

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7



Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Desarrollo del Módulo de Gestión de Tesis del Sistema Integral

de Gestión Administrativa de la Facultad 7

Autor: Alexander Guedes Hernández

Tutores: Ing. Maidelys Pulido Morera

Ing. Daniel Miranda Calzadilla

Ciudad de La Habana, Julio de 2010

“Año 52 de la Revolución”

Datos de Contacto

Maidelys Pulido Morera: Ingeniera en Ciencias Informáticas, graduada en la Universidad de las Ciencias Informáticas (UCI) en el curso 2006-2007. Profesora Instructora de la Facultad 7. Ha impartido las asignaturas Ingeniería de Software I y II. Ha participado como miembro de tribunal en eventos científicos, además ha presentado ponencias y trabajos obteniendo diferentes reconocimientos. Se desempeñó como Responsable del Segundo Perfil Informática para la Salud, Responsable del Grupo de Trabajo del Colectivo de Carrera de la Facultad 7, Jefa del Departamento de Práctica Profesional y Segundo Perfil y actualmente se desempeña como Subdirectora de Formación del Centro de Informática Médica (CESIM).

Correo electrónico: mpulido@uci.cu

Daniel Miranda Calzadilla: Ingeniero en Ciencias Informáticas, graduado en la Universidad de Ciencias Informáticas (UCI) en el curso 2007-2008. Instructor recién graduado en adiestramiento. Durante su trabajo como profesor ha impartido las asignaturas Introducción a la Programación y Programación I. Desarrolla su vinculación a la producción en el Departamento de Sistemas Especializados en Medicina (SEM) del Centro de Informática Médica (CESIM), donde se desempeña como Jefe de Módulo y Desarrollador en el Proyecto Rehabilitación.

Correo electrónico: dmiranda08@graduados.uci.cu

Resumen

El objetivo central de este trabajo es desarrollar una aplicación web que permita optimizar el trabajo y la documentación referente al Proceso de Tesis de Grado en la Facultad 7, brindando la posibilidad de llevar un control durante el transcurso de cada período sobre cada una de las tesis y otros eventos que se efectúan como parte del proceso, el cual se ejecuta hoy de forma manual, lo que influye negativamente en la calidad y efectividad del mismo.

El sistema brinda una amplia información sobre todo el tema referente a las tesis de grado, garantiza la centralización de los documentos generados entorno a este proceso, permite publicar y asignar los temas de tesis propuestos a los estudiantes de quinto año y mantener un seguimiento de la evaluación y evolución de las mismas. Esta aplicación presenta un requerimiento de hardware mínimo. Cuenta con una interfaz web amigable y fácil de usar permitiendo así que un gran número de usuarios puedan disfrutar de sus servicios.

Para el desarrollo de la aplicación se usó como metodología de desarrollo de software Rational Unified Process (RUP), lenguaje de modelado UML 2.0, herramienta Case Enterprise Architect 7.0, marco de trabajo Django, lenguaje de programación Python 2.6 y como Entorno Integrado de Desarrollo Wing IDE.

ÍNDICE

Introducción	1
Capítulo 1. Fundamentación Teórica	5
1.1. Conceptos Básicos relacionados con el Problema Planteado	5
1.2. Situación internacional de los sistemas de gestión de tesis.....	5
1.2.1. TESEO	5
1.2.2. TESIS.....	6
1.3. Situación nacional de los sistemas de gestión de tesis.	7
1.3.1. Tesis.uci.cu.....	7
1.3.2. Trabajo de diploma Akademos: Análisis y Diseño del Módulo de Gestión de Tesis.	7
1.3.3. Gestión de perfiles.....	8
1.4. Tendencias y tecnologías actuales utilizadas.....	9
1.4.1. Metodologías de Desarrollo.....	9
1.4.2. Metodología de desarrollo Rational Unified Process (RUP).....	9
1.5. Tecnologías de desarrollo.....	11
1.5.1. Lenguaje Unificado de Modelado (UML 2.0).....	11
1.5.2. Arquitectura.....	11
1.5.2. Ventajas del patrón MVC:	12
1.5.3. Lenguaje de programación Python.....	12
1.5.4. Herramienta de Modelado Enterprise Architect 7.0.	13
1.5.5. Framework Django.....	14
1.5.6. Entorno Integrado de Desarrollo Wing IDE.....	14
1.5.7. Sistema Gestor de Base de Datos PostgreSQL.....	14
Capítulo 2. Características del Sistema.....	16
2.1. Objeto de estudio.....	16
2.1.1. Flujo actual de los procesos involucrados en el campo de acción.....	16
2.1.2. Propuesta de solución.....	16

2.1.3. Descripción de los procesos que serán objeto de automatización.....	16
2.2. Modelo del Negocio	17
2.2.1.Descripción Textual de los Casos de Uso del Negocio.	19
2.3. Especificación de los requisitos de software.....	24
2.4 Definición de los casos de uso	30
Capítulo 3. Diseño del Sistema.....	35
3.1. Patrones de arquitectura.....	35
3.1.1. Descripción de la Arquitectura.....	35
3.2. Tratamiento de errores	36
3.3. Diseño del sistema.....	36
3.4. Modelo de diseño.....	37
3.5. Diagramas de clases del diseño.....	37
3.6. Diagramas de interacción.	40
3.7. Diagrama de secuencia.....	41
3.8. Diagrama de colaboración.	41
3.9. Patrones del diseño.....	50
Capítulo 4. Implementación.....	52
4.1. Modelo de datos.....	52
4.2. Diagrama de despliegue.	55
4.3 Diagramas de componentes.....	56
4.4. Tratamiento de errores.....	61
Conclusiones	63
Recomendaciones.....	64
Referencias Bibliográficas	65
Bibliografía	67
Glosario de Términos.....	70

Introducción

El término tesis procede de los idiomas griego y latín indistintamente, su origen puede enmarcarse alrededor del año 1433 (Siglo XV) en que los griegos lo definían como *théma*; por la idea de un tema de conversación. Posteriormente hacia el Siglo XVII el latín lo toma del griego y lo define como una conclusión mantenida por razonamientos.

En la antigüedad, la tesis consistía en una hoja de papel grande impresa por un solo lado, comenzando por la dedicatoria al padrino, que generalmente era el Virrey, el Arzobispo, algún gran señor o el fraile superior de un convento. Después de la dedicatoria venían los puntos del texto que exigen los estatutos y las conclusiones que iba a defender, afirmar o negar el estudiante, terminando con el nombre del Decano en turno y la fecha y hora del solemne acto.

En la actualidad el término tesis posee múltiples acepciones pero comúnmente se encuentra relacionado con la formación académica en las universidades y demás entidades de estudios superiores y técnicos como un producto que se elabora y se somete a la evaluación de un jurado calificado.

En Cuba se trabaja el término de tesis como un informe escrito cuyo propósito es regir la investigación desarrollada por el autor. En el mismo se plasma de forma ordenada las ideas, conceptos tratados y las soluciones que se sugieren. Sirve como fuente evaluativa de los conocimientos adquiridos por el autor durante un período de enseñanza determinado.

De lo antes expuesto se infiere que es un término muy antiguo cuyo significado ha evolucionado a través del tiempo pero que en esencia no dista del concepto milenario que le dio origen. En la actualidad por el avance tecnológico que presentan algunos centros universitarios se viene manejando un nuevo término, que es el de Tesis Digitales. Este término básicamente consiste en la forma de elaborar el documento que respalda la investigación ya que éste se elabora totalmente con ayuda de las Tecnologías de la Informática y las Comunicaciones.

Tesis Digitales: Son documentos digitales que pueden contener cualquier morfología de la información expresada en uno o varios medios (multimedia) y en uno o varios textos (hipertexto). Estos documentos son almacenados lógicamente y físicamente en un sistema de cómputo electrónico digital o en otro medio de almacenamiento, de manera tal que puedan ser recuperados por uno o varios usuarios simultáneamente.

La Universidad de las Ciencias Informáticas (UCI), cuya misión es formar ingenieros en Ciencias Informáticas se diferencia del resto de las universidades del país por ser una universidad con un mayor énfasis en la producción, además de que también posee un elevado nivel tecnológico. A la vez, posee elementos comunes con el resto de las universidades nacionales, como son: establecer la tesis como ejercicio académico para la culminación de los estudios del alumnado, según el Reglamento para el Trabajo Docente y Metodológico en la Educación Superior. La Universidad posee otras características particulares que inciden de manera sustancial en el proceso de elaboración de tesis de sus estudiantes, entre ellas: la amplia matrícula perteneciente a una misma especialidad, el régimen interno de los mismos y la representatividad de todas las provincias y municipios del país en el estudiantado.

A pesar de que se dispone de un elevado nivel tecnológico, en la Facultad 7 de la Universidad de las Ciencias Informáticas se carece de un sistema para la gestión de estos documentos que garantice la calidad del mencionado proceso y lograr que el mismo termine en el tiempo requerido.

En la Universidad de las Ciencias Informáticas cuando se gestiona una tesis, se le asigna al estudiante el tema y el tutor, hay que planificar los cortes de tesis, distribuir los profesores por cada tribunal, almacenar las notas de los cortes y tener constancia de todos los documentos relacionados con el desarrollo de la tesis. Todo este proceso se lleva a cabo de forma manual, lo que trae como consecuencia que se cometan errores, se deterioren los archivos físicos, que exista retraso en la información, que ocurran pérdidas de documentos importantes, que exista congestión de la red o que se saturen los buzones de correo, entre otros inconvenientes. Por otro lado, para realizar dicha asignación se necesita mucho tiempo demorando días para que los estudiantes puedan tener su tema de tesis y comiencen a trabajar en él.

Los directivos de las facultades, encargados de realizar la asignación de los perfiles de tesis y evaluación de los mismos se enfrentan a situaciones como:

-Se asignan perfiles de tesis que luego, por distintas razones, cambian o se eliminan y los encargados del proceso no están conscientes de dichos cambios.

- Se realizan propuestas de perfiles por parte de los estudiantes, que mientras no se controlen pueden ser evaluados, sin embargo más adelante pudiera comprobarse que no tienen alcance como trabajo de diploma. Así en un momento dado aparecen estudiantes sin temas de tesis ya en un período avanzado del curso.

-La revisión y corrección de los trabajos, por parte de los tutores, tiende a demorar el avance del mismo, ya que depende del tiempo disponible de las partes implicadas (tutor y tesista), para reunirse y realizar los señalamientos pertinentes.

- No se cuenta con un control preciso sobre los profesores vinculados a las tesis, ni respecto a los que han sido ubicados en comisiones de tesis, de modo que no es posible determinar quiénes de ellos se encuentran disponibles para desempeñar una u otra tarea.

Por todo lo anterior se plantea como problema a resolver: ¿Cómo viabilizar el proceso de gestión de la información que realiza el profesor para asignar tesis en la facultad 7? Este problema se enmarca en el objeto de estudio: proceso de gestión de la información relacionada con la gestión de tesis en la Universidad de las Ciencias Informáticas. Su campo de acción se enfoca en el proceso de gestión de la información relacionada con la gestión de tesis en la facultad 7 de la Universidad de las Ciencias Informáticas.

Por todo lo planteado, para darle solución al problema, se plantea como objetivo general: Desarrollar un sistema informático que permita viabilizar la gestión de la información que se genera en los procesos relacionados con la gestión de tesis en la facultad 7 de la Universidad de las Ciencias Informáticas (UCI).

Para poder dar cumplimiento al objetivo planteado, se proponen las siguientes tareas de la investigación:

- Analizar los procesos de negocio asociados a la gestión de la información relacionada con la gestión de tesis en la facultad 7 de la Universidad de las Ciencias Informáticas (UCI).
- Realizar un análisis acerca de los sistemas informáticos existentes a nivel nacional e internacional referentes a la gestión de la información relacionada con la gestión de tesis.
- Asimilar la arquitectura definida para el desarrollo del Sistema Integral de Gestión Administrativa.
- Analizar las necesidades de funcionamiento de la aplicación describiendo la Especificación de Requisitos de Software.

- Implementar el sistema informático siguiendo las necesidades de funcionamiento establecidas en la Especificación de Requisitos de Software.
- Diseñar el sistema informático utilizando la arquitectura definida para el desarrollo del Sistema Integral de Gestión Administrativa.

El desarrollo de este trabajo está organizado en cuatro capítulos, que a continuación se describen:

Capítulo 1. Fundamentación teórica: En este capítulo se realiza un estudio del arte de todos los Sistemas de Gestión de Tesis tanto a nivel nacional como a nivel internacional. Se exponen las tendencias, técnicas, tecnologías, metodología y los lenguajes de programación existentes usados para su desarrollo.

Capítulo 2. Características del sistema: Se presenta con argumentos el objeto de estudio, la descripción de los procesos del negocio del problema planteado y se especifican los requisitos no funcionales y funcionales del sistema.

Capítulo 3. Diseño del sistema: Se determinan las clases que se utilizarán en la implementación del sistema y la relación entre ellas. La realización de los diagramas de interacción de los casos de uso más significativos.

Capítulo 4. Implementación: Muestra el modelo de datos, el despliegue y los componentes utilizados a través de diagramas.

Capítulo 1. Fundamentación Teórica

En este capítulo se presenta el marco teórico y conceptual asociado a la problemática a resolver, aborda los antecedentes de los sistemas de gestión de tesis, tanto en el ámbito internacional como en el nacional. Se describen las tecnologías, herramientas y lenguajes a utilizar en el desarrollo del objetivo propuesto.

Las actividades asociadas a la realización y obtención de resultados, y al seguimiento, control y evaluación de cada tesis, tienen un objetivo fundamental dentro del conjunto de todas estas. Las tareas se organizan y se ejecutan paulatinamente, conformando el proceso en toda su amplitud del control de gestión de las tesis. La automatización de este proceso, lleva implícito una serie de conceptos que resultan de vital importancia destacar. Conociéndolos se puede lograr un mejor entendimiento y dominio del campo de acción.

Las Tecnologías de la Informática y Comunicaciones brindan herramientas y metodologías útiles para el desarrollo del sistema que se propone. Por tanto es necesario que se realice un análisis de las tendencias actuales y el desarrollo de las mismas para seleccionar las más convenientes y eficaces.

1.1. Conceptos Básicos relacionados con el Problema Planteado

Tesis: es una proposición que se expone, se defiende y tiene como corolario lógico una conclusión; implica entonces una evidente unidad entre proposición, demostración y conclusión.

1.2. Situación internacional de los sistemas de gestión de tesis.

1.2.1. TESEO

TESEO es un sistema de gestión académica, que se encarga de gestionar los resúmenes de las tesis doctorales, posee una base de datos que mantiene el registro de las tesis doctorales declaradas aptas. Todo el resultado del módulo de grabación es un fichero de texto que se envía por correo y que es procesado por el Consejo de Universidades para incorporar los datos a una base de datos temporal, y una ficha impresa que se envía por correo. El Consejo coordina los datos de la ficha con los del fichero, los pasa de la base de datos temporal a la base de datos definitiva, momento en que estarán disponibles para su consulta.

El sistema TESEO cuenta con las siguientes funcionalidades:

- Permite añadir, modificar y consultar los resúmenes de las tesis doctorales.
- Genera un fichero con los datos de la universidad a la que pertenece el estudiante y los datos de la dirección de la universidad.
- Imprime la ficha generada con los datos de los estudiantes.

1.2.2. TESIS

Universidad de Castilla-La Mancha, en esta universidad se implementó la aplicación informática TESIS, que fue desarrollada por los Servicios Informáticos de la Universidad de Castilla-La Mancha en el año 1998, a propuesta del Archivo General Universitario, en colaboración con la Secretaría General y el Vicerrectorado de Ordenación Académica. La aplicación permite realizar, de forma compartida y automatizada, la gestión administrativa de las tesis doctorales presentadas en la institución.

Mediante TESIS se controla el trámite y la localización física de la tesis doctoral a lo largo de todo el procedimiento para su lectura, desde el trámite de presentación al de archivo, incluyendo además otras posibilidades como es el análisis del contenido o la información relativa a la posterior publicación de estos trabajos de investigación. Las funciones de la aplicación informática TESIS se pueden resumir, atendiendo a la finalidad que cumplen en lo que se ha convenido denominar unidades administrativas, del siguiente modo:

Secretaría General:

- Registro de tesis doctorales admitidas a trámite para su lectura.
- Control del trámite de exposición pública y lectura de tesis.
- Control de la remisión al Archivo General de la Universidad para su archivo.

Vicerrectorado de Ordenación Académica:

- Control de la tramitación y contenido de las tesis doctorales.

Archivo General Universitario:

- Control de la transferencia y recepción de las tesis doctorales.

- Control del almacenamiento de las tesis doctorales.
- Control del almacenamiento de los expedientes de lectura de tesis doctorales.
- Control de la información contenida en las tesis doctorales.
- Control de la publicación de las tesis doctorales.
- Control de la consulta de las tesis doctorales.

1.3. Situación nacional de los sistemas de gestión de tesis.

1.3.1. Tesis.uci.cu.

En la Universidad de las Ciencias Informáticas (UCI) existe la aplicación Tesis.uci.cu que se encarga de reunir a tesistas, investigadores y tutores, para facilitarle organización en cuanto a la información, herramientas de comunicación, formación en línea, apoyo y asesoría en los procesos de desarrollo y tutoría de los trabajos de grado, tesis y trabajos de investigación.

1.3.2. Trabajo de diploma Akademos: Análisis y Diseño del Módulo de Gestión de Tesis.

En el proyecto Akademos de la UCI se realizó un trabajo de diploma que tiene como propósito analizar los procesos de gestión de tesis, por lo que se plantea como objetivo el Análisis y el Diseño de un módulo que gestione los procesos de tesis. Este trabajo de diploma queda como base para la futura implementación de los procesos de gestión de tesis mediante los cuales se podrá llevar el control de la información referente a los tesistas, temas de tesis y sus tutores, la creación de los tribunales de tesis y las evaluaciones que existan en el período de trabajo de diploma, así como su impresión.

Este trabajo de diploma planteó en el Análisis y Diseño realizado los siguientes procesos de tesis:

- Gestionar Problema.
- Buscar Problema.
- Gestionar Tribunal.
- Gestionar Estructura de Tribunal.
- Gestionar Asignación de Problema.
- Asignar Tribunal de Tesis.
- Gestionar Asignación de Tutor.
- Modificar Perfil.
- Aprobar Perfil.

- Gestionar Evaluación.

Estas aplicaciones han hecho aportes en cuanto a los pasos hacia la implementación de los sistemas de gestión de tesis en la UCI de acuerdo al marco en que han sido utilizados, sin embargo están muy lejos aún de las expectativas que se han trazado en este campo para la Facultad 7. Por tal motivo se ha decidido implementar el sistema, que comparte ciertas semejanzas con los ya existentes, pero sobretodo, ajustado a los intereses, estructura organizacional, estrategia de trabajo y perfil de estudio de la Facultad 7 de la Universidad de las Ciencias Informáticas.

Lo primero que se pretende lograr con esta aplicación es proporcionar un sistema totalmente libre de herramientas propietarias que ayude a mejorar el proceso de gestión de tesis, que brinde facilidades como, el hecho de mostrar un número de reportes considerables que le permitan al personal involucrado en el manejo de las tesis, llevar un constante y detallado control de las mismas, de modo que el flujo de información pueda hacerse de forma no presencial, además debe permitir tener constancia de las evaluaciones obtenidas en los diferentes cortes.

Además, de que estará disponible tanto para el personal de dirección: jefes de tribunales de discusión, vicedecano de producción, decano docente; como para los tutores y tesisas. O sea, que en principio no es una aplicación restringida sólo al alcance de la dirección de la facultad o para agilizar el trabajo de la misma, sino que es de alcance público. Los tutores podrán acceder y realizar las revisiones pendientes a las tesis, haciendo los señalamientos pertinentes sin necesidad de que tengan que concertar encuentros con los tesisas para debatir las modificaciones recomendadas.

1.3.3. Gestión de perfiles.

Los antecedentes planteados tanto a nivel nacional como a nivel internacional no presentan una solución capaz de resolver el problema planteado, el sistema TESEO sólo gestiona los resúmenes de las tesis doctorales. La aplicación TESIS perteneciente a la Universidad de Castilla-La Mancha tampoco se ajusta al proceso de negocio de la Facultad 7. La aplicación Tesis.uci.cu no cumple con los objetivos propuestos en este trabajo de diploma, ni tiene solución al problema planteado. El trabajo de diploma realizado en Akademos no brinda ninguna funcionalidad que permita poder llevar a cabo el proceso de gestión de tesis.

Por lo que es necesario realizar un sistema que sea capaz de gestionar trabajos de diplomas y que tenga las funcionalidades requeridas para un óptimo desarrollo de la aplicación.

Estos sistemas han hecho aportes en cuanto a los pasos hacia la implementación de sistemas de gestión de tesis en la UCI de acuerdo al marco en que han sido utilizados, sin embargo están muy lejos aún de las expectativas que se han trazado en este campo para la Facultad 7. Por tanto se ha decidido implementar el Módulo de Gestión de Tesis, que comparte ciertas semejanzas con los sistemas ya existentes, pero sobretodo ajustado a los intereses, estructura organizacional, estrategia de trabajo y perfil de estudio de la Facultad 7.

1.4. Tendencias y tecnologías actuales utilizadas.

1.4.1. Metodologías de Desarrollo.

Metodologías de Desarrollo: Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. [1] Las metodologías de desarrollo son importantes para el desarrollo del software ya que ofrecen documentación formal. Estas son útiles para garantizar la eficacia según los requisitos iniciales y la eficiencia al minimizar las pérdidas de tiempo en el proceso de generación de software.

1.4.2. Metodología de desarrollo Rational Unified Process (RUP).

Proceso para el desarrollo de un proyecto de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. [2] RUP es una metodología muy utilizada porque sirve de guía a los equipos de proyecto en cómo administrar el desarrollo iterativo, describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica de acuerdo a los requerimientos y a la arquitectura. Se caracteriza por tener 4 fases de desarrollo de software: Inicio, Elaboración, Construcción, y Trasmisión; y 9 disciplinas o flujos de trabajo, dividiéndose en 6 flujos ingenieriles (Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y 3 flujos de apoyo (Gestión de la Configuración y Cambio, Gestión de Proyecto, y Ambiente).

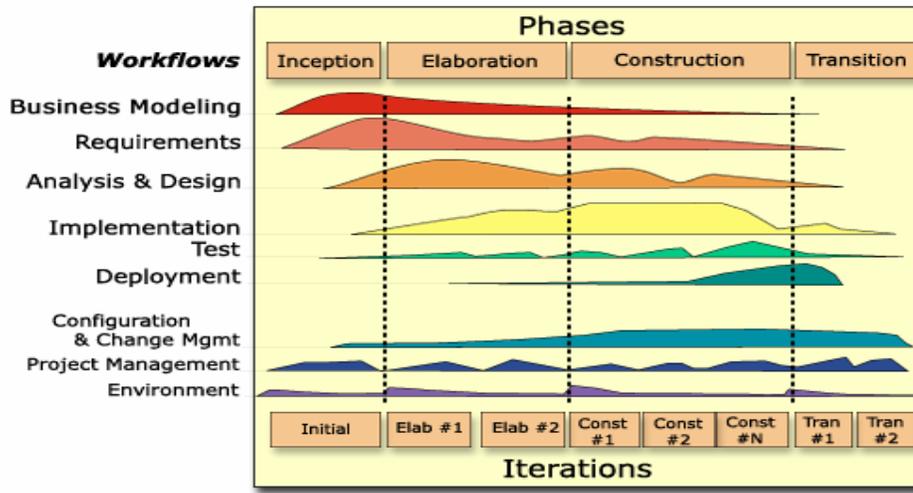


Figura 1. Ciclo de vida de RUP.

Entre las principales características del ciclo de vida de RUP se encuentran:

- Dirigido por casos de uso: Los casos de uso visualizan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de este momento los casos de uso guían el proceso de desarrollo, porque los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema en toda su amplitud, en la que el equipo de proyecto y los usuarios deben estar completamente de acuerdo, por eso se describen los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.
- Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, irá creciendo incrementalmente en cada iteración.

1.5. Tecnologías de desarrollo.

1.5.1. Lenguaje Unificado de Modelado (UML 2.0).

Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el RUP), pero no especifica en sí mismo qué metodología o proceso usar. [3].

UML no puede compararse con la programación estructurada debido a que no es un lenguaje de programación sino un lenguaje de modelado, que permite graficar una situación del mundo real en un requerimiento, mientras que la programación estructurada es una forma de programar al igual que la programación orientada a objetos. Sin embargo, la programación orientada a objetos es un complemento perfecto de este lenguaje de modelado, pero no por eso se usa sólo para lenguajes orientados a objetos. Este lenguaje gráfico está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que UML es un lenguaje, cuenta con reglas para combinar tales elementos.

Es importante recalcar que 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.

1.5.2. Arquitectura.

Patrón Modelo Vista Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.[4]

Vista: Es el responsable de la lógica de presentación y captura de datos del sistema al exterior y viceversa, presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Es el responsable de la lógica operacional de negocio, traslada las peticiones de la Vista al Modelo, y según la respuesta, la redirecciona o no a la Vista; además, carga objetos y opera con ellos.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

1.5.2. Ventajas del patrón MVC:

- MVC nos aporta una construcción de software mantenible, en la que se pueden localizar los errores de forma ágil. Supone un diseño modular y muy poco acoplado, favoreciendo la reutilización.
- Facilita la labor de todo el equipo: diseñadores gráficos, programadores y diseñadores de base de datos.
- Existen marcos de trabajo (frameworks) ya programados que facilitarán el trabajo de los miembros del equipo de trabajo.
- Al separar la presentación de la programación (o lógica de negocio), la aplicación es más fácil de modificar en el futuro.
- El resultado es más claro y la asignación de tareas dentro del equipo de trabajo es más fácil; la depuración de la aplicación es más sencilla y, finalmente, puede utilizarse un marco de trabajo (framework) bien testeado.

1.5.3. Lenguaje de programación Python.

Python es un lenguaje de scripting (secuencia de comandos) independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa desde aplicaciones Windows a servidores de red o incluso páginas web. Es un lenguaje interpretado, o sea no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e evita inconvenientes como una menor velocidad. [5]

Python permite dividir el programa en módulos reutilizables desde otros sistemas implementados con Python. Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas o como ejemplos para empezar a aprender Python. También hay módulos incluidos que

proporcionan entrada y salida de ficheros, llamadas al sistema, sockets y hasta interfaces a GUI (interfaz gráfica con el usuario). Python se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa.

Características:

- Python es un lenguaje simple y sencillo de aprender.
- Es libre y se distribuyen libremente copias de este software, lo cual permite leer su código fuente, hacerle cambios y usar partes del mismo en nuevos programas libres.
- Es un lenguaje de alto nivel.
- Es orientado a objetos, lo que permite agrupar en estructuras encapsuladas tanto sus datos como los métodos o funciones que manipulan esos datos.
- Tiene una librería extendida.

1.5.4. Herramienta de Modelado Enterprise Architect 7.0.

Enterprise Architect es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software (análisis y diseño orientados a objetos, construcción, pruebas, y despliegue). [6] El mismo combina el poder de la última especificación UML 2.0 con alto rendimiento e interfaz intuitiva, para traer modelado avanzado al escritorio y para el equipo completo de desarrollo e implementación. Las bases de Enterprise Architect están construidas sobre la especificación de UML 2.0 pero no se detiene ahí. Usa Perfiles UML para extender el dominio de modelado, mientras que la Validación del Modelo asegura integridad. Combina Procesos de Negocio, Información y Flujos de trabajo en un modelo usando extensiones gratuitas para BPMN y el perfil Eriksson-Penker.

Características:

- Permite tanto la ingeniería directa como inversa, soportada en varios lenguajes de programación.
- Brinda un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Presenta un lenguaje estándar común a todo el equipo de desarrollo, el cual facilita la comunicación.

1.5.5. Framework Django.

Django es un framework web de código abierto escrito en Python, que permite construir aplicaciones web más rápido y con menos código. Fue inicialmente desarrollado para gestionar aplicaciones web de páginas orientadas a noticias de Word Online, más tarde se liberó bajo licencia BSD. Django se centra en automatizar todo lo posible y se adhiere al principio DRY (Don't Repeat Yourself). [7].

Django está fuertemente inspirado en la filosofía de desarrollo Modelo Vista Controlador, sus desarrolladores declaran públicamente que no se sienten especialmente atados a observar estrictamente ningún paradigma particular y en cambio prefieren hacer "lo que les parece correcto". Gracias al poder de las capas mediator y foundation, Django permite que los desarrolladores se dediquen a construir los objetos Entity y la lógica de presentación y control para ellos.

1.5.6. Entorno Integrado de Desarrollo Wing IDE.

Wing IDE es un entorno de desarrollo totalmente integrado para Python, muy veloz y dotado de potentes herramientas de edición y depuración de código, así como buenas aptitudes de búsqueda optimizadas, lo que permite reducir en gran medida las probabilidades de errores y hace más fácil la navegación a través del código Python. [8].

Características:

- Potente entorno de desarrollo.
- Manejo fácil en integración de aplicaciones.
- Manejo sencillo de las pruebas de software.

1.5.7. Sistema Gestor de Base de Datos PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRE. Como muchos otros proyectos de código abierto, el desarrollo del mismo no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). [9].

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

PostgreSQL es un poderoso sistema manejador de bases de datos, es decir, un sistema diseñado para administrar grandes cantidades de datos, siendo la base de datos de código abierto más avanzada del mundo.

Características:

- Se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL.
- Permite aproximar los datos a un modelo objeto-relacional, es capaz de manejar complejas rutinas y reglas, tiene soporte para lenguajes procedurales internos y usa una arquitectura proceso-por-usuario cliente/servidor.
- Soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario; posibilita la integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

En este capítulo se realizó un análisis del tema de la investigación a nivel nacional e internacional, además, de una investigación de las principales metodologías de desarrollo de software, con el fin de justificar la empleada teniendo en cuenta las características del sistema a desarrollar. Por otra parte se realizó un estudio de las principales herramientas y tecnologías a utilizar para la realización de este trabajo teniendo en cuenta criterios que hagan óptimo al sistema.

Capítulo 2. Características del Sistema

En el presente capítulo se abordarán las características del sistema a construir. Se hará una descripción de cómo funciona el proceso de gestión de tesis en la Universidad de las Ciencias Informáticas. También se hará una propuesta del sistema y se presentará el modelo de casos de uso del negocio, los diagramas de actividades, la descripción de los casos de uso, y el modelo de objetos. También se mencionarán los requisitos tanto funcionales como no funcionales del software.

2.1. Objeto de estudio.

2.1.1. Flujo actual de los procesos involucrados en el campo de acción.

En las universidades la asignación de las tesis de los estudiantes se realiza a través de un encuentro entre estudiante y profesor o correo electrónico, lo que no posibilita un intercambio directo y rápido, ya que hay que ponerse de acuerdo para verse estudiante y profesor, además el correo se usa para muchas otras cosas, y pueden existir confusiones, entre otras causas porque a veces el buzón está lleno y no llegan los mensajes.

2.1.2. Propuesta de solución.

Un sistema online para el intercambio de información del proceso de gestión de tesis. Esto resultaría en una forma directa de comunicación entre el estudiante y profesor. Brindaría una fuente de información relacionada para cada tesis, así como las notas de los cortes y la planificación de los mismos.

2.1.3. Descripción de los procesos que serán objeto de automatización.

Se propone realizar un sistema para viabilizar el proceso de gestión de tesis, el sistema brindará funcionalidades para asignar tesis a los estudiantes, planificar los cortes, almacenar las notas de los mismos y distribuir los profesores por los tribunales de tesis. A este sistema accederán plenamente el planificador de la tesis, los tutores de cada tesis, y los tribunales existentes accederán para ver los cortes de tesis, los estudiantes podrán acceder para verificar su tema y ver su nota. Se ofrecerán opciones como: asignar tesis, planificar los cortes, almacenar las notas de los cortes y construir los tribunales por cada tesis. De esta manera se ganará mucho en cuanto a tiempo y organización en cuanto a gestión de tesis se refiere, ya que no será necesario que se vean directamente el profesor y el estudiante para que el

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

estudiante conozca el tema de tesis que le asignaron, entre muchas otras ventajas. Además no se corre el riesgo de que al estudiante o al profesor no le llegue la información debido a que tenga su correo lleno.

El sistema automatizará el registro de los datos que se encuentran en el proceso de gestión de tesis, en este caso toda la información referente a la planificación de los cortes de tesis, la asignación de temas de tesis, la creación y asignación de tribunales entre otros, esto será modificado por el Responsable de Tesis de la Facultad, además del Responsable de Tesis del Departamento de Producción. La aplicación posibilitará también la obtención de reportes de los cortes de tesis con las notas de los estudiantes y en caso de ausentarse con la causa, entre otras ventajas.

2.2. Modelo del Negocio

El modelo del negocio describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización. [10]. Este tipo de modelo tiene fronteras bien establecidas, donde se puede ver claramente quienes son las personas que lo inician, los beneficiados con cada uno de estos procesos, y además las personas que desarrollan las actividades en cada uno de estos procesos.

Actor del Negocio: Cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa. [11]

Usuario	Justificación
Estudiante	Es el principal beneficiado del proceso de tesis ya que se le asigna el Tema de Tesis definiéndosele el tutor y el Perfil de Tesis, en el cual se le define el alcance del trabajo que debe realizar. Es ubicado en un tribunal de defensa de tesis donde realiza los cortes de tesis, revisándosele el avance y la calidad de su trabajo, y en caso necesario se le realizan recomendaciones para mejorar las deficiencias detectadas.

Tabla 1: Relación de actores del negocio.

Trabajador del negocio: Un trabajador define el comportamiento y las responsabilidades de un individuo que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Trabajador	Justificación
Responsable de Tesis de la Facultad	Es el responsable de aprobar los perfiles de tesis en la facultad 7, de crear los tribunales de tesis, de planificar los cortes y de generar un documento con los resultados de las notas de los cortes de tesis. Así como de controlar el proceso de tesis de la facultad.
Responsable de Tesis del Departamento de Producción	Es el encargado de asignar el tema de tesis, el tutor y el oponente de la tesis. Elabora los perfiles de tesis de cada estudiante, también es quien le presenta el Perfil de Tesis al Responsable de Tesis de la Facultad para que sea aprobado, también debe conocer el resultado de los cortes de cada estudiante de su área temática.
Directivo	Es el encargado de pedir al responsable de tesis de la facultad, que comience con el proceso de la asignación y la planificación de cada tesis.
Tribunal de Tesis	Es el responsable de realizar el corte de tesis, donde hace recomendaciones y emite una nota.

Tabla 2: Relación de trabajadores del negocio.

Diagrama de Casos de Uso del Negocio: representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.

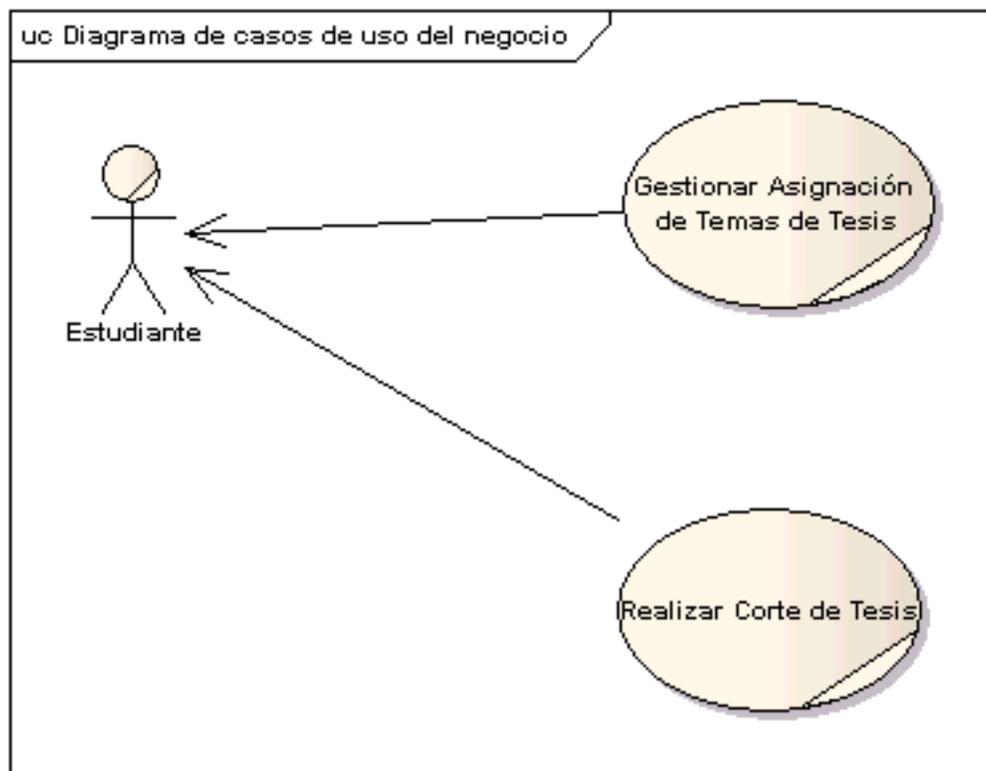


Figura 2: Diagrama de Casos de Uso del Negocio.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.2.1. Descripción Textual de los Casos de Uso del Negocio.

Caso de Uso:	Asignar tema de tesis (Inicia).
Actores:	Estudiante.
Trabajadores:	Directivo, Responsable de Tesis de la Facultad, Responsable de Tesis por Área Temática.
Propósito:	Asignar perfil de tesis al estudiante.
Resumen:	El caso de uso se inicia cuando el Directivo le pide al Responsable de Tesis de la Facultad que realice el proceso de asignación de tesis, el Responsable de Tesis de la Facultad a su vez le solicita al Responsable de Tesis del Departamento de Producción que realice la asignación de los temas de tesis del Departamento de Producción al que pertenece y que elabore los perfiles de tesis.
Casos de uso asociados:	No se aplica.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
	1.1) El Directivo le pide al Responsable de Tesis por Facultad que realice el proceso de asignación de tesis.
	1.2) El Responsable de Tesis de la Facultad le ordena al Responsable de Tesis del Departamento de Producción que proceda con la asignación de tesis.
	1.3) El Responsable de Tesis del Departamento de Producción concluye y le entrega la lista de asignación de tesis al Responsable de Facultad.
	1.4) El Responsable a nivel Facultad le pide al Responsable de Producción que elabore los perfiles de tesis.
	1.5) El Responsable de Producción consulta con los estudiantes los perfiles de tesis.
2) El estudiante evalúa el perfil de tesis y dice si	3) El Responsable de Producción le envía los perfiles de tesis al Responsable a nivel Facultad.
	3.1) El Responsable a nivel Facultad dictamina la aprobación o desaprobación de los perfiles.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

está de acuerdo o no.	3.2) El Responsable a nivel Facultad crea los tribunales de tesis.
	3.3) El Responsable a nivel Facultad le envía los perfiles de tesis al Directivo.
	3.4) El Directivo le envía los perfiles de tesis a los estudiantes.
4) El estudiante recibe el perfil de tesis.	5) Termina el caso de uso.
Flujo Alternativo de Eventos	
Viene de la sección 2 del flujo normal de eventos. 1.6) Se actualiza el perfil y se retorna al paso 2).	
Viene de la sección 3.1 del flujo normal de eventos 3.1.1) Se actualiza el perfil y se retorna al paso 3.1).	
Mejoras:	No se aplica

Tabla 3. Descripción del caso de uso del negocio "Asignar Tema de Tesis".

Caso de Uso:	Realizar cortes de tesis (inicia).
Actores:	Estudiante.
Trabajadores:	Directivo, Responsable de Tesis de la Facultad, Responsable de Tesis por Área Temática, Tribunal de Tesis.
Propósito:	Realizar corte de tesis.
Resumen:	Resumen: El caso de uso se inicia cuando el Directivo le pide al Responsable de Tesis de la Facultad que realice el proceso de planificación de los cortes de tesis, este a su vez verifica los horarios del Tribunal de Tesis y planifica los cortes. El Tribunal de Tesis ejecuta el corte y una vez terminado el Responsable de Tesis del Departamento de Producción le envía las notas de los estudiantes en el corte al Responsable de Tesis de la Facultad especificando las causas de los estudiantes evaluados de 2 puntos y/o de los No Evaluados. El Responsable de Tesis de la Facultad elabora un resumen de los resultados del corte de tesis y se lo envía al Directivo y finalmente el Directivo les envía los resultados del corte a los estudiantes.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Casos de uso asociados:	No se aplica.		
Flujo Normal de Eventos			
Acción del Actor	Respuesta del Negocio		
	1.1) El Directivo le pide al Responsable de Tesis de la Facultad que realice el proceso de planificación de los cortes de tesis.		
	1.2) El Responsable de Tesis de la Facultad verifica los horarios de los profesores del tribunal.		
	1.3) El Responsable a nivel Facultad realiza la planificación de los cortes.		
	1.4) El Responsable a nivel Facultad le envía la planificación de los cortes al Directivo.		
	1.5) El Directivo le envía la planificación de los cortes de tesis a los estudiantes.		
2) El estudiante recibe la planificación de los cortes de tesis.	3) El tribunal de tesis ejecuta el corte.		
	3.1) El Responsable de Tesis del Departamento de Producción le envía las notas de los estudiantes en el corte al Responsable a nivel Facultad.		
	3.2) El Responsable a nivel Facultad elabora un resumen de los resultados del corte.		
	3.3) El Responsable a nivel Facultad le envía los resultados al Directivo.		

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

	3.4) El Directivo le envía los resultados a los estudiantes.
4) El estudiante recibe el resultado del corte de tesis.	5) El caso de uso termina.
Mejoras:	No se aplica.

Tabla 4. Descripción del caso de uso del negocio “Realizar Tema de Tesis”.

Diagrama de Actividades: contiene estados en que puede hallarse una actividad. Un estado de actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo.

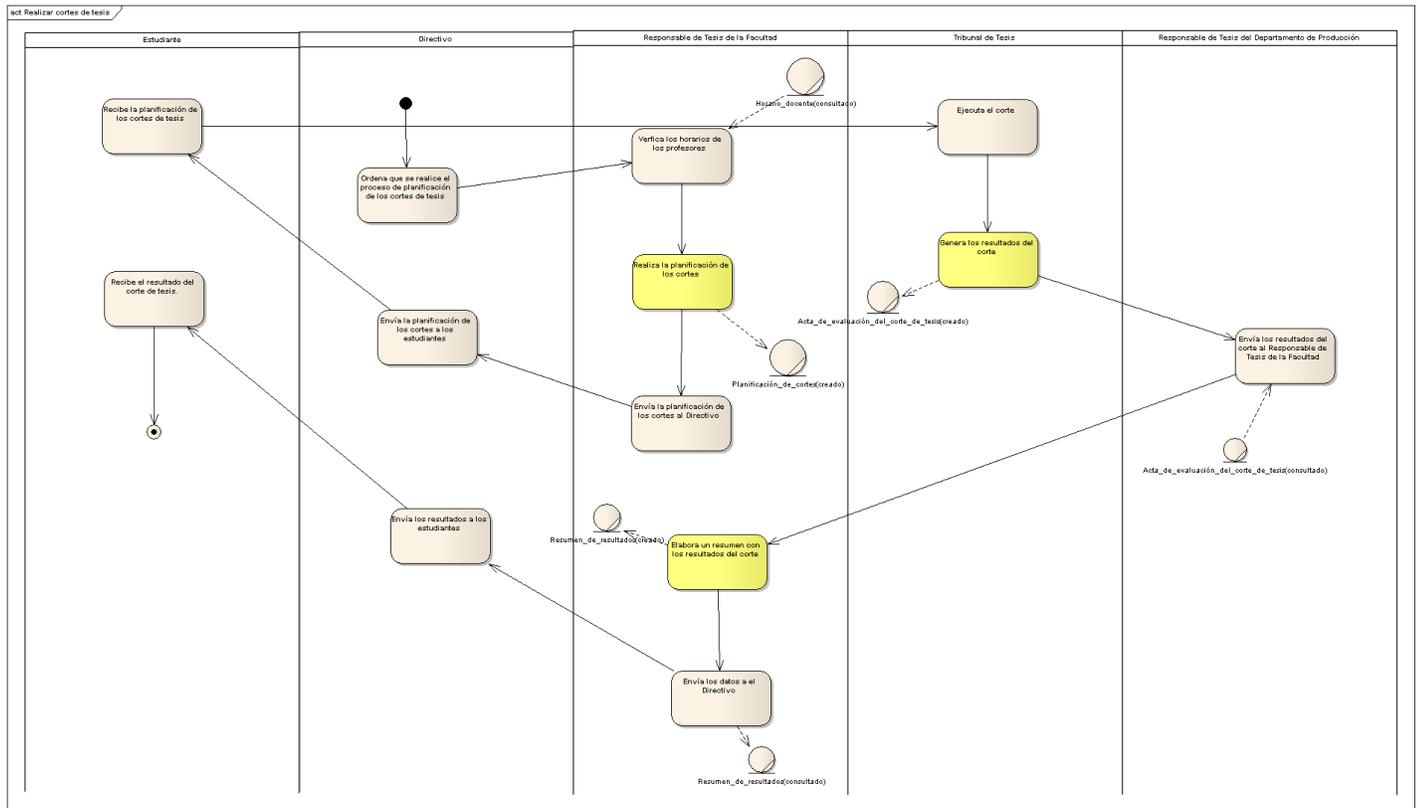


Figura 3: Diagrama de actividades: Realizar Corte de Tesis

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

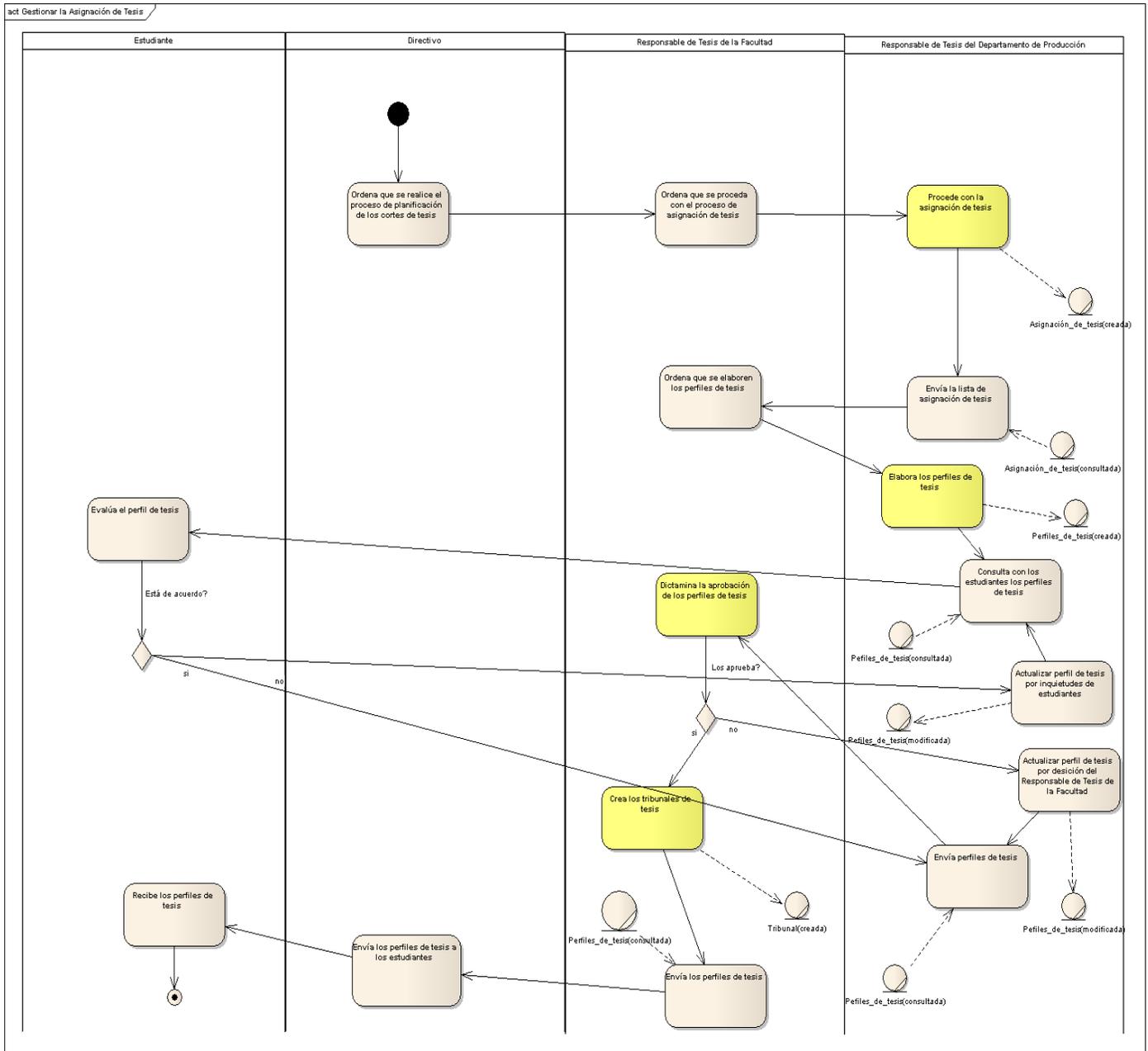


Figura 3: Diagrama de actividades: Gestionar la Asignación de Tesis.

Modelo de Objetos: muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos. [12].

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

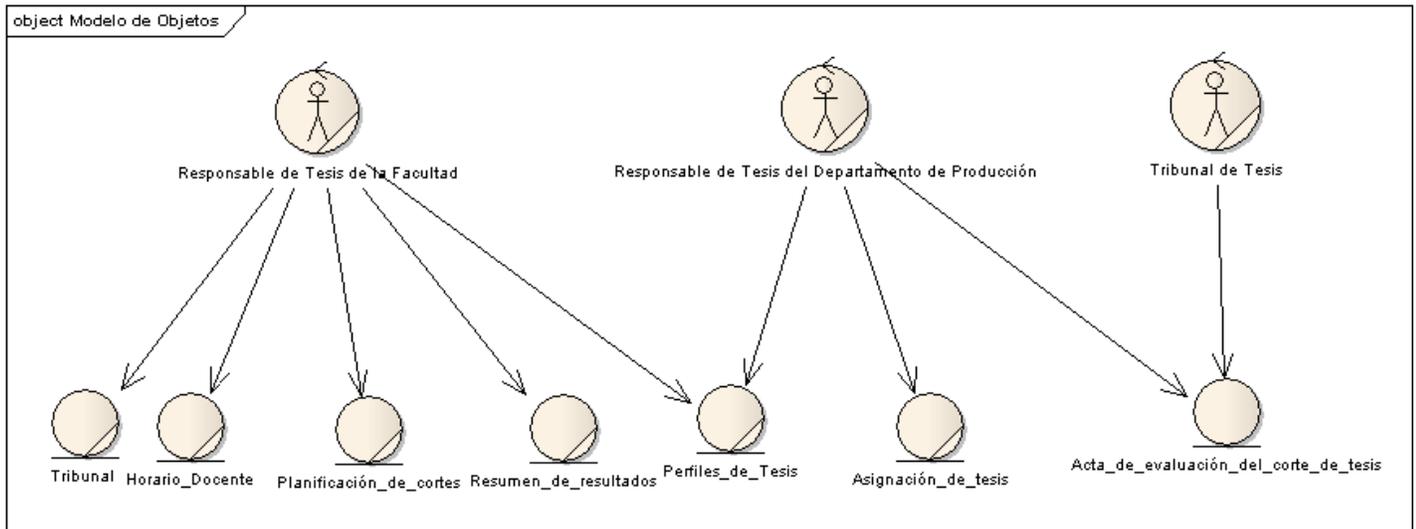


Figura 5: Modelo de Objetos

2.3. Especificación de los requisitos de software.

Requerimiento: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. [13]. Los requerimientos de software son las condiciones necesarias que se debe tener para el desarrollo de un sistema. Estos requerimientos junto a una interfaz de usuario definen el ámbito de un sistema.

Los requerimientos se dividen en dos tipos:

- **Requerimientos Funcionales:** son capacidades o condiciones que el sistema debe tener.

Los requerimientos funcionales del Sistema de Gestión de Tesis posibilitaron que se definiera el alcance del sistema teniendo en cuenta las acciones a realizar y el intercambio de datos entre sus diferentes funciones.

Los requerimientos funcionales del sistema propuesto:

RF1. Gestionar Tema de Tesis.

RF1.1. Insertar Tema de Tesis.

RF1.2. Buscar Tema de Tesis.

RF1.3. Modificar Tema de Tesis.

RF1.4. Eliminar Tema de Tesis.

RF2. Gestionar Perfil de Tesis.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RF2.1. Insertar Perfil de Tesis.

RF2.2. Buscar Perfil de Tesis.

RF2.3. Modificar Perfil de Tesis.

RF2.4. Eliminar Perfil de Tesis.

RF3. Gestionar Tribunal de Tesis.

RF3.1. Insertar Tribunal de Tesis.

RF3.2. Buscar Tribunal de Tesis.

RF3.3. Modificar Tribunal de Tesis.

RF3.4. Eliminar Tribunal de Tesis.

RF4. Gestionar la planificación del corte de tesis.

RF4.1. Insertar planificación del corte de tesis.

RF4.2. Buscar planificación del corte de tesis.

RF4.3. Modificar corte de tesis.

RF4.4. Eliminar corte de tesis

RF5. Gestionar Acta de Corte de Tesis.

RF5.1. Insertar Acta de Corte de Tesis.

RF5.2. Buscar Acta de Corte de Tesis.

RF5.3. Modificar Acta de Corte de Tesis.

RF5.4. Eliminar Acta de Corte de Tesis.

RF6. Reporte de los resultados de los cortes de tesis.

RF 7. Gestionar evaluación.

RF 7.1 Insertar evaluación.

RF 7.2. Buscar evaluación.

RF 7.3. Eliminar evaluación.

RF 8. Gestionar clasificación del Área de Desarrollo.

RF 8.1 Insertar clasificación del Área de Desarrollo.

RF 8.2. Buscar clasificación del Área de Desarrollo.

RF 8.3. Eliminar clasificación del Área de Desarrollo.

RF 8. Gestionar clasificación.

RF 8.1 Insertar clasificación.

RF 8.2. Buscar clasificación.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

RF 8.3. Eliminar clasificación.

- **Requerimientos no funcionales:** son propiedades o cualidades que el producto debe tener. Estos se vinculan directamente con los funcionales y forman una parte significativa de la Especificación de los Requisitos.

Apariencia o Interfaz Externa: Describe la apariencia del producto y se especifica cómo va a quedar la interfaz externa del producto. Se tiene en cuenta la intercomunicación con los otros sistemas.

Nº: RNF 1	
Nombre	Apariencia o Interfaz Externa
Descripción	<ul style="list-style-type: none">- Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.- La interfaz contará con imágenes y menús desplegables que faciliten y aceleren su utilización.- La entrada de datos incorrecta será detectada claramente e informada al usuario detalladamente.- Todos los textos y mensajes en pantalla aparecerán en idioma español.- El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, lo que minimizará los pasos a dar en cada proceso.- La interfaz será sencilla, amigable, intuitiva y de fácil navegación por el usuario, con el objetivo de evitar la resistencia humana al uso del nuevo sistema.

Tabla 5. Requerimientos No Funcionales: Apariencia o Interfaz Externa

Usabilidad: Describe los niveles apropiados de usabilidad, dados los usuarios finales del producto. Para ello debe revisarse la especificación de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

Nº: RNF 2	
Nombre	Usabilidad
Descripción	<ul style="list-style-type: none">- El sistema proporciona un acceso fácil y rápido al usuario, con el objetivo de proveer familiarización con el uso de la misma.

Tabla 6. Requerimientos No Funcionales: Usabilidad

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Rendimiento: Se relaciona con tiempos de respuesta estimados, requeridos y esperados para la ejecución en línea de procesos del sistema. Tiene como base la plataforma tecnológica y escenarios específicos a los que en teoría el sistema estará expuesto y frente a los que deberá responder.

Nº: RNF 3	
Nombre	Rendimiento
Descripción	<ul style="list-style-type: none">- El sistema proporcionará tiempo de respuesta aceptable para los procesos en línea.- Garantizará la velocidad de navegación para los usuarios.

Tabla 7. Requerimientos No Funcionales: Rendimiento

Seguridad: Se relaciona con la confidencialidad de los datos en la transmisión y en el almacenamiento, junto con las necesidades del sistema. Posibilita evitar intrusiones no autorizadas al mismo y la capacidad para seguir eventos que comprometan esta seguridad a través del tiempo.

Nº: RNF 4	
Nombre	Seguridad
	-La información queda protegida de los usuarios que no estén autorizados a acceder a la aplicación.

Tabla 8 Requerimientos No Funcionales: Seguridad

Disponibilidad: Significa que a los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

Nº: RNF 5	
Nombre	Disponibilidad
Descripción	<ul style="list-style-type: none">- El sistema estará accesible las 24 horas del día.

Tabla 9 Requerimientos No Funcionales: Disponibilidad

Software: Se especifican las condiciones o capacidades que el sistema debe cumplir. El software que se usará tanto en la parte del servidor como en la del cliente.

Nº: RNF 6	
Nombre	Software

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Descripción	<ul style="list-style-type: none">- El servidor deberá contar con el Sistema Operativo Windows XP o superior, GNU/Linux, o Mac OS, también deberá tener instalado el Apache Server 2.2 con los módulos ModPython o WSGI.- La computadora cliente deberá contar con un navegador, se recomienda que sea Internet Explorer 8.0 o superior o Mozilla Firefox con la versión 3.0 como mínimo .
--------------------	---

Tabla 10 Requerimientos No Funcionales: Software

Hardware: Elementos de hardware de que se disponen.

Nº: RNF 7	
Nombre	Hardware
Descripción	<ul style="list-style-type: none">- Para garantizar el correcto funcionamiento de la aplicación, se necesita una computadora con un procesador Pentium IV ó superior y una memoria RAM de 256 megabytes ó más.

Tabla 11 Requerimientos No Funcionales: Hardware

Restricciones en el diseño y la implementación: Especifica o restringe la codificación o construcción de un sistema. Son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

Nº: RNF 8	
Nombre	Restricciones en el Diseño y la Implementación.
Descripción	<ul style="list-style-type: none">- Se utilizará el patrón de diseño MVC.- Para el desarrollo del análisis y el diseño del sistema deberá ser utilizada la metodología RUP, usando el lenguaje de modelado UML y como herramienta case para llevarlo a cabo el Enterprise Architect versión 7.0 o superior.- Para la implementación se utilizará como lenguaje de programación Python, Entorno Integrado de Desarrollo Wing, framework de desarrollo Django y Sistema Gestor de Base de Datos POSTGRESQL 8.3.

Tabla 12 Requerimientos No Funcionales: Restricciones en el Diseño y la Implementación

Ayuda y documentación: Documentos ó enlaces que sirvan de guía para el manejo de las funciones del sistema. Esto posibilita que se facilite su uso.

Nº: RNF 9	
Nombre	Ayuda y documentación

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Descripción	- Presentará mensajes de ayuda para orientar en el uso de la interfaz, así como un asterisco en los campos que son obligatorios.
--------------------	--

Tabla 13 Requerimientos No Funcionales: Ayuda y documentación

- **Diagrama de casos de uso del Sistema**

El Modelo de Casos de Uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requerimientos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El mismo contiene actores, casos de uso y sus relaciones y constituye una entrada para el análisis, el diseño, la implementación y las pruebas. [14].

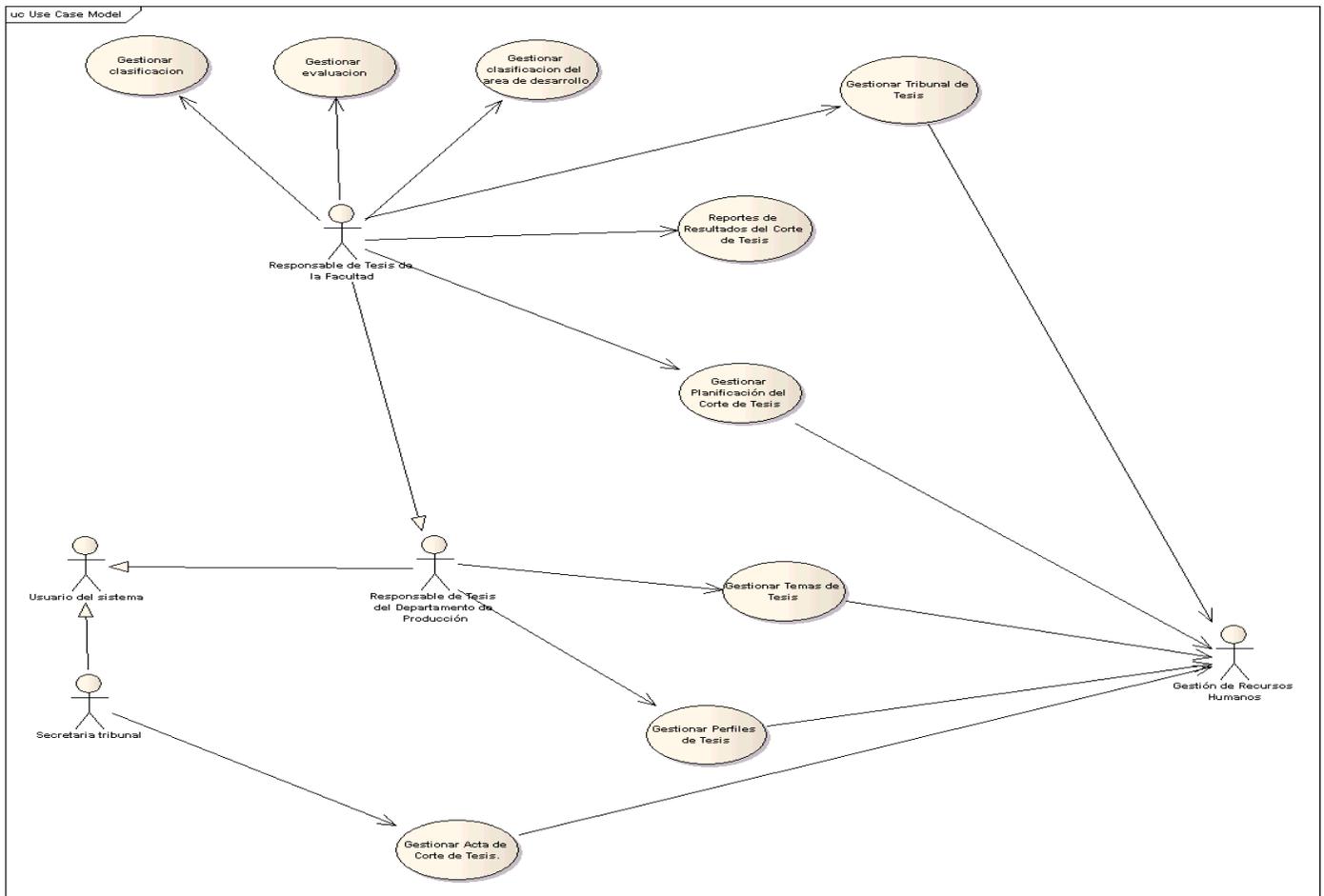


Figura 5: Diagrama de Casos de Uso del Sistema.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.4 Definición de los casos de uso

- Definición de los actores

Actores	Justificación
Secretario de Tribunal	Es el encargado de gestionar las actas de las tesis, de guardar las evaluaciones y las correcciones que se generen durante los cortes de tesis.
Responsable de Tesis de la Facultad	Es el encargado de gestionar los tribunales de tesis, la planificación de los cortes, los objetivos a evaluar en los cortes y crear un reporte de los resultados de los cortes.
Responsable de Tesis del Departamento de Producción	Es el encargado de gestionar los temas de tesis y los perfiles de tesis.
Gestión de Recursos Humanos	Es un sistema que se encarga de buscar los datos de los usuarios, en este caso estudiantes y profesores de la Universidad de las Ciencias Informáticas.

Tabla 14. Actores del Sistema.

- Listado de casos de uso

Los casos de uso son definidos por los actores del dominio y para el establecimiento de los casos de uso se tuvo un conocimiento previo de los requerimientos del sistema a desarrollar. Ellos definen verdaderamente lo que se quiere y así se podría concretar toda la información en casos de uso.

CU-1	Gestionar Acta del Corte de Tesis
Actor	Secretario de Tribunal
Descripción	El caso de uso Gestionar Acta Corte de Tesis se inicia cuando el secretario del tribunal gestiona la información de los cortes mediante el acta de los cortes de tesis y finaliza cuando la secretaria del tribunal ha gestionado la información.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Referencia	RF 5.1; RF 5.2; RF 5.3; RF 5.4.
-------------------	---------------------------------

Tabla 15. Casos de Uso del Sistema: Gestionar Acta del corte de tesis.

CU-2	Gestionar Tribunal de Tesis
Actor	Responsable de Tesis Facultad
Descripción	El caso de uso Gestionar Tribunal de Tesis se inicia cuando el Responsable de Tesis por la Facultad gestiona los tribunales de tesis, o sea: los crea, los elimina, los modifica o los busca y termina cuando el responsable de tesis por la Facultad realiza completamente cualquiera de estas operaciones.
Referencia	RF 3.1; RF 3.2; RF 3.3; RF 3.4.

Tabla 16. Casos de uso del Sistema: Gestionar Tribunal de Tesis.

CU-3	Gestionar Planificación de Corte de Tesis.
Actor	Responsable de Tesis Facultad
Descripción	El caso de uso Gestionar Planificación de Corte de Tesis se inicia cuando el responsable de tesis por la facultad gestiona la planificación de los cortes de tesis, o sea: los crea, los elimina los modifica o los busca, y finaliza cuando se realiza completamente cualquiera de estas operaciones.
Referencia	RF 4.1; RF 4.2; RF 4.3; RF 4.4.

Tabla 17. Casos de Uso del Sistema: Gestionar Planificación de Corte de Tesis.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

CU-4	Gestionar Tema de Tesis
Actor	Responsable Tesis del Departamento de Producción
Descripción	El caso de uso Gestionar Tema de Tesis se inicia cuando el Responsable de Tesis del Departamento de Producción gestiona los temas de tesis, o sea: los crea, los elimina los modifica o los busca. El caso de uso termina cuando el Responsable de Tesis del Departamento de Producción termina de realizar cualquier operación de gestión.
Referencia	RF 1.1; RF 1.2; RF 1.3; RF 1.4.

Tabla 18. Casos de Uso del Sistema: Gestionar Tema de Tesis.

CU-5	Gestionar Perfil Tesis
Actor	Responsable Tesis del Departamento de Producción
Descripción	El caso de uso Gestionar Perfil de Tesis se inicia cuando el Responsable de Tesis del Departamento de Producción gestiona los perfiles de tesis, los crea, los elimina los modifica o los busca y termina cuando el Responsable de Tesis del Departamento de Producción completa cualquiera de estas operaciones.
Referencia	RF 3.1; RF 3.2; RF 3.3; RF 3.4.

Tabla 19. Casos de Uso del Sistema: Gestionar Perfiles Tesis.

CU-6	Reporte de Resultados del Corte de Tesis
Actor	Responsable de Tesis Facultad
Descripción	El caso de uso Reporte de Resultados del Corte de Tesis se inicia cuando el Responsable de Tesis de la Facultad va a redactar un reporte con todas las notas de los estudiantes en los cortes de tesis y termina cuando el Responsable de Tesis de la Facultad termina de

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

	redactar el reporte.
Referencia	RF6.

Tabla 20. Casos de Uso del Sistema: Reporte Resultado Corte de Tesis.

CU-7	Gestionar Evaluación
Actor	Responsable de Tesis Facultad
Descripción	El caso de uso Gestionar Evaluación se inicia cuando el Responsable de Tesis de la Facultad va a insertar, modificar o eliminar una evaluación y termina cuando el Responsable de Tesis de la Facultad termina de insertar, modificar o eliminar la evaluación.
Referencia	RF7.1; RF7.3; RF7.3.

Tabla 21. Casos de Uso del Sistema: Gestionar Evaluación.

CU-9	Gestionar Clasificación del Área de Desarrollo
Actor	Responsable de Tesis Facultad
Descripción	El caso de uso Gestionar Clasificación del Área de Desarrollo se inicia cuando el Responsable de Tesis de la Facultad va a insertar, modificar o eliminar una clasificación del Área de Desarrollo y termina cuando el Responsable de Tesis de la Facultad termina de insertar, modificar o eliminar la Clasificación del Área de Desarrollo.
Referencia	RF8.1; RF8.2; RF8.3.

Tabla 22. Casos de Uso del Sistema: Gestionar Clasificación del Área de Desarrollo.

CU-8	Gestionar Clasificación
-------------	-------------------------

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Actor	Responsable de Tesis Facultad
Descripción	El caso de uso Gestionar Clasificación se inicia cuando el Responsable de Tesis de la Facultad va a insertar, modificar o eliminar una clasificación para un trabajo de diploma, y termina cuando el Responsable de Tesis de la Facultad termina de insertar, modificar o eliminar la clasificación.
Referencia	RF9.1; RF9.2; RF9.3.

Tabla 23. Casos de Uso del Sistema: Gestionar Clasificación.

En el presente capítulo se analizaron las características fundamentales del sistema. En tal sentido y a partir del análisis de los procesos de negocio se obtuvo un listado con los requerimientos funcionales y no funcionales para el desarrollo del sistema, que se representaron en el Diagrama de Casos de Uso del Sistema y finalmente se describieron todas las acciones de los actores del sistema dentro de los procesos con los que se vinculan.

Capítulo 3. Diseño del Sistema

Este capítulo abordará sobre el diseño del sistema propuesto. Se describe la arquitectura que se utiliza, la cual se verá reflejada en los diagramas de análisis y diseño, a través de la utilización de las tecnologías, lenguaje y metodología antes descritos.

3.1. Patrones de arquitectura.

3.1.1. Descripción de la Arquitectura.

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema, establece los fundamentos para que analistas, diseñadores y programadores, trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. [15] Aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño.

El patrón de diseño Modelo Vista Controlador o MVC describe una forma, muy utilizada en la web, de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

Modelo - Componente encargado del acceso a datos.

Vista - Define la interfaz de usuario, HTML+CSS... enviados en el navegador.

Controlador - Responde a eventos y modifica la vista y el modelo.

Ya que la "C" la maneja el propio framework y la mayoría de lo excitante de Django sucede en los modelos, plantillas y vistas, a Django se le ha definido como framework MTV. En el patrón de desarrollo MTV:

- La "M" se refiere al modelo, la capa de acceso de datos. Esta capa contiene todo lo referido a los datos: cómo acceder a ellos, cómo validarlos, qué comportamiento tienen y las relaciones entre ellos.

- La "T" se refiere a las plantillas (*templates*), la capa de presentación. Esta capa contiene las decisiones relacionadas con la presentación: cómo debería mostrarse algo en una página web u otro tipo de documento.
- La "V" se refiere a la vista, la capa de la lógica de negocio. Esta capa contiene el acceso al modelo y delega en las plantillas apropiadas. Puede pensar en ella como el puente entre los modelos y las plantillas.

El sistema va a estar compuesto por 8 clases modelos: Acta, Perfil_de_Tesis, Tema_de_Tesis, Corte_de_Tesis, Tribunal, Evaluacion, Clasificacion y Clasificacion_Area_de Desarrallo. Además va a heredar de una plantilla principal y a partir de ahí se derivan las otras plantillas. En la clase Views.py se albergarán las funciones necesarias para el desarrollo de las funcionalidades.

3.2. Tratamiento de errores

El tratamiento de los errores es un paso indispensable para el buen funcionamiento del sistema. Los errores más comunes que se pueden presentar están relacionados con el trabajo con la base de datos, fundamentalmente en los procesos de inserción, eliminación y/o modificación de datos. Es por esto que se han establecido mecanismos que visualicen la información para evitar en la medida de lo posible la introducción manual errónea de los datos, y así minimizar los errores que pueda tener la aplicación. Además se muestran mensajes indicando al usuario los errores cometidos, permitiéndole a la vez corregir su entrada con facilidad y continuar. También se le muestran al usuario mensajes de confirmación cuando va a eliminar un objeto.

3.3. Diseño del sistema

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, o sea, cómo cumple el sistema sus objetivos. El mismo debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. El resultado final más importante en esta etapa del desarrollo será el modelo de diseño.

3.4. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Sirve de abstracción de la implementación y es utilizado como entrada fundamental de las actividades de implementación.

3.5. Diagramas de clases del diseño

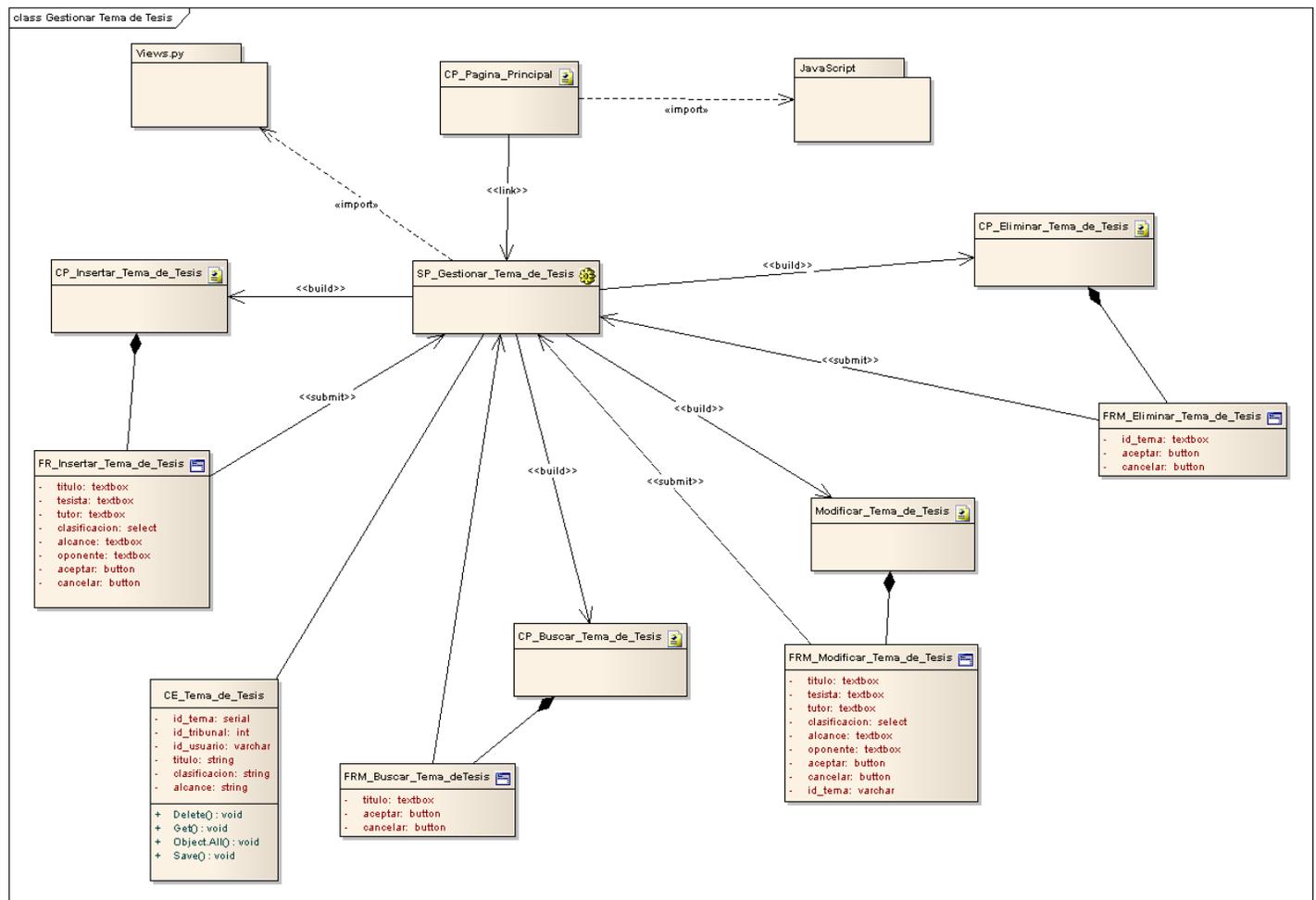


Figura 6: Diagrama de Clases del Diseño del CU Gestionar Tema de Tesis.

CAPÍTULO 3. DISEÑO DEL SISTEMA

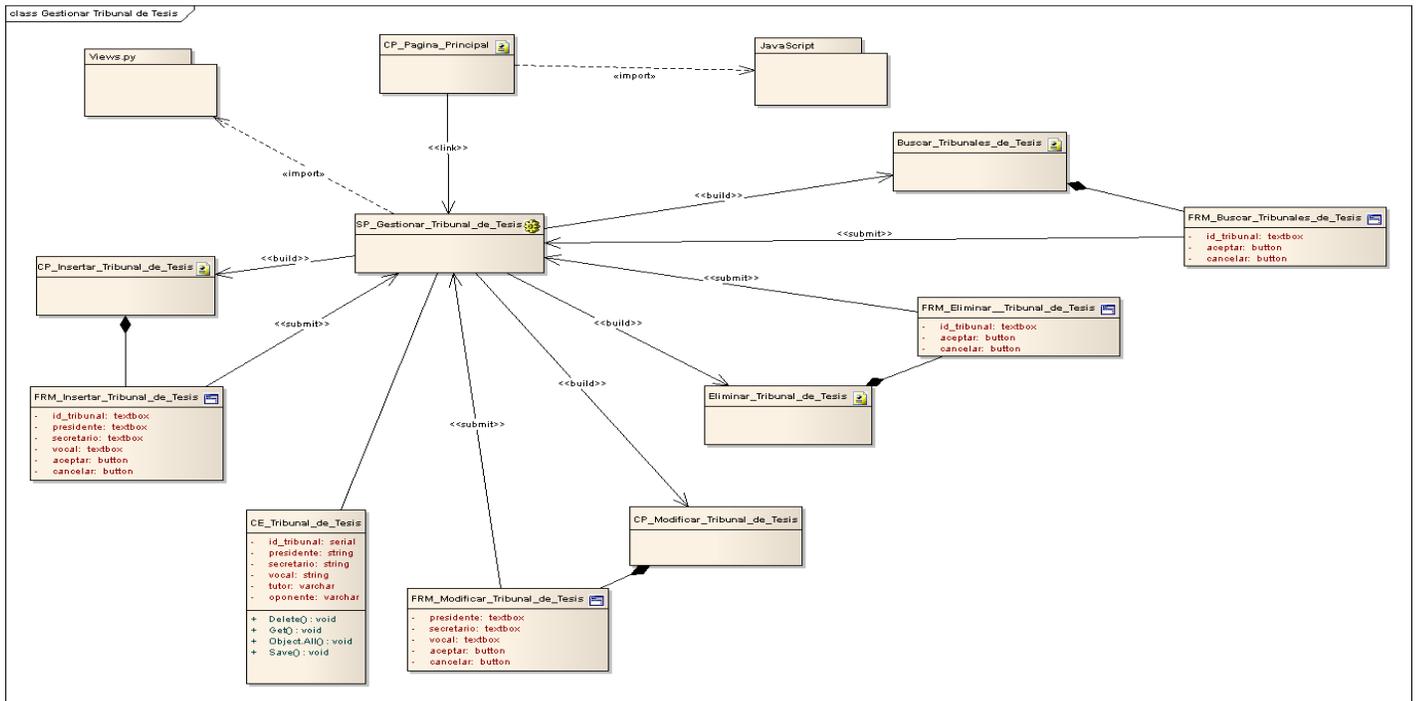


Figura 7: Diagrama de Clases del Diseño del CU Gestionar Tribunal de Tesis.

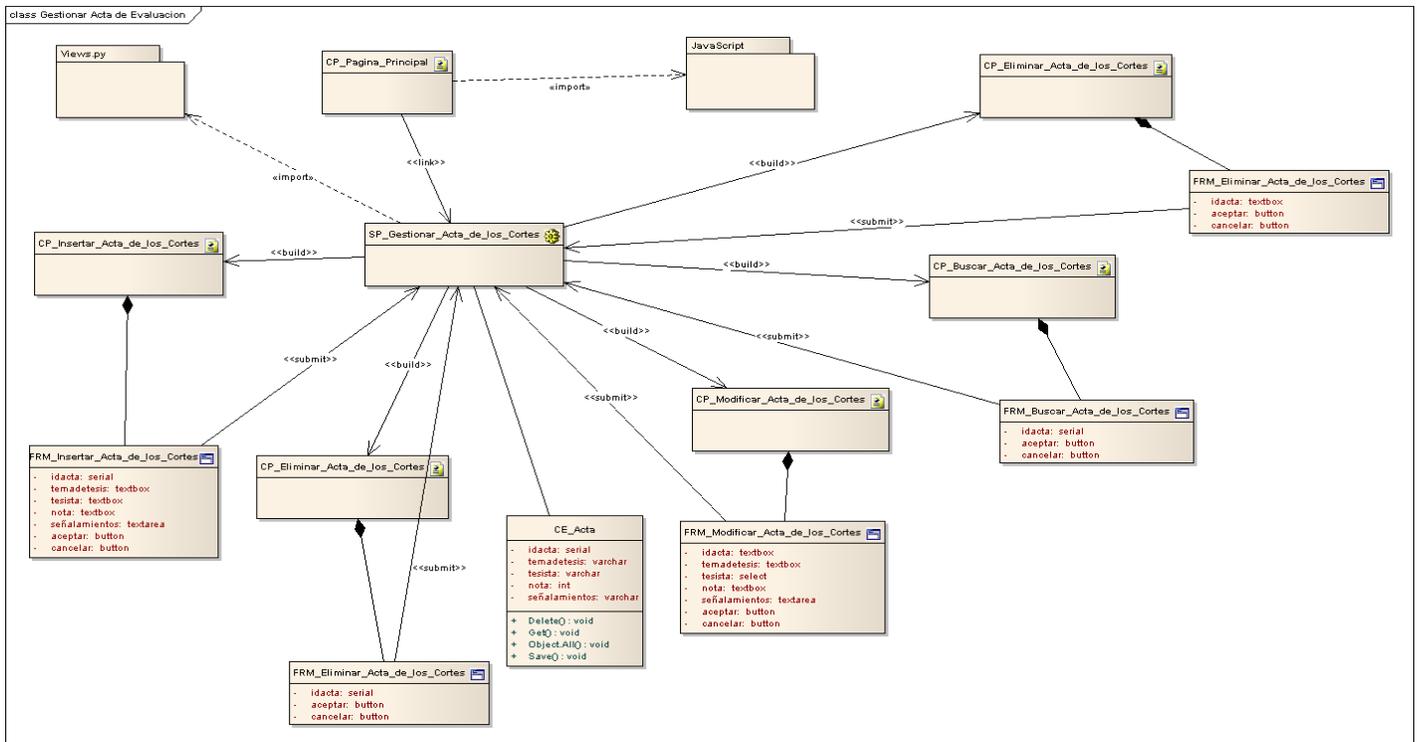


Figura 8: Diagrama de Clases del Diseño del CU Gestionar Acta de Corte de Tesis.

CAPÍTULO 3. DISEÑO DEL SISTEMA

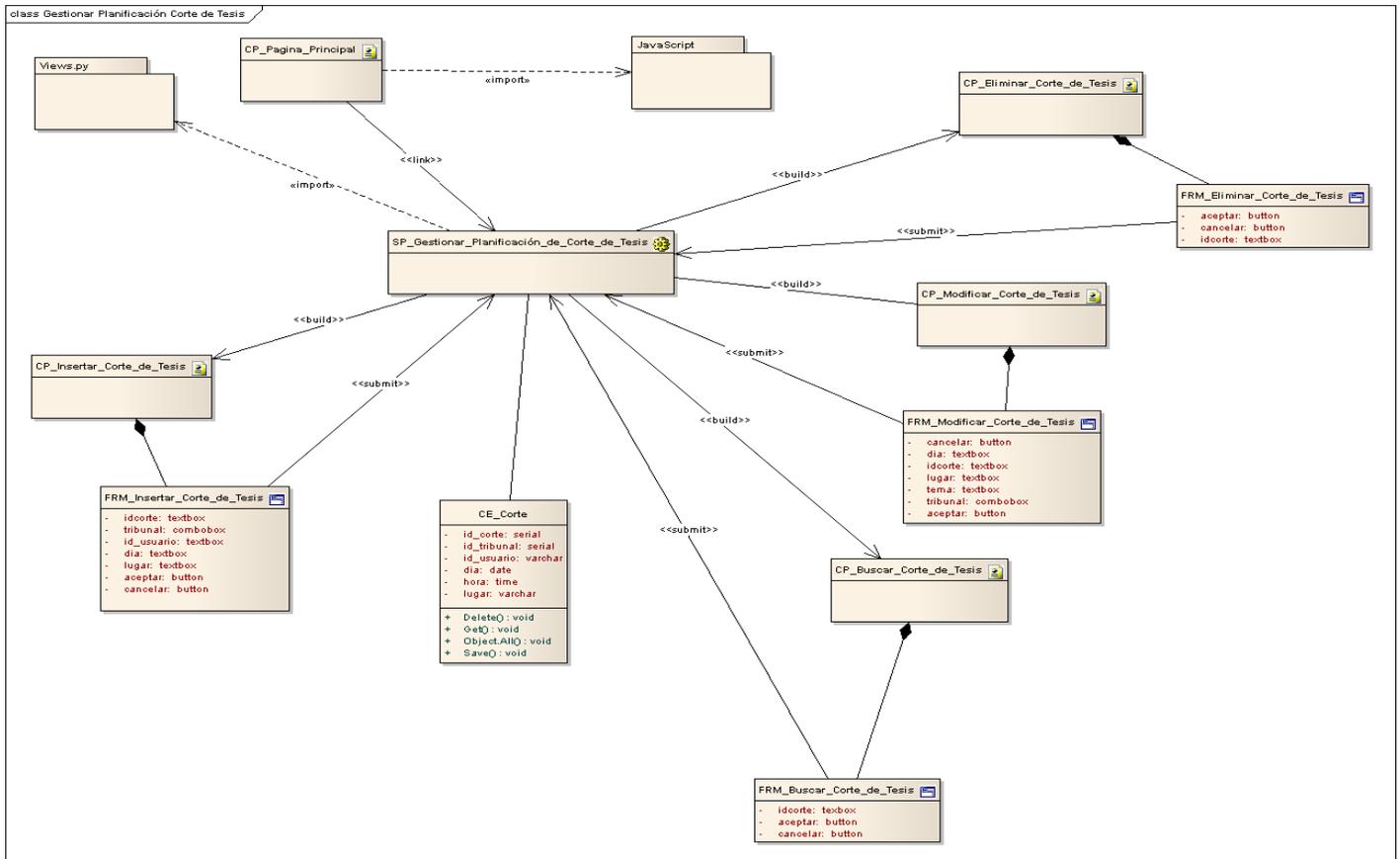


Figura 9: Diagrama de Clases del Diseño del CU Gestionar Planificación de Corte de Tesis.

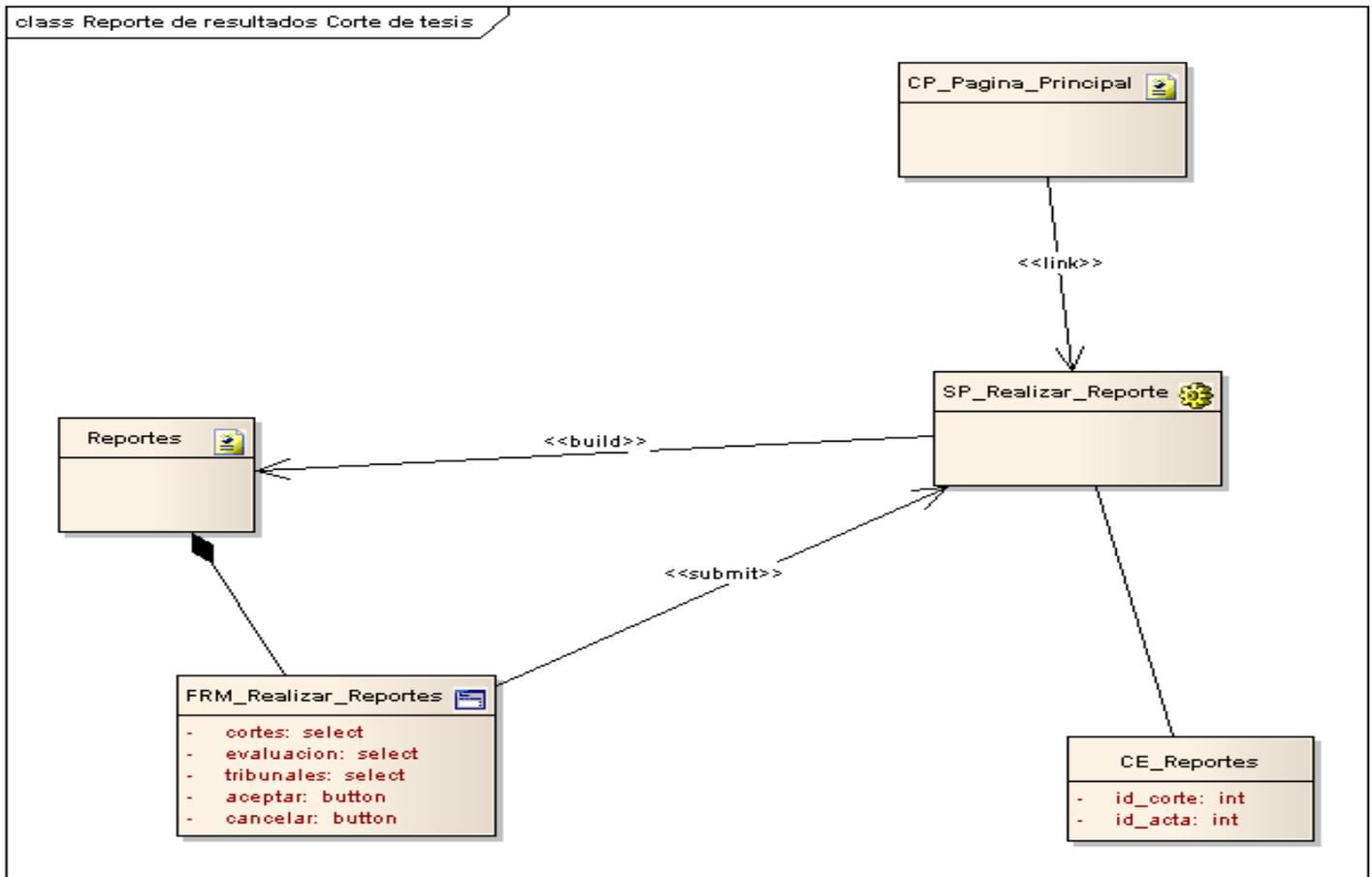


Figura 10: Diagrama de Clases del Diseño del CU Reporte de los resultados de los cortes de tesis.

3.6. Diagramas de interacción.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva a modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. [16] Estos diagramas se expresan de dos formas, una en diagramas de colaboración, que son los que muestran las relaciones entre los objetos y los mensajes que intercambian y, otra en diagramas de secuencia, que muestran las interacciones expresadas en función de secuencias temporales.

3.7. Diagrama de secuencia.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etcétera) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren. [17]

3.8. Diagrama de colaboración.

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia. [18]

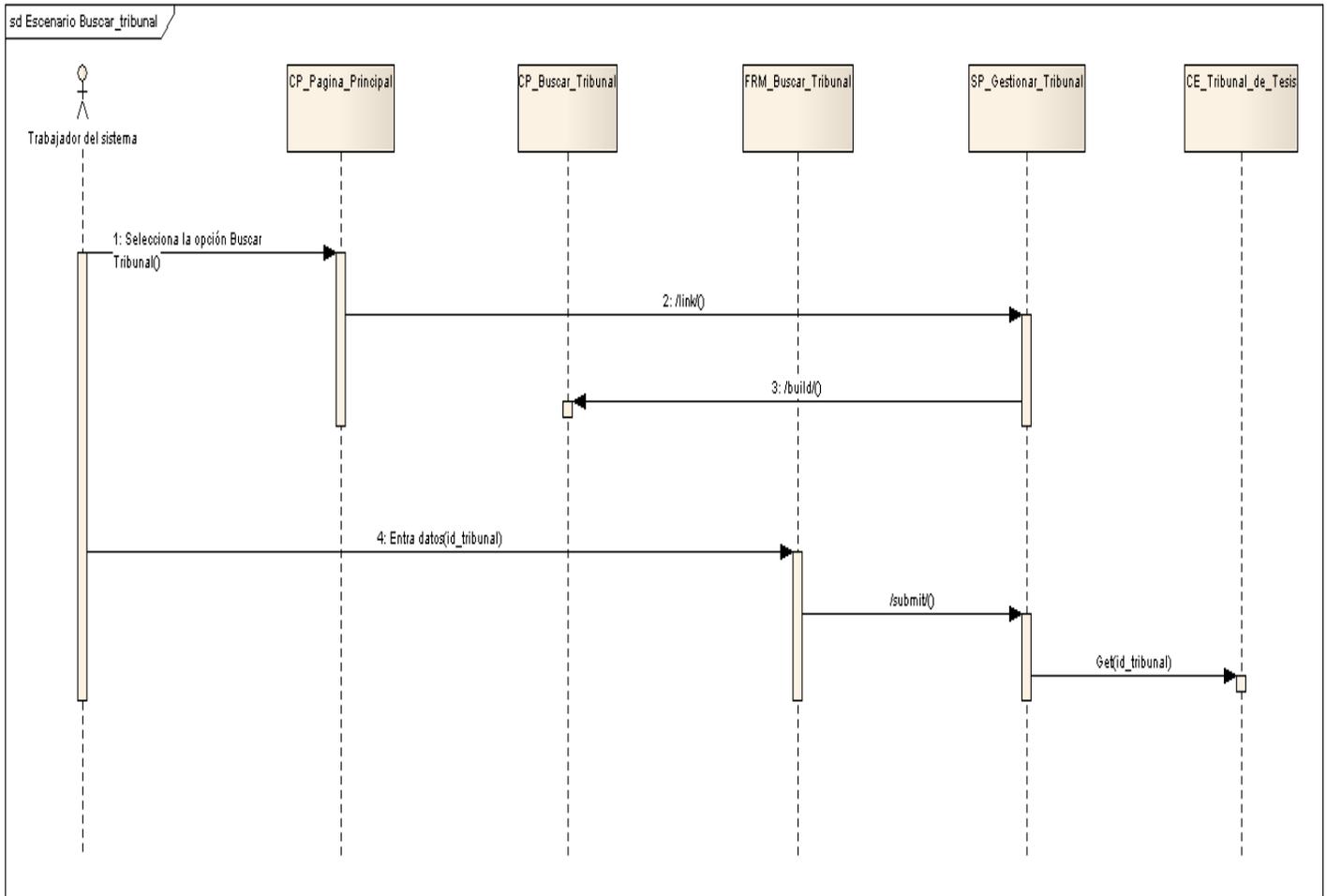


Figura 11: Diagrama de secuencia del CUS “Gestionar tribunal”. Sección “Buscar Tribunal”

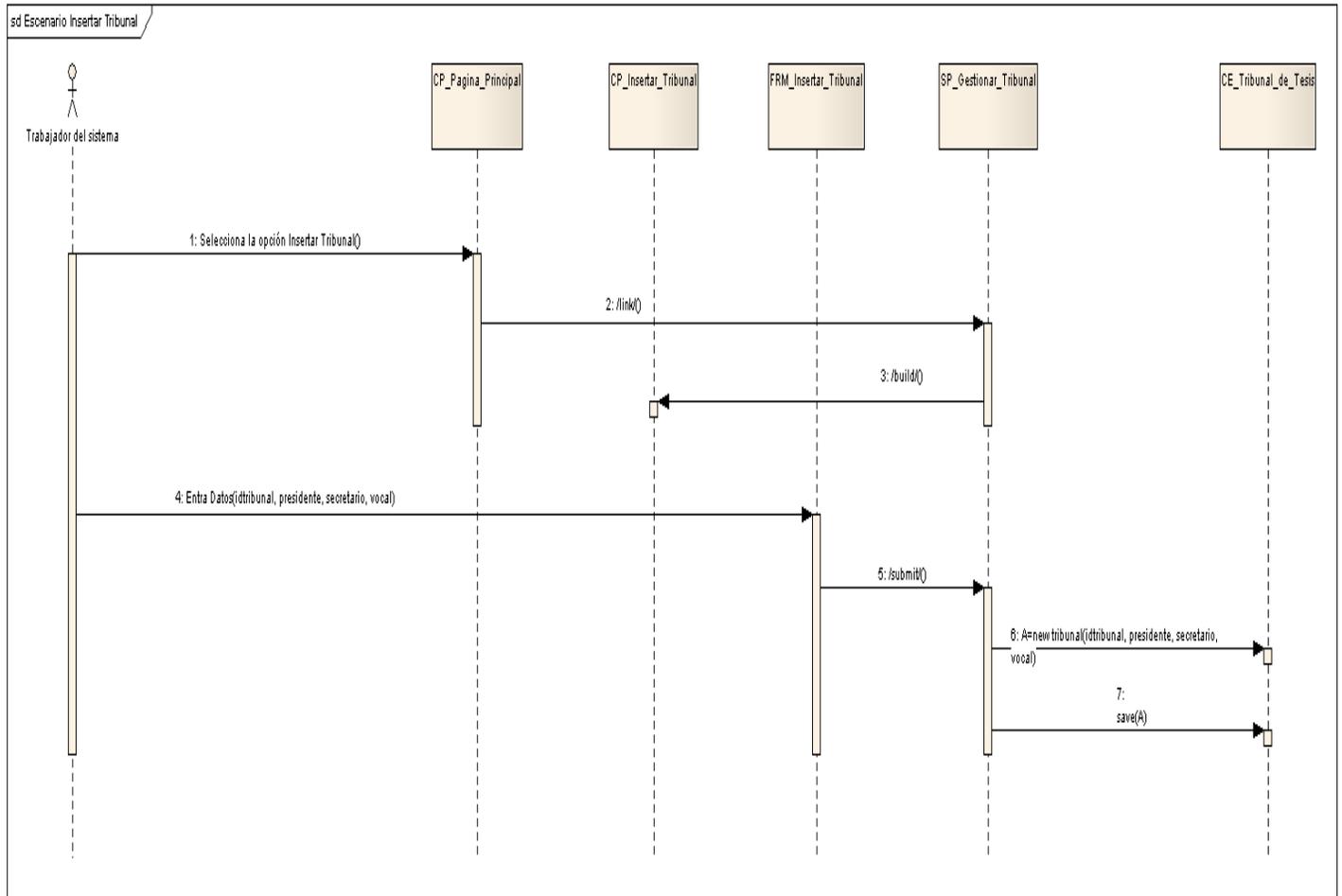


Figura 12: Diagrama de secuencia del CUS “Gestionar tribunal”. Sección “Insertar Tribunal”

CAPÍTULO 3. DISEÑO DEL SISTEMA

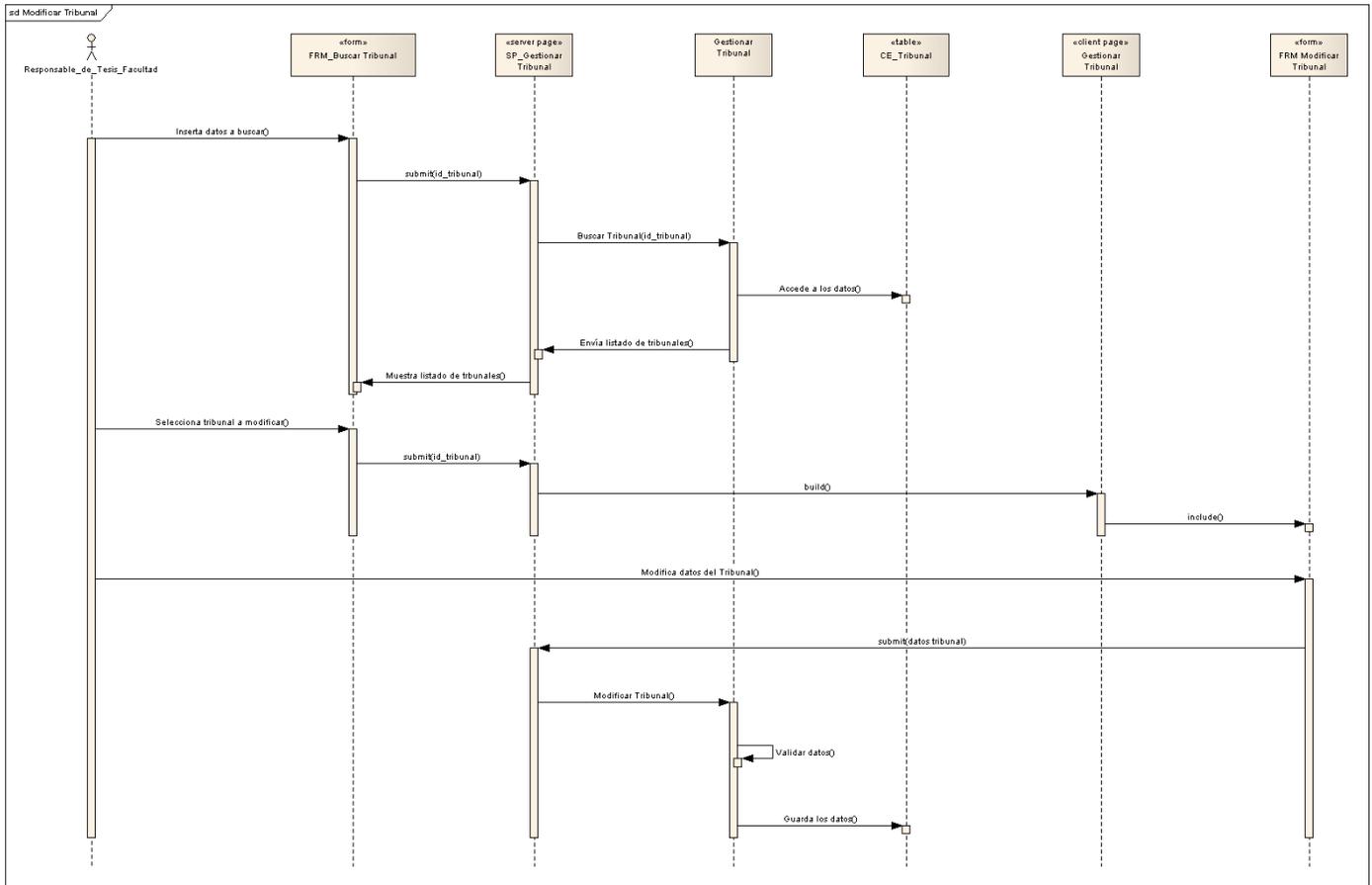


Figura 13: Diagrama de secuencia del CUS “Gestionar tribunal”. Sección “Modificar Tribunal”

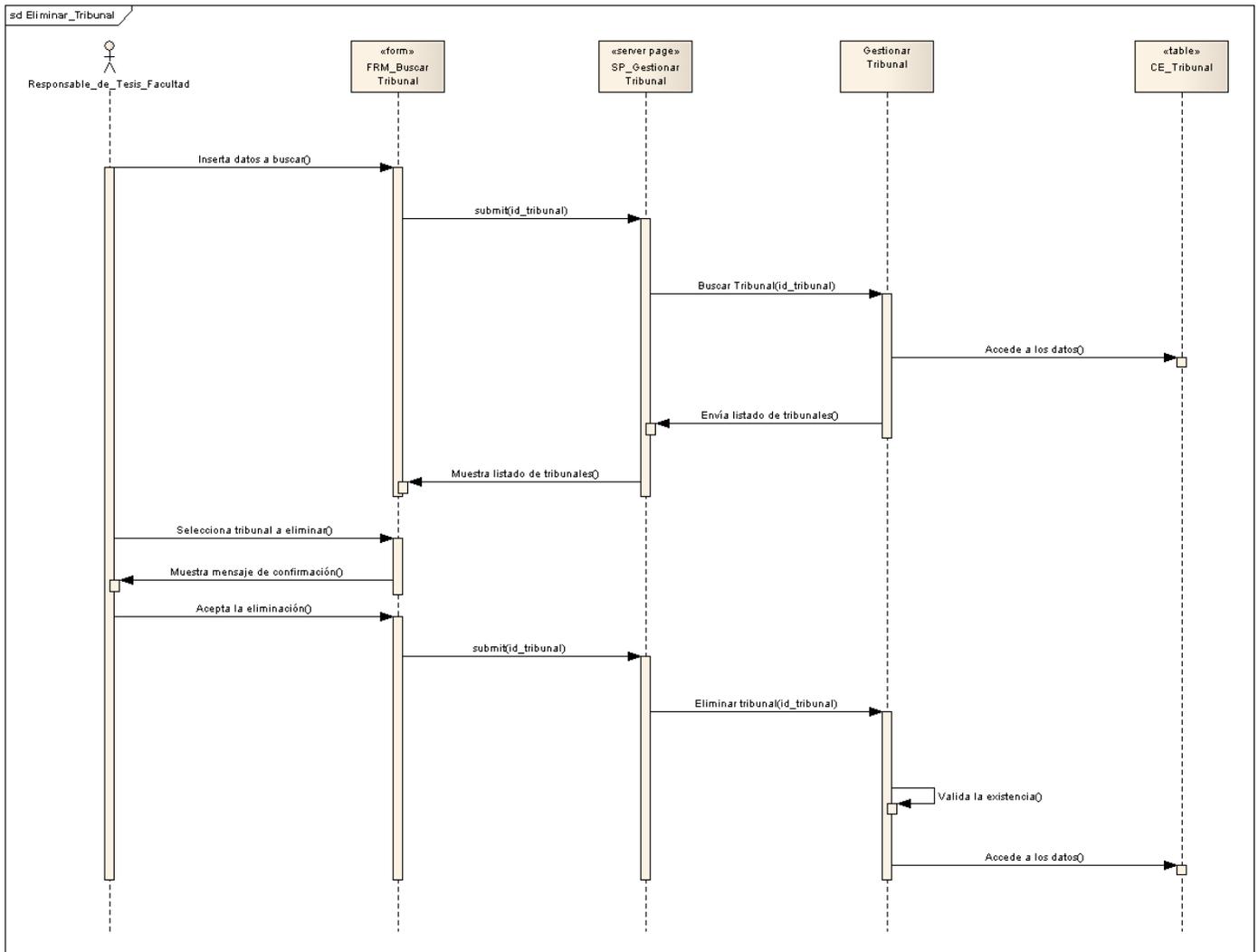


Figura 14: Diagrama de secuencia del CUS “Gestionar tribunal”. Sección “Eliminar Tribunal”

CAPÍTULO 3. DISEÑO DEL SISTEMA

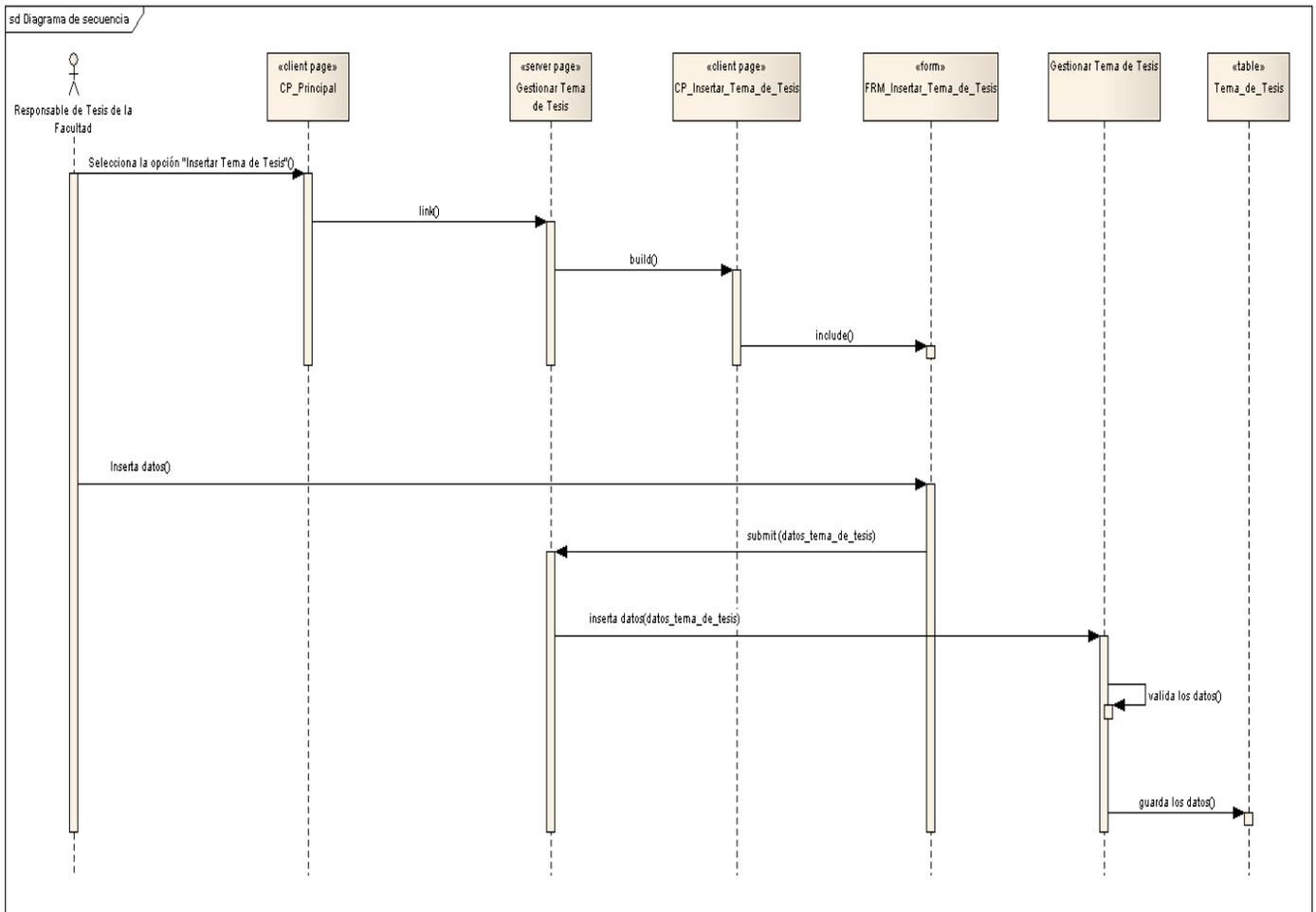


Figura 15: Diagrama de secuencia del CUS “Gestionar Tema de Tesis”. Sección “Insertar Tema de Tesis”

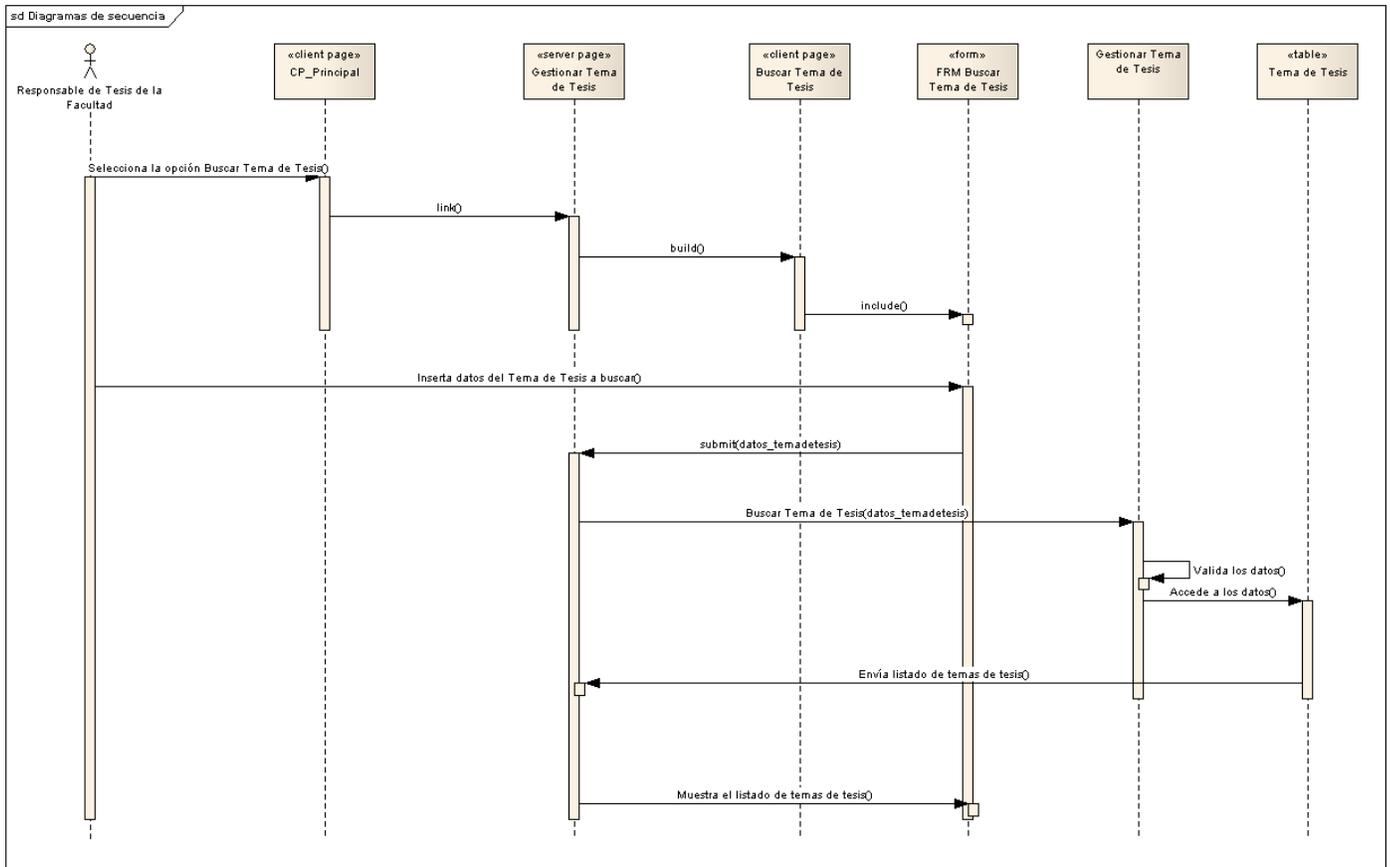


Figura 16: Diagrama de secuencia del CUS “Gestionar Tema de Tesis”. Sección “Buscar Tema de Tesis”

CAPÍTULO 3. DISEÑO DEL SISTEMA

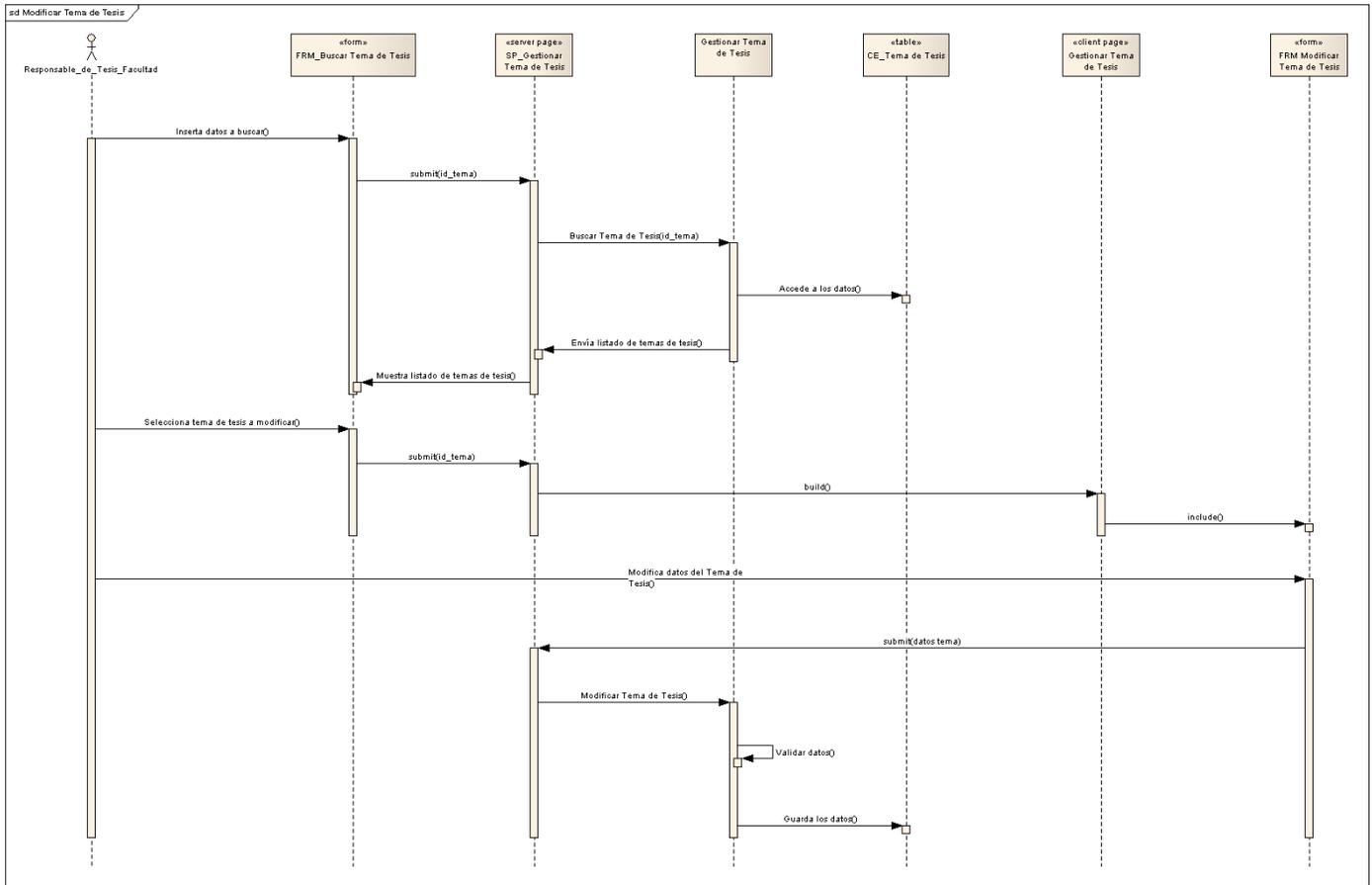


Figura 17: Diagrama de secuencia del CUS “Gestionar Tema de Tesis”. Sección “Modificar Tema de Tesis”

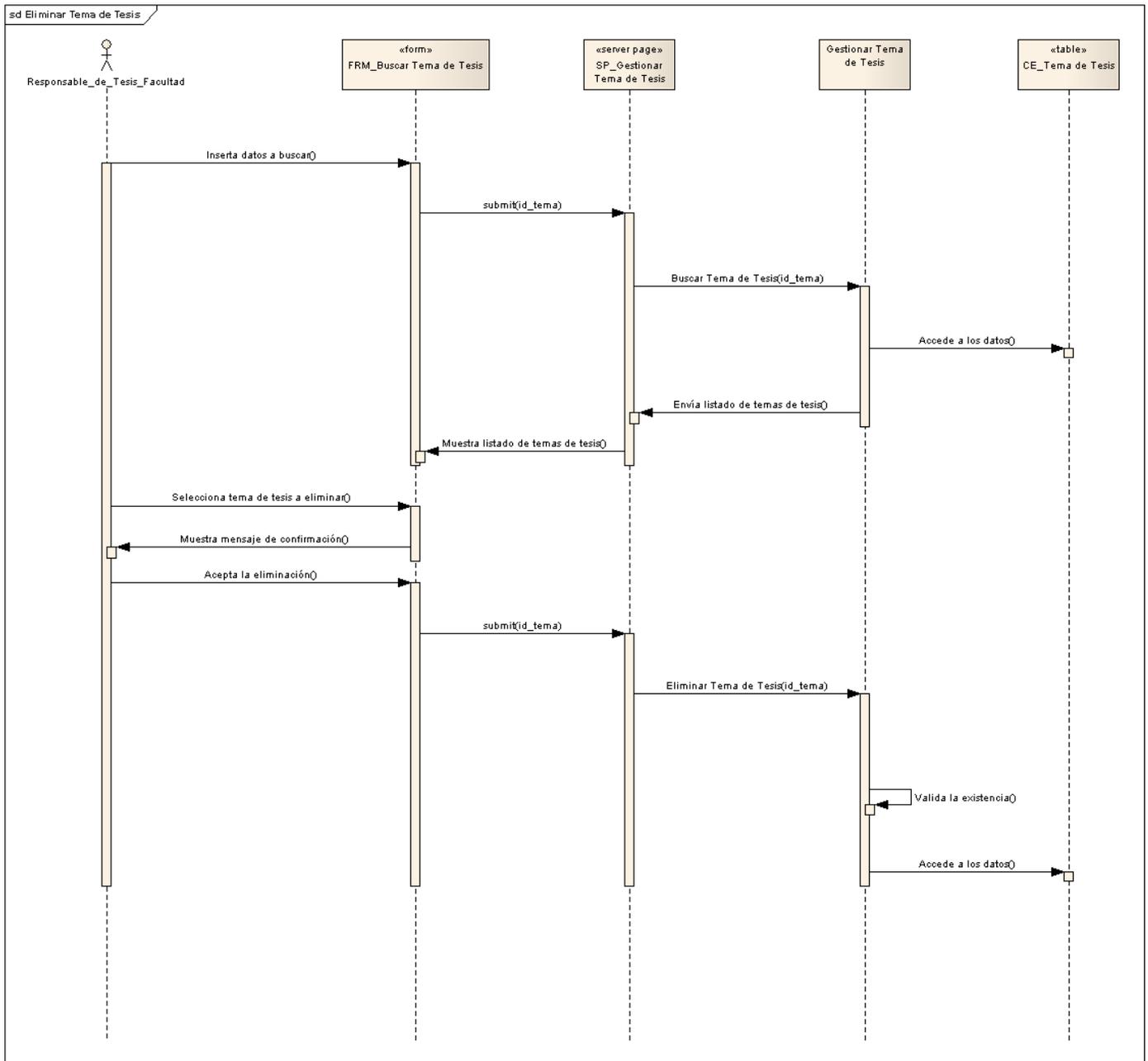


Figura 18: Diagrama de secuencia del CUS “Gestionar Tema de Tesis”. Sección “Eliminar Tema de Tesis”

3.9. Patrones del diseño.

Los principios del diseño que se requieren para construir buenos diagramas de interacción y de forma general el modelo de diseño, pueden codificarse, explicarse y utilizarse en forma metódica. Esta manera de entender y usar los principios del diseño se funda en los patrones con que se asignan las responsabilidades [19] Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades). Se pueden destacar 5 patrones principales que son:

Experto.

Creador.

Alta cohesión.

Bajo acoplamiento.

Controlador.

Para lograr una asignación eficiente de las responsabilidades, a las clases utilizadas, se le aplicó el patrón Experto, el cual consiste en asignar una responsabilidad al experto en información, o sea, a la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este es un patrón que se usa más que cualquier otro al asignar responsabilidades, ya que es un principio básico que suele utilizarse en el diseño orientado a objetos.

Además, el framework que se utiliza para desarrollar la presente aplicación, Django, fue creado desde un punto de vista técnico y arquitectónico, entre otros aspectos, con la idea de garantizar un poco acoplamiento, o sea, mientras menos componentes dependan de otro, más reusable y flexible se torna el sistema, ya que las responsabilidades están distribuidas entre las clases, de manera que su colocación no incrementa el acoplamiento, pero a su vez se mantienen enfocadas y relacionadas las responsabilidades de las mismas, lo que responde al patrón de Alta Cohesión.

CAPÍTULO 3. DISEÑO DEL SISTEMA

En este capítulo se definieron y describieron las clases y las relaciones entre ellas, a través de los diagramas de clases, tanto para el proceso de análisis como del diseño de la aplicación, lo cual permitió obtener una visión avanzada del sistema basada en la comprensión de los requisitos funcionales y no funcionales. Se modelaron además los diagramas de interacción correspondientes a los escenarios más significativos, de modo que quedó descrito más claramente el flujo de los mensajes entre los objetos del sistema y la dinámica de la aplicación.

Capítulo 4. Implementación

En este capítulo se describen los elementos necesarios para la implementación, partiendo del resultado obtenido del diseño. Se muestra la distribución del sistema en nodos mediante el diagrama de despliegue y la organización de los componentes y las relaciones lógicas entre ellos a través del diagrama de componentes, quedando así conformado el modelo de implementación.

4.1. Modelo de datos.

Un modelo de datos es aquel que describe la representación lógica y física de los datos persistentes de la base de datos. Básicamente consiste en una descripción de algo conocido como contenedor de datos (donde se guarda la información), así como los métodos a almacenar y recuperar información de dichos contenedores. Consiste en:

Objetos (entidades que existen y que se manipulan).

Atributos (características básicas de estos objetos).

Relaciones (forma en que se enlazan los distintos objetos entre sí).

A continuación se muestra el modelo de datos resultante de la base de datos y posteriormente la descripción de las tablas:

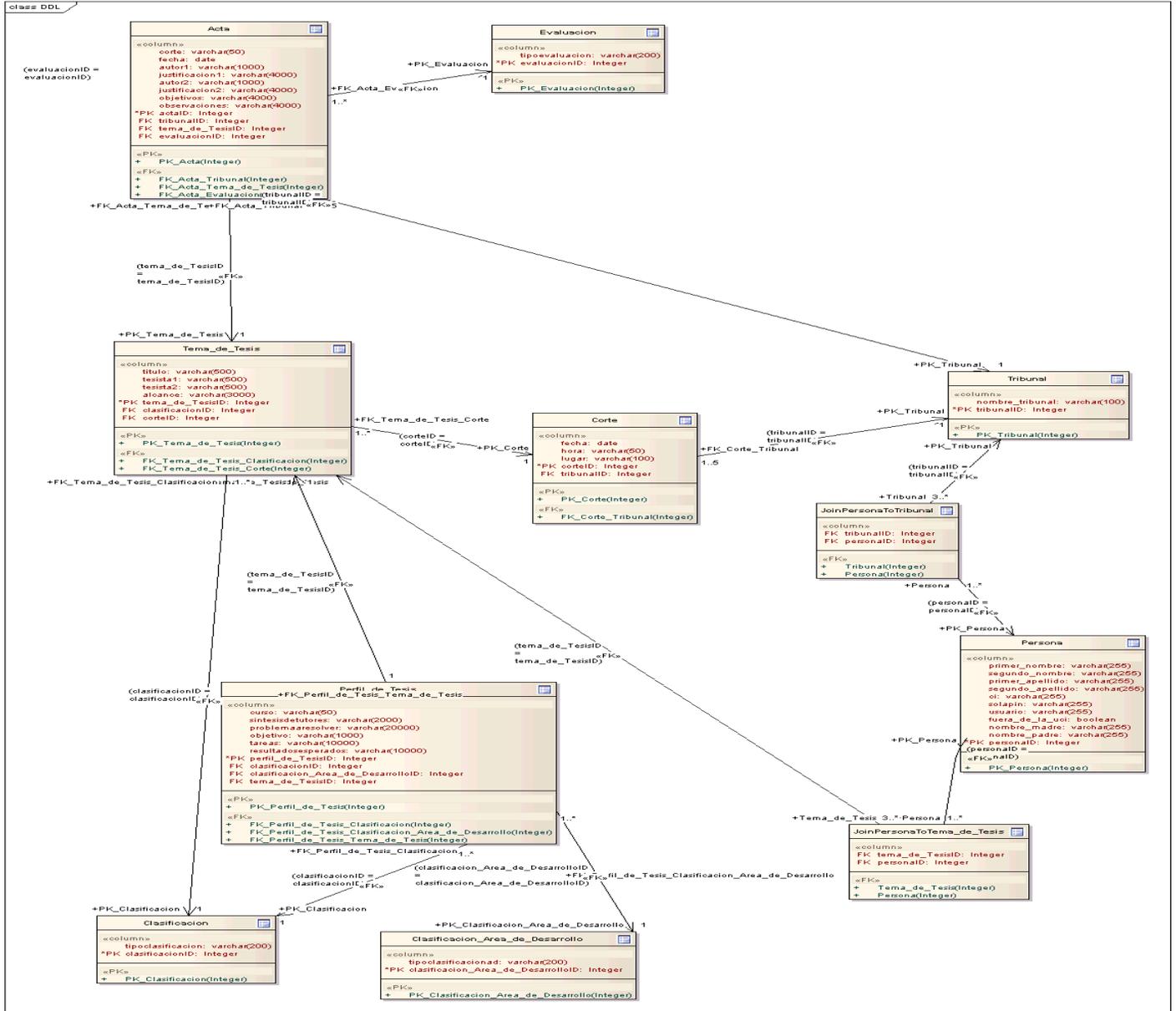


Figura 19: Modelo de datos.

CAPÍTULO 4. IMPLEMENTACIÓN

Descripción de las tablas del Modelo de Datos.

Nombre: Tribunal		
Descripción: Contiene los datos de los integrantes de los tribunales de tesis.		
Atributo	Tipo	Descripción
nombre_tribunal	varchar(100)	Representa el nombre del tribunal

Tabla 24. Descripción de la tabla “Tribunal” del Modelo de Datos.

Nombre: Persona		
Descripción: Contiene datos reales de trabajadores de la Universidad de las Ciencias Informáticas.		
Atributo	Tipo	Descripción
primer_nombre	varchar(255)	Representa el primer nombre de un trabajador.
segundo_nombre	varchar(255)	Representa el segundo nombre de un trabajador.
primer_apellido	varchar(255)	Contiene el primer apellido de un trabajador.
segundo_apellido	varchar(255)	Contiene el segundo apellido de un trabajador.
ci	varchar(255)	Contiene el número de carnet de identidad de un trabajador.
solapín	varchar(255)	Contiene el número de solapín de un trabajador.
usuario	varchar(255)	Contiene el usuario de un trabajador.
fueradelauci	boolean	Contiene información sobre un trabajador, si es interno o si no es interno.
nombre_padre	varchar(255)	Contiene el nombre del padre de un trabajador.
nombre_madre	varchar(255)	Contiene el nombre de la madre de un trabajador.

Tabla 25. Descripción de la tabla “Persona” del Modelo de Datos.

Nombre: Acta		
Descripción: Contiene los datos de las actas generadas en los cortes de tesis.		
Atributo	Tipo	Descripción
corte	varchar(50)	Representa el corte en que se redacta el acta.
fecha	date	Representa la fecha en que se redacta el acta.
autor1	varchar(1000)	Contiene el nombre del primer autor.
justificacion1	varchar(4000)	Contiene la justificación del primer autor en caso de que no se presente en el corte u obtenga una nota menor de 3 puntos.
autor2	varchar(1000)	Contiene el nombre del segundo autor.
justificacion2	varchar(4000)	Contiene la justificación del segundo autor en caso de que no se presente en el corte u obtenga una nota menor de 3 puntos.
objetivos	varchar(4000)	Contiene los objetivos del corte de tesis en que se redacta el acta.
observaciones	varchar(4000)	Contiene observaciones basadas en el avance del trabajo de diploma por parte de los autores.

Tabla 26. Descripción de la tabla “Acta” del Modelo de Datos.

CAPÍTULO 4. IMPLEMENTACIÓN

Nombre: Corte		
Descripción: Contiene los datos que tienen que ver con la planificación de los cortes de tesis.		
Atributo	Tipo	Descripción
fecha	date	Representa la fecha para el corte de tesis.
hora	varchar(50)	Representa la hora para el corte de tesis.
lugar	varchar(50)	Contiene el lugar donde se efectuará el corte de tesis.

Tabla 27. Descripción de la tabla "Corte" del Modelo de Datos.

Nombre: Tema de Tesis		
Descripción: Contiene los datos de los temas de tesis.		
Atributo	Tipo	Descripción
título	varchar(500)	Contiene los títulos de las tesis.
tesista1	varchar(500)	Contiene el nombre del primer autor.
tesista2	varchar(500)	Contiene el nombre del segundo autor.
alcance	varchar(500)	Contiene los datos del alcance de las tesis.

Tabla 28. Descripción de la tabla "Tema de Tesis" del Modelo de Datos.

Nombre: Evaluación		
Descripción: Nomenclador que contiene todas las posibles evaluaciones que se pueden establecer.		
Atributo	Tipo	Descripción
tipoevaluacion	varchar(200)	Representa la evaluación que se puede establecer.

Tabla 29. Descripción de la tabla "Evaluación" del Modelo de Datos.

Nombre: Clasificación		
Descripción: Nomenclador que contiene todas las posibles clasificaciones que pueden tener las tesis en dependencia de su alcance.		
Atributo	Tipo	Descripción
tipoclasificacion	varchar(200)	Representa la clasificación que se puede establecer.

Tabla 30. Descripción de la tabla "Clasificación" del Modelo de Datos.

Nombre: Clasificacion_Area_de_Development		
Descripción: Nomenclador que contiene todas las posibles clasificaciones que se pueden establecer de parte del Área de Desarrollo.		
Atributo	Tipo	Descripción
tipoclasificacionad	varchar(200)	Representa la clasificación que se puede establecer de parte del Área de Desarrollo.

Tabla 31. Descripción de la tabla "Tribunal" del Modelo de Datos.

4.2. Diagrama de despliegue.

El diagrama de despliegue es utilizado para modelar el hardware empleado en las implementaciones de sistemas y las relaciones entre sus componentes. [20] El modelo 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 cómputo. A continuación se explican los recursos presentes en cada nodo del diagrama:

- Nodo PC Cliente a través del cual los clientes se conectarán al sistema y podrán operar en el mismo. Esta PC deberá contar con un navegador, preferiblemente Internet Explorer 8.0 al menos o Mozilla Firefox 3.0 o superior que le permitirán al cliente tener acceso a la aplicación.
- Nodo Servidor Web: puede tener instalado como Sistema Operativo Windows XP o superior, GNU /Linux o Mac OS. Además debe tener instalado el Apache Server 2.2 con el módulo ModPython o WSGI instalados.
- Nodo Servidor Base de Datos, este servidor deberá presentar como Sistema Gestor de Base de Datos PostgreSQL 8.3.

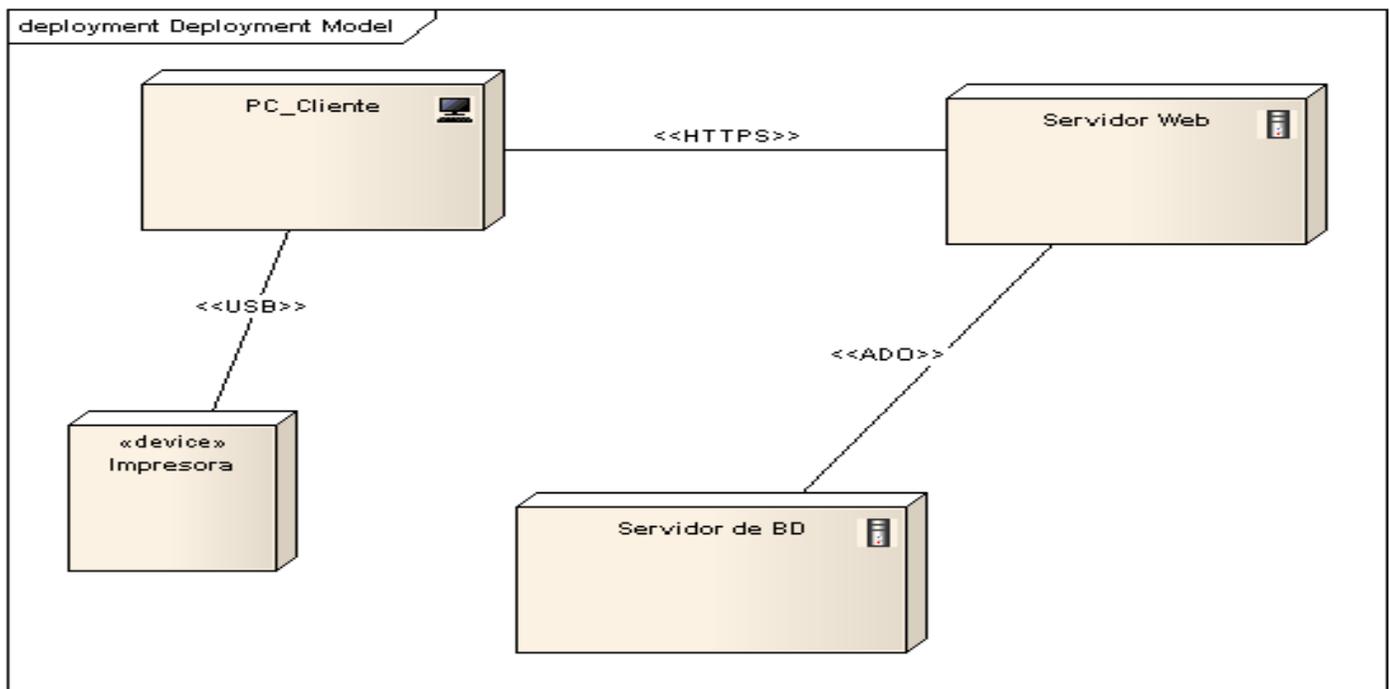


Figura 20: Diagrama de despliegue.

4.3 Diagramas de componentes.

Los diagramas de componentes siguientes muestran un conjunto de elementos del modelo de implementación tales como componentes y sus relaciones. Se utilizan para modelar la vista estática del

CAPÍTULO 4. IMPLEMENTACIÓN

sistema y muestran la organización y las dependencias lógicas entre los componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables.

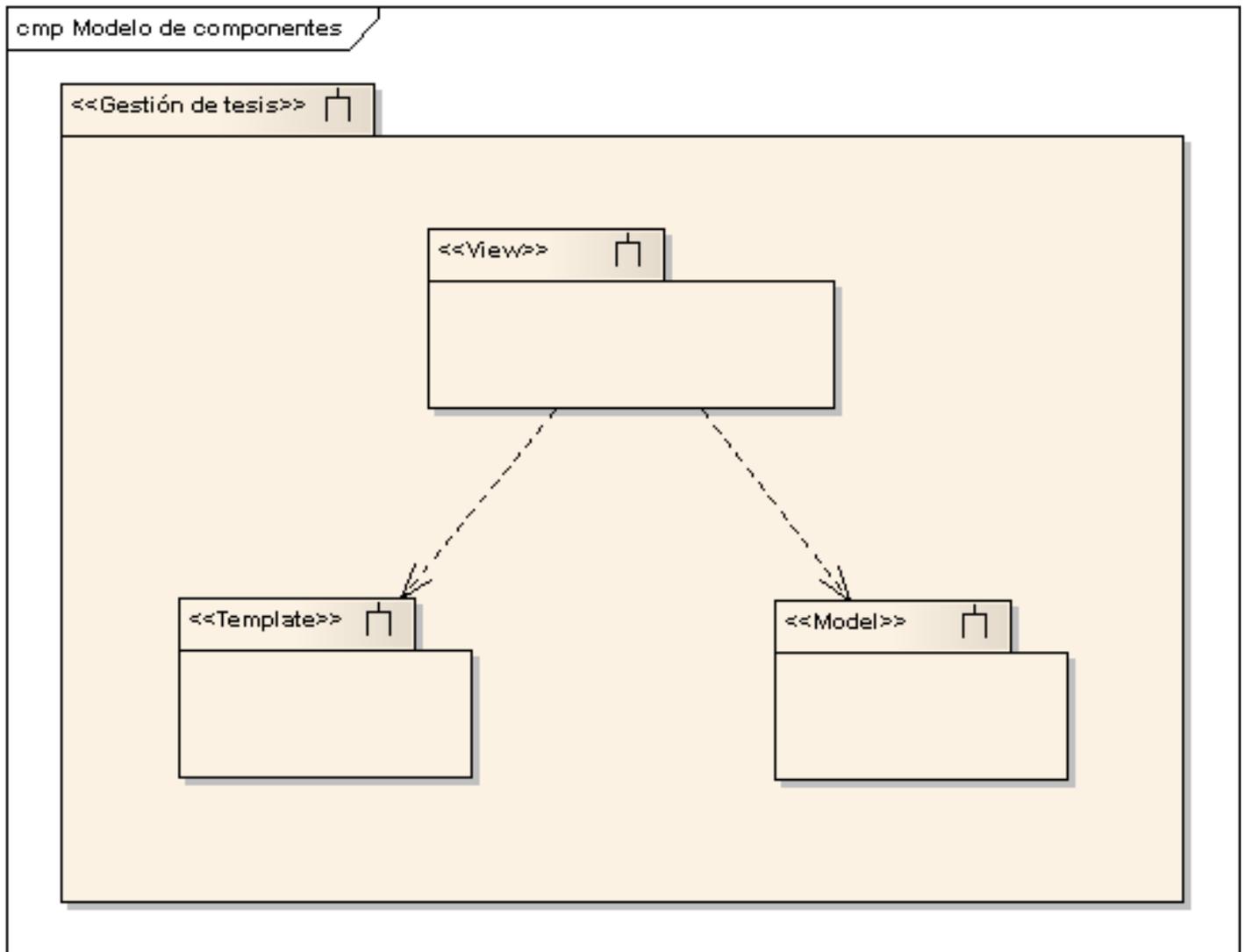


Figura 21: Diagrama de Componentes.

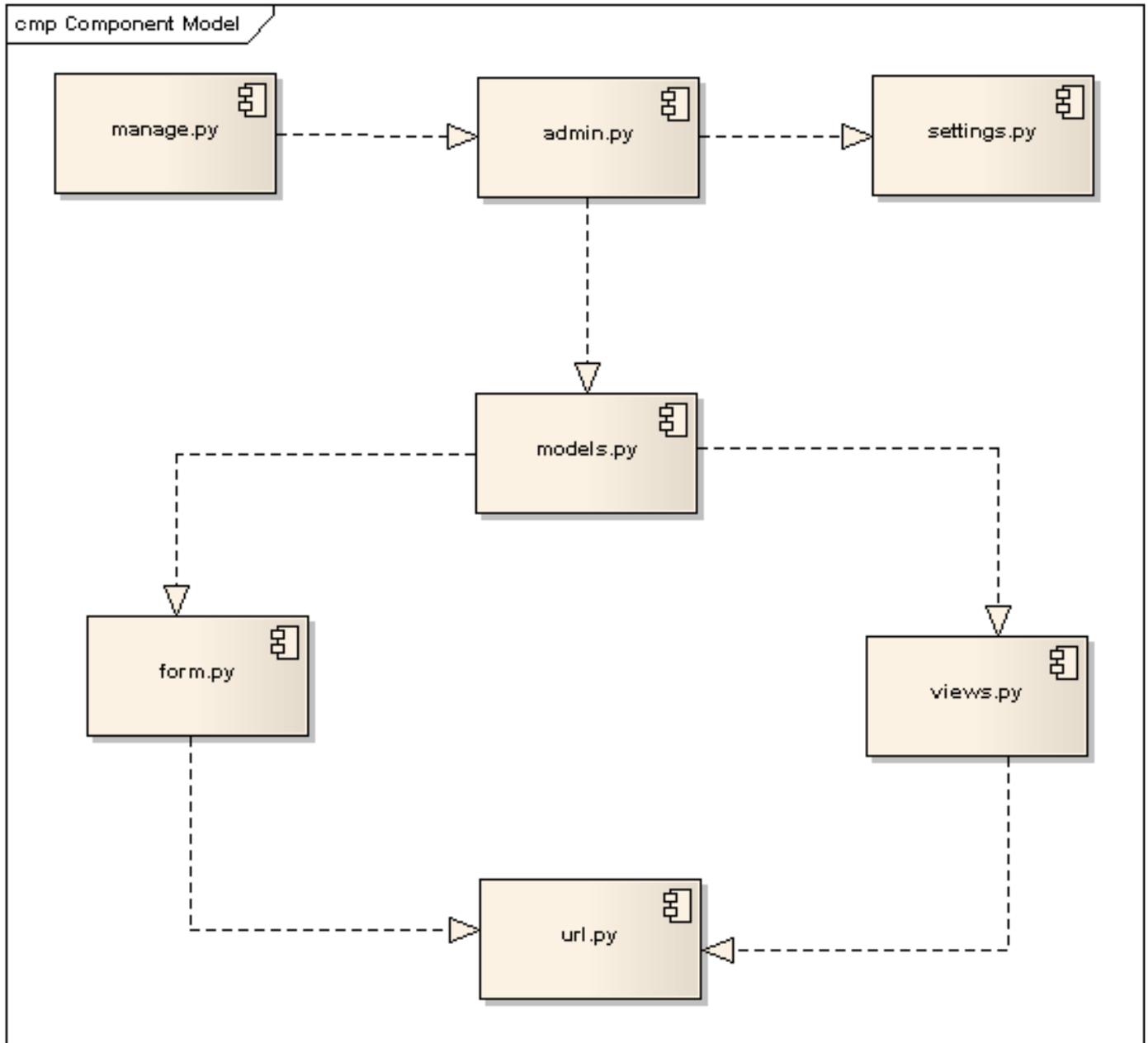


Figura 22: Subsistema View.

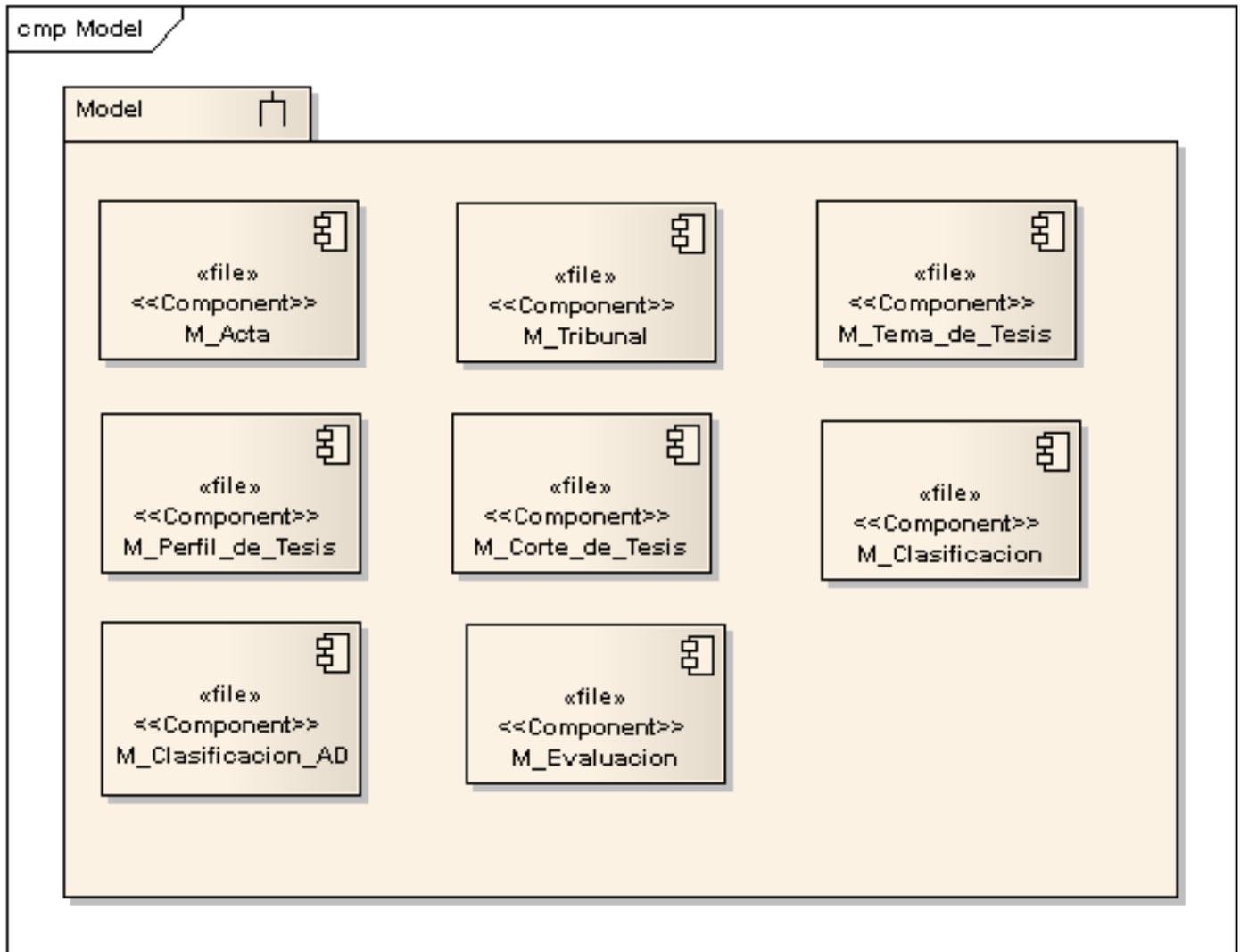


Figura 23: Subsistema Model.

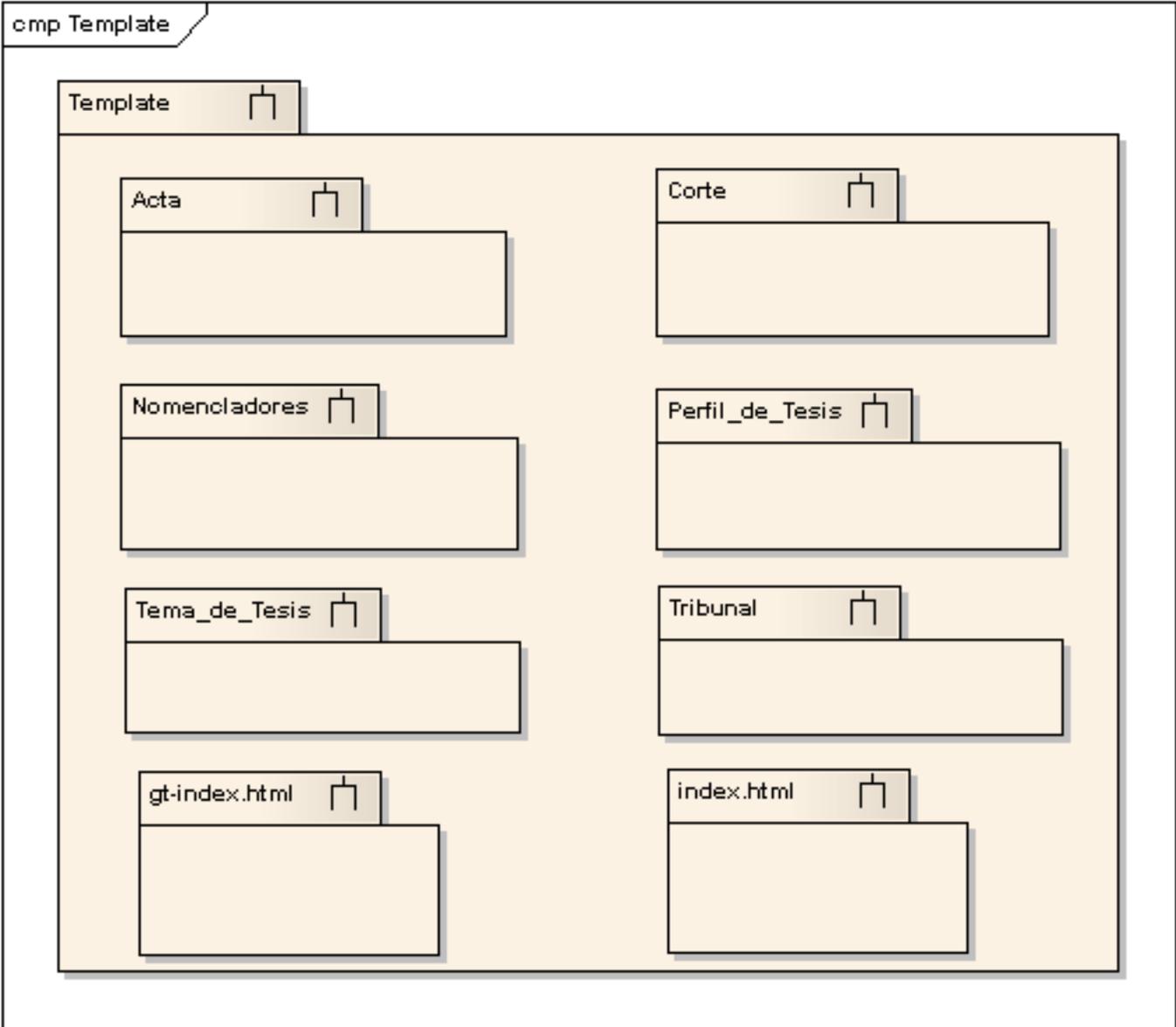


Figura 24: Subsistema Template.

4.4. Tratamiento de errores.

El tratamiento de los errores es un paso indispensable para el buen funcionamiento del sistema. Los errores más comunes que se pueden presentar están relacionados con el trabajo con la base de datos, fundamentalmente en los procesos de inserción, eliminación y/o modificación de datos. Es por esto que se han establecido mecanismos que visualicen la información para evitar en la medida de lo posible la introducción manual de los datos y así minimizar los errores que pueda tener la aplicación. Además se muestran mensajes indicando al usuario los errores cometidos, permitiéndole a la vez corregir su entrada con facilidad y continuar.

El manejo de excepciones es una estructura de control de los lenguajes de programación diseñada para manejar condiciones anormales que pueden ser tratadas por el mismo programa que se desarrolla. Por ejemplo, un programa puede admitir cierto número de errores en el formato de los datos y continuar su proceso para producir el mejor resultado posible en lugar de producir una salida aparatosa llena de mensajes de error probablemente incomprensibles para el usuario. Muchas veces, la acción asociada a una excepción es simplemente producir un mensaje informativo y terminar; otras veces, es sólo indicación de la necesidad de un cambio en la estrategia de resolución del problema.

Una de las mejores características del framework Django es el nivel de detalle que aporta en el tratamiento de errores, dándole al programador información muy útil y exacta sobre que tipo de error es el que se produce y donde.

La aplicación está capacitada para mostrarle al usuario información detallada sobre el tipo de dato que debe insertar en caso de que el que inserte no sea el correcto, también muestra mensajes en caso de que se dejen campos en blanco que no se pueden dejar. Además se le muestran mensajes de confirmación cuando se dispone a eliminar algún objeto.

CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo quedó conformado el modelo de implementación, ya que se realizó la distribución del sistema en nodos mediante el diagrama de despliegue y se mostraron las organizaciones y las dependencias lógicas entre componentes a través de los diagramas de componentes.

Conclusiones

La realización del presente trabajo ha posibilitado cumplir con los objetivos propuestos, por lo que se pueden plantear las siguientes conclusiones:

1. Se analizaron los procesos de negocio asociados a la gestión de la información relacionada con la gestión de tesis en la Facultad 7 de la Universidad de las Ciencias Informáticas (UCI).
2. Se realizó un análisis acerca de los sistemas existentes a nivel nacional e internacional referentes a la gestión de la información relacionada con la gestión de tesis, llegando a la conclusión de que no se ajustaban a las exigencias en este campo de la Facultad 7.
3. Se asimiló la arquitectura definida para el desarrollo del Sistema Integral de Gestión Administrativa.
4. Se implementó el sistema informático siguiendo las necesidades de funcionamiento establecidas en la Especificación de Requisitos de Software.

Recomendaciones

Las recomendaciones de la investigación están dirigidas a sugerir acciones para complementar el producto obtenido. Por lo que para el buen desempeño y puesta en marcha de la aplicación se hace la siguiente recomendación:

1. Extender el software a las demás facultades de la Universidad de las Ciencias Informáticas.
2. Desarrollar un Manual de Usuario que sirva de ayuda para los usuarios finales de la aplicación.

Referencias Bibliográficas

[1] Alarcos-inf-cr. Tema04. *Metodologías de Desarrollo* [Online] [Citado: 12 11 2009.]

<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>

[2] Gómez Gallego, Juan. Scribd.RUP. *Fundamento de la metodología RUP*

<http://www.scribd.com/doc/297224/RUP> [Online] [Citado: 12 11, 2009.]

[3] Lenguaje Unificado de Modelado, Primer Párrafo. [Citado: 13, 11, 2009.]

<http://www.scribd.com/doc/8963141/lenguaje-unificado-de-modelado>

[4] Nombre del sitio: BuenMaster.com. Modelo Vista Controlador. Primer Párrafo. [Citado: 13, 11, 2009.]

<http://buenmaster.com/?a=536>

[5] Nombre del sitio: DesarrolloWeb.com. Autor: Álvarez, Miguel Angel. Fecha de publicación: 19, 11, 2003. [Citado: 13, 11, 2009.]

<http://www.desarrolloweb.com/articulos/1325.php>

[6] Nombre del sitio: Sparx Systems. [Citado: 13, 11, 2009.]

<http://www.sparxsystems.com.ar>

[7] Nombre del sitio: Django en español. [Citado: 13, 11, 2009.]

<http://django.es/>

[8] Nombre del sitio: Softonic. [Citado: 14, 11, 2009.]

<http://wing-ide.softonic.com/linux>

[9] Autor: González Pérez, Reisel. Título: Introducción al Sistema de Gestión de Base de Datos PostgreSQL. Primer Párrafo. [Citado: 14, 11, 2009.]

<http://www.monografias.com/trabajos-pdf2/sistema-gestion-base-datos-postgresql/sistema-gestion-base-datos-postgresql.pdf>

REFERENCIAS BIBLIOGRÁFICAS

- [10] 26. Unidad IV: Introducción a la programación. [Citado: 14, 11, 2009.]
<http://ingunderworld.bloginom.com/>
- [11] Nombre del sitio: EcuRed. Capítulo Actores del Negocio. [Citado: 15, 11, 2009.]
http://www.ecured.cu/index.php/Flujo_de_Trabajo_Modelo_del_Negocio#Actores_del_Negocio
- [12] Nombre del documento: Anexo. Capítulo E: Resumen del Tema. [Citado: 16, 11, 2009.]
<http://www.eumed.net/libros/2009c/587/Anexo.pdf>
- [13] Ingeniería de Software 1: Introducción a la Ingeniería de Software."Conferencia # 1": UCI, 2010
- [14] Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Primera edición 1999. Pág. 187.
- [15] Nombre del sitio: SlideShare. Capítulo 3[Citado: 17, 11, 2009.]
<http://www.slideshare.net/masa832/clase-de-fds22>
- [16] Ingeniería de Software II."Continuación del FT Análisis de Diseño." Conferencia # 1.s.l.: UCI, 2009 - 2010.
- [17] Nombre del sitio: Clickear.com. Capítulo II.5.1 Diagrama de Secuencia. [Citado: 13, 1, 2010.]
<http://www.clickear.com/manuales/uml/diagramasinteraccion.aspx>
- [18] Nombre del sitio: Clickear.com. Capítulo II.5.2 Diagrama de Colaboración. [Citado: 13, 1, 2010.]
<http://www.clickear.com/manuales/uml/diagramasinteraccion.aspx>
- [19] Ingeniería de Software 1: Introducción a la Ingeniería de Software."Conferencia # 1": UCI, 2010
- [20] Autor: Pinelo, David. Nombre del documento: Introducción a UML. Capítulo Diagrama de despliegue. marzo-abril de 2009.
<http://www.pinelo.com/blog/postimgs/2009/03/26/UML.pdf>

Bibliografía

Metodologías de Desarrollo de Software.

<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>

Fundamentos de la metodología RUP.

<http://www.scribd.com/doc/297224/RUP>

Artículo Qué es Python.

<http://www.desarrolloweb.com/articulos/1325.php>

Enterprise Architect : Herramienta de Modelado UML.

<http://www.sparxsystems.com.ar/>

Definición de modelo de negocio.

<http://definicion.de/modelo-de-negocio/>

Sistema de Gestión Académica TESEO.

https://www.gestion.uco.es/gestion/aplicaciones/docs/sga_TESEO.pdf

Usabilidad y arquitectura del software.

<http://www.desarrolloweb.com/articulos/1622.php>

M-V-C Modelo Vista Controlador.

<http://www.programacionweb.net/articulos/articulo/?num=505>

Capítulo 1 Introducción a JavaScript.

<http://www.elcodigo.net/tutoriales/javascript/javascript1.html>

El lenguaje HTML.

<http://www.juntadeandalucia.es/averroes/iesgaviota/informatica/html.html>

Guía breve de CSS.

<http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>

Model reference.

<http://www.djangoproject.com/documentation/0.96/model-api/>

Django.

<http://code.djangoproject.com/wiki/DjangoOnWindowsWithIISAndSQLServer>

Using SQL reserved names.

http://www.djangoproject.com/documentation/models/reserved_names/

Writing your first Django app.

<http://docs.djangoproject.com/en/dev/intro/tutorial01/>

Working with Excel files in Python.

<http://www.python-excel.org/>

Conectividad Python/PostgreSQL.

http://www.postgresql.cl/desarrollo/python/conectividad_python_postgresql_por_Rodrigo_Ancavil_del_Pino

Inserting Data into a PostgreSQL Database.

http://python.about.com/od/pythonanddatabases/ss/dbpostgresql_5.htm

Model instance reference.

<http://docs.djangoproject.com/en/dev/ref/models/instances/>

Framework.

<http://www.tecnolinkinformatica.com/definicion/Framework>

Sistema de Gestión Académica TESEO.

http://www.gestion.uco.es/gestion/aplicaciones/docs/sga_TESEO.pdf

Model examples.

<http://www.djangoproject.com/documentation/models/>

Generic views.

<http://docs.djangoproject.com/en/dev/ref/generic-views/>

Built-in templates, tags and filters.

<http://docs.djangoproject.com/en/dev/ref/templates/builtins/>

The Django Template language: For Python programmers

<http://docs.djangoproject.com/en/dev/ref/templates/api/>

The Django Template language: For template authors.

<http://www.djangoproject.com/documentation/0.96/templates/>

Working with forms.

<http://docs.djangoproject.com/en/dev/topics/forms/>

Form fields.

<http://docs.djangoproject.com/en/dev/ref/forms/fields/>

Entradas sobre "forms".

<http://django.es/blog/tag/forms/>

Python documentation.

<http://www.python.org/doc/>

Inmersión en Python.

<http://es.diveintopython.org/>

Glosario de Términos

ActiveX Data Objects (ADO) es uno de los mecanismos que usan los programas de computadoras para comunicarse con las bases de datos, darles órdenes y obtener resultados de ellas.

Aplicación o Sistema Informático: Programas con los cuales el usuario final interactúa a través de una interfaz y que realizan tareas útiles para éste.

Cliente Servidor: Modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

Componente: Parte física y reemplazable de un sistema que se ajusta a, y proporciona la realización de, un conjunto de interfaces.

Conductor Universal en Serie (CUS), abreviado comúnmente **USB**, es un puerto que sirve para conectar periféricos a un ordenador.

Framework: Es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

HTTPS (Hypertext Transfer Protocol Secure - Protocolo seguro de transferencia de hipertexto): Es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

Informática: Disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales.

Informatizar: Proceso de aplicar sistemas o equipos informáticos al tratamiento de la información.

Javascript: Es un lenguaje de scripting basado en objetos no tipado y liviano, utilizado para acceder a objetos en aplicaciones.

LDAP (Lightweight Directory Access Protocol - Protocolo Ligero de Acceso a Directorios): Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos optimizada para las operaciones de lectura y búsqueda, a la que pueden realizarse consultas.

Paquete: Mecanismo de propósito general para organizar elementos en grupos.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre.

Servicio: Unidad de software que encapsula alguna funcionalidad de negocio y proporciona estas a otros servicios a través de interfaces públicas bien definidas.

Software: Conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Subsistema: Agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por los elementos contenidos.

Anexos

Anexo 1. Plantilla del Acta de Evaluación del Corte de Tesis

Acta de Evaluación del Corte de Tesis

Corte: _____

Fecha: _____

Título: _____

Participantes:

Cargo	Nombre y Apellidos	Firma	Justificación de los ausentes
Autor			
Autor			
Tutor			
Tutor			
Oponente			
Presidente			
Secretario			
Vocal			
Asesor			

Objetivos del Corte de Tesis:

Observaciones:

Evaluación: _____

Anexo 2. Plantilla generalizada de un Perfil de Tesis.

Perfil del Trabajo de Diploma

Curso docente:

Título:

Autores:

Tutor:

Clasificación:

Clasificación del área de desarrollo:

Síntesis del Tutor:

Problema a resolver:

Objetivo:

Tareas a cumplir por los estudiantes:

Resultados esperados:

Nombre del estudiante
Autor

Nombre del estudiante
Autor

Nombre del tutor
Tutor

Nombre del tutor
Tutor