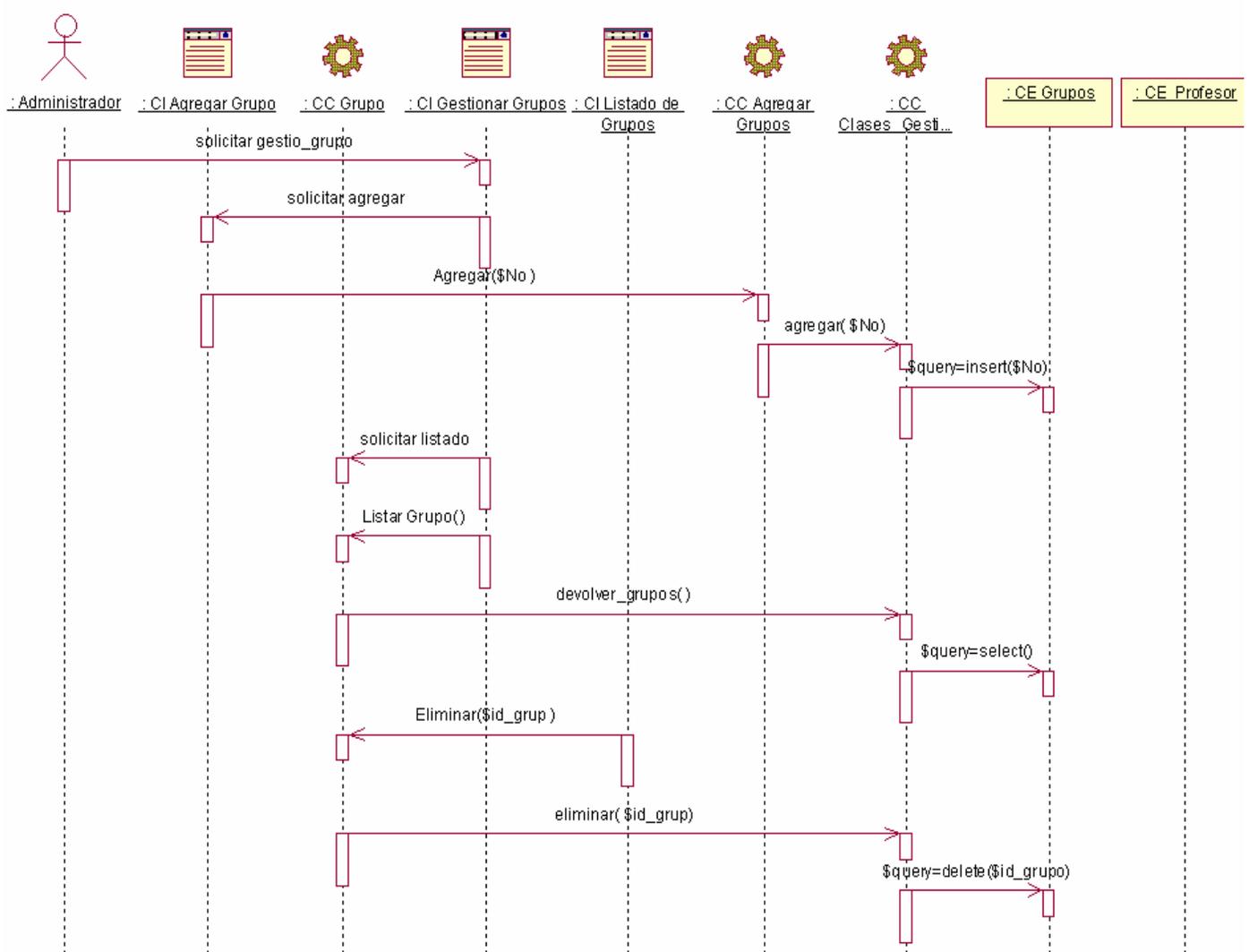


Diagrama de colaboración. CU Gestionar Grupo.



GLOSARIO DE TÉRMINOS

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

UML: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

Lenguajes de Programación: Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.

Apache: es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.

Html: *HyperText Markup Language*. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

PHP: *Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

XML: *Extensible Markup Language*. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.

Navegador Web, ojeador o browser es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Internet. Esta red de documentos es denominada World Wide Web (WWW) o Telaraña Mundial. Los navegadores actuales permiten mostrar o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

Servidor Web: Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear www.wikipedia.org en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Web Service: Es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

MVC: El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Maneja la visualización de la información.

Controlador: Controla el flujo entre la vista y el modelo (los datos). Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

Patrones: Son una disciplina de resolución de problemas reciente en la ingeniería del software que ha emergido en mayor medida de la comunidad de orientación a objetos, aunque pueden ser aplicados en cualquier ámbito de la informática y las ciencias en general.

Diagramas: Es la representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas.

Diagrama de clases: Conjunto de clases, interfaces y colaboraciones; así como sus colaboraciones.

Diagrama de objetos: Conjunto de objetos y sus relaciones.

Diagrama de casos de uso: Conjunto de casos de uso y actores y sus relaciones.

Diagramas de interacción (secuencia y colaboración): Objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.

Diagrama de actividad: Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Diagrama de componentes: Organización y las dependencias entre un conjunto de componentes.

Diagrama de despliegue: Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

Clase: Conjunto de objetos que comparten atributos, operaciones, relaciones y semántica.

Colaboración: Define la interacción entre los elementos que proporcionan un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos.

Caso de uso: Conjunto de secuencia de acciones que un sistema ejecuta y que produce un resultado observable para un actor.

Componente: Es una parte física y reemplazable de un sistema que conforman un conjunto de interfaces y proporciona la implementación de dicho conjunto.

Interacción: Conjunto de mensajes intercambiados entre un conjunto de objetos para alcanzar un propósito específico.

Dependencia: relación semántica que indica que un cambio en un elemento afecta a la semántica de otro elemento.