

**Universidad de las Ciencias Informáticas
Facultad #4**



**Título: “Propuesta de diseño de una aplicación
Web para el Control de Trabajos de Diploma”**

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

Autor: Yurien Ramos García

Tutor: Ing. Loan Joa Matos

Co-tutor: Ing. Líber Luis González Vilela

Junio 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yurien Ramos García

Firma del Autor

Loan Joa Matos

Firma del Tutor

Líber Luis González Vilela

Firma del Co-Tutor

DATOS DE CONTACTO

Tutor: Ing. Loan Joa Matos.

- ✓ Ingeniero en informática en la Universidad José Antonio Echeverría (CUJAE).
- ✓ Profesor de la Universidad de las Ciencias Informáticas, en la Disciplina de Práctica Profesional.
- ✓ Cuenta con 4 años de experiencia en la Educación Superior.
- ✓ Jefe de Proyecto Productivo en la Facultad #4.

Correo electrónico: ljoa@uci.cu

Co-Tutor: Ing. Líber Luis González Vilela.

- ✓ Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2007.
- ✓ Profesor de la Universidad de las Ciencias Informáticas, en la Disciplina de Programación 3 desde el año 2007.
- ✓ Cuenta con 1 año de trabajo en la Educación Superior.
- ✓ Trabaja en el MINFAR y en la Universidad de las Ciencias Informáticas.

Correo electrónico: liber@uci.cu

AGRADECIMIENTOS

A la **Revolución** por permitirme la educación y formación profesional, por hacerme una persona digna de estos tiempos.

A la **UCI** por ser mi hogar, por darme la posibilidad de crecerme como persona, por educarme, por darme tantos amigos.

A mi tutor **Loan** por asesorarme y ayuda incondicional.

A mi co-tutor **Liber** y su compañera **Yaima** por asesorarme y ayuda incondicional.

A los **profes** por contribuir a mi formación profesional y revolucionaria.

A **todos mis compañeros de aula** que me hicieron sonreír y aportaron granos de felicidad a mi estancia.

A **todos** los que me ayudaron de una u otra forma, gracias.

A mi **Madre y mi Padre** por su apoyo y sacrificio, estar siempre a mi lado en todos los momentos de mi vida, por hacer de mí lo que soy hoy, por ese amor desmedido que me han dado.

A mi hermano **Yadier** por estar a mi lado todo este tiempo, se que estas orgulloso de mi.

A mi esposa **Yusleimi (La China)** por amarme, ser mí amiga, guía y compañía en estos cinco años y para toda la vida. Te amo.

A mis **tíos(a)** por demostrarme que la vida da muchas oportunidades, por quererme tanto y ayudarme en todas las esferas de mi vida.

A mis **primos(a)** por ser mis amigos y hermanos. Por todos sus consejos y ayuda en todo los años de mi vida. Los quiero mucho.

A mis **suegros** por su amor y ayuda en estos cinco años.

A mi abuelita **Evangelina** por amarme tanto, porque sé que eres mi ángel de la guarda, que me guías y que estas orgullosa de mi y de mi padre. Te amo.

A mi abuela **Clara** por amarme tanto. Te amo abuelita.

A mis abuelos **Francisco y Restituto** por amarme y enseñarme, por prepararme para la vida.

A mis otros papás **Orestes y Zenaida** por amarme tanto y soportar todas las malcriadeces de mi infancia. Sé que están muy orgullosos de mí. Los amo mucho.

A **Lourdes y Ernesto** por educarme y guiarme en la vida.

A mi primo **Mandy** por quererme tanto y apoyarme. Que Dios te tenga en la gloria.

A **Felicó, Alieski y Yassier**, por su ayuda y consejos.

A **Carmita, Ariel, Yudermis, Pepito, Omero, Silvia, Dayami, Osley, Caridad, Pepe, Ivette, Dainelys** por confiar en mí, por su apoyo y ayuda. Los quiero mucho.

DEDICATORIA

A mis padres, mi hermano, mi esposa y a mi abuela Evangelina que aunque ya no está sé que hubiese disfrutado mucho de este triunfo en mi vida.

A toda aquella persona que en un futuro va a utilizar esta tesis para su desarrollo profesional.

RESUMEN

El presente trabajo está enmarcado en el control de los trabajos de diploma en la facultad cuatro, de la Universidad de las Ciencias Informáticas. En la actualidad el desarrollo tecnológico es constante, y por estas variaciones dinámicas en el ambiente en el que se desarrollan las organizaciones modernas, se plantea la necesidad de adaptarse a transformaciones de manera inmediata. La innovación, la rapidez, el servicio de gran calidad y el ritmo al que se mejora y emplea el conocimiento constituyen la nueva política a aplicar. La dedicación y la aportación de fuerza laboral son decisivas y la información es el arma esencial en esta lucha, la cual es recopilada mediante un sistema informativo que abarca su recepción, elaboración, registro y transmisión. Hoy en día el procesamiento y gestión de la información que se lleva a cabo en la Universidad de las Ciencias Informáticas es realizado de forma manual. Esto posibilita la deficiencia de información, el intercambio de información es lento y además poco confiable pues puede perderse información en el manejo de los trabajos realizados y en cuestión. Este trabajo tiene como objetivo principal realizar una investigación profunda a los procesos de control de tesis en la facultad cuatro, cuyos resultados permitan desarrollar un software que garantice que la actualización, procesamiento y obtención de la información relacionada con el proceso de control de tesis en la facultad sea eficiente en cuanto a rapidez, confiabilidad y seguridad.

PALABRAS CLAVES

Control de tesis, Perfil de tesis, Trabajos de diploma, RUP, PHP, Web.

TABLA DE CONTENIDO	
AGRADECIMIENTOS	IV
DEDICATORIA	V
RESUMEN	VI
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	3
1.1 Introducción.....	3
1.2 Sistemas vinculados al campo de acción.....	3
1.3 La Web.	3
1.4 Diferencias en las Aplicaciones Web y Sitios Web.....	4
1.5 Tecnologías actuales a utilizar.....	5
1.6 Metodologías de Trabajo.....	10
1.7 Lenguaje Unificado de Modelado (UML).	14
1.8 Sistema de Gestión de Bases de Datos (SGBD).	15
1.9 Herramientas CASE.....	17
1.10 Selección de las tecnologías a utilizar en la propuesta de solución.....	20
1.11 Conclusiones.	21
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	22
2.1 Introducción.....	22
2.2 Objeto de estudio.....	22
2.3 Modelo de negocio.	24
2.4 Especificación de los requisitos de software.	41
2.5 Definición de los casos de uso.....	45
2.6 Conclusiones.....	63
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	64
3.1 Introducción.....	64
3.2 Descripción de la arquitectura.	64
3.3 Análisis.	65
3.4 Diseño.	72
3.5 Diseño de la BD.....	81
3.6 Conclusiones	88
CONCLUSIONES	89
RECOMENDACIONES	90
REFERENCIAS BIBLIOGRÁFICAS	91
ANEXOS	94
ANEXO I. Los niveles en capas por lo que se identifica una aplicación Web.	94
ANEXO 2. Arquitectura Cliente/Servidor.	95
ACRÓNIMOS	99

INTRODUCCIÓN

En la Universidad de las Ciencias Informáticas se está llevando a cabo la informatización de muchos sectores, lo que posibilita la agilidad y rapidez de trabajo por parte del personal. Las facultades y la dirección de los proyectos productivos son los encargados de asignar temas de trabajo de diploma según el perfil e interés de la institución. Esta tarea se hace actualmente de forma manual lo que imposibilita un control estricto y facilidad en el manejo de los datos. De los procesos llevados a cabo por este organismo depende en gran manera la organización de los trabajos de diploma y el desarrollo satisfactorio de los mismos.

En la Universidad de las Ciencias Informáticas se brinda poca información a los usuarios acerca de los trabajos de diploma realizados y los que se encuentran en ejecución. Las facultades no cuentan con una herramienta para el control y seguimiento de los trabajos de diploma que en ellas se desarrollan por lo que se dificulta poder brindar una información detallada de los temas desarrollados. Esto provoca que no se tenga conocimiento del alcance que tuvieron los mismos por lo que es muy difícil darle continuidad. Por otra parte existe la posibilidad de que se estén replicando innecesariamente trabajos de diplomas.

Existe entonces el siguiente **problema**: ¿Cómo lograr en la Universidad de las Ciencias Informáticas un control sistemático de los trabajos de diploma que posibilite controlar la información requerida?

Este problema se enmarca en el **objeto de estudio**: Trabajos de diploma de la facultad cuatro.

El **campo de acción** abarcado es: Control y seguimiento de los trabajos de diploma en la facultad.

Por tanto el **objetivo general** del trabajo es el siguiente:

- Diseñar una propuesta de aplicación Web que permita controlar los trabajos de diploma que se desarrollan en la Universidad de las Ciencias Informáticas.

Objetivos Específicos:

- Identificar los procesos para el control de tesis que tienen lugar en la facultad cuatro, en la Universidad de las Ciencias Informáticas.
- Modelar los procesos de control de tesis que tienen lugar en la facultad cuatro, en la Universidad de las Ciencias Informáticas.
- Diseñar una aplicación informática para el control de los trabajos de diploma en la facultad.

Para guiar la investigación se plantea la siguiente **hipótesis**:

- Si se desarrolla un sistema que controle los trabajos de diploma en la Universidad de Ciencias Informáticas, se lograría reducir la réplica de los mismos y brindar una información detallada.

Para llevar a cabo este objetivo es necesario realizar las siguientes **tareas**:

- Analizar el diseño de propuestas de otros desarrolladores en otras instituciones para la solución de este problema.
- Definir las tecnologías informáticas a utilizar para desarrollar el sistema.
- Realizar el análisis y diseño del sistema informático utilizando el proceso de software seleccionado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

El presente capítulo aborda conceptos relacionados con el desarrollo de las aplicaciones Web tales como: qué es una Web y por qué una Web. También se realizará un análisis de las tendencias y tecnologías existentes en la actualidad en las que nos apoyaremos para el desarrollo del sistema. Se describen aspectos fundamentales sobre la Web como, la diferencia entre aplicación Web y sitio Web, lenguajes de programación más usados en el desarrollo de dichas aplicaciones y una panorámica de los Sistemas Gestores de Bases de Datos a utilizar. Además se brinda una explicación de la metodología a seguir para el análisis, diseño e implementación del sistema informático a desarrollar y por último se expone una propuesta del ciclo a seguir para el desarrollo de la aplicación.

1.2 Sistemas vinculados al campo de acción.

La importancia que han adquirido en la sociedad los procesos docentes educativos en los centros universitarios, ha provocado la existencia de sistemas o aplicaciones para la gestión y el control eficiente de los procesos educacionales. A continuación se hace un estudio crítico sobre los principales sistemas existentes en el ámbito nacional e internacional para el control de tesis.

En el instituto superior José Antonio Echeverría (CUJAE) no existen antecedentes de aplicaciones para el control de tesis de los estudiantes, no se ha realizado ningún aporte a esta rama. En la Universidad de la Habana pasa un caso similar, los estudiantes, comienzan con la realización y estructuración de los trabajos de diplomas desde cursos anteriores, temas que son propuestos por un tutor y después seleccionados por uno o dos estudiantes. Por lo que se hace imperiosa la necesidad de implementar una aplicación para el Control de Tesis.

Por otra parte en algunas Universidades del mundo existen bases de datos donde se guarda trabajos de diploma, maestrías, etc. pero no hay un sistema de gestión que controle los trabajos de diploma desde su inicio y controle el avance, además de monitorear la réplica de trabajos. Si existen muchos sitios donde se publica información y trabajos importantes. Esta funcionalidad de publicar trabajos de diploma terminados va a quedar plasmada en nuestro sistema.

1.3 La Web.

La World Wide Web o la Web fue creada alrededor de 1990 por el inglés Tim Berners-Lee y el belga Robert Cailliau mientras trabajaban en el CERN en Ginebra, Suiza. Desde entonces, Berners-Lee ha jugado un papel activo guiando el desarrollo de estándares Web (como los lenguajes de marcado con los que se crean las páginas Web).

La Web es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza sitios Web, forjados de páginas Web que pueden contener texto, imágenes u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los webmails, wikis, weblogs, tiendas en línea y la Wikipedia misma son ejemplos bien conocidos de aplicaciones web.

1.4 Diferencias en las Aplicaciones Web y Sitios Web.

En ingeniería del software una **aplicación Web** es aquella que los usuarios usan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

En una aplicación Web a los usuarios se le permite ejecutar lógica de negocio a través de un navegador, en otras palabras modificar el estado del negocio. Los mismos utilizan tecnologías que generan contenidos dinámicos y permiten a los usuarios del sistema modificar la lógica del negocio en el servidor, de no ser así considerado un sitio Web.

Un sitio Web está estructurado por un Servidor Web, una Red y un cliente o navegador. Una aplicación Web incluye además una aplicación en el Servidor, que va hacer quien permitirá manejar lógica de negocio y tener un estado.

Las aplicaciones Web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores comunes como HTML (HyperText Markup Language) o XHTML (EXtensible Hypertext Markup Language).

Sus interfaces poseen ciertas limitantes en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla, arrastrar o soltar no están soportados por las tecnologías Web estándar. Los desarrolladores utilizan generalmente lenguajes interpretados del lado del cliente para obtener una mayor funcionalidad, así como tecnologías para al lado del servidor para no tener que recargar la página en su totalidad, algo que molesta mucho a los usuarios.

Una aplicación Web está estructurada como una aplicación de tres-capas, el Web browser o navegador es la primera capa, un motor usando alguna tecnología Web dinámica (ejemplo: CGI, PHP, Java o ASP) es la capa del medio, y una base de datos como última capa. El Web browser manda

peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario. (Ver **ANEXO I**).

1.5 Tecnologías actuales a utilizar.

1.5.1 Arquitectura cliente\servidor.

"Es un 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".(1)

Por lo tanto la arquitectura cliente\servidor es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores". (Ver **ANEXO 2**).

El concepto de cliente\servidor proporciona una forma eficiente de utilizar todos estos recursos de máquina de tal forma que la seguridad y fiabilidad que proporcionan los entornos mainframe se traspa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

1.5.1.1 Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

1.5.1.2 Algunas ventajas:

- **Aumento de la productividad:** Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos. Mediante la integración de las aplicaciones cliente/servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes. Una interfaz gráfica de usuario consistente reduce el tiempo de aprendizaje de las aplicaciones.
- **Bajo costo de desarrollo:** Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, el compartir los servidores (habitualmente caros) y dispositivos periféricos (como impresoras) entre máquinas clientes permite un mejor rendimiento del conjunto. Proporcionan un mejor acceso a los datos. La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información. El movimiento de funciones desde un ordenador central hacia servidores o clientes locales origina el desplazamiento de los costes de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.
- **Mejora en el rendimiento de la red:** Las arquitecturas cliente/servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios. Tanto el cliente como el servidor pueden escalarse para ajustarse a las necesidades de las aplicaciones.
- **La existencia de varias UCPs proporciona una red más fiable:** Un fallo en uno de los equipos no significa necesariamente que el sistema deje de funcionar. En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros con lo que

pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema. La arquitectura modular de los sistemas cliente/servidor permite el uso de ordenadores especializados (servidores de base de datos, servidores de ficheros, etc.). Permite centralizar el control de sistemas que estaban descentralizados, como por ejemplo la gestión de los ordenadores personales que antes estuvieran aislados.(2)

1.5.2 Servidor Web Apache.

El servidor Web Apache es uno de los mayores triunfos del software libre. Entre sus características fundamentales podemos destacar:

- Es flexible, rápido y eficiente.
- Continuamente actualizado y evoluciona a mayor velocidad.
- Multiplataforma.
- Se desarrolla de forma abierta.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Extensible: Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.(3)

1.5.3 Lenguajes de programación para la Web.

Existe una multitud de lenguajes concebidos para la Web. Cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útil para desarrollar distintas aplicaciones. Se pueden clasificar de dos formas diferentes; de acuerdo a la arquitectura Cliente/Servidor, los lenguajes del lado del servidor son aquellos reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él (entre estos lenguajes podemos encontrar **(PHP, ASP, JSP, Perl, Python)**); por otro lado, los lenguajes del lado del cliente (entre los cuales no sólo se encuentra el HTML sino también el VBScript y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente comprendidos por el navegador y no necesitan un pretratamiento.

Cada uno de estos tipos tiene por supuesto sus ventajas y sus inconvenientes, por ejemplo: un lenguaje del lado del cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Inversamente, un lenguaje del lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a

las versiones del mismo; por otra parte, los scripts son almacenados en el servidor quien los ejecuta y traduce a HTML por lo que permanecen ocultos para el cliente. Este hecho puede resultar a todas luces una forma legítima de proteger el trabajo intelectual realizado.

1.5.3.1 Del lado del cliente:

JavaScript.

Es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica. Permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, JavaScript pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.(4)

HTML.

HTML es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir la forma en la que presentar el texto y otros elementos de la página. El HTML es un lenguaje de programación muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida pueda enfrentarse a la tarea de crear una web.

Este lenguaje se escribe en un documento de texto, por eso se necesita un editor de textos para escribir una página Web. Así pues, el archivo donde está contenido el código HTML es un archivo de texto, con una peculiaridad, que tiene extensión .html o .htm (es indiferente cuál utilizar). De modo que cuando se programa en HTML se realiza con un editor de textos, lo más sencillo posible y guarda los trabajos con extensión .html.(5)

VBScript.

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de JavaScript.

Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están muy inspiradas en él. Sin embargo, no todo lo que se puede hacer

en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.

El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas Web es muy similar al utilizado en JavaScript y los recursos a los que se puede acceder también son los mismos: el navegador.(6)

1.5.3.2 Del lado del servidor:

Personal Home Page (PHP).

El PHP (acrónimo de PHP: Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. También ofrece la integración con varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa, desde generar documentos en pdf, hasta analizar código XML.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día en comparación con el código de otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.(7)

JSP.

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es una tecnología orientada a crear páginas Web con programación en Java. Con JSP podemos crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual. En JSP se crean páginas de manera parecida a como se crean en ASP o PHP. Se generan archivos con extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor. Antes de que sean funcionales los archivos, el motor JSP lleva a cabo una fase de traducción de esa página en un servlet, implementado en un archivo class (Byte codes de Java). Esta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de

la página .jsp, aunque existe la opción de precompilar en código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.(8)

ASP.

ASP (Active Server Pages) es una tecnología para la creación de páginas dinámicas del lado del servidor desarrollada por Microsoft. Con las ASP podemos realizar muchos tipos de aplicaciones distintas. Permite acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor. También tenemos la posibilidad de comprar componentes ActiveX fabricados por distintas empresas de desarrollo de software que sirven para realizar múltiples usos, como el envío de correo, generar gráficas dinámicamente, y otras funcionalidades.(9)

PERL.

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para Web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito.(10)

1.6 Metodologías de Trabajo.

Todo desarrollo de software se torna riesgoso y difícil de controlar, pero si no se utiliza una metodología, existe una mayor exposición a obtener clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Muchas veces no se toma en cuenta la utilización de una metodología adecuada. En ocasiones, ocurre que se realiza el diseño del software de manera rígida con los requerimientos que el cliente solicitó, de tal manera que cuando el cliente en la etapa final (etapa de prueba), solicita un cambio se hace muy difícil realizarlo, pues si se hace, altera muchas cosas que no se han previsto, y es justo éste, uno de los factores que ocasiona un atraso en el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido. Obviamente para evitar estos incidentes se debe llegar a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique el desarrollo del mismo, así como escoger una metodología adecuada para el desarrollo del software que sirva como guía para realizar de forma disciplinada y eficiente el producto deseado.

Para dar una idea de que metodología podemos usar y cuál será la más adaptable a nuestro sistema, se hará una breve descripción de las siguientes: RUP, XP y MSF.

Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado de Desarrollo (RUP) es una propuesta de proceso para el desarrollo de software orientado a objeto que utiliza UML para describir un sistema. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. Este proceso tiene como características fundamentales que está dirigido por casos de uso, es iterativo e incremental y centrado en la arquitectura. Está constituido por 5 flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba, los que tienen lugar sobre 4 fases: inicio, elaboración, construcción y transición.

- **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener el reléase del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

El ciclo de vida que se desarrolla por cada iteración, es llevado bajo dos disciplinas:

Disciplina de Desarrollo.

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Trasladando las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Trasladando los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte.

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

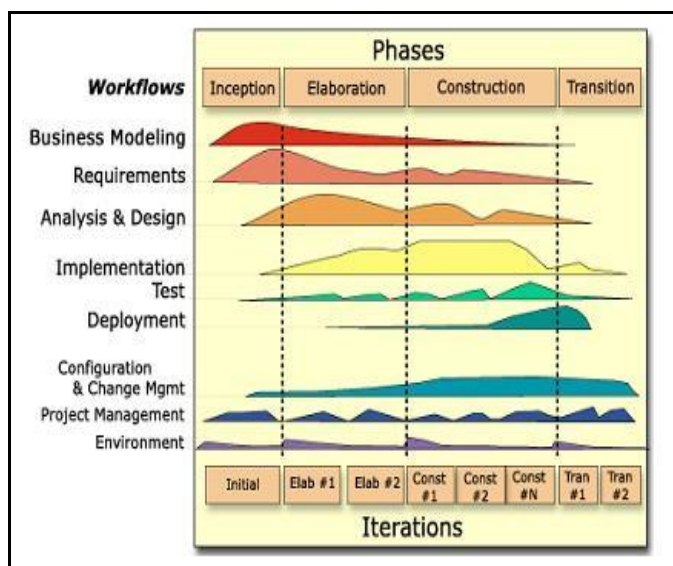


Figura 1.1 Fases e Iteraciones de la Metodología RUP.

Los elementos del RUP son:

- **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores:** Vienen hacer las personas o gentes involucradas en cada proceso.
- **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Características de RUP (Rational Unified Process).

Dirigido por Casos de Uso: Esto quiere decir que el proceso de desarrollo lleva un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba. Los casos de uso guían la arquitectura del sistema.

Centrado en la arquitectura: La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se refleja en los casos de uso. Incluye los aspectos más significativos del sistema y se ve influida por muchos otros factores como la plataforma en la que tiene que funcionar el software (arquitectura hardware, sistema operativo, sistema de gestión de base de datos), las consideraciones de implantación, sistemas heredados y requisitos no funcionales.

Iterativo e incremental: Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la

arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque.(11)

1.6.1 Extreme Programming (XP).

Es una de las metodologías de desarrollo de software más exitosa en la actualidad utilizada para proyectos de corto plazo, pequeño equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. La metodología se basa en:

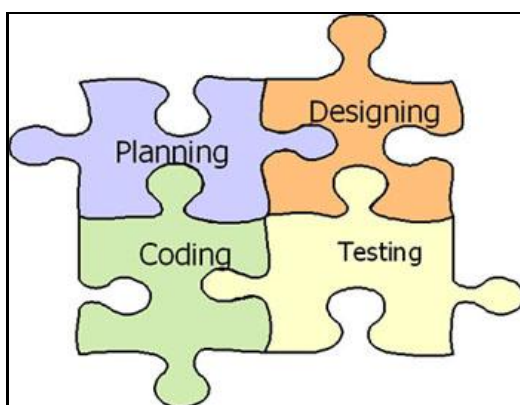


Figura 1.2 Metodología Extreme Programming.

- Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- Refabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en una parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias

- El cliente o el usuario se convierten en miembros del equipo.(11)

1.6.2 Microsoft Solution Framework (MSF).

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.



Figura 1.3 Metodología MSF.

MSF tiene las siguientes características:

Adaptable: Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

Escalable: Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.

Flexible: Es utilizada en el ambiente de desarrollo de cualquier cliente.

Tecnología Agnóstica: Porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación.(11)

1.7 Lenguaje Unificado de Modelado (UML).

Es el lenguaje gráfico para modelado de sistemas con tecnología orientada a objeto que permite especificar, visualizar, construir y documentar. Soporta todo el ciclo de vida de desarrollo de software: especificaciones de analistas, arquitectura, diseño, implementación e implantación. Soporta además distintas áreas de aplicación: sistemas distribuidos, tiempo real, aplicaciones mono proceso, sistemas de información corporativos, Banca/Finanzas, Telecomunicaciones, Defensa/Espacio, Transporte, Distribución, Electro medicina, Ciencia, entre otros. Es un lenguaje de modelado visual fácil de aprender pero rico en significado, es estándar, estable y configurable.(12)

Es importante remarcar que UML es un "lenguaje" para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas:

- Diagramas de Casos de Uso para modelar los procesos del negocio.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

1.8 Sistema de Gestión de Bases de Datos (SGBD).

Un Sistema de Gestión de Bases de Datos es aquel que se encarga de controlar el acceso concurrente, evitar redundancia, que se cumplan las restricciones y reglas de integridad, usar elementos que aceleren el acceso físico a los datos (índices, agrupamientos, funciones de dispersión), distribuir los bloques del disco del modo más adecuado para el crecimiento y uso de los datos, controlar el acceso y los privilegios de los usuarios y recuperar ante fallos.(13)

PostgreSQL.

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. A continuación se muestra brevemente una lista de algunas de esas características:

- Soporte SQL Comprensivo: PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia y arrays.
- Altamente Extensible: PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor.(14)

MySQL.

MySQL ha pasado de ser una pequeña base de datos a una completa herramienta. Su rápido desarrollo se debe en gran medida a la contribución de mucha gente al proyecto, así como la dedicación del equipo de MySQL.

Lo que en un tiempo se consideró como un sencillo juguete para uso en sitios Web, se ha convertido en la actualidad en una solución viable y de misión crítica para la administración de datos. MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios como: Oracle, SQL Server y Db2. MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. Puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no ha oído hablar nunca. MySQL utiliza el lenguaje de consulta estructurado (SQL).

MySQL es rápido, y una solución accesible para administrar correctamente los datos de una empresa. MySQLAB es la compañía responsable del desarrollo de MySQL, dispone de un sistema de asistencia eficiente y a un precio razonable, y como ocurre con la mayor parte de las comunidades de código abierto, se puede encontrar una gran cantidad de ayuda en la Web. Son muchas las razones para escoger a MySQL como una solución de misión crítica para la administración de datos:

- Asistencia: MySQLAB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.

- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- **Portabilidad:** MySQL se ejecuta en la inmensa mayoría de sistemas operativos y la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- **Facilidad de uso:** MySQL resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.(15)

Oracle.

Oracle es un sistema de administración de base de datos, fabricado por Oracle Corporation, básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas Web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server y otros. Oracle es sin duda una de las mejores bases de datos que tenemos en el mercado, es un sistema gestor de base de datos robusto, tiene muchas características que nos garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de gestores de bases de datos comerciales y de la oferta de otros con licencia Software Libre como PostgreSQL o FireBird.(15)

1.9 Herramientas CASE.

Se puede definir a las Herramientas CASE como: un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.

Clasificación:

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Podrían clasificarse atendiendo a:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que cubren.
- La arquitectura de las aplicaciones que producen.
- Su funcionalidad.

Algunas ventajas de su utilización:

- Aumento de la productividad y de la calidad de las aplicaciones desarrolladas.
- Las herramientas CASE ayudan a los desarrolladores a expresar gráficamente los procesos de negocio y sus correspondientes modelos de datos, validando la integridad y consistencia de los mismos y generando en muchos casos el código necesario para convertir dichos modelos en nuevas aplicaciones.
- Pueden producir rápidamente prototipos de las aplicaciones a desarrollar, los cuales se pueden probar, corregir e incorporar a la aplicación final.
- La tecnología CASE también permite partir los proyectos en múltiples piezas que se pueden trabajar individualmente y que luego pueden ser reensambladas en una única aplicación que funciona perfectamente.
- Suponen una gran ayuda a la hora de gestionar la documentación asociada al desarrollo de los sistemas de información, ya que proporcionan un entorno donde centralizar dicha documentación, además de generar buena parte de la misma.(16)

Visual Paradigm-UML.

Visual Paradigm para UML es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Visual Paradigm-UML también proporciona características tales como generación del código, ingeniería reversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

ER/Studio.

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos.

ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes.

ER/Studio ofrece las siguientes funcionalidades:

- Capacidad fuerte en el diseño lógico.
- Sincronización bidireccional de los diseños lógico y físico.
- Construcción automática de Base de Datos.
- Reingeniería inversa de Base de Datos.
- Documentación basada en HTML.
- Un Repositorio para el modelado.

Diagramas.

La creación de diagramas es clara y rápida. Tiene la posibilidad de realizar diagramas con desempeño rápido. También es posible cambiar el estilo de las líneas, los colores, tipos de letra, niveles de acercamiento, y modelos de despliegue. Es posible crear subvistas para separar y manejar áreas importantes. ER/Studio automáticamente mantiene todas las dependencias entre subvistas y el diagrama completo.

Ayuda.

Ya sea que se inicie un nuevo diseño o se mantenga uno existente, ER/Studio está equipado con elementos de ayuda para hacer el trabajo de manera efectiva. Las barras de herramientas tienen algunas sugerencias para el uso de las mismas, además de contar con ayuda en línea sensible al contexto.

Esquema de la base de datos.

Las capacidades de diseño que contiene, ayudan a crear un diseño lógico que puede transformarse en cualquier número de diseños físicos. Como resultado, se puede mantener un diseño lógico normalizado mientras se desnormalizan los diseños físicos para su desempeño. ER/Studio mantiene ligaduras entre todos los niveles de su diseño por lo tanto puede mezclar cambios en cualquier dirección entre ellos. ER/Studio revisa la normalización y la compilación con la sintaxis de la plataforma de la base de datos.

Ingeniería hacia delante y de reversa.

Ingeniería hacia delante:

Una vez que se ha diseñado la base de datos, se puede construir o generar código fuente para todo o para parte de los diseños de la base de datos. Propiamente hace la secuencia de la creación de tipos de objetos diferentes para asegurar eficiencia, y construir bases de datos libres de errores.

Actualiza una base de datos del diagrama. ER/Studio permite aplicar cambios de diseño del modelo de datos directamente a la base de datos. Cuando se comparan las diferencias entre los dos, formula una estrategia de alteración inteligente que implementa el diseño de las modificaciones mientras se preserva la tabla con los datos existentes, privilegios de objetos, y dependencias en la base de datos.

Ingeniería de reverso:

Cuenta con ingeniería de reverso, cuando necesite iniciar un trabajo de una base de datos existente, ER/Studio puede hacer una ingeniería de reverso al esquema completo para cualquier plataforma de bases de datos. La operación de la ingeniería de reverso extrae eficientemente definiciones de objetos y construye un modelo de datos gráfico.

Sincronización de la Base de Datos.

Sincronización entre el diagrama físico y el lógico. Mezcla entre cualquier par de diagramas físicos para la misma plataforma de bases de datos. Comparación lado-a-lado de las diferencias. El usuario puede decidir qué diferencias mezclar o ignorar.

- Objetos reusables. Construir atributos reusables. Aplicarlos a atributos y columnas. Propagación global de actualizaciones. Construir tipos de datos personalizables.
- Submodelado. Crear cualquier número de subvistas personalizables sobre un diagrama físico o lógico. Cualquier objeto puede existir en cualquier número de subvistas (relaciones de muchos a muchos entre objetos y subvistas).
- Crear rápidamente subvistas eligiendo un área del diagrama.
- Control independiente sobre el despliegue de la subvista, incluyendo posición del objeto, colores y letras.
- Utilidad de búsqueda rápida. Editores en tabla para evitar la necesidad de poner en modo cascada los diálogos.
- Diferenciación de color de llaves primarias y secundarias inherentes. Sombreado de cajas de la entidad.(17)

1.10 Selección de las tecnologías a utilizar en la propuesta de solución.

Luego de haber realizado un estudio de las tecnologías más avanzadas se concluyó con una propuesta de solución en la cual se utilizará PHP como lenguaje de programación usado generalmente para la creación de contenido para sitios web, teniendo en cuenta algunas de sus ventajas como ser un lenguaje multiplataforma, con capacidad de conectarse con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, capacidad de expandir su potencial utilizando la enorme cantidad de módulos, es libre, por lo que se presenta como una alternativa de fácil acceso para todos; y lenguaje del lado del cliente JavaScript. Como sistema gestor de base de datos se escogió

PostgreSQL por ser rápido, confiable y fácil de usar, ofrece hoy una rica variedad de funciones, su conectividad, velocidad y seguridad lo hacen altamente satisfactorio para acceder bases de datos. Se tiene también como propuesta a utilizar la metodología de desarrollo RUP con notación UML debido a sus características y organización del trabajo. Herramienta CASE de Modelado UML: Visual Paradigm v6.1 y Servidor Web: Apache. ER/Studio v7.1 para el diseño lógico y físico de la base de datos.

Se decidió utilizar como lenguajes de programación para la implementación del software PHP y Java Script.

1.11 Conclusiones.

En la primera parte de este capítulo se hizo referencia a conceptos relacionados con el objeto de estudio definido en la introducción del documento. En la segunda parte, se desarrolla el estudio de algunas de las tecnologías actuales que serán empleadas durante el desarrollo del software, mencionando ciertas características y ventajas que las mismas poseen; la decisión de utilizarlas fue tomada siguiendo fundamentalmente la política de uso de herramientas con soporte multiplataforma y licencias de utilización libre las cuales fueron escogidas por nuestros clientes. Fueron utilizadas además algunas herramientas de software propietario para apoyar la realización del trabajo.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción.

Para poder comprender lo que es el sistema, es importante examinar las características del mismo que lo diferencian de otras cosas que los hombres pueden construir, pues el producto que finalmente se obtiene no se traduce en algo físico sino lógico. El software es un elemento del sistema que se desarrolla; no se fabrica. En el presente capítulo abordaremos aspectos relacionados con el objeto de automatización; la información que se maneja, es decir, todo lo relacionado con los documentos específicos que se procesen; una descripción general de la propuesta de sistema y cómo debe funcionar. Estaremos abordando además, el modelado del negocio, la especificación de los requisitos funcionales y no funcionales así como la definición de los casos de uso.

2.2 Objeto de estudio.

2.2.1 Problema y situación problemática.

En la Universidad de las Ciencias Informáticas las facultades son las encargadas de dirigir y controlar los procesos de docencia y producción. Muchos de los procesos que controla este organismo orientados al cumplimiento de la preparación general de los alumnos se realizan de forma manual, sobre todo los que tienen lugar en los órganos de Gestión y Organización de Trabajos de Diploma. Para cada facultad se hace engorrosa la tarea de brindar temas de investigación, mantener reportes sistemáticos, medir el alcance e impedir replicas de investigaciones, además de Generar la Planificación para todo el control del trabajo en cuestión. Por el grado de importancia que tiene esta tarea se hace necesaria la utilización de herramientas automatizadas que faciliten el flujo de información haciendo su organización y transmisión más rápida y eficaz, sin embargo con los medios existentes hasta el momento la obtención de estos informes se hace demasiado lento, así como su almacenamiento que debe ser organizado y clasificado atendiendo a diversos perfiles de trabajo, además puede que existan márgenes de error en determinadas informaciones. Asimismo la recopilación de grandes cantidades de documentos (perfiles de tesis) trae consigo dificultades a la hora de obtener alguna información transmitida, o realizar una búsqueda. La falta de medios automatizados que permitan llevar a cabo las actualizaciones necesarias así como el envío y recepción de datos y los perfiles de trabajos de forma eficiente que garantice la confidencialidad y la integridad de la información de los trabajos en cuestión y desarrollados; es decir la gestión de la misma viene a ser la situación problemática que se nos presenta en las facultades de la Universidad de las Ciencias Informáticas.

2.2.2 Objeto de automatización.

Entre los procesos que serán objeto de automatización se encuentra la entrada y recepción de datos y perfil de tesis. Los coordinadores de tesis ingresan de forma ordenada los datos del estudiante, perfil de tesis, tutor(s), y el tribunal al que pertenece. El plan de trabajo de este organismo también brindará la oportunidad a profesores de publicar temas de trabajos de diploma ya aprobados, donde el estudiante tendrá la posibilidad de escoger o al menos inscribirse en dicha investigación publicada, luego se aprobará o no su registro por el coordinador de tesis conjuntamente con el profesor o tutor; que en muchos casos coincide con la misma persona. Todo este proceso se realiza basándose en cubrir las necesidades de cada facultad. También será objeto de automatización el proceso de Generar la Planificación de Cortes de Tesis, donde se tiene en cuenta varias reglas. De este modo se mantendrá de forma organizada el transcurso y control de los trabajos de diploma, vinculando esto con la posibilidad de automatizar un mecanismo genérico de reportes estadísticos. Esto permitirá mantener informado a todo el personal implicado y conocer el estado de los trabajos de diploma en la facultad.

2.2.3 Información que se maneja.

El conjunto organizado de datos que se maneja es característico del proceso en cuestión, existen diferentes documentos:

Comité de Tesis: Constituye el documento en el que se maneja la información de los profesores implicados en el trabajo de aprobación de Perfiles de Tesis, esto se refiere a un Jefe de Comité de Tesis y varios miembros.

Perfil de Tesis: Documento que maneja toda la información acerca del trabajo de diploma en cuestión, proporciona un conocimiento previo de la investigación, incluye entre sus datos la Clasificación del trabajo de diploma en cuestión.

Tribunal de Tesis: Documento en el cual se muestra cada tribunal por polos de investigación.

2.2.4 Propuesta de sistema.

Después de haber realizado un profundo, amplio estudio y análisis de los problemas existentes en los órganos de Gestión y Organización de los Trabajos de Diploma en los niveles correspondientes con el manejo y la administración de toda la información que se necesita controlar, buscar y obtener; y teniendo en cuenta de que no existe un sistema anterior que de cumplimiento total a lo requerido, y que cubra en su totalidad todas las necesidades, se propone desarrollar un sistema que aporte soluciones reales, palpables, eficientes y satisfactorias a las necesidades planteadas. El cual estará ubicado en las facultades de la Universidad de las Ciencias Informáticas, y que su principal administrador será el Coordinador de Tesis de cada facultad. Ofreciendo la posibilidad de recibir datos de Perfiles de Tesis, publicar Perfiles de Trabajos de Diplomas aprobados, generar reportes

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

estadísticos de cada corte por individual o general y además de lo antes mencionado generar una Planificación de Cortes de Tesis. El sistema quedará elaborado de forma tal que muestre al usuario solo aquellas opciones a la que está autorizado a acceder y realizar en las interfaces solo los cambios que le sean permitidos dependiendo de las responsabilidades que posea. Debe ser fácil de usar y debe mantener todos los datos archivados con la mayor seguridad disponible, así como la integridad de los mismos.

2.3 Modelo de negocio.

2.3.1 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa y desempeña un rol determinado dentro del negocio para beneficiarse de sus resultados. Representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, por otro lado un mismo usuario puede actuar como diferentes actores.

Tabla 2.1 Descripción de los actores del negocio.

Actores del negocio	Justificación
Estudiante	Este actor representa todos los estudiantes que se encuentran desarrollando trabajos de diploma.
Dirección de la Facultad	Este actor representa la Dirección de la Facultad. Es el encargado de llevar el control de toda la información referente a los procesos de solicitud, asignación y control de trabajos de diploma. Además es quien realiza todos los procesos organizativos en el ciclo de desarrollo y terminal de los proyectos de tesis.
Profesor	Este actor representa a todos los profesores que presentan las propuestas de Perfiles de Tesis, y que después serán aprobadas o no en el Comité de Tesis de cada facultad.

2.3.2 Trabajadores del negocio.

Un trabajador del negocio representa a personas o sistemas dentro del negocio que son los que realizan las actividades que están comprendidas dentro de un caso de uso.

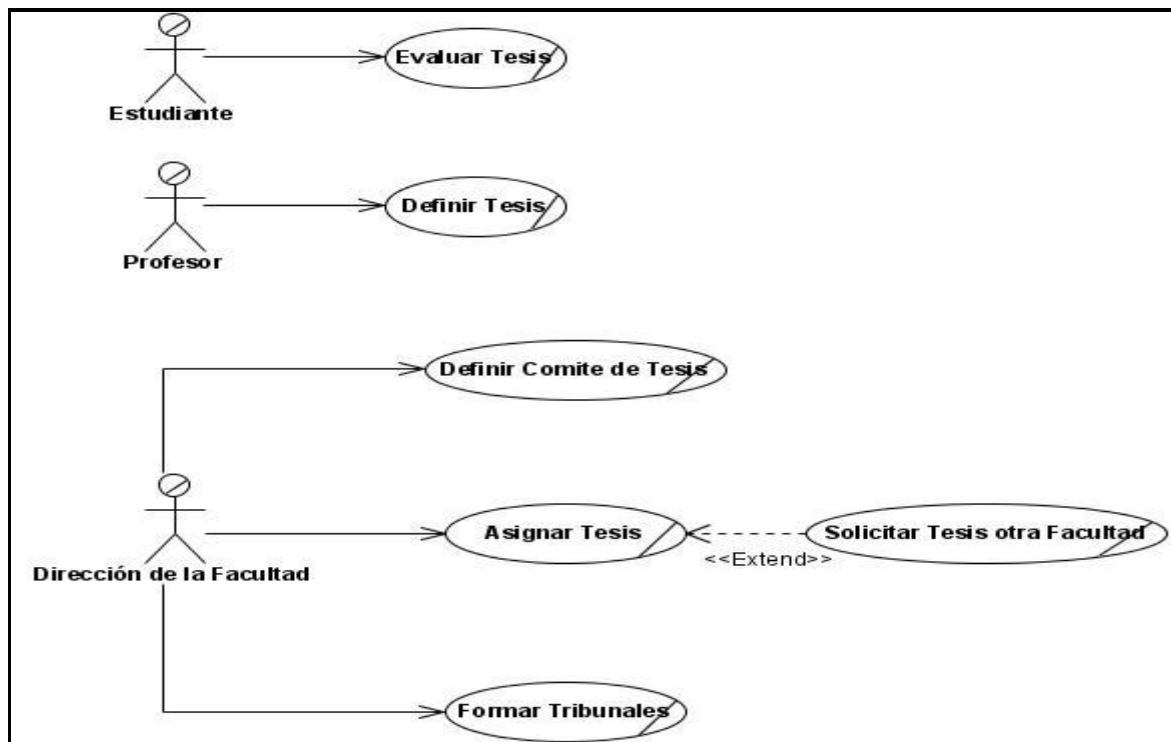
Estos trabajadores están dentro de la frontera del negocio, son los candidatos a convertirse en un futuro en usuarios del sistema que se quiere construir.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tabla 2.2 Descripción de los trabajadores del negocio.

Trabajadores del negocio	Justificación
Jefe del Departamento de Ingeniería de Software (ISW)	Es el encargado de definir en la facultad el comité de tesis integrados por profesores. Además de definir los polos de investigación a los cuales están asignados los perfiles de tesis.
Jefe del Comité de Tesis	Es el encargado de definir y aprobar los perfiles de tesis que serán asignados a los estudiantes en la facultad y conformar los tribunales.
Presidente del Tribunal	Es el encargado de evaluar los trabajos de diploma, además de hacer señalamientos y sugerencias al autor(es). Tiene la facultad de medir el alcance y avance de la investigación en cuestión.
Tribunal	Es el encargado de evaluar a los estudiantes en cada uno de los cortes de tesis que se realizan.

2.3.3 Diagrama de casos de uso del negocio.



CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Figura 2.1 Diagrama de casos de uso del negocio

2.3.4 Descripción en formato extendido de los casos de uso del negocio.

*Tabla 2.3 Especificación de caso de uso: **Evaluar Tesis.***

Caso de Uso:	Evaluar Tesis.	
Actores:	Estudiante.	
Trabajadores:	Presidente del Tribunal, Tribunal.	
Resumen:	El CU inicia cuando la facultad hace un seguimiento de las tesis para llevar una constancia del nivel de avance de cada una de ellas, para esto la facultad asigna cada tesis a un tribunal según su perfil, según el estado en que se encuentre la facultad propone una evaluación para estas tesis que puede ser bien (B), regular (R) y mal (M). Además aquí también se hacen señalamientos y sugerencias al tesista. También se mide el alcance del trabajo en cuestión.	
Precondiciones:		
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Negocio	
1. El estudiante(s) presenta el avance de las tesis mediante la exposición de los cortes de tesis llevados a cabo en la facultad.	2. El Tribunal verifica el cumplimiento de los objetivos y realiza preguntas relacionadas con el corte.	
3. El estudiante responde las preguntas hechas por el Tribunal.	4. El Tribunal analiza en alcance del trabajo en cuestión, en caso de no tener alcance ver flujo alternativo 1a.	
	5. El Tribunal realiza sugerencias y señalamientos que puedan ayudar al estudiante en su tesis.	
	6. El Tribunal propone una nota de acuerdo con lo expuesto por el estudiante(s) en cada corte de tesis.	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	7. El Presidente del Tribunal recibe la evaluación del corte de tesis.
	8. El Presidente del Tribunal archiva la evaluación realizada y las expone para que los estudiantes tengan conocimiento previo.
9. El estudiante recibe la(s) nota(s) del corte de tesis.	10. El presidente del tribunal verifica que no existan estudiantes pendientes en la evaluación, en caso contrario ver el flujo alternativo 2a.
Flujos Alternos	
Flujo alternativo 1a. no tiene alcance el perfil de tesis.	
Acción del Actor	Respuesta del Negocio
	4. El tribunal rechaza el Perfil de Tesis.
5. El estudiante modifica el Perfil de Tesis junto al tutor.	
Flujos Alternos	
Flujo alternativo 2a. si existen estudiantes pendientes de evaluación.	
Acción del Actor	Respuesta del Negocio
	10. El Presidente del Tribunal planifica el horario para los pendientes en la evaluación.
11. El estudiante recibe la planificación para evaluarse en el corte que tiene pendiente.	
Poscondiciones	

Tabla 2.4 Especificación de caso de uso: Definir Tesis.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Caso de Uso:	Definir Tesis.	
Actores:	Profesor.	
Trabajadores:	Jefe del Comité de Tesis.	
Resumen:	El CU inicia cuando el Profesor crea propuestas de posibles perfiles de tesis y las presenta ante el Jefe del Comité de Tesis, este recibe las propuestas de perfiles de tesis. El comité de tesis es el encargado de aprobar o no estos perfiles. En caso de presentar errores el comité manda al profesor a modificar el tema y ser reanalizado después. Define y organiza todos los perfiles que fueron aprobados para conocimiento de los profesores de la facultad.	
Precondiciones:		
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Negocio	
1. El profesor(es) crea las propuestas de posibles perfiles de Trabajos de Diploma y se le presenta al Jefe del Comité de Tesis.	2. El Jefe del Comité de Tesis recibe por parte de los profesores las propuestas de Perfiles de Trabajos de Diploma.	
	3. El Jefe del Comité de Tesis conjuntamente con el comité de tesis aprueba el perfil de tesis, en caso contrario ver el flujo alternativo 3a.	
	4. El Jefe del Comité de Tesis se encarga de archivar perfil(es) de tesis aprobado.	
	5. El Jefe del Comité de Tesis envía notificación de aprobación al profesor.	
Flujos Alternos		
Flujo alternativo 3a. no se aprueba el perfil de tesis.		
Acción del Actor	Respuesta del Negocio	
	1. El Jefe del Comité de Tesis le envía	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	al profesor el perfil de tesis que tiene error.
2. El profesor recibe Perfil de Tesis con error y modifica el perfil de tesis.	3. El Jefe del Comité de Tesis recibe el perfil de tesis corregido.
	4. El Perfil de Tesis es analizado nuevamente.
Poscondiciones	

Tabla 2.5 Especificación de caso de uso: Definir Comité de Tesis.

Caso de Uso:	Definir Comité de Tesis.	
Actores:	Dirección de la Facultad.	
Trabajadores:	Jefe del Departamento de ISW.	
Resumen:	El CU inicia cuando la facultad solicita la confección del comité de tesis. Un comité de tesis está conformado por profesores, este comité es el encargado de aprobar y asignar las tesis a los estudiantes. El Jefe del Dpto. de ISW es el encargado de conformar el comité de tesis.	
Precondiciones:		
Flujo Normal de Eventos		
Sección ""		
Acción del Actor	Respuesta del Negocio	
1. La Dirección de la Facultad solicita al Jefe del Dpto. de ISW la definición del Comité de Tesis.	2. El Jefe del Dpto. de ISW hace un estudio de los profesores que mejores condiciones tienen en la facultad que puedan conformar el Comité de Tesis.	
	3. El Jefe del Dpto. de ISW crea el Comité de Tesis.	
	4. El Jefe del Dpto. de ISW archiva los datos del Comité de Tesis y le informa a la Dirección de la facultad.	
5. La Dirección de la Facultad recibe la		

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

estructura del comité de tesis.	
---------------------------------	--

*Tabla 2.6 Especificación de caso de uso: **Formar Tribunales.***

Caso de Uso:	Formar Tribunales.
Actores:	Dirección de la Facultad.
Trabajadores:	Jefe del Comité de Tesis.
Resumen:	El CU inicia cuando la Dirección de la Facultad solicita la creación de los tribunales de Tesis. El Jefe del Comité de Tesis de la facultad es el encargado de la confección de los mismos, que están integrados por profesores. Dichos tribunales están conformados por, un presidente, dos vocales y un secretario. Los tribunales se distribuyen por áreas de producción y desarrollo.
Precondiciones:	
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Negocio
1. La Dirección de la Facultad solicita la creación de los Tribunales de Tesis.	2. El Jefe del Comité de Tesis realiza un estudio de los posibles tribunales de tesis en dependencia de las áreas de producción y desarrollo.
	3. El Jefe del Comité de Tesis confecciona los diferentes Tribunales de Tesis, con todos sus integrantes.
4. La Dirección de la Facultad recibe la información y la distribuye a los implicados.	

*Tabla 2.7 Especificación de caso de uso: **Asignar Tesis.***

Caso de Uso:	Asignar Tesis.
Actores:	Dirección de la Facultad.
Trabajadores:	Jefe del Comité de Tesis.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Resumen:	El CU inicia cuando el Jefe del Comité de Tesis desea conocer la cantidad de estudiantes que están cursando el quinto año académico. Se da una propuesta del estudiante(s) y del tutor(es) que van a llevar a cabo cada Perfil de Tesis aprobado y el resultado final es distribuido por el Jefe del Comité de Tesis. Si aún quedan estudiantes sin Perfiles de Tesis asociados, se le exige a la Facultad por nuevos Perfiles de Tesis. Los existentes se guardan.	
Precondiciones:		
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Negocio	
	1. El Jefe del Comité de Tesis solicita el conocimiento de la cantidad de estudiantes que cursan el quinto curso académico.	
2. La Dirección de la Facultad le informa al Jefe del Comité de Tesis la cantidad de estudiantes existentes con esta condición.	3. El Jefe del Comité de Tesis recibe la información.	
	4. El Jefe del Comité de Tesis da una propuesta de los posibles estudiantes y tutores que puedan desarrollar los diferentes Perfiles de Tesis. En caso contrario ver flujo alterno 4a.	
	5. El Jefe del Comité de Tesis distribuye la información asociada a los implicados.	
6. La Dirección de la Facultad recibe la información.	7. Guarda los Perfiles de Tesis conjuntamente con el estudiante(s) y tutor(s) relacionados.	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Flujos Alternos	
Flujo alternativo 4a. hay estudiantes en la facultad sin Perfiles de Tesis.	
Acción del Actor	Respuesta del Negocio
	1. El Jefe del Comité de Tesis exige a la facultad Perfiles de Tesis para los estudiantes vacantes de temas.
2. La Dirección de la Facultad, solicita nuevos Perfiles de Tesis a los profesores.	3. Recibe las nuevas propuestas de Perfiles de Tesis.
	4. El Jefe del Comité de Tesis distribuye los nuevos Perfiles de Tesis a los estudiantes vacantes.
5. La Dirección de la Facultad, recibe la información.	6. Guarda todos los Perfiles de Tesis, con el estudiante(s) y tutor(es) asociados.
Poscondiciones	

Tabla 2.8 Especificación de caso de uso: Solicitar Tesis a otra Facultad (CU extendido).

Caso de Uso:	Solicitar Tesis a otra Facultad.
Actores:	Dirección de la Facultad.
Trabajadores:	Jefe del Comité de Tesis.
Resumen:	El CU inicia en caso de que falten estudiantes en la facultad por temas de tesis, se les solicitan temas a otras facultades y los mismos son guardados después de que se presenten.
Precondiciones:	
Flujo Normal de Eventos	
Sección ""	
Acción del Actor	Respuesta del Negocio
	1. El Jefe del Comité de Tesis informa de la ausencia de propuestas de Perfiles de Tesis en la Facultad.
2. La Dirección de la Facultad recibe la información preliminar.	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

3. La Dirección de la Facultad solicita nuevos Perfiles de Tesis a otras facultades.	4. El Jefe del Comité de Tesis recibe las propuestas de Perfiles de Tesis realizadas por otras facultades.
	5. El Jefe del Comité de Tesis conjuntamente con el Comité de Tesis aprueba el perfil. En caso contrario ver flujo alternativo 5a.
	6. Archivar Perfil(es) de Tesis aprobados y enviar información a la Dirección de la Facultad.
7. La Dirección de la Facultad recibe la información.	
Flujos Alternos	
Flujo Alterno 5a: No se aprueba en la facultad el Perfil de Tesis propuesto desde otra facultad.	
Acción del Actor	Respuesta del Negocio
	1. El Jefe del Comité de Tesis envía a la Dirección de la Facultad la no aprobación del Perfil de Tesis solicitado.
2. La Dirección de la Facultad recibe la información.	
3. La Dirección de la Facultad envía información a la(s) facultad(es) correspondientes con cada perfil.	
Poscondiciones	

2.3.5 Diagramas de Actividades.

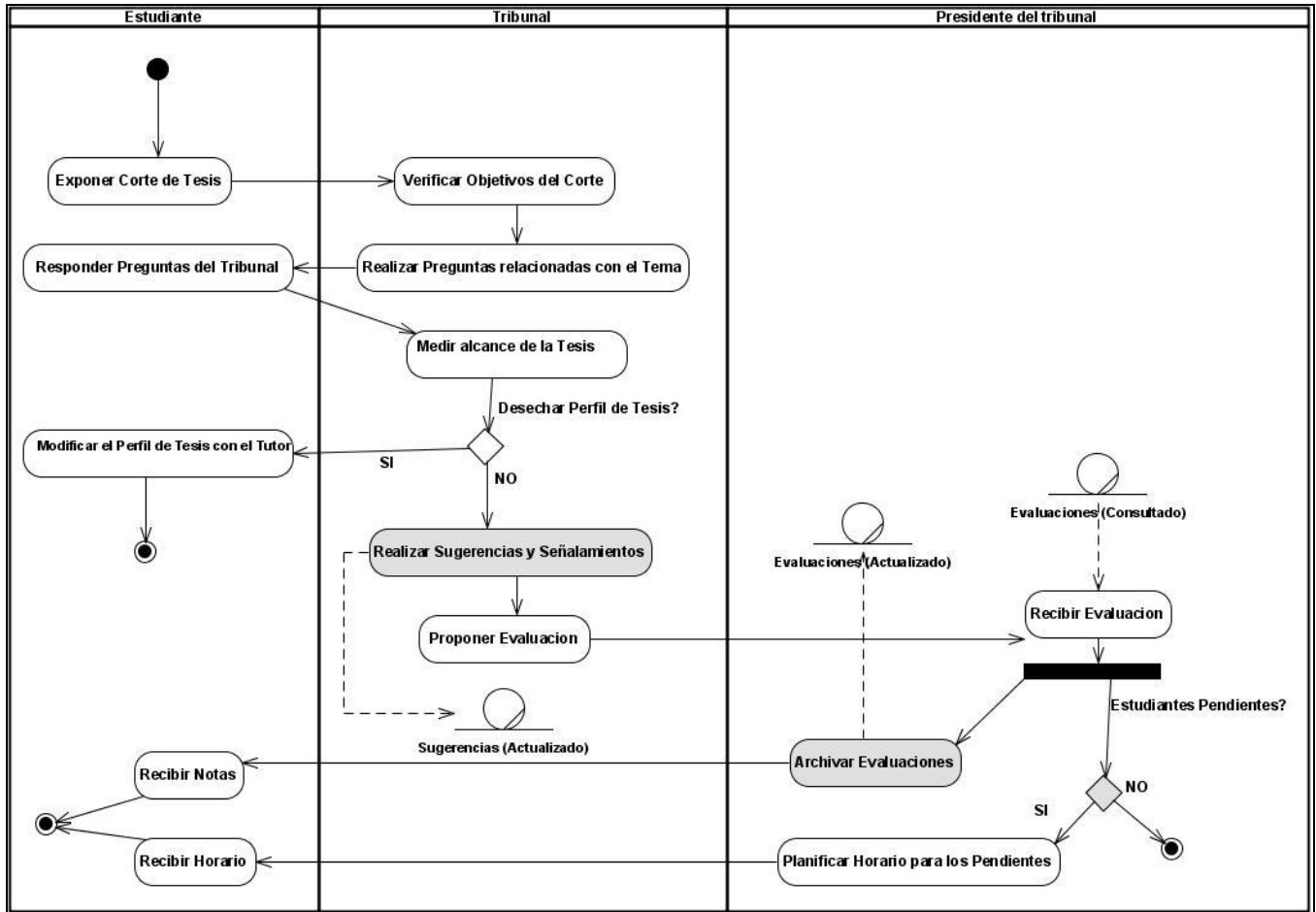


Figura 2.2 Diagrama de Actividad CU: *Evaluar Tesis*.

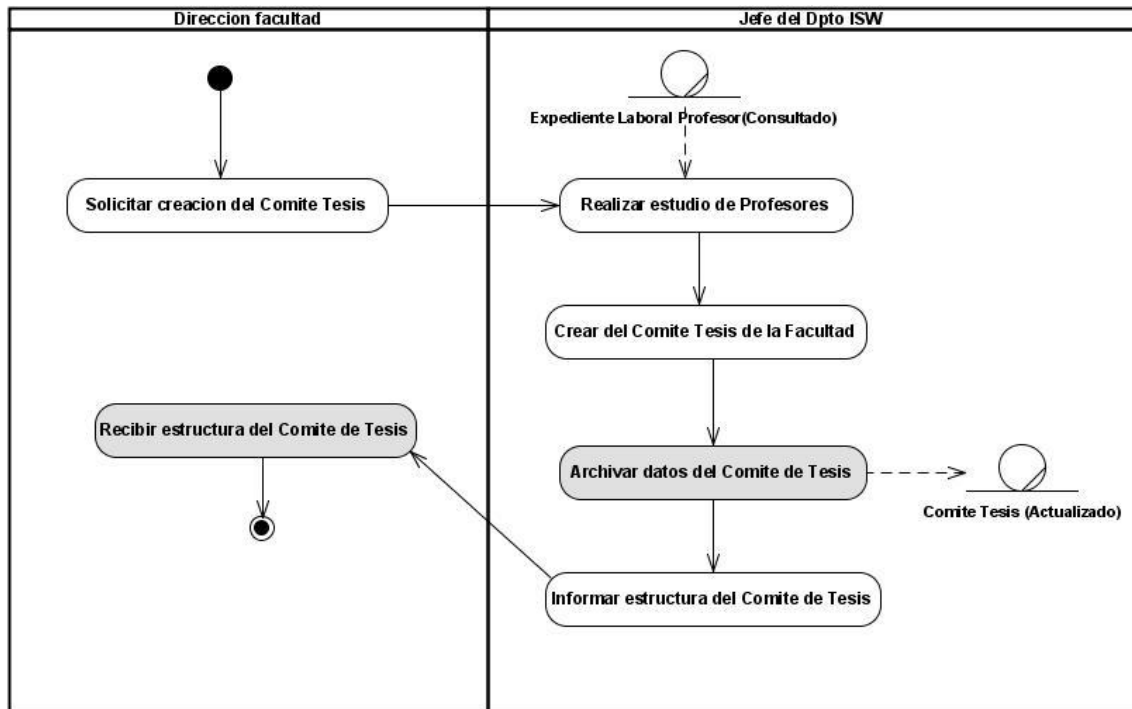


Figura 2.4 Diagrama de Actividad CU: Definir Comité Tesis.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

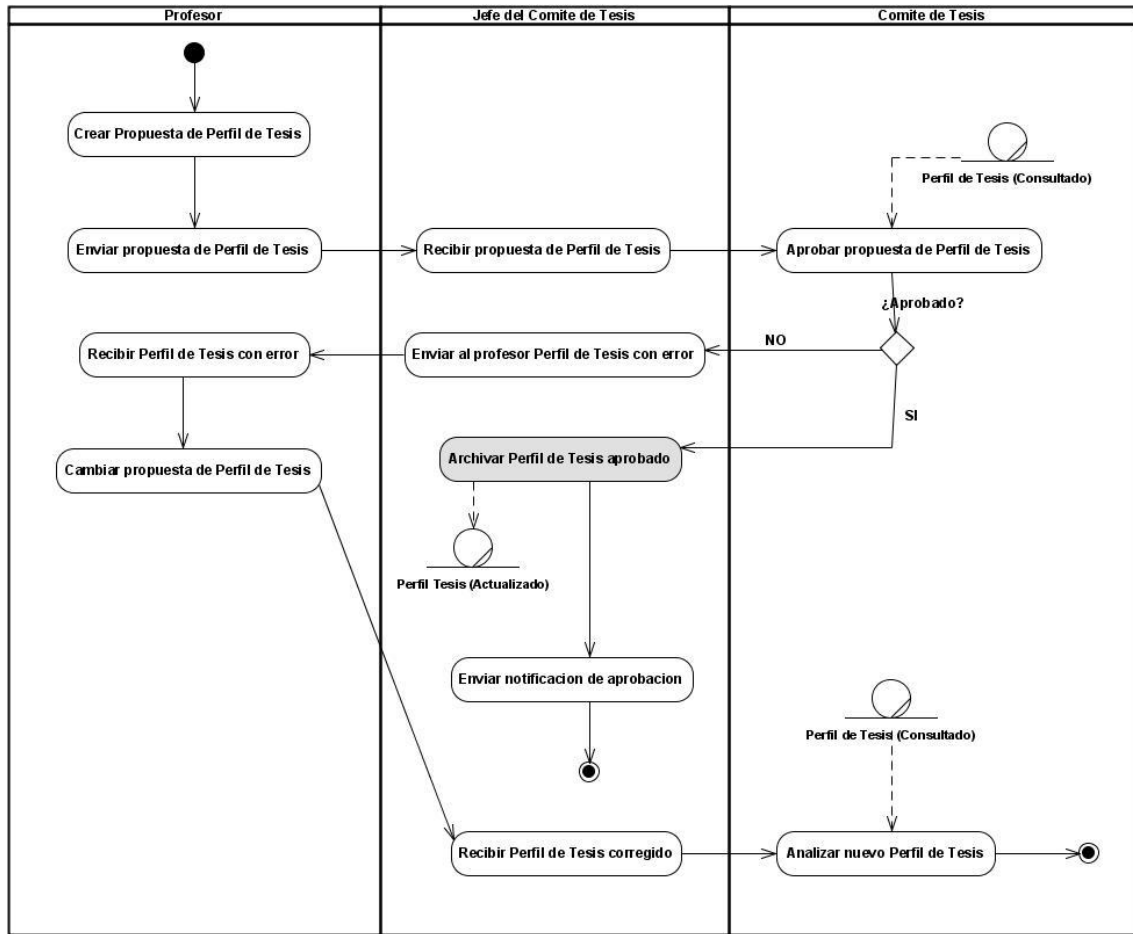


Figura 2.3 Diagrama de Actividad CU: *Definir Tesis*.

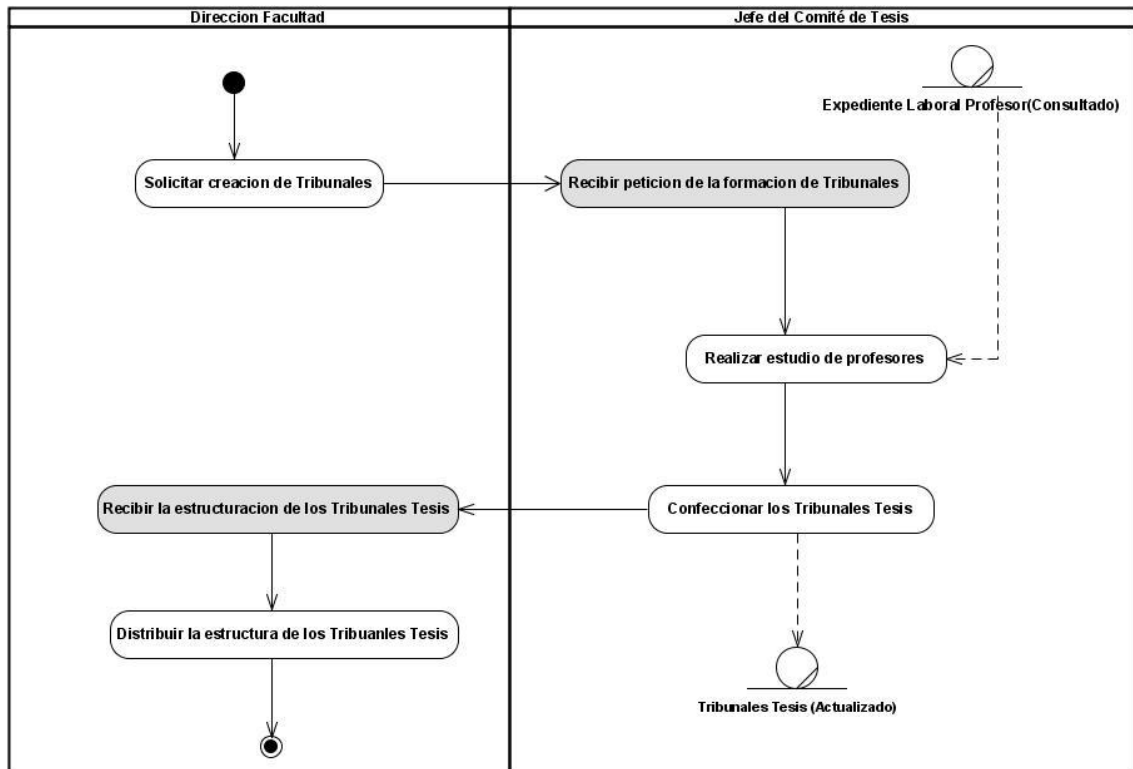


Figura 2.5 Diagrama de Actividad CU: Formar Tribunales.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

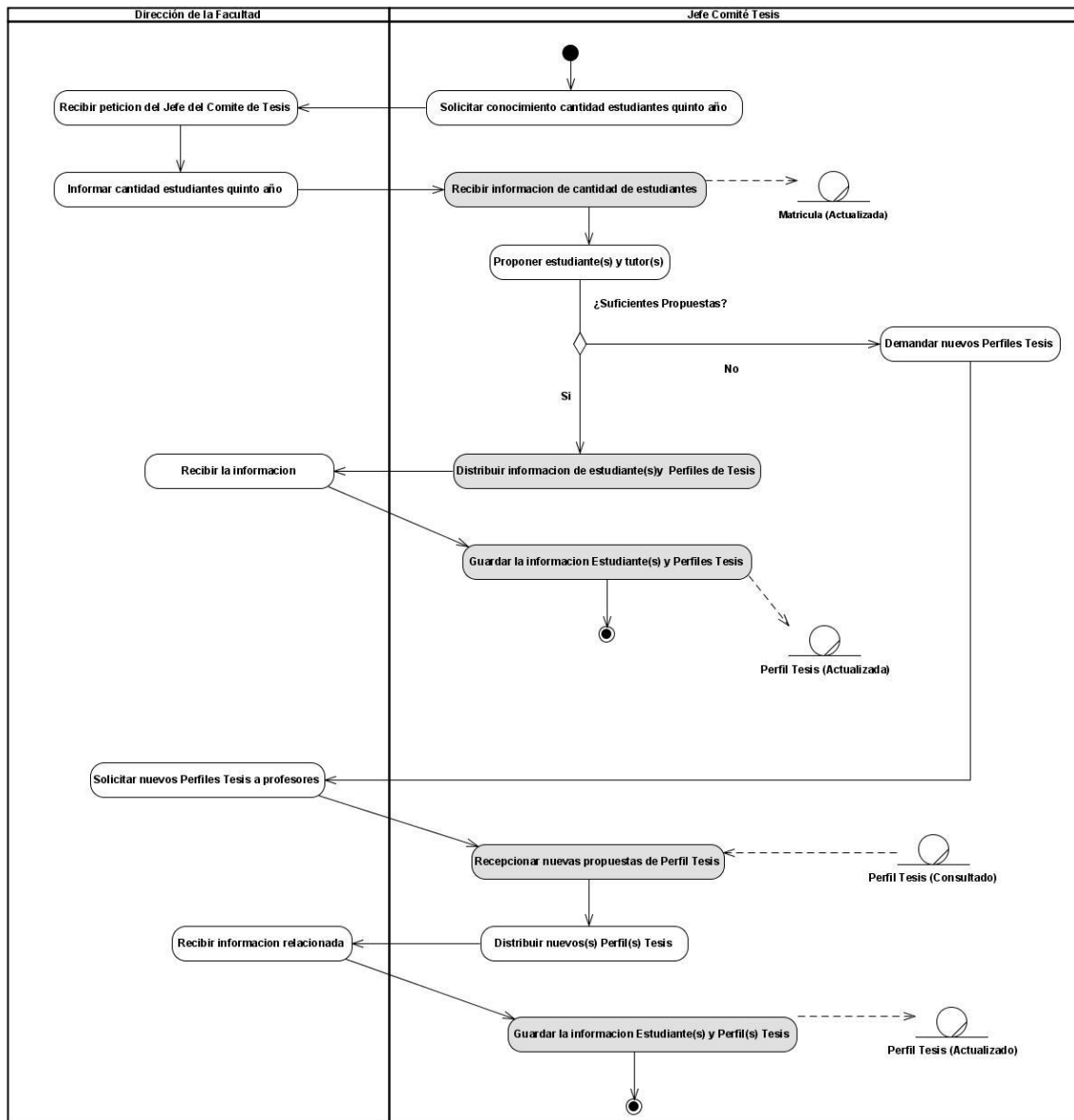


Figura 2.6 Diagrama de Actividad CU: *Asignar Tesis*.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

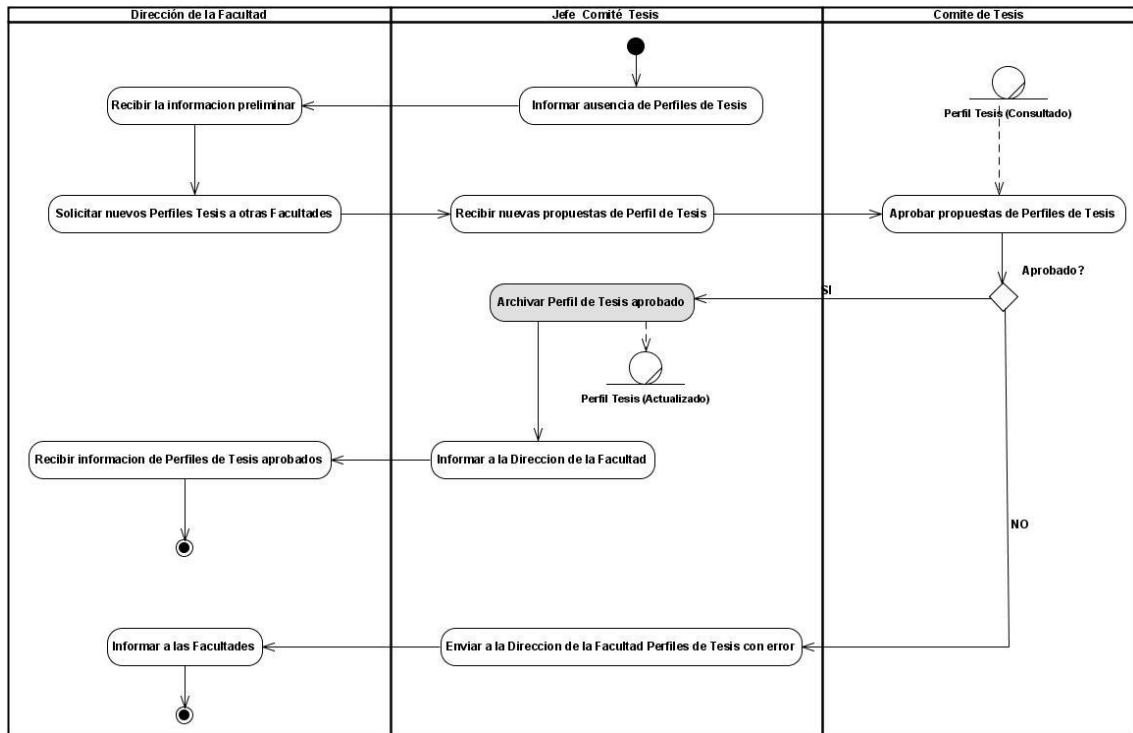


Figura 2.7 Diagrama de Actividad CU: Solicitar Tesis otra Facultad.

2.3.6 Diagrama de clases del modelo de objetos.

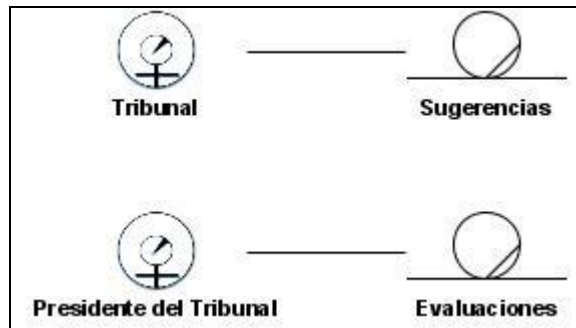


Figura 2.8 Modelo de objetos del caso de uso: Evaluar Tesis.

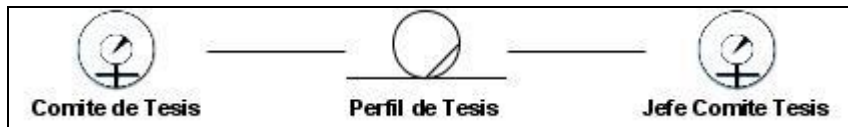


Figura 2.9 Modelo de objetos del caso de uso: Definir Tesis.

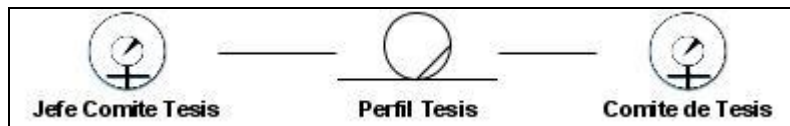


Figura 2.10 Modelo de objetos del caso de uso: Solicitar Tesis a otra Facultad.

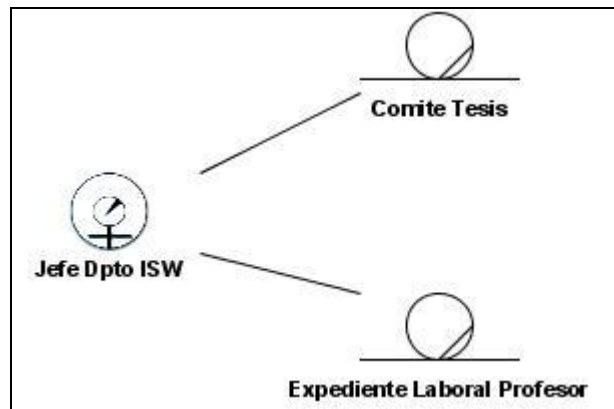


Figura 2.11 Modelo de objetos del caso de uso: *Definir Comité Tesis.*

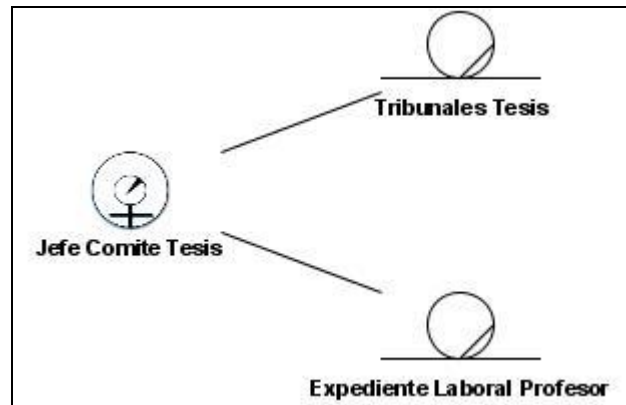


Figura 2.12 Modelo de objetos del caso de uso: *Formar Tribunales.*

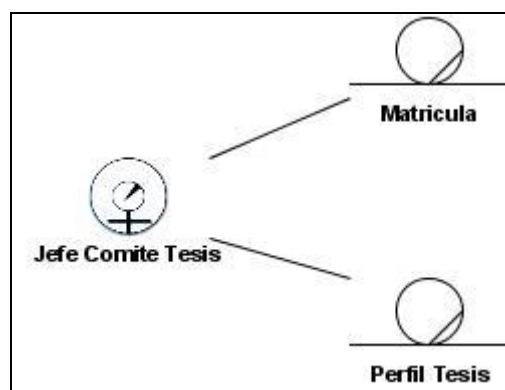


Figura 2.13 Modelo de objetos del caso de uso: *Asignar Tesis.*

2.4 Especificación de los requisitos de software.

2.4.1 Dependencias y relaciones con otro software.

El sistema tiene relación con otros sistemas como es el directorio universitario a través de ldap.dll del cliente PHP que devuelve todos los estudiantes y profesores del dominio UCI. Es importante tener en cuenta que la aplicación tiene la posibilidad de autenticar los usuarios que estén en el dominio UCI, además de brindar privilegios de administrativos al personal requerido para un buen funcionamiento del sistema.

2.4.2 Requerimientos Funcionales.

El sistema deberá ser capaz de:

R1. Administrar usuarios.

Esta información se refiere a:

- Jefe del Dpto. de ISW.
- Presidente del Tribunal de Tesis.
- Jefe del Comité de Tesis.
- Profesor.

R2. Autenticar usuarios.

Esta información se refiere a:

- Todos los usuarios registrados en el dominio UCI.

R3. Actualizar Datos Comité de Tesis.

R3.1 Insertar datos Comité Tesis.

Esta información se refiere a:

- Nombre del jefe de Comité de Tesis.
- Varios miembros.

R3.2 Modificar datos Comité Tesis.

R3.3 Eliminar datos Comité Tesis.

R4. Actualizar Datos Perfil de Tesis.

R4.1 Insertar datos Perfil Tesis.

Esta información se refiere a:

- Curso docente.
- Título.
- Autor.
- Tutor(s).

- Cotutor.
- Clasificación.
- Síntesis del Tutor.
- Problema a resolver.
- Objetivos Generales.
- Objeto de Estudio.
- Campo de Acción.
- Tareas a cumplir por los estudiantes.
- Posibles resultados.

R4.2 Modificar datos Perfil Tesis.

R4.3 Eliminar datos Perfil Tesis.

R5. Actualizar Datos Cortes de Tesis.

R5.1 Registrar evaluaciones.

R5.2 Registrar sugerencias.

R5.3 Registrar señalamientos.

R6. Actualizar Datos Tribunales de Tesis.

R6.1 Insertar datos Tribunales Tesis.

Esta información se refiere a:

- Presidente.
- Secretario.
- Vocal.
- Especialidad.

R6.2 Modificar datos Tribunales Tesis.

R6.3 Eliminar datos Tribunales Tesis.

R7. Asignar Tribunal de Tesis a Perfiles de Tesis.

R7.1 Modificar Asignación.

R7.2 Eliminar Asignación.

R7.3 Generar un calendario de discusión por corte de tesis.

R8. Inscribirse en un Perfil de Tesis.

R9. Asignar Oponente.

R10. Publicar Trabajo de Diploma.

R11. Realizar Búsquedas.

R11.1 Buscar Estudiante.

R11.2 Buscar Comité Tesis.

R11.3 Buscar Tribunal Tesis.

R11.4 Buscar Tribunal de Tesis y Perfil de Tesis.

R11.5 Buscar Corte Tesis.

R11.6 Buscar Perfil de Tesis.

R11.7 Buscar Oponente.

R11.8 Buscar Trabajo de Diploma.

R12. Visualizar Datos.

2.4.3 Requerimientos no funcionales.

Apariencia o interfaz externa.

- El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucho entrenamiento por el usuario.
- Estará diseñado para la resolución deseada por el usuario.
- Empleo de imágenes y colores identificados con el negocio donde se implantará el sistema.

Usabilidad.

- El sistema podrá ser utilizado por personas que tengan un conocimiento mínimo en el manejo de las computadoras.
- El software tendrá siempre visible la opción de Ayuda, lo que posibilitará una mejor explotación por parte de los usuarios de sus funcionalidades.
- El sistema estará disponible las 24 h del día.
- Empleo de barras de progreso u otros para indicar el estado del proceso en cuestión.
- Barra de herramientas de acceso directo con las opciones más usadas.

Rendimiento.

- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 20 segundos para las recuperaciones.

Soporte.

- Se necesita un servidor de bases de datos que soporte volúmenes de datos (PostgreSQL).
- Se elaborará un paquete de instalación.

Portabilidad.

- El sistema será multiplataforma (Linux o Windows).

Seguridad.

- El usuario debe autenticarse antes de entrar al sistema.
- Garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información.

Software.

Para el cliente:

- Utilizar cualquier navegador.
- Sistema operativo Windows 98 o superior, o Linux.

Para el servidor:

- Sistema operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones.
- Un servidor Apache con módulo PHP disponible.
- Un servidor de bases de datos PostgreSQL.

Hardware.

Para el servidor:

- Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM.
- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

Para el cliente:

- Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.
- Tarjeta de red.

Restricciones para el diseño e implementación.

- Utilizar los estándares establecidos de codificación, diseño, entre otros.
- Emplear como servidores web y de bases de datos Apache y PostgreSQL respectivamente.
- Utilizar como lenguaje del lado del servidor al PHP y del lado del cliente el JavaScript.

Políticos-culturales.

- El sistema solo podrá ser utilizado en territorio cubano.
- El producto no debe contener palabras en otros idiomas.
- El producto debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Legales.

El sistema debe ajustarse y regirse por las leyes estipuladas para dar cumplimiento a los procesos que se automatizarán.

Confiabilidad.

- Deben establecerse los mecanismos necesarios para el restablecimiento del sistema ante fallos de comunicación u otros, los tiempos mínimos para ello no deben exceder las 6hrs.
- Deben montarse sistemas de respaldo eléctrico en los locales de los servidores para mantener la vitalidad de los servicios.

Ayuda y documentación en línea.

- Debe contar con la ayuda necesaria para que el usuario final pueda estar informado acerca de las principales funcionalidades que brinda el sistema.
- La ayuda debe ser fácil de entender y utilizar.

2.5 Definición de los casos de uso.

2.5.1 Definición de los actores.

Cada trabajador del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

Tabla 2.9 Actores del Sistema.

Actores	Justificación
Usuario	Representa a todas las personas que pueden acceder al sistema con la posibilidad de visualizar e imprimir todos los reportes y publicar en el sistema según sus permisos.
Jefe Dpto. ISW	Es el encargado de definir el Comité de Tesis de la facultad, y actualizar constantemente cambios en el sistema.
Jefe Comité Tesis	Es el encargado de; primeramente aprobar los perfiles de tesis conjuntamente con el Comité de Tesis y publicar la aprobación. También gestiona y forma los tribunales tomando en cuenta los perfiles correspondientes. Asigna temas de investigación a los estudiantes y

	tribunal correspondiente a cada perfil de tesis.
Presidente Tribunal	Es el encargado de confeccionar las sugerencias y detallar en cada corte de tesis la evaluación correspondiente.
Estudiante	Representa a todos los estudiantes de la Facultad.
Profesor	Representa a todos los profesores de la Facultad.
Administrador	Es el encargado de administrar los privilegios en el sistema de Control de Tesis.
Buscador	Representa a todas las personas que entran en el sistema para buscar cualquier información.

2.5.2 Diagrama de casos de uso.

Vista de gestión del modelo. Paquetes del Sistema.

Después de haber analizado la complejidad del sistema decidimos dividirlo en paquetes con el fin de darle una mayor organización al mismo. La división de los paquetes fundamentales fue hecha teniendo en cuenta los tipos de actores que interactúan con el sistema.

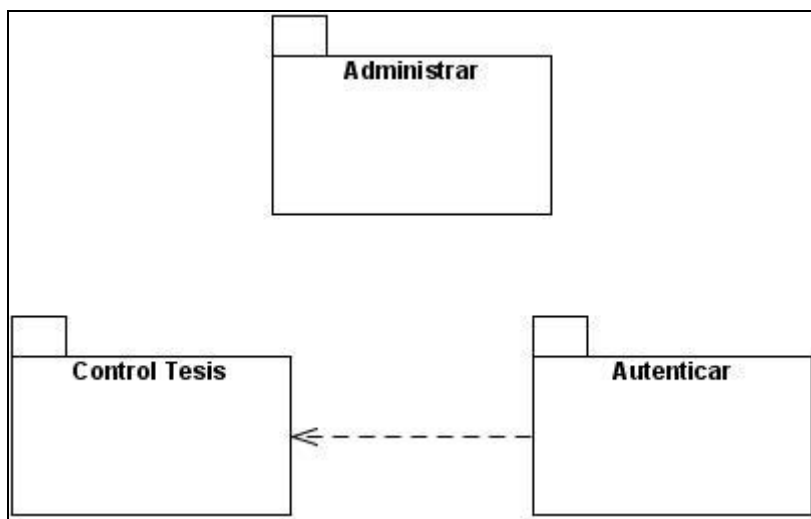


Figura 2.14 Paquetes Fundamentales del Sistema

Casos de Uso.

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

Paquete Administrar.

En este paquete se encuentran los casos de uso que son iniciados por el actor Administrador.

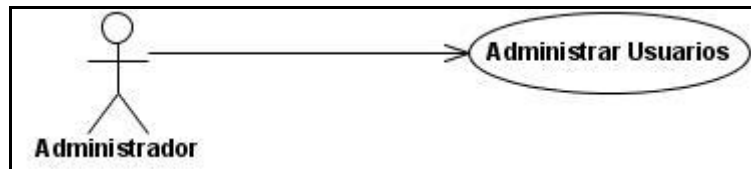


Figura 2.15 Casos de Uso del Paquete Administrar.

Paquete Autenticar.

En este paquete se encuentra los casos de uso que son iniciados el actor Usuario.

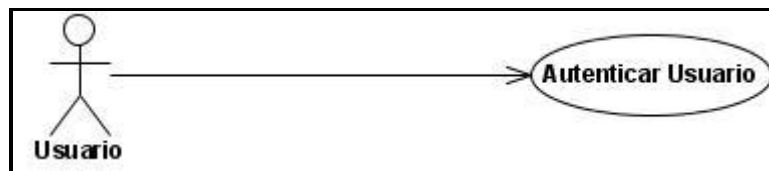


Figura 2.16 Casos de Uso del Paquete Autenticar.

Paquete Control Tesis.

En este paquete se encuentran los casos de uso que tienen que ver con las actualizaciones del sistema.

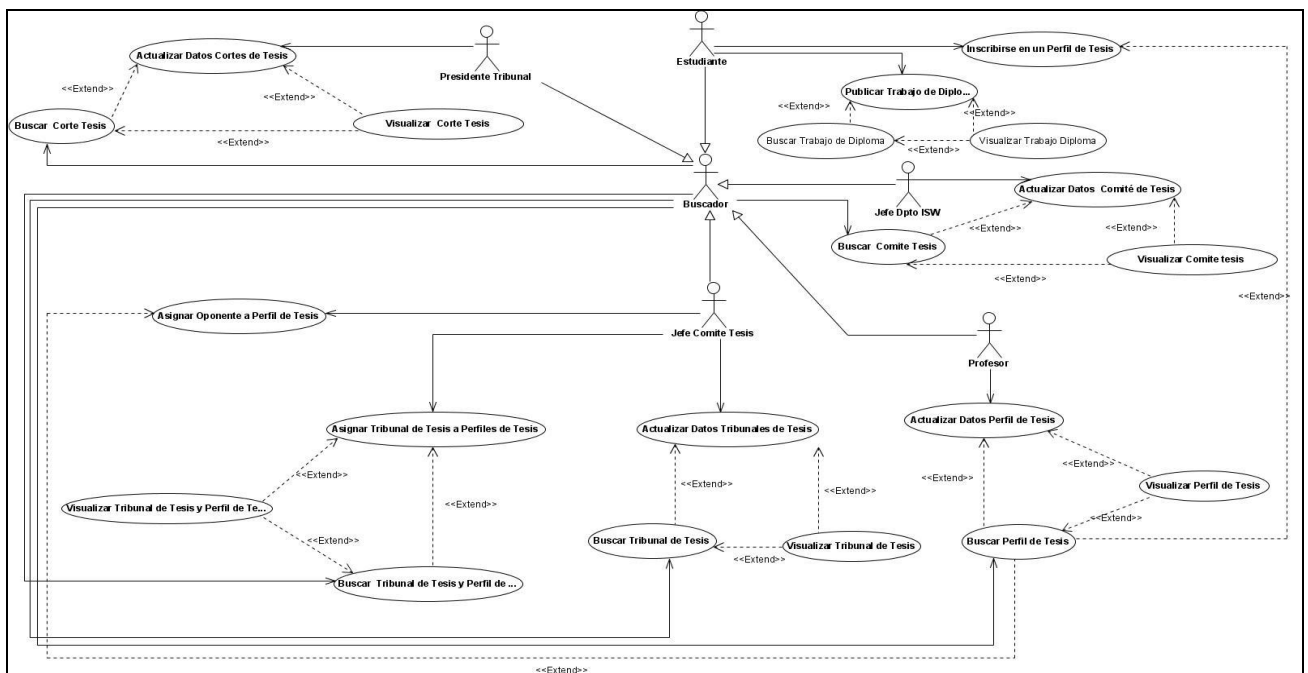


Figura 2.17 Casos de Uso del Paquete Actualización.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.5.3 Casos de uso por ciclo.

Tabla 2.10 Casos de uso por ciclo.

Cód.	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Administrar Usuarios	Administrar	Crítico
CU-2	Autenticar Usuarios	Autenticar	Crítico
CU-3	Actualizar Datos Comité de Tesis	Control Tesis	Crítico
CU-4	Actualizar Datos Perfil de Tesis	Control Tesis	Crítico
CU-5	Actualizar Datos Cortes de Tesis	Control Tesis	Crítico
CU-6	Actualizar Datos Tribunales de Tesis	Control Tesis	Crítico
CU-7	Asignar Tribunal de Tesis a Perfiles de Tesis	Control Tesis	Crítico
CU-8	Inscribirse en un Perfil de Tesis	Control Tesis	Crítico
CU-9	Asignar Oponente	Control Tesis	Crítico
CU-10	Publicar Trabajo de Diploma	Control Tesis	Crítico
CU-11	Buscar Tribunal de Tesis y Perfil de Tesis	Control Tesis	Secundario
CU-12	Buscar Tribunal de Tesis	Control Tesis	Secundario
CU-13	Buscar Perfil de Tesis	Control Tesis	Secundario
CU-14	Buscar Comité Tesis	Control Tesis	Secundario
CU-15	Buscar Corte Tesis	Control Tesis	Secundario
CU-16	Visualizar Comité tesis	Control Tesis	Secundario
CU-17	Visualizar Perfil de Tesis	Control Tesis	Secundario
CU-18	Visualizar Tribunal de Tesis	Control Tesis	Secundario
CU-19	Visualizar Tribunal de Tesis y Perfil de Tesis	Control Tesis	Secundario
CU-20	Visualizar Corte Tesis	Control Tesis	Secundario

2.5.4 Descripción de los Casos de uso.

A continuación se mostrarán las descripciones de algunos de los casos de uso críticos.

Tabla 2.10 Caso de uso del paquete Administrar – Administrar Usuarios.

Caso de Uso	Administrar Usuarios.
Actores	Administrador.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Propósito	Asignar a los usuarios del sistema los lugares.	
Resumen	Esta opción permite al administrador configurar el uso de los usuarios con los lugares definidos en el sistema.	
Responsabilidades	R1	
CU asociados		
Precondiciones	El actor debe haberse autenticado como Administrador.	
Requisitos especiales		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El actor selecciona del menú principal la opción Administrar usuario.	1.1 El sistema muestra la interfaz con los usuarios y lugares definidos para que el acto pueda asociar a cada usuario el lugar o los lugares con lo que puede trabajar	
2. El actor selecciona el usuario y selecciona el lugar o los lugares a los que tiene acceso y oprime el botón Agregar.	2.1 El sistema agrega el lugar o los lugares al usuario seleccionado.	
Sección Eliminar		
3. El actor selecciona del grid el usuario acompañado del lugar y oprime el botón "Eliminar".	3.1 El sistema envía un mensaje de que si está seguro que desea eliminar ese registro.	
4. El actor selecciona "Aceptar".	4.1 El sistema elimina el registro.	
Cursos Alternos		
Acción del Actor	Respuesta del Sistema	
4. El actor selecciona cancelar.	4.1 El sistema no realiza ninguna acción.	
Prioridad	Crítico.	

Tabla 2.11 Caso de uso del paquete Autenticar – Autenticar Usuario.

Caso de Uso	Autenticar Usuario.
Actores	Usuario.
Propósito	Autenticar usuarios en el sistema.
Resumen	El caso de uso inicia cuando el usuario desea entrar al

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	sistema.
Responsabilidades	R2
CU asociados	
Precondiciones	El actor debe ser parte del dominio UCI.
Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor introduce usuario y contraseña.	1.1 El sistema devuelve el usuario y la contraseña. 1.2 El sistema verifica los datos introducidos, si son correctos, el sistema envía mensaje de “Acceso permitido”.
Cursos Alternos	
Acción del Actor	Respuesta del Sistema
	1.2 Si los datos son incorrectos el sistema envía un mensaje de “Acceso denegado”
2. El actor introduce usuario y contraseña nuevamente.	2.1 El sistema sigue su ejecución a partir de la línea 1.1 del caso de uso.
Prioridad	Crítico.

Tabla 2.12 Caso de uso del paquete Control Tesis – Actualizar Datos de Comité Tesis.

Caso de Uso	Actualizar Datos de Comité Tesis.
Actores	Jefe Dpto. ISW.
Propósito	Permitir Insertar, Modificar y Eliminar datos de Comité de Tesis.
Resumen	El caso de uso inicia cuando el Jefe el Dpto. de ISW entra en el sistema con el objetivo de actualizar información.
Responsabilidades	R3: (R3.1 al R3.3)
CU asociados	El CU Buscar Comité Tesis es un Extend. El CU Visualizar Comité Tesis es un Extend.
Precondiciones	El actor debe haberse autenticado como Jefe del Dpto. de ISW.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Jefe Dpto. ISW Inserta, Modifica y Elimina datos del Comité de Tesis.	El sistema ejecuta alguna de las siguientes acciones: <ul style="list-style-type: none"> a) Para Insertar un Comité Tesis, ir a la sección "Adicionar Comité Tesis" b) Para Modificar datos, ir a la sección "Modificar". c) Para Eliminar datos, ir a la sección "Eliminar".
Sección "Adicionar Comité Tesis"	
Acción del Actor	Respuesta del Sistema
2. El Jefe Dpto. ISW entra los datos del Comité de Tesis para realizar su registro en la aplicación.	2.1 El sistema verifica que los campos del Nombre del jefe de Comité de Tesis, Varios Miembros, estén llenos. 2.2 El Comité Tesis se almacena en el sistema. 2.3 Se muestra un mensaje informándole al Jefe Dpto. ISW que ya ha sido efectuado el registro del Comité Tesis y finaliza el caso de uso.
Curso Alternativo	
	2.1 Se emite un mensaje para que llene los campos obligatorios. 2.2 Si el Comité Tesis existe se muestra un mensaje informativo y finaliza el caso de uso.
Sección " Modificar "	
Acción del Actor	Respuesta del Sistema
2. El Jefe Dpto. ISW selecciona el Comité Tesis para modificar	2.1 El sistema brinda la posibilidad de modificar los datos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

datos.	
3. El Jefe Dpto. ISW realiza las actualizaciones deseadas.	3.1 Se verifica que los campos obligatorios estén llenos. 3.2 Se actualiza la Información y finaliza el caso de uso.
Curso Alternativo	
	3.1 Se emite un mensaje para que llene los campos obligatorios.
Sección “Eliminar”	
Acción del Actor	Respuesta del Sistema
2. El Jefe Dpto. ISW selecciona en del grid el Comité Tesis a eliminar.	2.2 El sistema pide confirmación. 2.3 El sistema elimina el documento Comité Tesis.
Prioridad	Crítico.

Tabla 2.13 Caso de uso del paquete Control Tesis – Actualizar Datos de Perfil de Tesis.

Caso de Uso	Actualizar Datos Perfil de Tesis.
Actores	Profesor.
Propósito	Permitir Insertar, Modificar y Eliminar datos de Perfiles de Tesis.
Resumen	El caso de uso inicia cuando el Profesor entra en el sistema con el objetivo de actualizar información.
Responsabilidades	R4: (R4.1 al R4.3)
CU asociados	El CU Buscar Perfil de Tesis es un Extend. El CU Visualizar Perfil de Tesis es un Extend.
Precondiciones	El actor debe haberse autenticado como Profesor.
Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Profesor, Modifica y Elimina datos de los Perfiles de Tesis.	El sistema ejecuta alguna de las siguientes acciones: a) Para Insertar un Perfil de Tesis, ir a la

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<p>sección "Adicionar Perfil de Tesis"</p> <p>b) Para Modificar datos, ir a la sección "Modificar".</p> <p>c) Para Eliminar datos, ir a la sección "Eliminar".</p>
Sección "Adicionar Perfil de Tesis"	
Acción del Actor	Respuesta del Sistema
2. El Profesor entra los datos del Perfil de Tesis para realizar su registro en la aplicación.	<p>2.1 El sistema verifica que los campos del Curso docente, Título, Autor, Tutor(s), Cotutor, Clasificación, Síntesis del Tutor, Problema a resolver, Objetivos Generales, Objeto de Estudio, Campo de Acción, Tareas a cumplir por los estudiantes, Posibles resultados, estén llenos. El Perfil de Tesis se almacena en el sistema.</p> <p>2.2 Se muestra un mensaje informándole al Profesor que ya ha sido efectuado el registro del Perfil de Tesis y finaliza el caso de uso.</p>
Curso Alternativo	
	<p>2.3 Se emite un mensaje para que llene los campos obligatorios: Curso docente, Título, Tutor(s), Cotutor, Clasificación, Síntesis del Tutor, Problema a resolver, Objetivos Generales, Objeto de Estudio, Campo de Acción, Tareas a cumplir por los estudiantes, Posibles resultados.</p> <p>2.4 Si el Perfil de Tesis existe se muestra un mensaje informativo y finaliza el caso de uso.</p>
Sección " Modificar "	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2. El Profesor selecciona el Perfil de Tesis para modificar datos.	2.1 El sistema brinda la posibilidad de modificar los datos.
3. El Profesor realiza las actualizaciones deseadas.	3.1 Se verifica que los campos obligatorios estén llenos. 3.2 Se actualiza la Información y finaliza el caso de uso.
Curso Alternativo	
	3.1 Se emite un mensaje para que llene los campos obligatorios.
Sección "Eliminar"	
Acción del Actor	Respuesta del Sistema
2. El Profesor selecciona del grid el Perfil de Tesis a eliminar.	2.1 El sistema pide confirmación. 2.2 El sistema elimina el documento Perfil de Tesis.
Prioridad	Crítico.

Tabla 2.14 Caso de uso del paquete Control Tesis – Actualizar Datos Cortes de Tesis.

Caso de Uso	Actualizar Datos Cortes de Tesis.
Actores	Presidente Tribunal.
Propósito	Permitir registrar Sugerencias, Señalamientos y Evaluaciones en cada Corte de Tesis.
Resumen	El caso de uso inicia cuando el Presidente Tribunal entra en el sistema con el objetivo de actualizar Corte de Tesis.
Responsabilidades	R5: (R5.1, R5.2, R5.3)
CU asociados	El CU Buscar Corte Tesis es un Extend. El CU Visualizar Corte Tesis es un Extend.
Precondiciones	El actor debe haberse autenticado como Presidente Tribunal.
Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Presidente Tribunal registra Sugerencias, Señalamientos y	1.1 El sistema muestra un formulario con los datos necesarios para el registro.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Evaluaciones del Corte de Tesis.	Sugerencias, Señalamientos y Evaluaciones del Corte de Tesis.
2.El Presidente Tribunal entra las sugerencias y evaluación a un Corte de Tesis para realizar su registro en la aplicación.	<p>2.1 El sistema verifica que el campo donde se registra la evaluación, esté lleno. Por consiguiente no es obligatorio hacer sugerencias si el Trabajo es magistral.</p> <p>2.2 El Corte de Tesis se almacena en el sistema.</p> <p>2.3 Se muestra un mensaje informándole al Presidente Tribunal que ya ha sido efectuado el registro del Corte de Tesis y finaliza el caso de uso.</p>
Curso Alternativo	
	<p>2.1 Se emite un mensaje para que llene los campos obligatorios, Evaluaciones del Corte de Tesis.</p> <p>2.2 Si el Corte de Tesis existe se muestra un mensaje informativo y finaliza el caso de uso.</p>
Prioridad	Crítico.

Tabla 2.15 Caso de uso del paquete Control Tesis – Actualizar Datos Tribunales de Tesis.

Caso de Uso	Actualizar Datos de Tribunales de Tesis.
Actores	Jefe Comité Tesis.
Propósito	Permitir Insertar, Modificar y Eliminar datos de Tribunales de Tesis.
Resumen	El caso de uso inicia cuando el Jefe Comité Tesis entra en el sistema con el objetivo de actualizar información.
Responsabilidades	R6: (R6.1 al R6.3)
CU asociados	El CU Buscar Tribunales de Tesis es un Extend. El CU Visualizar Tribunales de Tesis es un Extend.
Precondiciones	El actor debe haberse autenticado como Jefe Comité Tesis.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.El Jefe Comité Tesis Inserta, Modifica y Elimina datos de Tribunales de Tesis.	El sistema ejecuta alguna de las siguientes acciones: <ul style="list-style-type: none"> a) Para Insertar un Tribunales de Tesis, ir a la sección "Adicionar Tribunales de Tesis" b) Para Modificar datos, ir a la sección " Modificar". c) Para Eliminar datos, ir a la sección "Eliminar".
Sección "Adicionar Comité Tesis"	
Acción del Actor	Respuesta del Sistema
2.El Jefe Comité Tesis entra los datos del Tribunal de Tesis para realizar su registro en la aplicación.	2.1 El sistema verifica que los campos del Presidente, Secretario, Vocal, Especialidad, estén llenos. 2.2 El Tribunal de Tesis se almacena en el sistema. 2.3 Se muestra un mensaje informándole al Jefe Comité Tesis que ya ha sido efectuado el registro del Tribunal de Tesis y finaliza el caso de uso.
Curso Alternativo	
	2.2 Se emite un mensaje para que llene los campos obligatorios. 2.3 Si el Tribunal de Tesis existe se muestra un mensaje informativo y finaliza el caso de uso.
Sección " Modificar "	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2. El Jefe Comité Tesis selecciona el Tribunal de Tesis para modificar datos.	2.1 El sistema brinda la posibilidad de modificar los datos.
3. El Jefe Comité Tesis realiza las actualizaciones deseadas.	3.1 Se verifica que los campos obligatorios estén llenos. 3.2 Se actualiza la Información y finaliza el caso de uso.
Curso Alternativo	
	3.1 Se emite un mensaje para que llene los campos obligatorios.
Sección "Eliminar"	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis selecciona del grid el Tribunal de Tesis a eliminar.	2.1 El sistema pide confirmación. 2.2 El sistema elimina el documento Tribunal de Tesis.
Prioridad	Crítico.

Tabla 2.16 Caso de uso del paquete Control Tesis – Asignar Tribunal de Tesis a Perfiles de Tesis.

Caso de Uso	Asignar Tribunal de Tesis a Perfiles de Tesis.
Actores	Jefe Comité Tesis.
Propósito	Permitir asignar a varios Perfiles de Tesis un Tribunal de Tesis según su clasificación o polo de investigación.
Resumen	El caso de uso inicia cuando el Jefe Comité Tesis entra en el sistema con el objetivo de asignar Tribunal de Tesis a Perfiles de Tesis.
Responsabilidades	R7: (R7.1, R7.2, R7.3)
CU asociados	El CU Buscar Tribunal de Tesis y Perfiles de Tesis es un Extend. El CU Visualizar Tribunal de Tesis y Perfiles de Tesis es un Extend.
Precondiciones	El actor debe haberse autenticado como Jefe Comité Tesis.
Requisitos especiales	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Jefe Comité Tesis Asigna, Modifica y Elimina Tribunales de Tesis a Perfiles de Tesis.	<p>El sistema ejecuta alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> a) Para Asignar un Tribunal de Tesis a Perfiles de Tesis, ir a la sección "Asignar Tribunal" b) Para Modificar datos, ir a la sección "Modificar". c) Para Eliminar datos, ir a la sección "Eliminar". d) Generar un calendario de discusión por corte de tesis, ir a la sección "Generar Calendario de Corte".
Sección "Asignar Tribunal"	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis asigna Tribunales a Perfiles de Tesis.	<p>2.1 El sistema verifica que los campos obligatorios estén llenos.</p> <p>2.2 Se muestra un mensaje informándole al Jefe Comité Tesis que ya ha sido efectuado el registro del Tribunal de Tesis y Perfiles de Tesis y finaliza el caso de uso.</p>
Curso Alternativo	
	2.1 Se emite un mensaje para que llene los campos obligatorios.
Sección " Modificar "	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis selecciona el Perfil de Tesis o Tribunal de Tesis para modificar datos.	2.1 El sistema brinda la posibilidad de modificar los datos.
3. El Jefe Comité Tesis determina	3.1 Se verifica que los campos obligatorios

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tribunal de Tesis a Perfiles de Tesis nuevamente o viceversa las actualizaciones deseadas.	estén llenos. 3.2 Se actualiza la Información y finaliza el caso de uso.
Curso Alternativo	
	3.1 Se emite un mensaje para que llene los campos obligatorios.
Sección “Eliminar”	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis selecciona del grid el Perfil de Tesis a eliminar de un Tribunal de Tesis o viceversa.	2.1 El sistema pide confirmación. 2.2 El sistema elimina el archivo deseado.
Sección “Generar Calendario de Corte”	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis selecciona “Generar Calendario de Corte” en el sistema.	2.1 El sistema le Brinda la posibilidad de escoger un rango de días y horarios determinados para los cortes, lo cual es a nivel central en la Universidad.
3. El jefe Comité Tesis escoge el rango de días del corte de tesis y oprime generar.	3.1 El sistema genera un calendario de cortes en correspondencia del rango de días y de horarios escogidos por el Jefe Comité Tesis.
Prioridad	Crítico.

Tabla 2.17 Caso de uso del paquete Control Tesis – Inscribirse en un Perfil de Tesis.

Caso de Uso	Inscribirse en un Perfil de Tesis.
Actores	Estudiante.
Propósito	Inscribirse en un perfil de tesis determinado.
Resumen	El caso de uso inicia cuando el usuario entra en el sistema para inscribirse en un Perfil de Tesis.
Responsabilidades	R8
CU asociados	
Precondiciones	El actor debe haberse autenticado como Estudiante.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario accede al sistema insertando los datos de usuario y contraseña.	1.1 El sistema verifica que los datos sean válidos. 1.2 El sistema asigna los permisos en caso que los datos sean verdaderos.
2. El estudiante accede a la Sección Perfiles Libres y se inscribe en el que más importe.	2.1 El sistema guarda su petición.
Cursos Alternos	
Acción del Actor	Respuesta del Sistema
	1.1 Si los datos son incorrectos el sistema devuelve el mensaje “Disculpe. No se reconoce el nombre del usuario o la contraseña”.
Prioridad	Crítico.

Tabla 2.18 Caso de uso del paquete Control Tesis – Asignar Oponente a Perfil de Tesis.

Caso de Uso	Asignar Oponente a Perfil de Tesis.
Actores	Jefe Comité Tesis.
Propósito	Permitir Asignar, Modificar y Eliminar datos de oponente a un Perfil de Tesis.
Resumen	El caso de uso inicia cuando el Jefe Comité Tesis entra en el sistema con el objetivo de actualizar información.
Responsabilidades	R9
CU asociados	El CU Buscar Perfil Tesis es un Extend.
Precondiciones	El actor debe haberse autenticado como Jefe Comité Tesis.
Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

1. El Jefe Comité Tesis Asigna, Modifica y Elimina datos del Oponente a un Perfil de Tesis.	El sistema ejecuta alguna de las siguientes acciones: d) Para Asignar un Oponente, ir a la sección "Asignar Oponente" e) Para Modificar datos, ir a la sección "Modificar". f) Para Eliminar datos, ir a la sección "Eliminar".
Sección "Asignar Oponente"	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis Busca Perfiles de Tesis por Clasificación.	2.1 El sistema muestra el resultado de la Búsqueda.
3. El Jefe Comité Tesis asigna Oponentes según la Clasificación.	3.1 El Oponente se almacena en el sistema. 3.2 Se muestra un mensaje informándole al Jefe Comité Tesis que ya ha sido efectuada la asignación del Oponente y finaliza el caso de uso.
Curso Alternativo	
	3.1 Se emite un mensaje para que llene los campos obligatorios. 3.2 Si el Oponente en ese Perfil de Tesis existe se muestra un mensaje informativo y finaliza el caso de uso.
Sección " Modificar "	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis modifica Oponentes según la Clasificación.	2.1 El sistema brinda la posibilidad de modificar los datos.
3. El Jefe Comité Tesis realiza las actualizaciones deseadas.	3.1 Se verifica que los campos obligatorios estén llenos. 3.2 Se actualiza la Información y finaliza el

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	caso de uso.
Curso Alternativo	
	3.1 Se emite un mensaje para que llene los campos obligatorios.
Sección “Eliminar”	
Acción del Actor	Respuesta del Sistema
2. El Jefe Comité Tesis selecciona del grid el Oponente a eliminar.	2.1 El sistema pide confirmación. 2.2 El sistema elimina el documento Oponente.
Prioridad	Crítico.

Tabla 2.19 Caso de uso del paquete Control Tesis – Publicar Trabajo de Diploma.

Caso de Uso	Publicar Trabajo de Diploma.
Actores	Estudiante.
Propósito	Permitir publicar el Trabajo de Diploma ya acabado.
Resumen	El caso de uso inicia cuando el Estudiante entra en el sistema con el objetivo de publicar el documento de tesis.
Responsabilidades	R10
CU asociados	
Precondiciones	El actor debe haberse autenticado como Estudiante.
Requisitos especiales	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario accede al sistema insertando los datos de usuario y contraseña.	1.1 El sistema verifica que los datos sean válidos. 1.2 El sistema asigna los permisos en caso que los datos sean verdaderos.
2. El Estudiante accede a la Sección “Publicar” y registra su Trabajo de Diploma.	2.1 El sistema guarda el Trabajo de Diploma.
Prioridad	Crítico.

2.6 Conclusiones.

En este capítulo se realizó una descripción de los procesos que actualmente se llevan a cabo en la facultad, detallando el flujo de las actividades realizadas en dicho entorno de trabajo, ya sea con la descripción textual de los procesos como con los diagramas de actividades en cada caso, así como el modelado de los objetos que se utilizan. Se esclarecieron los requisitos necesarios para que el sistema funcione así como el diagrama y la descripción textual de los casos de uso que satisfacen al sistema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción.

El presente capítulo describe cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requisitos), con el objetivo de transformar los requisitos en el diseño del futuro sistema, así como una breve descripción de la arquitectura del mismo. Se representan las clases de análisis y diseño del sistema, el diseño de la base de datos conformada para el sistema y la definición del diseño a aplicar.

3.2 Descripción de la arquitectura.

La arquitectura de software es el conjunto de decisiones significativas sobre la organización de un sistema, la selección de los elementos estructurales y sus interfaces de los cuales el sistema está compuesto junto con su comportamiento. Describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.(18)

3.2.1 Estilo Arquitectónico: Arquitectura de 3 Capas.

Un estilo arquitectónico o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción. Los sistemas empresariales distribuidos pueden agrupar los siguientes estilos arquitectónicos:

- Modelo-Vista-Controlador (MVC).
- Arquitectura en Capas.
- Arquitectura Orientada a Objetos.
- Arquitectura Basada en Componentes.
- Arquitectura Orientada a Servicios.

La propuesta de diseño de una aplicación Web para la automatización del control de los trabajos de diploma en la UCI se desarrollará sobre la arquitectura de tres capas o niveles como también se le puede llamar.

Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma tal que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.(19)



Figura 3.1 Arquitectura de tres capas.

A continuación se describen las capas por la que está compuesta esta arquitectura:

Capa de Presentación: Esta capa contiene las clases interfaces de la aplicación encargadas de interactuar con el usuario y gestionar todas las peticiones que el mismo realice.

Capa Lógica de la Aplicación: Esta capa es básicamente el código al que recurre la capa de presentación para recuperar los datos deseados, es la que contiene toda la lógica del negocio, es decir, las clases controladoras del negocio que modelan las reglas del mismo.

Capa de Datos: Esta capa contiene los datos necesarios para la aplicación, aquí es donde se encuentra la base de datos física.

3.3 Análisis.

El análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales.

3.3.1 Diagrama de Clases de Análisis.

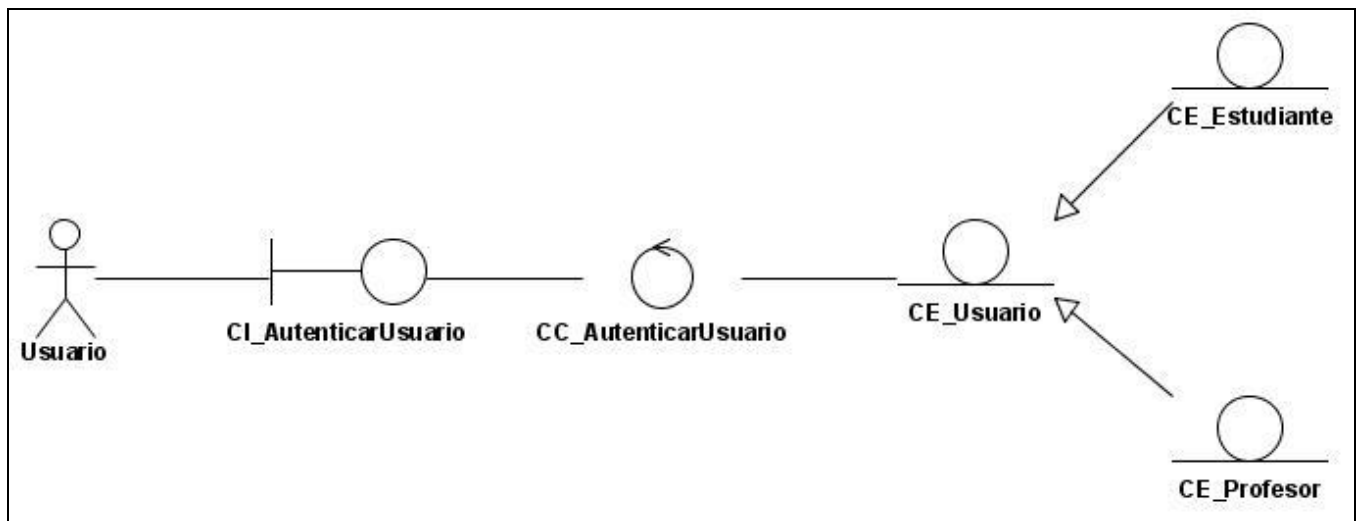


Figura 3.2 CU Autenticar Usuario.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

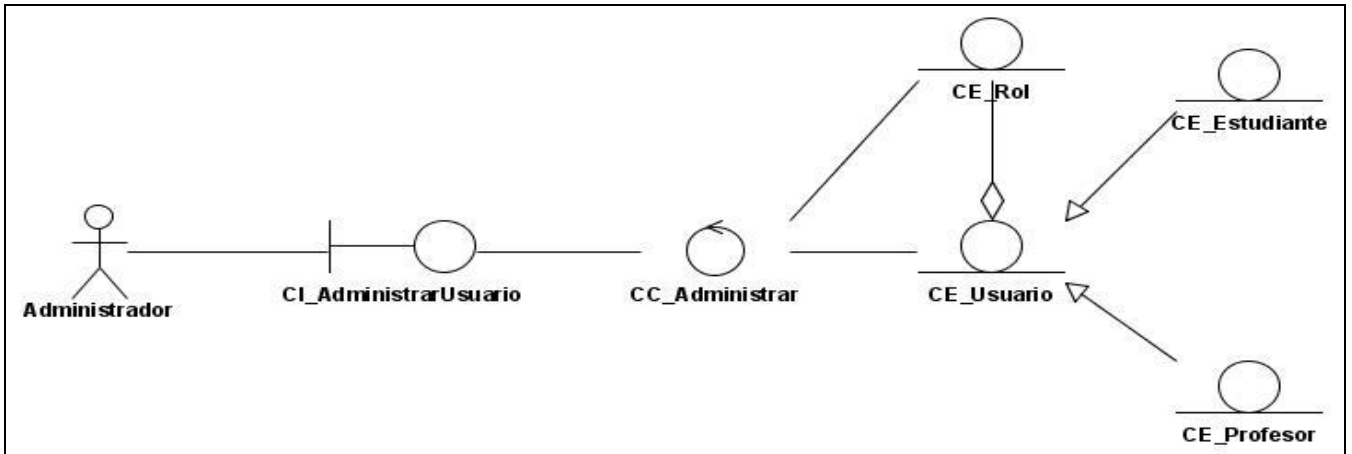


Figura 3.3 CU Administrar Usuario.

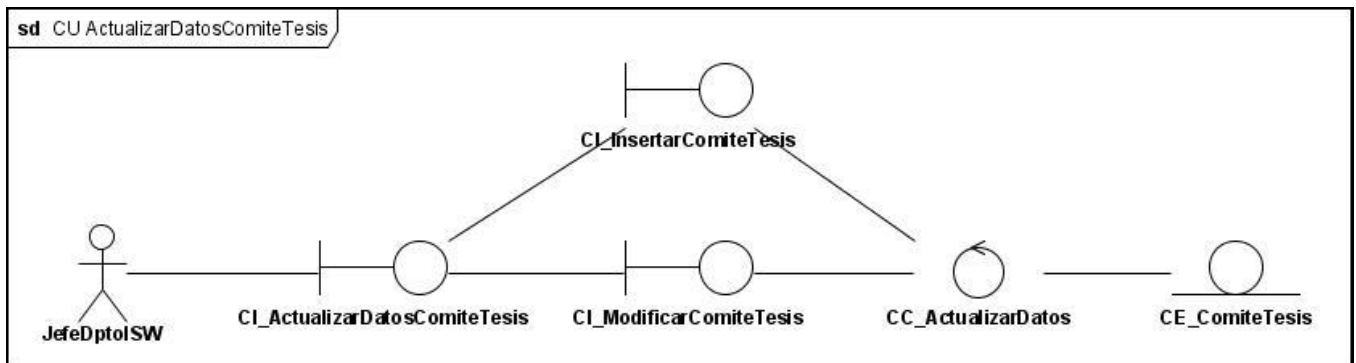


Figura 3.4 CU Actualizar Datos Comité de Tesis.

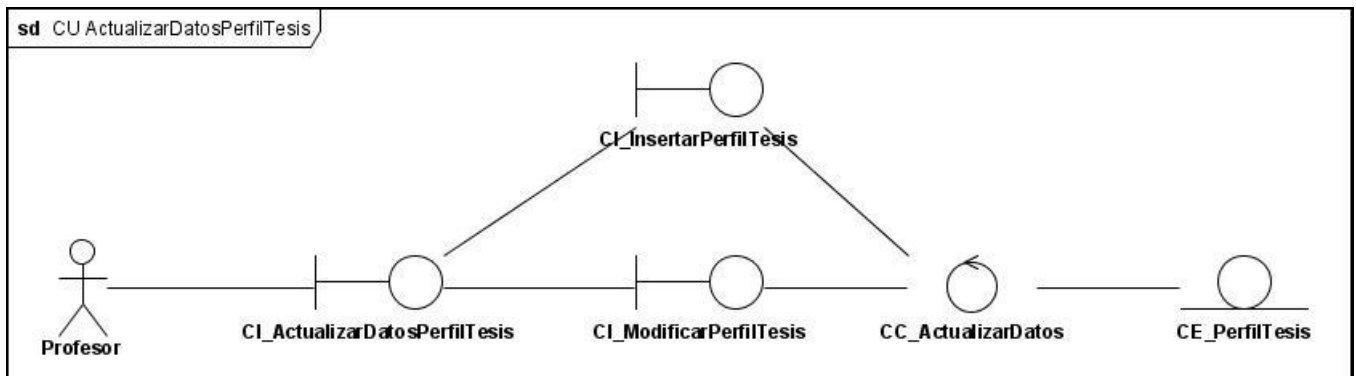


Figura 3.5 CU Actualizar Datos Perfil de Tesis.

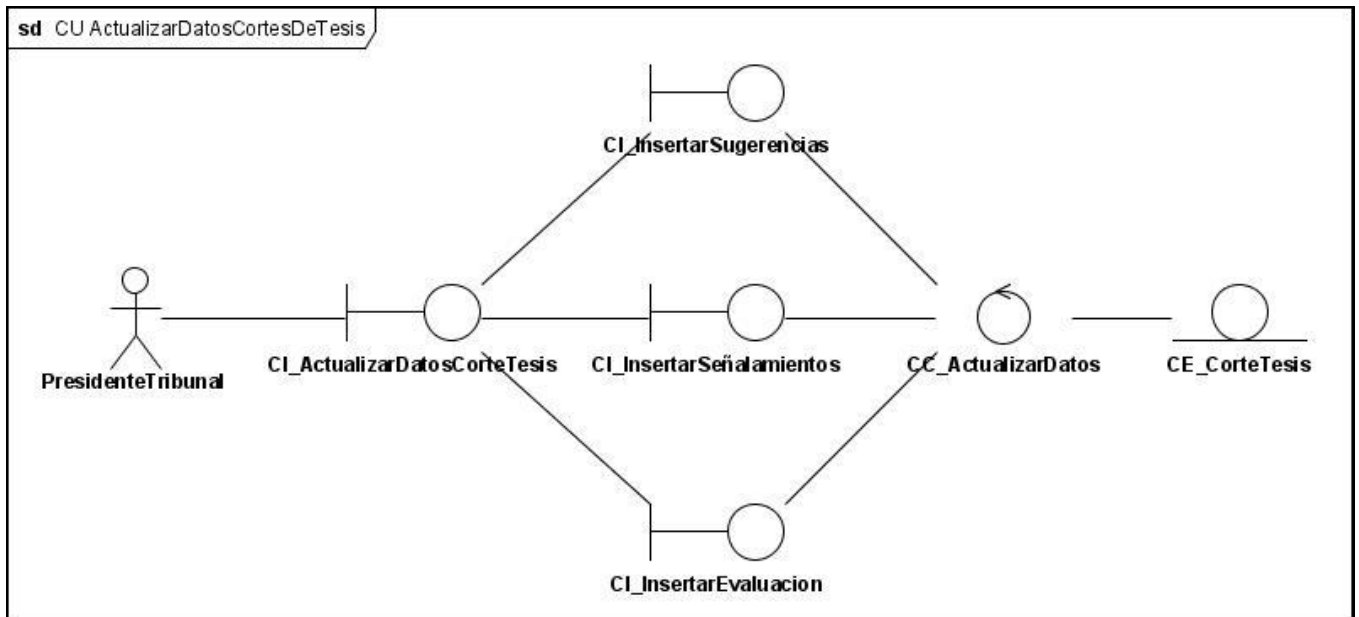


Figura 3.6 CU Actualizar Datos Corte de Tesis.

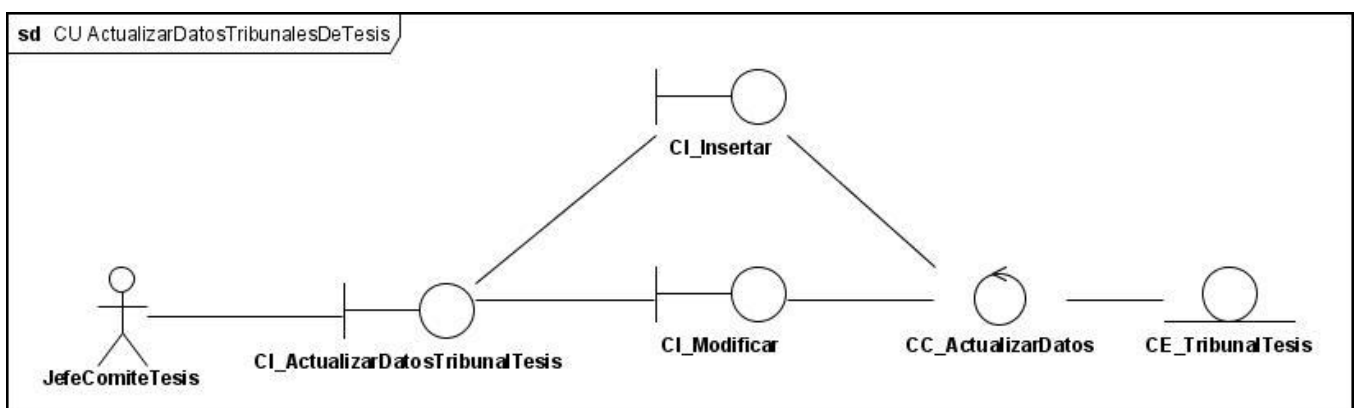
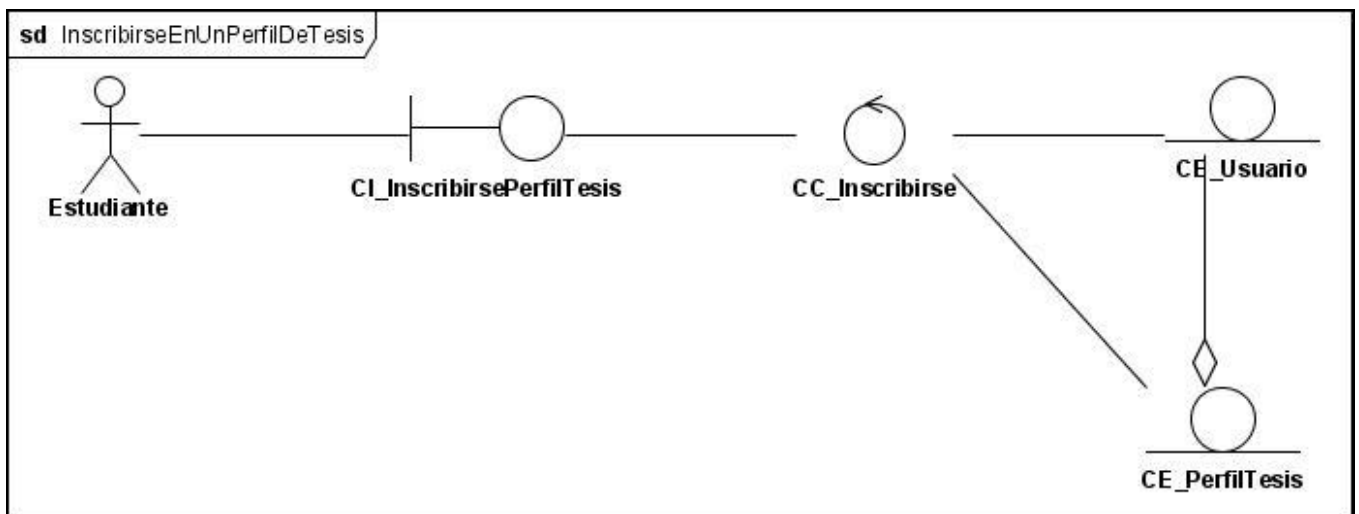


Figura 3.7 CU Actualizar Datos Tribunales de Tesis.



CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Figura 3.8 CU Inscribirse en un Perfil de Tesis.

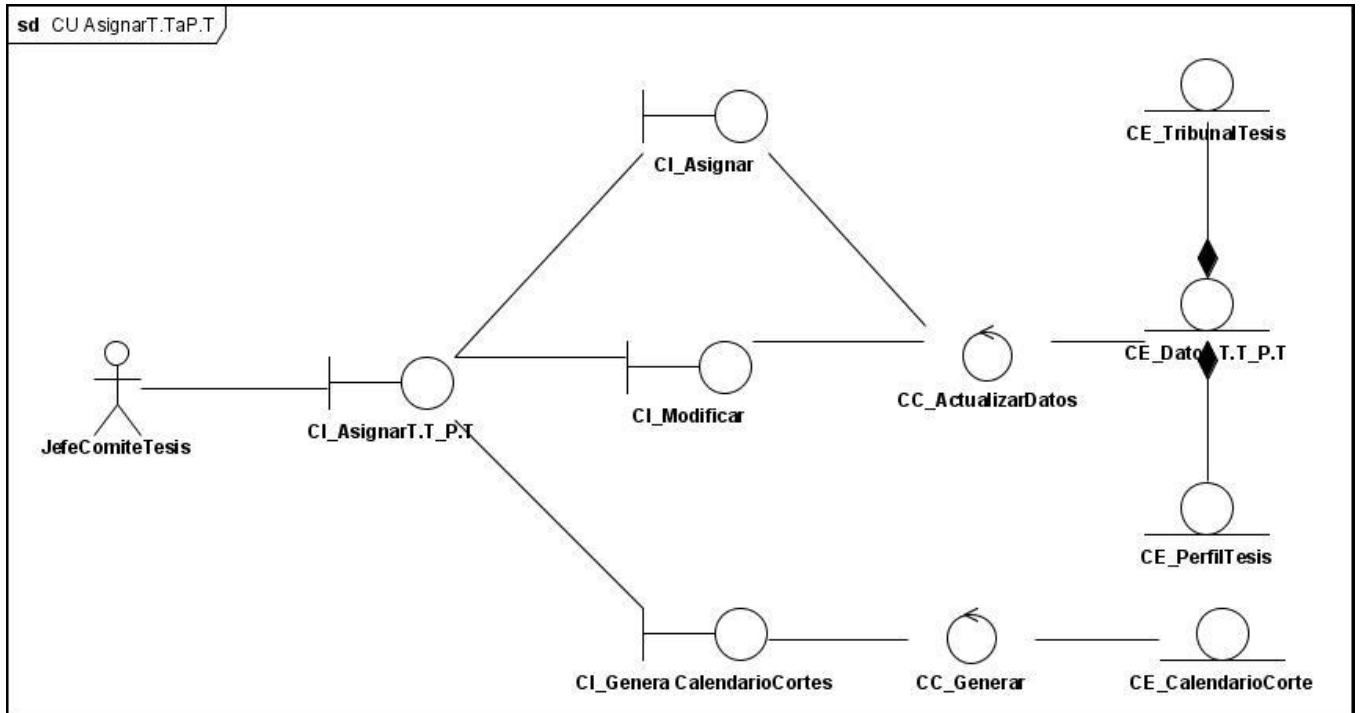


Figura 3.9 CU Asignar T.T a P.T.

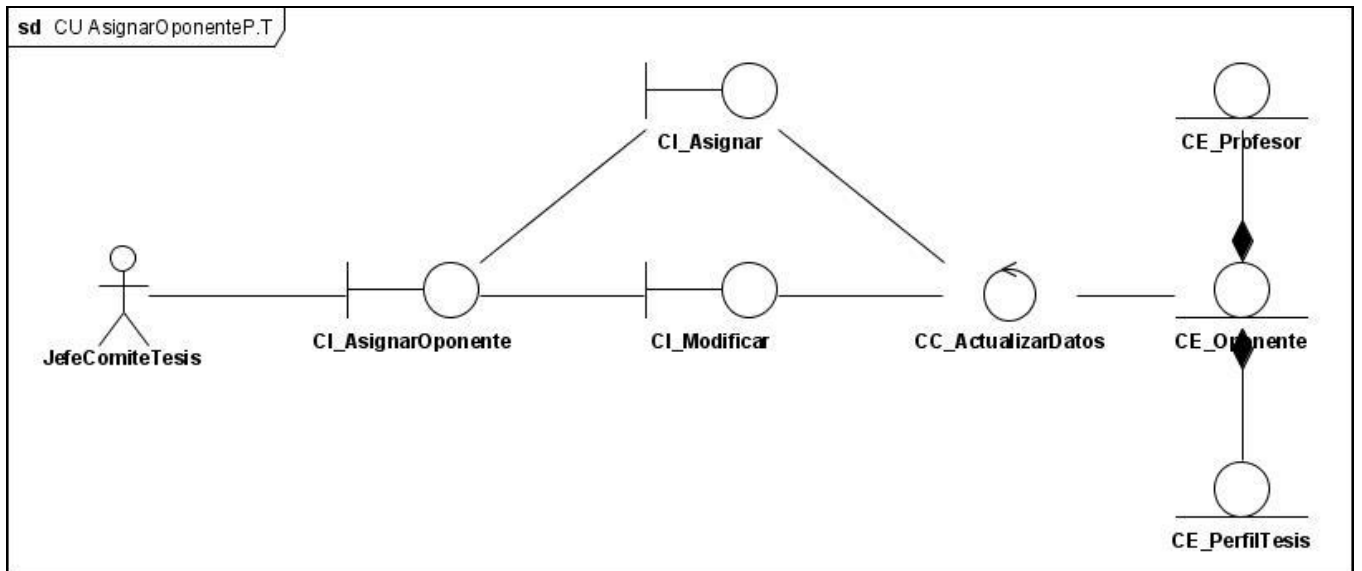


Figura 3.10 CU Asignar Oponente

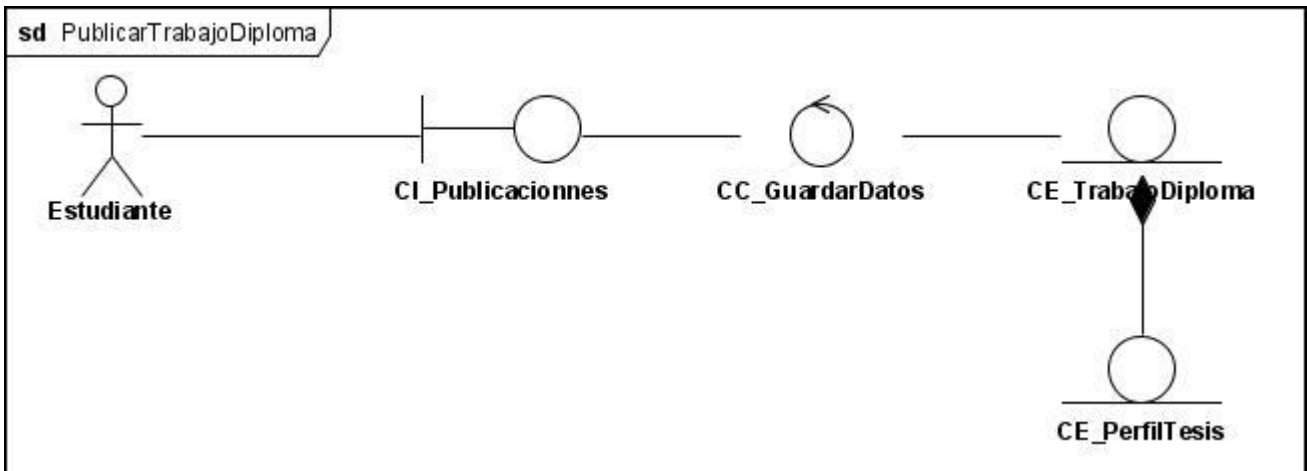


Figura 3.11 CU Publicar Trabajo de Diploma.

3.3.2 Diagramas de Interacción (Colaboración).

Se realizaron diagramas de colaboración por escenarios del caso de uso. En la documentación de este trabajo se muestran solamente los diagramas de colaboración de tres casos de uso.

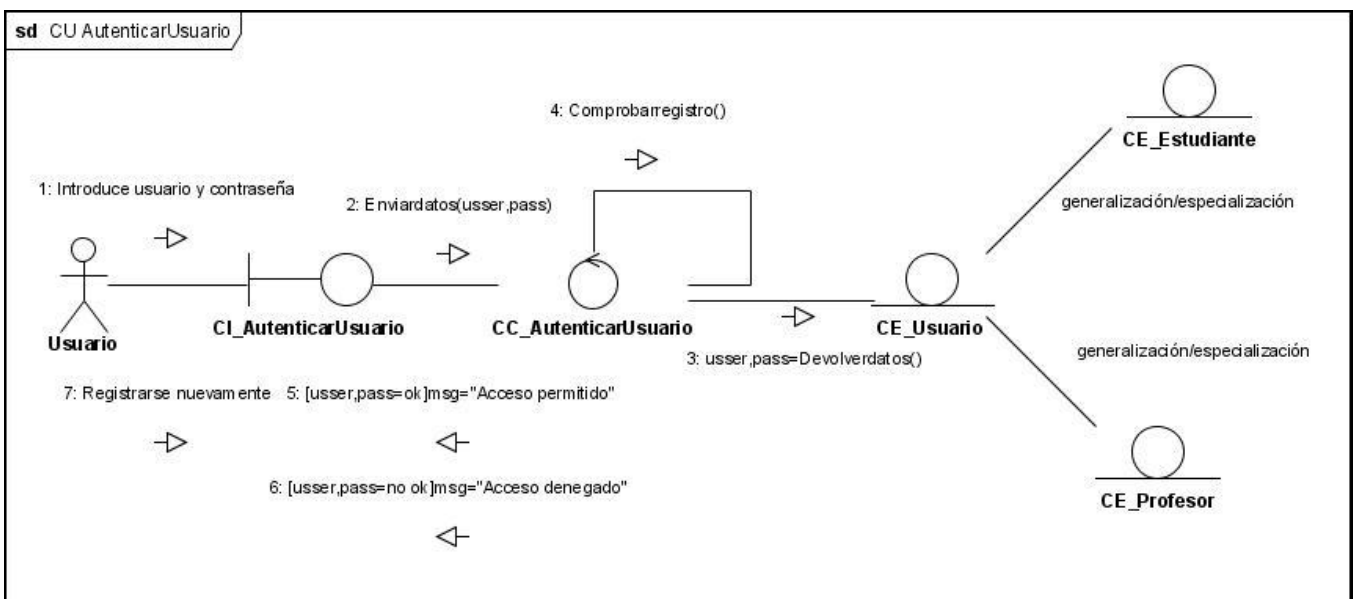


Figura 3.12 CU Autenticar Usuario.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

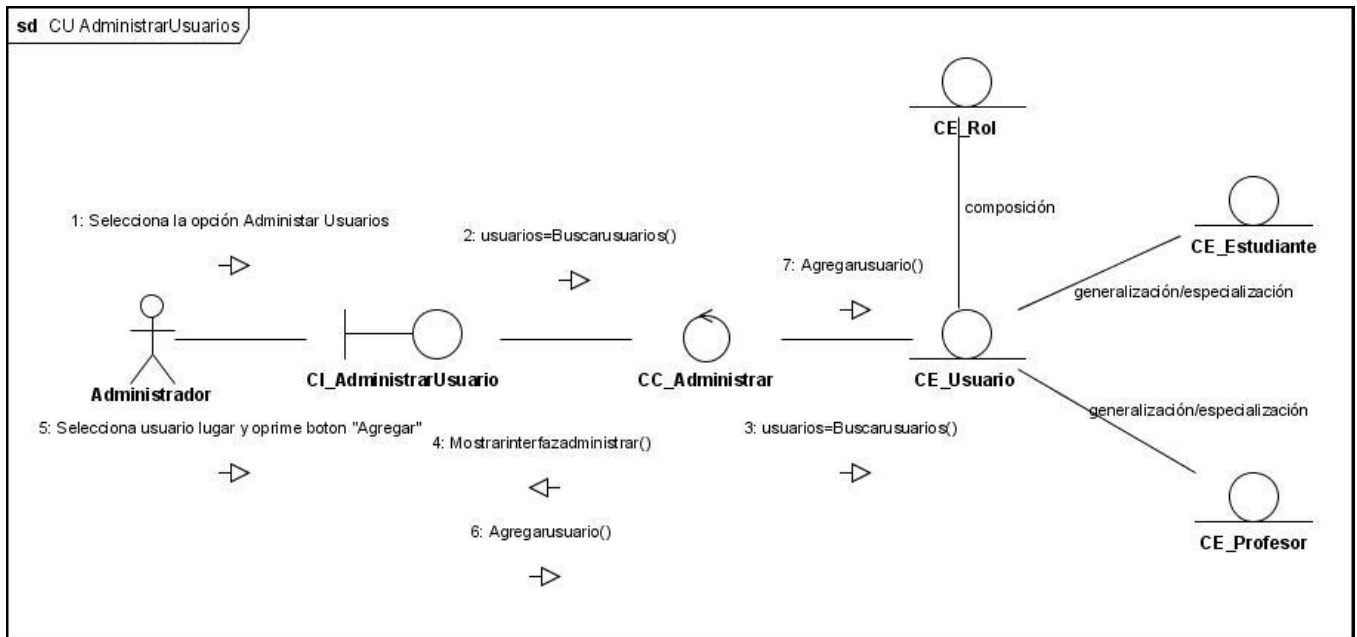


Figura 3.13 CU Administrar Usuarios.

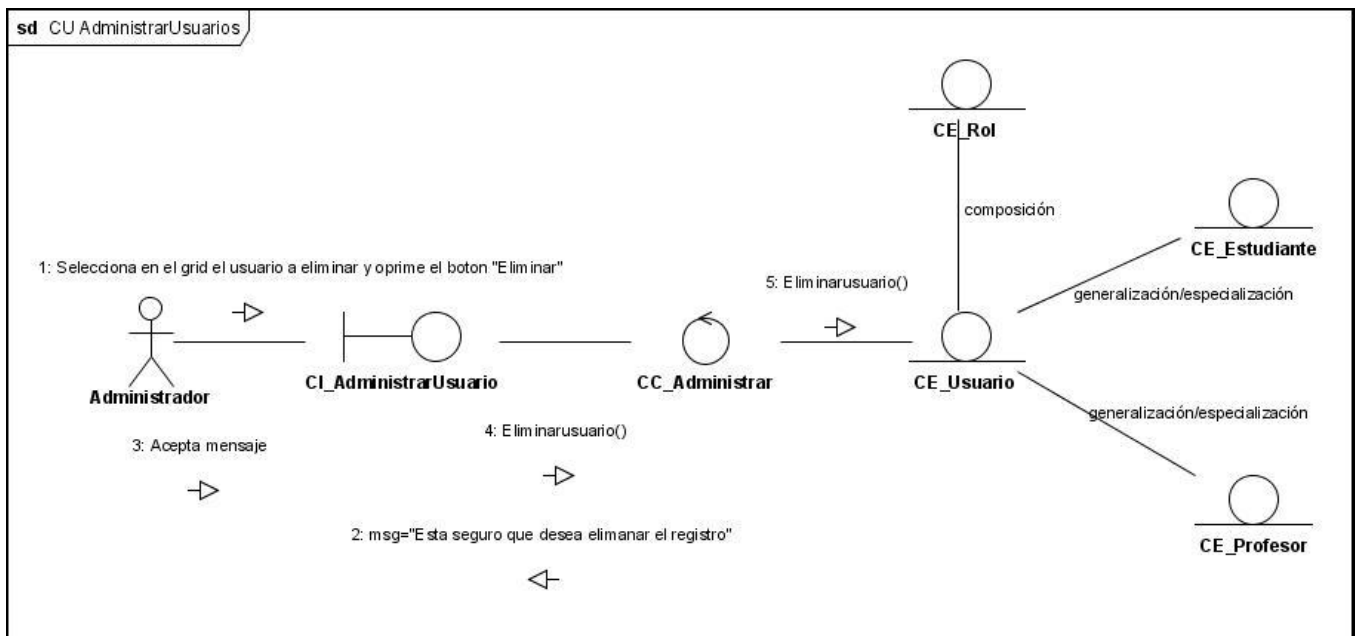


Figura 3.14 CU Administrar Usuarios (Escenario Eliminar).

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

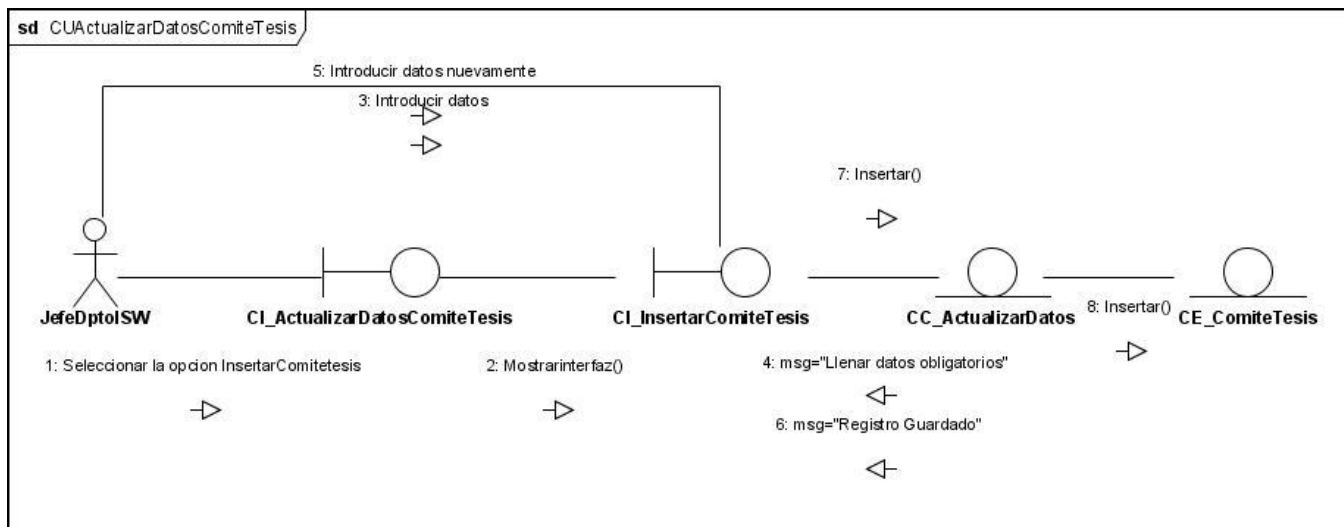


Figura 3.15 CU Actualizar Datos Comité Tesis (Escenario Insertar).

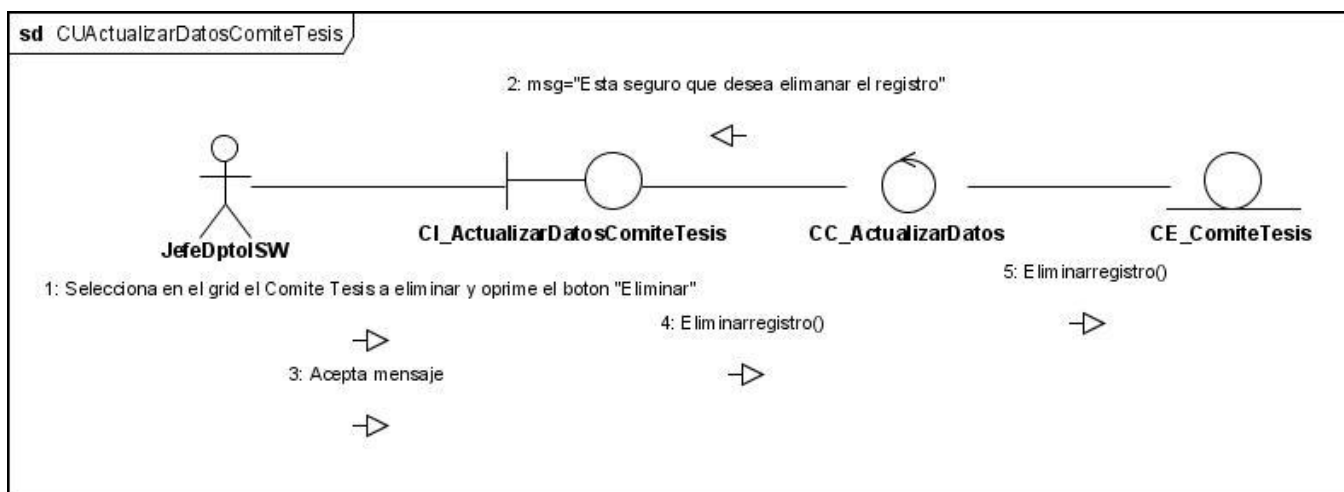


Figura 3.16 CU Actualizar Datos Comité Tesis (Escenario Eliminar).

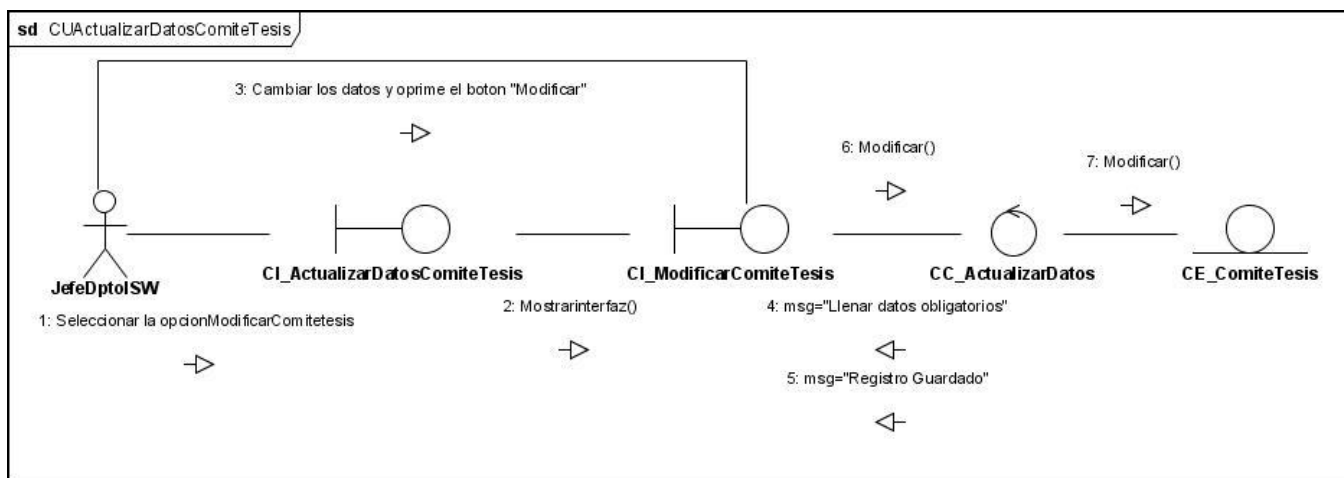


Figura 3.17 CU Actualizar Datos Comité Tesis (Escenario Modificar).

3.4 Diseño.

En el diseño se modela el sistema para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le imponen. Este debe adaptarse para satisfacer las características del entorno de implementación (lenguaje, plataforma) y los Requisitos No Funcionales, Seguridad, Escalabilidad, Usabilidad, Desempeño entre otros.

3.4.1 Diagrama de Clases de Diseño.

Cuando se crea un diagrama de clases, se está modelando una parte de los elementos y relaciones que configuran la vista de diseño del sistema. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, y se utilizan para modelar la vista de diseño estática de un sistema. Como el trabajo consiste en diseñar una aplicación Web hay que tener en cuenta que el diagrama de clases en las aplicaciones Web, difiere un poco de aquellas aplicaciones que se acostumbra a construir, debido a la necesidad de modelar las páginas, los enlaces entre estas y el contenido dinámico de las mismas. Para esto se utiliza el estereotipo Web, es decir las páginas cliente, las páginas servidoras y los formularios que se necesitan en el sistema.

Conexión: Clase encargada de establecer la conexión con el servidor de la BD. Se concibió aplicando el patrón Singleton el cual garantiza una única instancia para una clase y la creación de un mecanismo global (único) de acceso a dicha instancia.

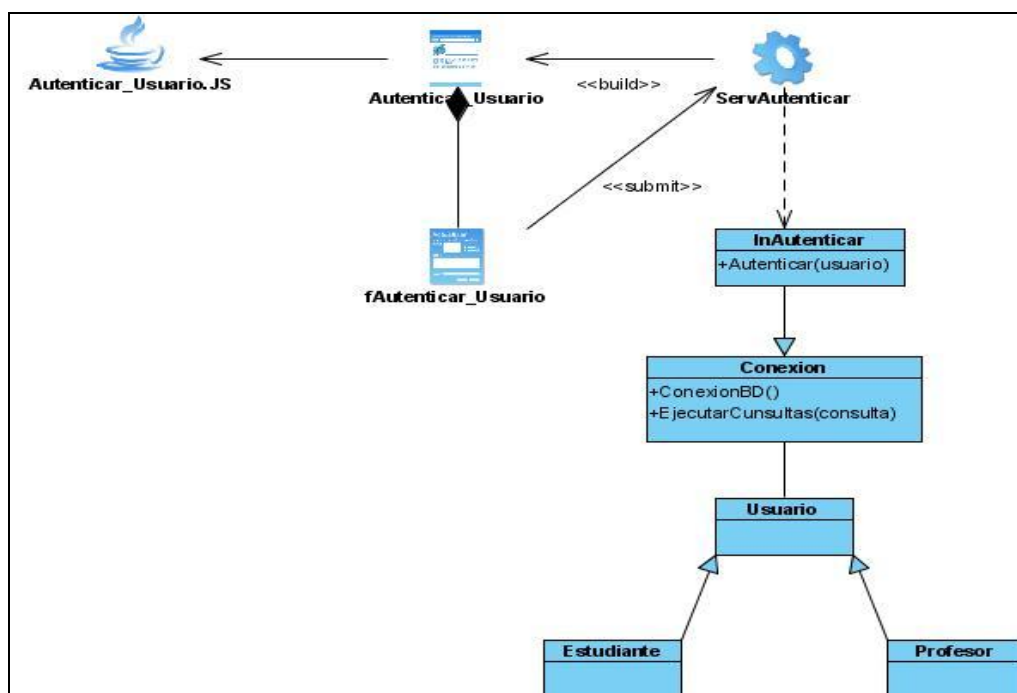


Figura 3.18 CU Autenticar Usuario.

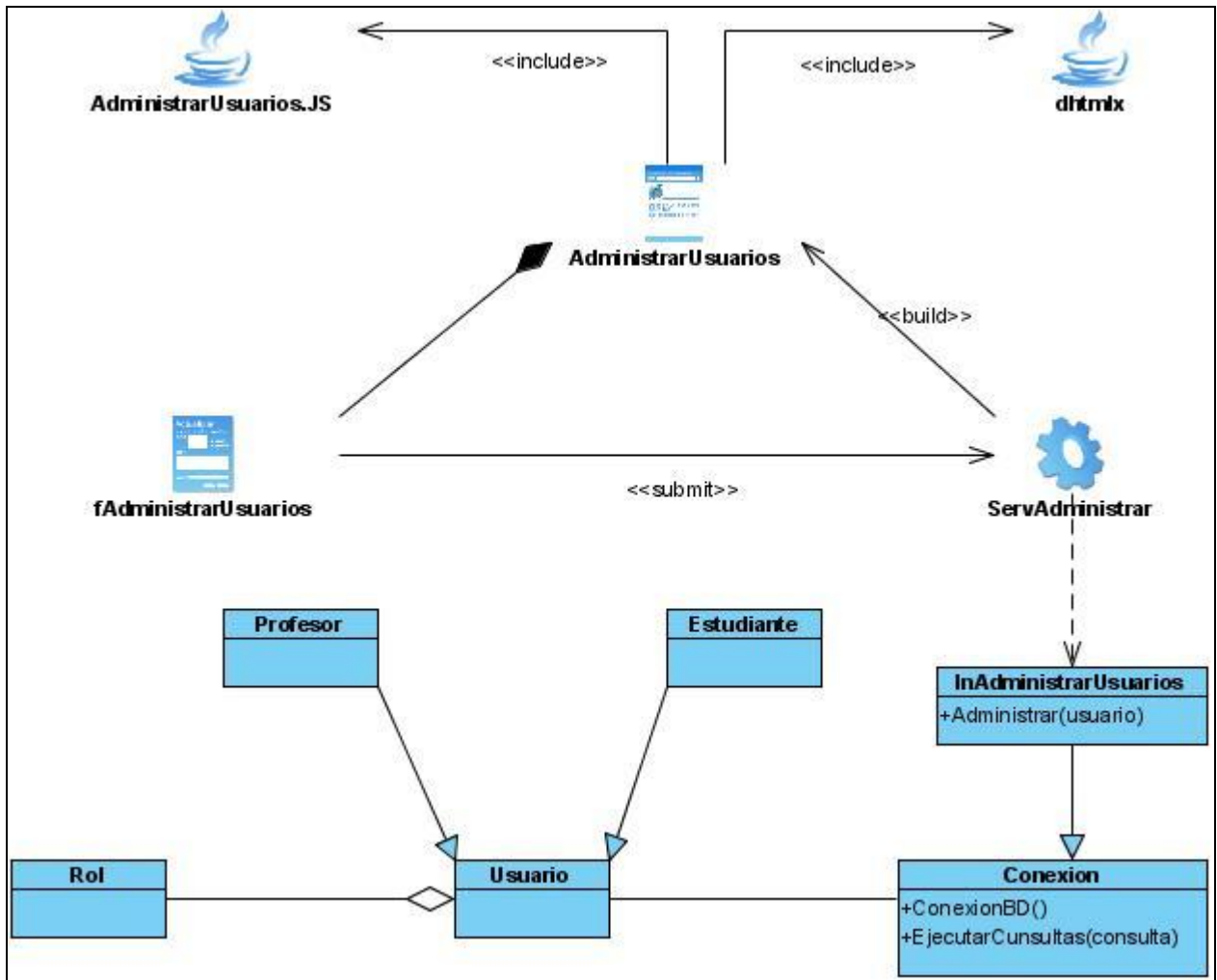


Figura 3.19 CU Administrar Usuario.

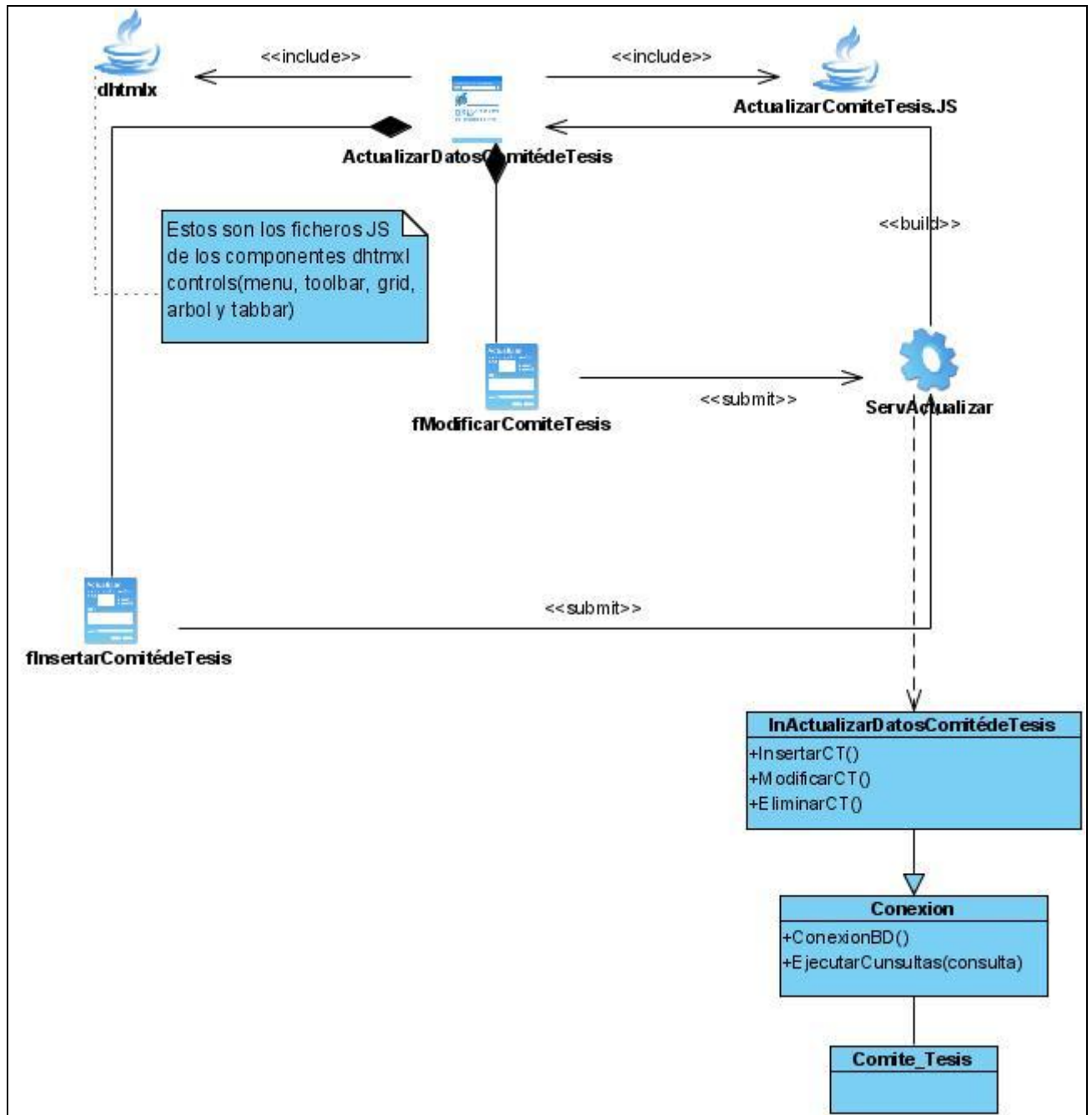


Figura 3.20 CU Actualizar Datos Comité de Tesis.

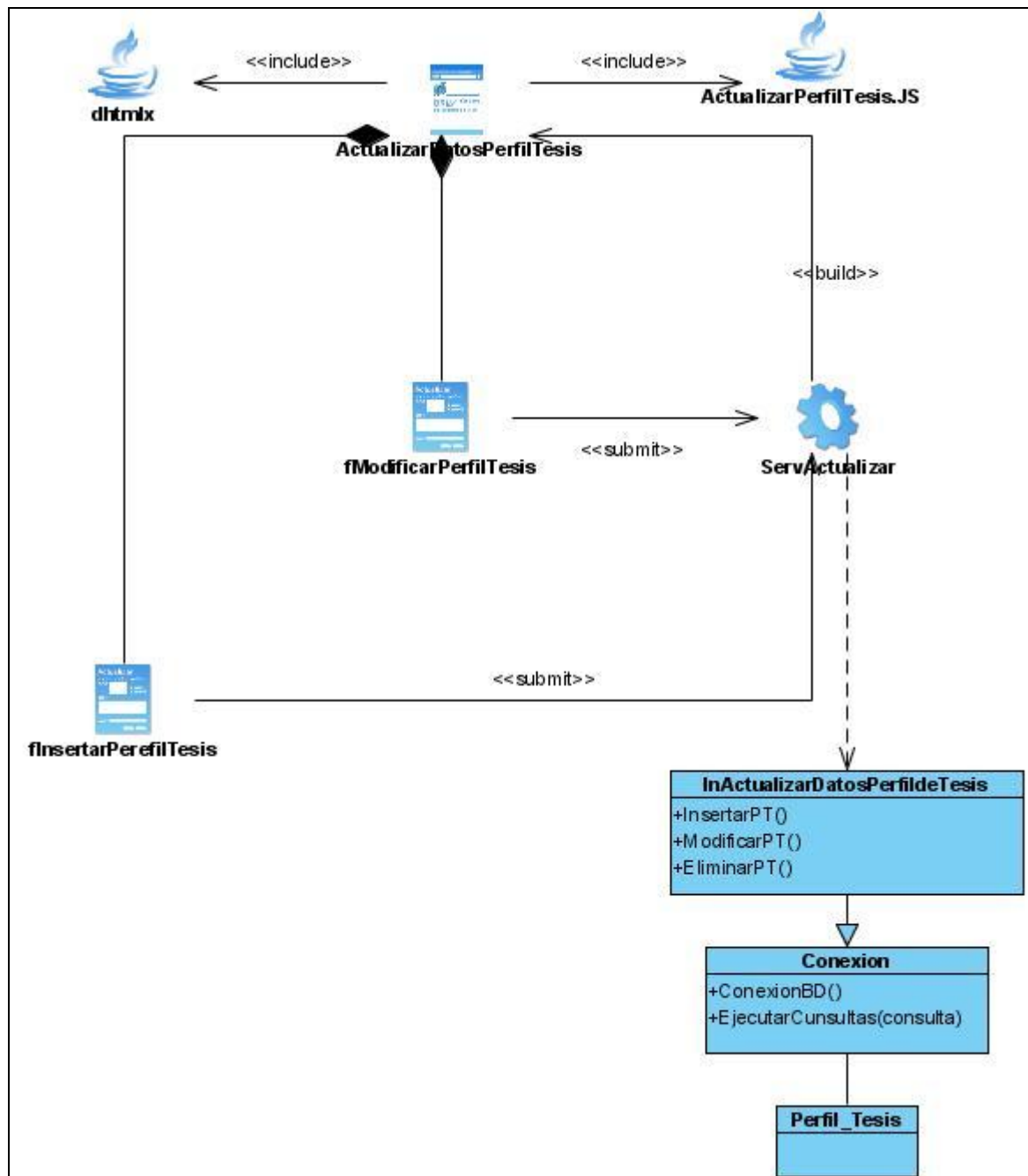


Figura 3.21 CU Actualizar Datos Perfil de Tesis.

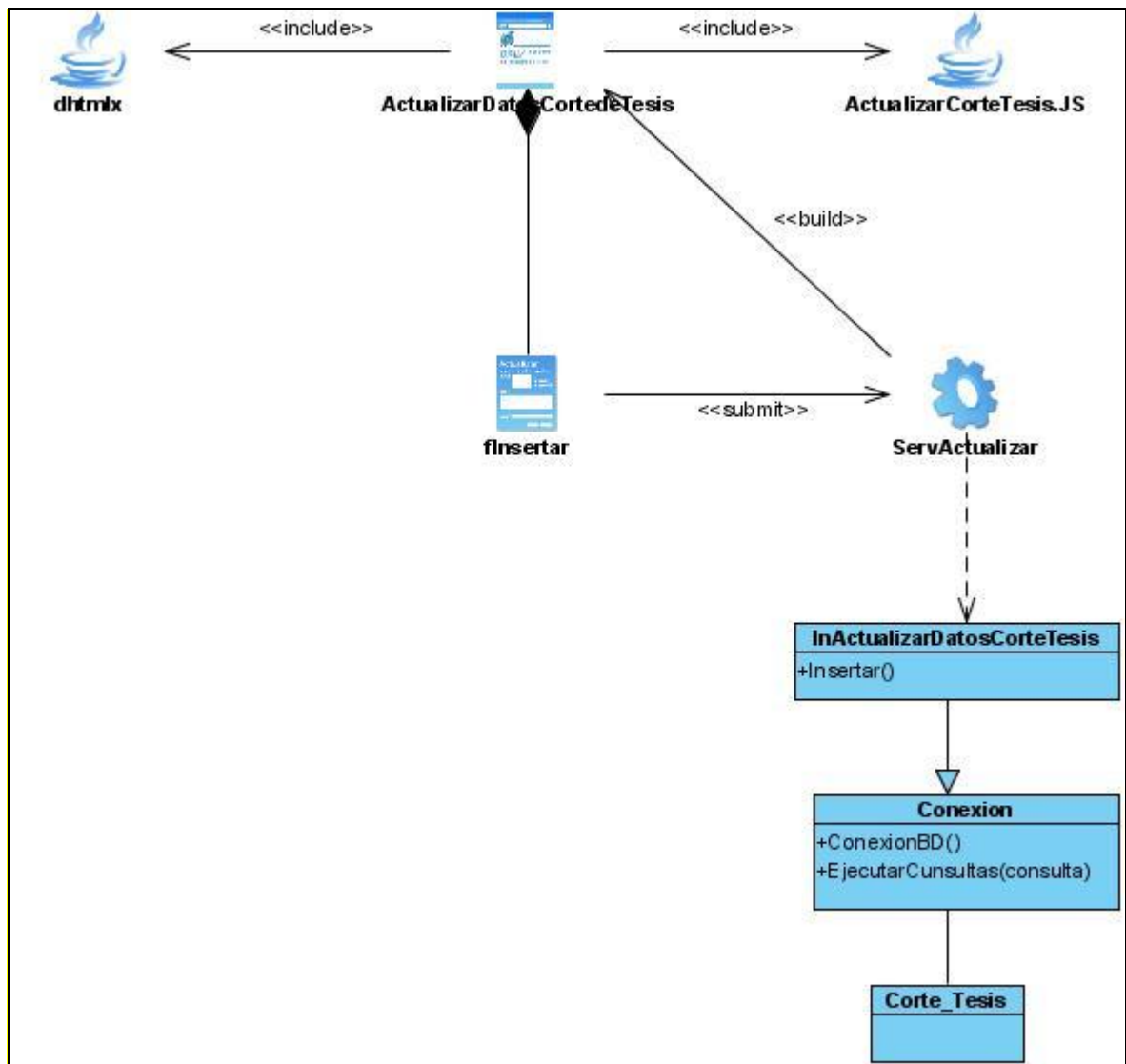


Figura 3.22 CU Actualizar Datos Corte de Tesis.

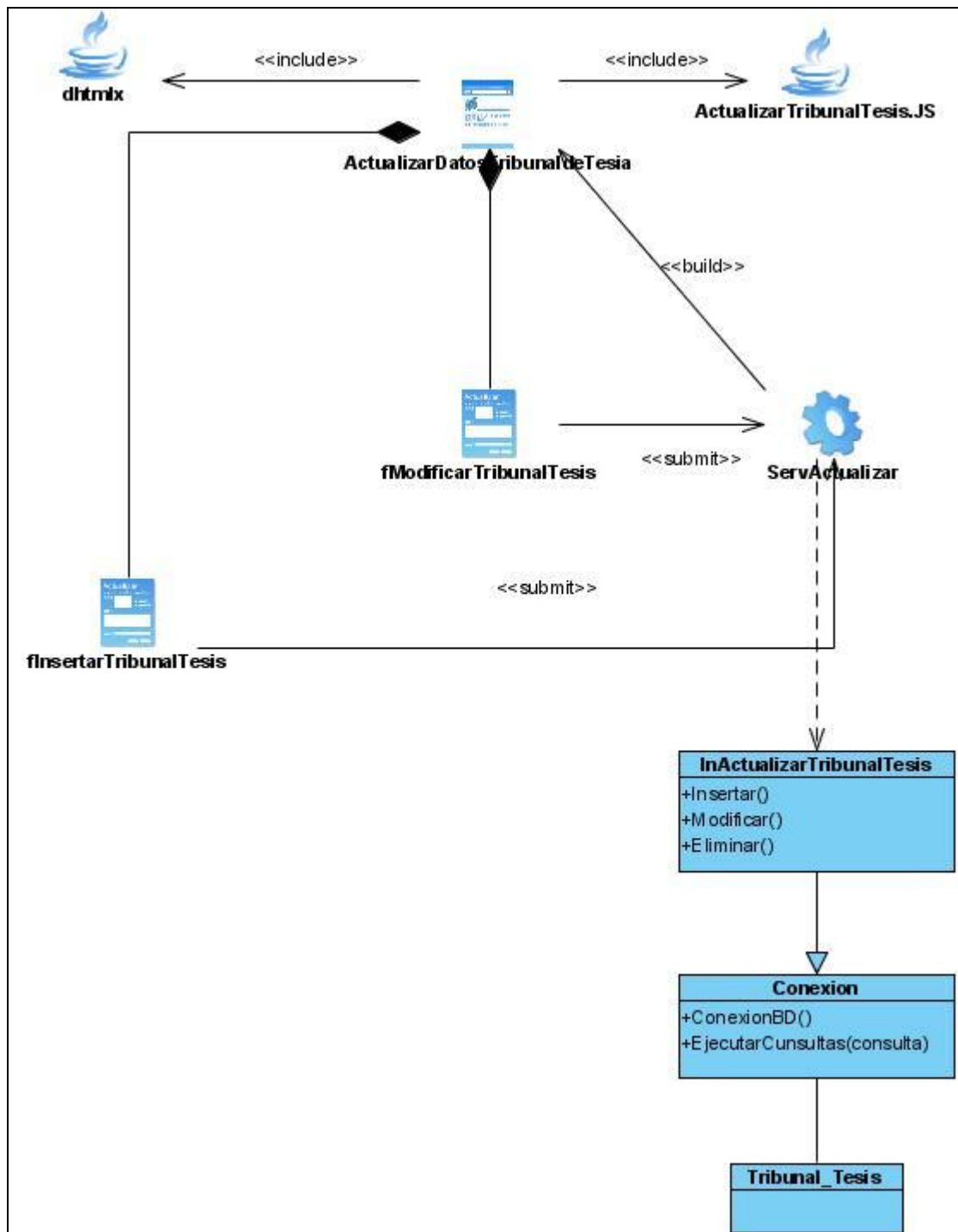


Figura 3.23 CU Actualizar Datos Tribunales de Tesis.

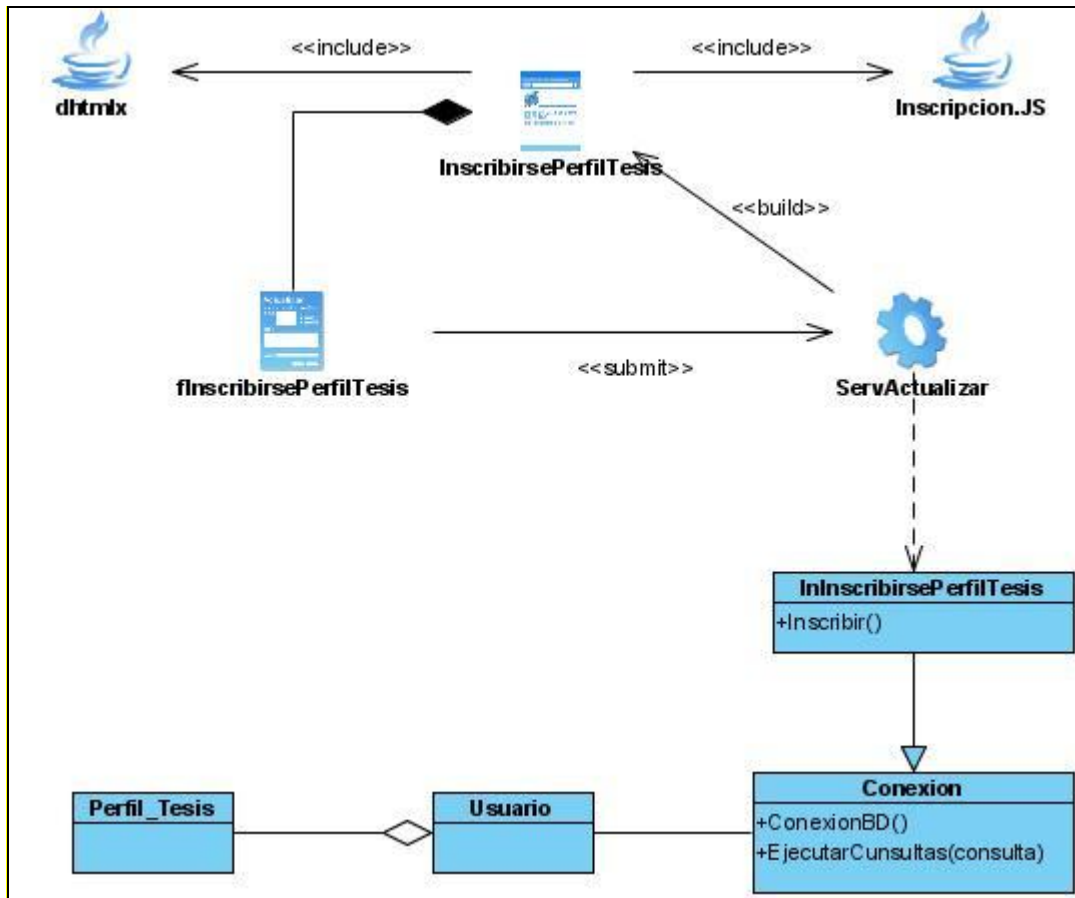


Figura 3.24 CU Inscribirse en un Perfil de Tesis.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

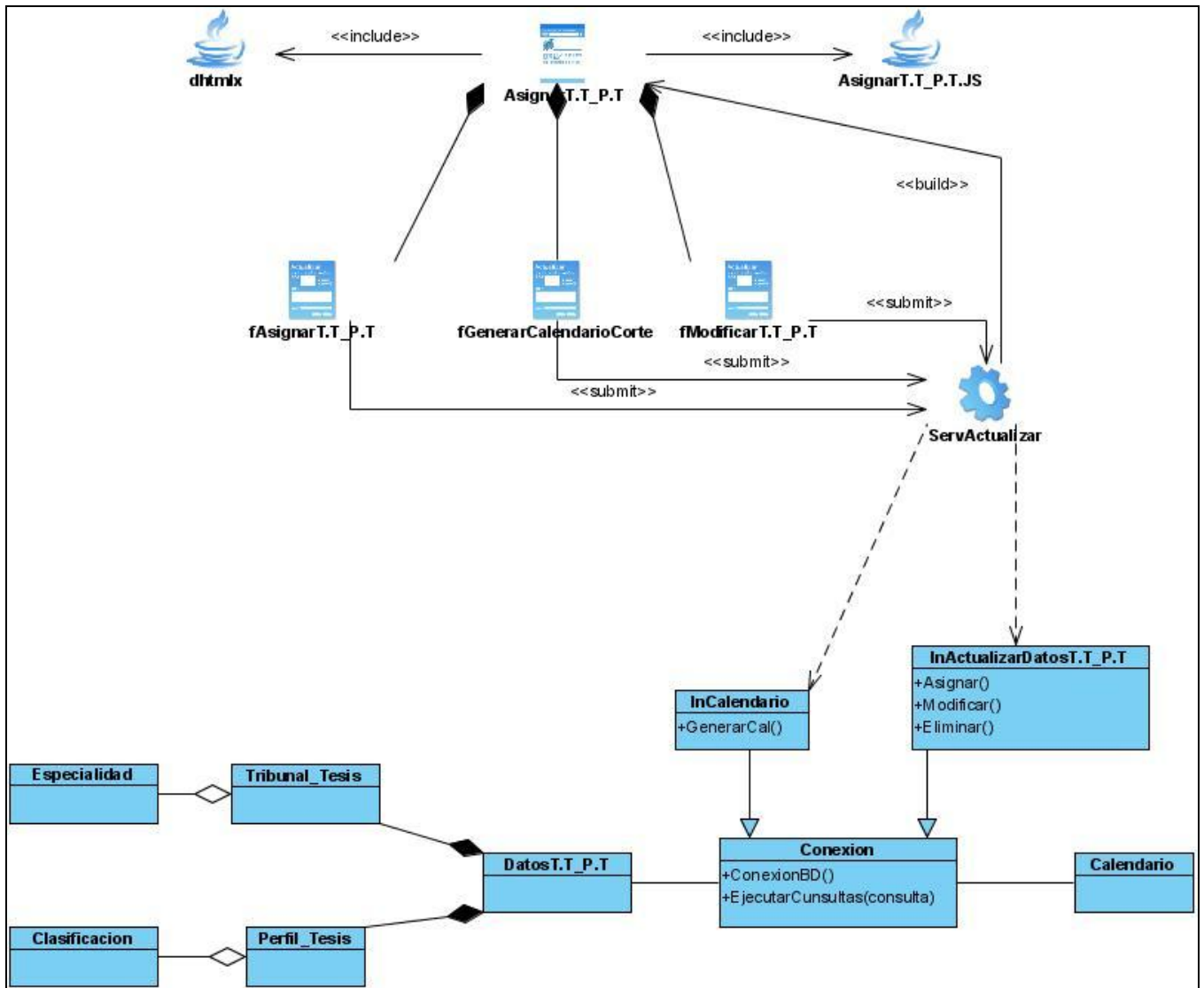


Figura 3.25 CU Asignar T.T a P.T.

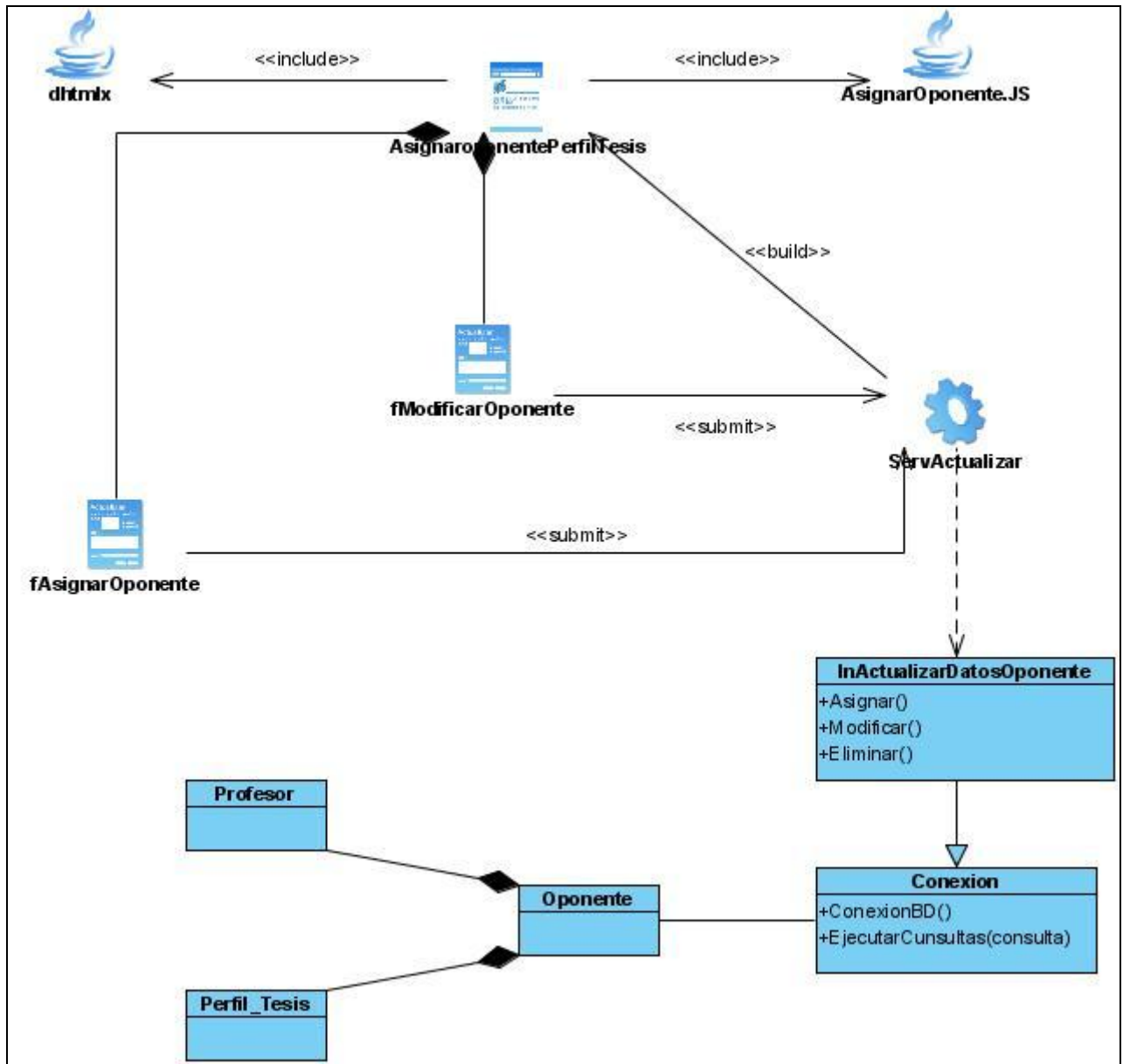


Figura 3.26 CU Asignar Oponente.

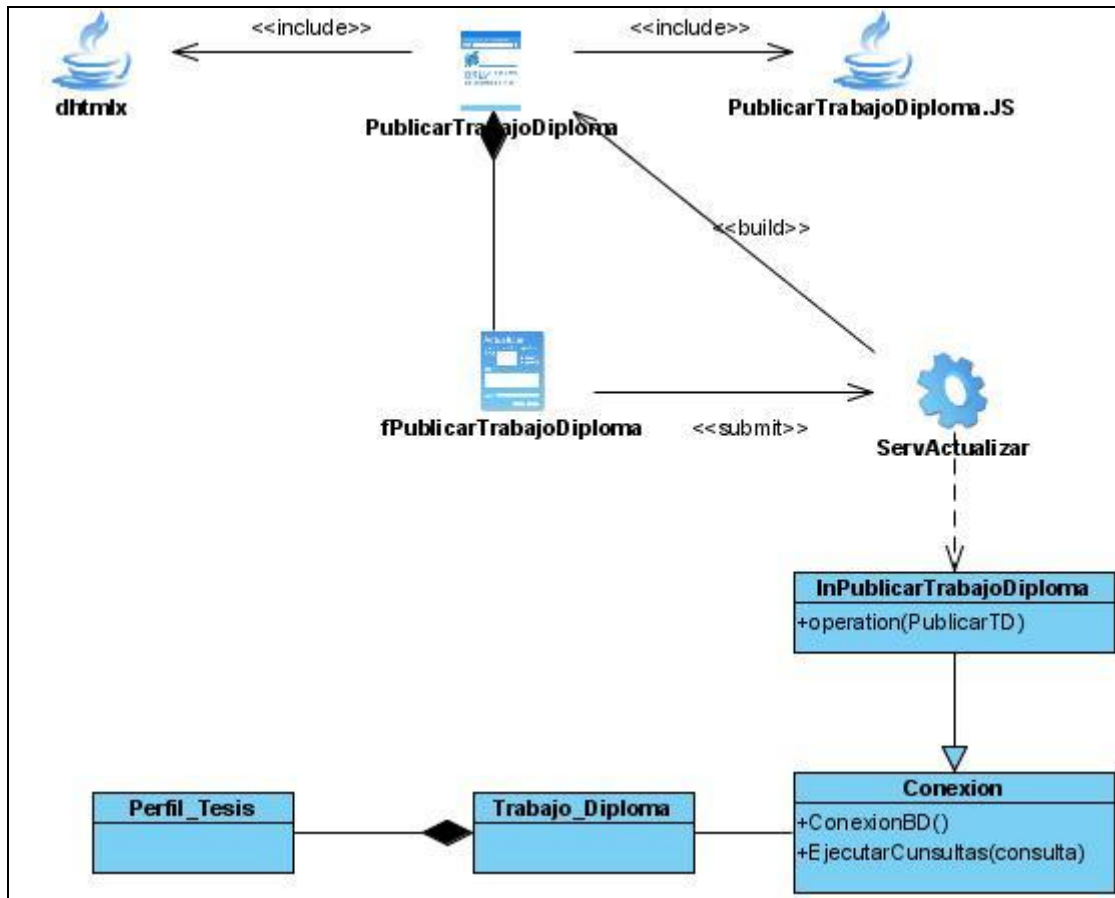


Figura 3.27 CU Publicar Trabajo de Diploma.

3.5 Diseño de la BD.

Para el diseño de la Base de Datos se tienen que seleccionar, de las clases entidades, cuáles son aquellas que necesitan ser almacenadas, en el caso de este trabajo existen trece clases entidades y las mismas requieren de almacenamiento. Luego de esto se identifican las relaciones entre estas clases que serán las clases persistentes formando parte de un modelo lógico de datos que seguidamente pasará a ser el modelo físico de la Base de Datos.

3.5.1 Modelo Lógico de Datos.

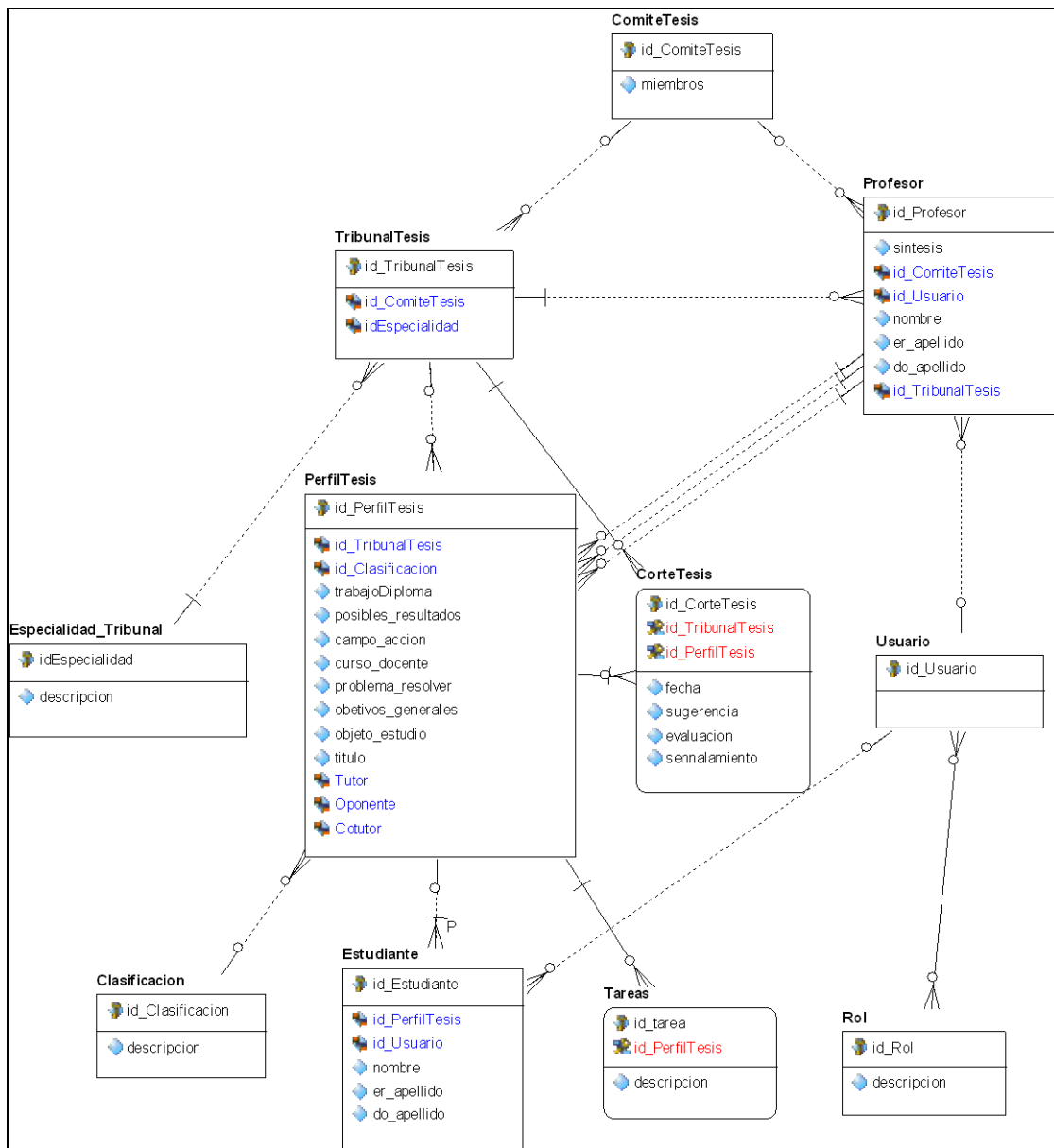


Figura 3.28 Diagrama de Clases Persistentes.

3.5.2 Modelo Físico de Datos.

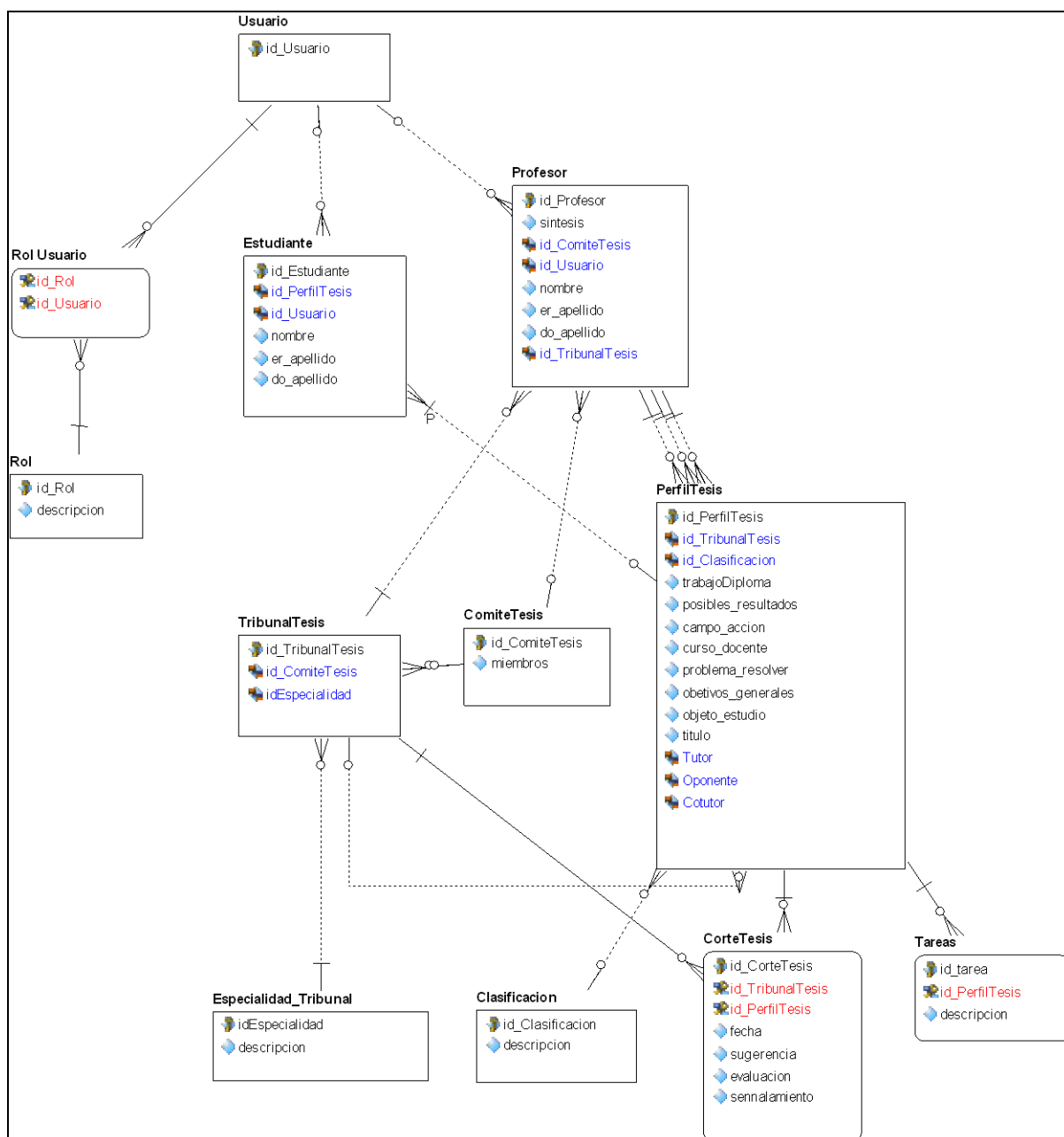


Figura 3.29 Diagrama de Base de Datos.

3.5.3 Descripción de las tablas de la Base de Datos.

Tabla3.1 Tabla "Perfil Tesis" de la BD.

Nombre: Perfil Tesis.
Descripción: Esta tabla recoge todos los datos que se necesitan sobre el Perfil de Tesis de un determinado trabajo de diploma, datos importantes como su título, sus

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

autores, tutor del trabajo, cuales son los objetivos, etc.		
Atributo	Tipo	Descripción
Id_PerfilTesis	SERIAL/INTEGER	Este es el atributo identificador.
curso_docente	DATE	En este campo se guarda la información del curso docente al que pertenece el perfil.
título	CHAR	En este campo se guarda el título del trabajo de diploma a defender.
problema_resolver	VARCHAR	Recoge un texto con el problema a resolver con dicho trabajo.
objetivos_generales	VARCHAR	Este campo recoge el objetivo general de la investigación.
objeto_estudio	VARCHAR	Recoge el objeto de estudio del trabajo.
campo_accion	VARCHAR	Recoge el campo de acción de la investigación en cuestión.
posibles_resultados	VARCHAR	Este campo recoge los posibles resultados que se pueden obtener después de concluida la investigación.
trabajoDiploma	IMAGE/LONG BINARY	Este campo recoge los trabajos de diplomas terminados en un formato concebido por la facultad.

Tabla3.2 Tabla "Comité Tesis" de la BD.

Nombre: Comité Tesis.		
Descripción: Esta tabla recoge todos los datos que se necesitan sobre el Comité de Tesis de la Facultad.		
Atributo	Tipo	Descripción
Id_ComiteTesis	SERIAL/INTEGER	Este es el atributo identificador.
miembros	VARCHAR	Ese campo recoge los miembros que tiene un comité de tesis.

Tabla3.3 Tabla "Clasificación" de la BD.

Nombre: Clasificación.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Descripción: Esta tabla recoge los prototipos de tesis que existen en la facultad.		
Atributo	Tipo	Descripción
id_Clasificación	SERIAL/INTEGER	Este es el atributo identificador.
descripción	CHAR	Este campo recoge la clasificación de un Perfil de Tesis determinado.

Tabla3.4 Tabla "Tribunal Tesis" de la BD.

Nombre: Tribunal Tesis.		
Descripción: Esta tabla recoge todos los datos que se necesitan sobre un tribunal de tesis de varios trabajos de diploma, datos importantes como su presidente, secretario y vocal.		
Atributo	Tipo	Descripción
Id_TribunalTesis	SERIAL/INTEGER	Este es el atributo identificador.

Tabla3.5 Tabla "Rol" de la BD.

Nombre: Rol.		
Descripción: Esta tabla recoge todos los datos que se necesitan para dar permisos en el sistema a distintos roles.		
Atributo	Tipo	Descripción
Id_Rol	SERIAL/INTEGER	Este es el atributo identificador.
descripción	CHAR	Este campo recoge la clasificación de los distintos usuarios que tienen privilegios en el sistema.

Tabla3.6 Tabla "Corte Tesis" de la BD.

Nombre: Corte Tesis.		
Descripción: Esta tabla recoge todos los datos que se necesitan sobre un Corte de Tesis de varios trabajos de diploma, datos importantes como su evaluación en el corte, sugerencias y señalamientos.		
Atributo	Tipo	Descripción
Id_CorteTesis	SERIAL/INTEGER	Este es el atributo identificador.
sugerencia	VARCHAR	En este campo se registran las sugerencias que se les hace a un perfil

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

		de tesis determinado.
evaluación	NUMERIC(5.0)	Registra la evaluación numérica en cada corte de tesis hasta la defensa del trabajo de diploma, lo que es un corte igualmente.
sennalamiento	VARCHAR	Registra los señalamientos en cada corte de tesis hasta la predefensa del trabajo de diploma.
fecha	DATE	Registra la fecha de los cortes de tesis, dato que es importante en el momento de generar calendario de cortes en la facultad.

Tabla3.7 Tabla "Tareas" de la BD.

Nombre: Tareas.		
Descripción: Esta tabla recoge todos los datos de las tareas de un perfil de tesis determinado.		
Atributo	Tipo	Descripción
Id_Tarea	SERIAL/INTEGER	Este es el atributo identificador.
descripción	VARCHAR	Almacena cada una de las tareas del Perfil de Tesis.

Tabla3.8 Tabla "Usuario" de la BD.

Nombre: Usuario.		
Descripción: Esta tabla recoge todos los datos que se necesitan sobre un usuario que accede al sistema.		
Atributo	Tipo	Descripción
Id_Usuario	SERIAL/INTEGER	Este es el atributo identificador.

Tabla3.9 Tabla "Estudiante" de la BD.

Nombre: Estudiante.		
Descripción: Esta tabla recoge todos los datos que se necesitan sobre un estudiante.		
Atributo	Tipo	Descripción

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Id_Estudiante	SERIAL/INTEGER	Este es el atributo identificador.
nombre	VARCHAR	Almacena el nombre del estudiante.
er_apellido	VARCHAR	Almacena el primer apellido del estudiante.
do_apellido	VARCHAR	Almacena el segundo apellido del estudiante.

Tabla3.10 Tabla "Profesor" de la BD.

Nombre: Profesor.		
Descripción: Esta tabla recoge todos los datos que se necesitan sobre un profesor.		
Atributo	Tipo	Descripción
Id_Profesor	SERIAL/INTEGER	Este es el atributo identificador.
nombre	VARCHAR	Almacena el nombre del profesor.
er_apellido	VARCHAR	Almacena el primer apellido del profesor.
do_apellido	VARCHAR	Almacena el segundo apellido del profesor.
síntesis	VARCHAR	En este campo se refleja una síntesis del tutor del Perfil de Tesis.

Tabla3.11 Tabla "Especialidad_Tribunal" de la BD.

Nombre: Especialidad_Tribunal.		
Descripción: Esta tabla recoge todos los datos que se necesitan sobre un Tribunal de Tesis en cuanto a las especialidades: (análisis y diseño, ciclo completo, investigación, etc.).		
Atributo	Tipo	Descripción
Id_Especialidad	SERIAL/INTEGER	Este es el atributo identificador.
descripción	CHAR	Este campo recoge la clasificación de los distintos tribunales en la facultad.

3.5.4 Principios de diseño.

El diseño de la interfaz es uno de los factores fundamentales a tener en cuenta en la creación de una aplicación Web, para esto se debe buscar la forma más amigable y comprensible posible que logre capturar y satisfacer las necesidades de los usuarios. Es muy importante también y no menos que el

diseño gráfico tener una buena navegabilidad, usabilidad y distribución del contenido. Para el diseño de la interfaz de usuario de este sistema se han seguido las siguientes primicias:

- Permitir utilizar el sistema por cualquier persona con conocimientos básicos de informática.
- Brindar a los usuarios una retroalimentación de las acciones que ejecutan, es decir mostrar aquellos mensajes que sean de confirmación, de explicación, y navegación entre otros, que sean claramente redactados de forma tal que el usuario tenga conocimiento de lo que está sucediendo.
- Evitar la sobrecarga de imágenes para favorecer la navegabilidad.
- Que el diseño sea atractivo para todos los usuarios.
- Que sea consistente con las expectativas e intuición del usuario.
- Que proporcione avisos eficaces y métodos de respuesta durante y tras la finalización de la tarea.

3.5.5 Tratamiento de errores.

Uno de los principios planteados en el diseño fue no dejar desinformado al usuario en cualquier acción que este ejecute, es por esto que como tratamiento de errores se tiene que el usuario sea redireccionado a una página con el mensaje correspondiente a dicha acción errada.

Por otra parte en el formulario se validan los datos que serán enviados mostrando mensajes en caso de invalidaciones o de confirmaciones, esto se realiza en la parte del cliente para no sobrecargar las acciones del servidor.

3.6 Conclusiones

En este capítulo se realizó una breve descripción de la arquitectura a utilizar en la realización de la aplicación, se representó gráficamente todo el análisis y el diseño del sistema, mencionando los principios de diseño que se llevarán a cabo, así como el tratamiento de errores para el correcto funcionamiento del sistema de gestión. Se representó y se describió la base de datos de la aplicación y se modelaron los nodos necesarios para la ejecución del sistema.

CONCLUSIONES

Al instante de haber realizado esta propuesta de solución para los procesos de control de trabajos de diploma que se llevan a cabo en la facultad, se puede llegar a las siguientes conclusiones:

- Se describieron las tecnologías adecuadas para la futura construcción de la aplicación, concluyéndose con la utilización de PHP como lenguaje de programación, como sistema gestor de base de datos PostgreSQL y como servidor Web Apache.
- Se utilizó RUP como metodología de desarrollo de software y UML como lenguaje de modelado para la visualización del sistema, transitando por las fases y flujos de trabajos del proyecto hasta llegar al diseño.
- Se describieron todas las actividades relacionadas con la información a procesar.
- Por último se cumplió con el objetivo propuesto de diseñar una aplicación Web, para el control de los trabajos de diploma de la facultad, lo que puede extenderse a todas las facultades de la Universidad de la Ciencias Informáticas. Capaz de garantizar el control y organización de perfiles de los trabajos de diploma, de brindar información sobre los temas, así como ofrecer una serie de funcionalidades para facilitar el control, la disponibilidad de trabajos e impedir repetición de trabajos ya desarrollados por los estudiantes. Además de mantener un control de los trabajos de diploma, una buena organización de los perfiles de tesis y los tribunales en correspondencia con la clasificación de la investigación en cuestión.

RECOMENDACIONES

Luego de concluir este trabajo se recomienda lo siguiente:

- Continuar el desarrollo del sistema, llevando a cabo el flujo de trabajo de Implementación que no se realizó aunque se deben tener en cuenta los cambios que puedan ocurrir.
- Profundizar en las tecnologías que con el continuo desarrollo de las mismas pueden surgir nuevas y mejores herramientas para la construcción del trabajo.
- Estudiar nuevamente los procesos que se realizan en la facultad en busca de cambios o nuevas tareas que requieran automatización.

REFERENCIAS BIBLIOGRÁFICAS

1. GUTIERREZ, J. G. *Definición arquitectura cliente servidor*. 2005, Disponible en: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>
2. IBAÑEZ, D. S. *Ventajas y Desventajas*. 2007, Disponible en: <http://www.csi.map.es/csi/silice/Global75.html>
3. FÉLIX, A. D. C. S. *El servidor de web Apache: Introducción práctica. Apache 1.x y 2.0 alpha*. 2000, Disponible en: <http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/apache.pdf>
4. ALVAREZ, M. A. *Qué es Java Script?* 2003, Disponible en: <http://www.desarrolloweb.com/articulos/25.php>
5. ---. *Qué es HTML?* 2003, Disponible en: <http://www.desarrolloweb.com/articulos/que-es-html.html>.
6. ---. *Qué es VBScript?* 2003, Disponible en: <http://www.desarrolloweb.com/articulos/717.php>.
7. S, C. V. D. H. *¿Qué es el PHP?* 2001, Disponible en: <http://www.maestrosdelweb.com/editorial/phpintro/>
8. ALVAREZ, M. A. *Que es JSP?* 2006, Disponible en: <http://www.desarrolloweb.com/articulos/831.php>
9. ---. *Qué es ASP?* 2006, Disponible en: <http://www.desarrolloweb.com/articulos/393.php>
10. ---. *Que es PERL?* 2006, Disponible en: <http://www.desarrolloweb.com/articulos/541.php>.
11. SANCHEZ, M. A. M. *Metodologías De Desarrollo De Software*. 2004, Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
12. MESTRAS, J. P. *Ingeniería del Software 2* 2004,
13. PESCADOR, J. S. *Manual: ¿Qué es una Base de Datos?*, Disponible en: <http://www.mihosting.net/areaclientes/knowledgebase.php?action=displayarticle&catid=12&id=63>.
14. WORSLEY, J. A. J. D. A., B. AND MICHAEL, H. *PostgreSQL Práctico*. 2001,
15. BURBANO PROAÑO, I. D. J. *Comparativa de base de datos, [PDF]*. 2006,
16. LÓPEZ, D. H. Y. F. R. *Ventajas de las Herramientas CASE*. 2000, Disponible en: <http://www.csi.map.es/csi/silice/Sgcase5.html>
17. ALVARADO, P. y LAITANO, R. *Modelo de base de datos con ER/studio*. Disponible en: <http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml#er>.
18. JACOBSON, I. G. B., ET AL. *El Proceso Unificado de Desarrollo de Software*. OTERO, A., Adison Wesley. 2000.1,

REFERENCIAS BIBLIOGRÁFICAS

19. ---. *El Proceso Unificado de Desarrollo de Software*. OTERO, A., Adison Wesley. 2000. 1.,

BIBLIOGRAFÍA

1. PRESSMAN, R. S. *Ingeniería del Software Un enfoque práctico*. 2005. vol. 1,
2. ---. *Ingeniería del Software Un enfoque práctico*. 2005. vol. 2,
3. RUP. "Rational Unified Process". 2001,
4. "Embarcadero ER/Studio" HELP. 2006,
5. Visual Paradigm for UML (Enterprise Edition) HELP. 2007,

ANEXOS

ANEXO I. Los niveles en capas por lo que se identifica una aplicación Web.

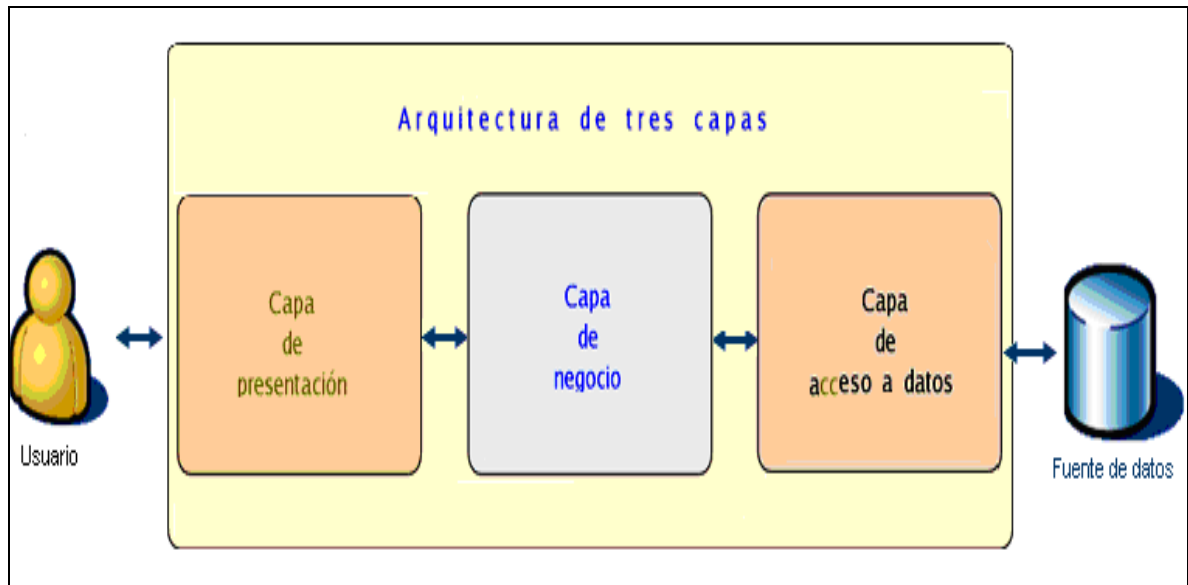


Figura 1. Los niveles en capas para una aplicación Web.

ANEXO 2. Arquitectura Cliente/Servidor.

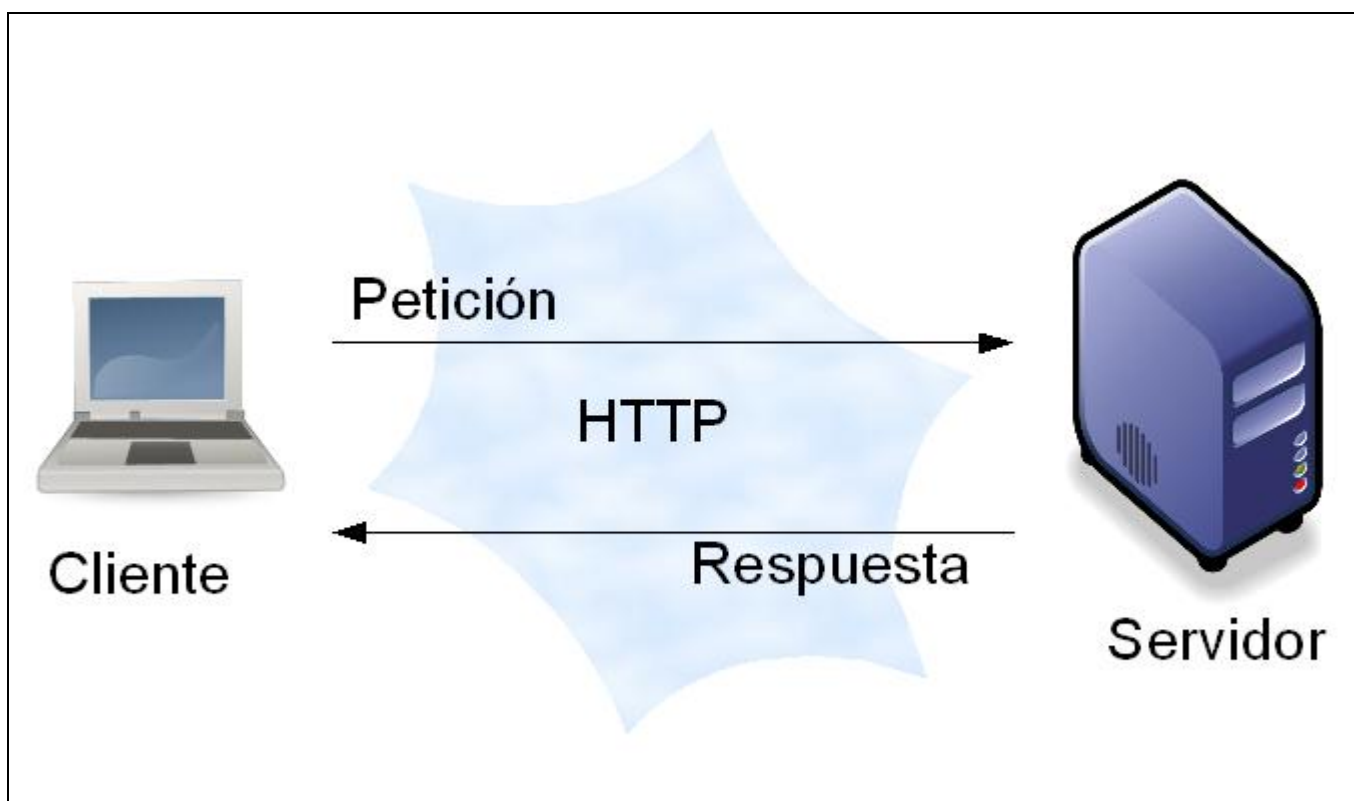


Figura 2. Arquitectura Cliente/Servidor.

GLOSARIO

A

Aplicación Web: Una aplicación Web es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet.

B

Base de datos: Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

C

Cliente/Servidor: La arquitectura cliente-servidor es una forma de dividir las responsabilidades de un Sistema de Información separando la interfaz de usuario (Nivel de presentación) de la gestión de la información (Nivel de gestión de datos).

Casos de Uso del Negocio: Representan un grupo de tareas relacionadas lógicamente que producen un resultado observable para ciertos actores del negocio.

D

Diagrama de actividad: Describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio.

F

Formulario: Conjunto de campos solicitados por un determinado programa, los cuales se almacenarán para su posterior uso o manipulación.

H

Herramientas: Dispositivo que provee una ventaja mecánica al realizar una determinada tarea.

I

Intranet: Es una red local que utiliza herramientas de Internet. Se puede considerar como una Internet privada que funciona dentro de una organización. Normalmente, dicha red local tiene como base el protocolo TCP/IP de Internet y utiliza un sistema firewall (cortafuegos) que no permite acceder a la misma desde el exterior.

Internet: Red de telecomunicaciones nacida en 1969 en los EE.UU. a la cual están conectadas centenares de millones de personas, organismos y empresas en todo el mundo.

L

Lenguaje de programación: Es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.

M

Metodología: La metodología es una etapa específica que dimana de una posición teórica y epistemológica y que da pie a la selección de técnicas concretas de investigación.

Multiplataforma: Es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

Modelo de Datos: Describe la representación lógica y física de los datos persistentes.

Modelo de Datos: Describe la representación lógica y física de los datos persistentes.

N

Negocio: Cualquier ambiente o entorno en el cual está enmarcado el problema.

P

Plataforma: Es un término de carácter genérico que designa normalmente una arquitectura de hardware, aunque también se usa a veces para sistemas operativos o para el conjunto de ambos.

S

Sistema: Conjunto de elementos interrelacionados y regidos por normas propias, de modo tal que pueden ser vistos y analizados como una totalidad. El sistema se organiza para producir determinados efectos, o para cumplir una o varias funciones.

Sistema Informático: Es la síntesis de hardware y software. Un sistema informático típico emplea un ordenador que usa dispositivos programables para almacenar, recuperar y procesar datos. El ordenador personal o PC resulta de por sí un ejemplo de un sistema informático.

Software: Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

Sitio: Punto de la red con una dirección única y al que pueden acceder los usuarios para obtener información.

Sitio Web: Colección de páginas Web dotada de una dirección Web única.

Servidor: Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la Red.

Servidor Web: Se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

Sistema Operativo: Es un conjunto de programas destinados a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera eficiente. Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos.

Script: El guión o archivo de procesamiento por lotes (en inglés "script") es un programa usualmente simple, que generalmente se almacena en un archivo de texto plano. Es un guión o conjunto de instrucciones. Permiten automatizar tareas creando pequeñas utilidades.

ACRÓNIMOS

API: Interfaz de Programación de Aplicaciones.

CERN: Organización Europea para la Investigación Nuclear.

CUJAE: Ciudad Universitaria José Antonio Echeverría.

XHTML: EXtensible Hypertext Markup Language o Lenguaje Extensible de Marcado de Hipertexto.

HTML: HyperText Markup Language o Lenguaje de Marcas de Hipertexto.

XML: EXtensible Markup Language o lenguaje extensible de marcado.

CGI: Common Gateway Interface.

ASP: Active Server Pages.

ORDBMS: “Object – Relational Data Base Management System”.

RDBMS: “Relational Data Base Management System”.

ISW: Ingeniería Del Software.