

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 9**



Sistema Gestor de Trabajos de Diploma.



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
INFORMÁTICA**

AUTOR(ES):

Zuzel Díaz Pérez
Adony González Cárdenas

TUTOR:

Lic. David Silva Barrera

Habana, 24 de junio del 2009
“Año del 50 aniversario del triunfo de la revolución”



“La gloria del mundo es transitoria, y no es ella la que da la dimensión de nuestra vida, sino la elección de seguir la leyenda personal, tener fe en tus propias utopías y luchar por los sueños tejidos”.

Paulo Coelho.

DEDICATORIA

Dedico este trabajo en especial a mi abuela Cocuca. Ella ha estado al tanto de mí en estos años y en toda mi vida como si fuera hija suya. Cuqui siempre ha confiado en mí cuando me he propuesto alguna meta difícil, no creo que pueda defraudarla nunca.

A mami, que ha sido una madre muy especial y una verdadera amiga. Ha velado por mí aconsejándome y mostrándome el buen camino. Nunca me ha dicho “no”. Soy independiente gracias a su educación y determinación de dejarme elegir por mí misma y asumir mis propios riesgos.

A papi, quien despertó en mí la creatividad y la lógica, quien me condujo en mis primeros pasos en el mundo de las computadoras. Me ha enseñado de todo un poco. Papi me ha brindado valiosos consejos y he tomado sus opiniones como mías.

A Denise, mi hermana grande. Tan buena como tan linda, que me quiere un mundo, así como yo la quiero a ella.

A mi tía linda, ella fue quien me enseñó a hacer pudines y a caminar en tacones.

A Pablito, siempre tan preocupado por mi alimentación aquí en la escuela. Me crió junto a Denise y fue quien me enseñó mis primeras zambullidas en la playa de Caletones.

A mis mejores amigas: Yaimara, Nerelys y Noris. Reímos y lloramos juntas. Las tres han sido maravillosas conmigo, siempre las recordaré como mis lindas hermanitas.

A David, mi mejor amigo, mi maestro de la vida y de lo profesional, mi pareja en todos los sentidos. Desde que lo conocí no he parado de superarme. Es todo un tesoro de virtudes, todavía no me creo que lo haya encontrado.

A todas las personas que participaron en mi formación personal y profesional.

Besos, Zuzel.

DEDICATORIA

A mi mamá Aida y a mi hermana Candy por ser parte esencial de mi vida.

A mi papá Antonio por tener siempre con su consejo y su ayuda.

A mi otro papá Ire por contar con él en los momentos difíciles que he tenido.

A la familia de Ovi y Marbelís antiguos compañeros míos por ser mis familias cuando estaba lejos de la mía.

A los profesores por encaminarme en los conocimientos que he adquirido en la universidad.

A mis compañeros de grupo por haber compartido conmigo todo este tiempo.

A todos los que de una forma u otra me ayudaron a transitar por mis estudios universitarios.

A todas las personas amantes del conocimiento y a aquellos al cual este Trabajo de Diploma le ayude en su vida profesional.

Adony

AGRADECIMIENTOS

En primer lugar queremos brindarle especiales agradecimientos a nuestro Tutor, el Lic. David. Él estuvo con nosotros en todo momento y a él le debemos haber podido llevar a feliz término nuestro Trabajo de Diploma.

También queremos agradecerle a nuestro oponente y cliente, el Msc. Febe, de quien tomamos valiosos consejos y quien estuvo al tanto de nuestro trabajo.

Queremos agradecer a todas nuestras amistades en especial a Hilda, Noris y Nerelys, quienes nos acompañaron interminables fines de semana de entrada de datos reales al sistema.

Agradecerle a los miembros del tribunal, en especial al presidente el Ing. Sandy, por su paciencia e interés para con nosotros.

Y por último pero no menos importante, agradecerle a nuestras familias. Siempre nos han apoyado, antes y después de llegar a esta universidad maravillosa que nos forjó en lo que somos.

Gracias a profesores y compañeros de estudio que estuvieron con nosotros estos 5 años tan decisivos en nuestras vidas en lo personal y en lo profesional.

Muchas gracias a todos.

Zuzel y Adony.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

Sistema Gestor de Trabajos de Diploma

y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 24 días del mes de junio del año 2009.

Zuzel Díaz Pérez

Adony Gonzáles Cárdenas

Tutor

Lic. David Silva Barrera.

RESUMEN

El proceso de evaluación, seguimiento y control de los trabajos de diploma (TD) en nuestra universidad es uno de los de mayor importancia dentro del proceso de formación docente. El cúmulo de personas implicadas no es pequeño; tesis y profesores vinculados a tesis, a tribunales y al comité de tesis de cada facultad suman un número considerable.

Planificar las evaluaciones previstas para cada corte y darle un seguimiento personalizado a cada tesis se convierte en un trabajo de mesa fuerte para los máximos responsables del éxito del proceso. De forma general, se gestionan TD, tesis, profesores, tribunales y evaluaciones.

La solución que se propone es un sistema gestor de TD que cuenta con una aplicación Web y una aplicación escritorio. Este documento que la ampara estará dividido en tres capítulos:

1. Fundamentación Teórica
2. Presentación de la Solución.
3. Construcción de la Solución.

PALABRAS CLAVES

Proceso, sistema de gestión.

INDICE

INTRODUCCIÓN	11
1.1 INTRODUCCIÓN	15
1.2 DESCRIPCIÓN GENERAL DEL OBJETO DE ESTUDIO.	15
1.3 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.	17
1.4 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES.	20
1.5 LAS TICS. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.	21
1.6.1 EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP) Y EL LENGUAJE UNIFICADO DE MODELADO (UML).	23
1.6.2 VISUAL PARADIMG COMO HERRAMIENTA COMPUTACIONAL DE MODELADO.	25
1.8 ARQUITECTURA DE UNA APLICACIÓN WEB	28
1.9 PRINCIPALES LENGUAJES Y TECNOLOGÍAS DE LA WEB.	30
1.9.1 LENGUAJES DE PROGRAMACIÓN PARA LA WEB	30
1.9.2 LA PLATAFORMA ASP.NET COMO SOPORTE DE LA IMPLEMENTACIÓN	32
1.9.3 VISUALSTUDIO.NET COMO PLATAFORMA DE DESARROLLO.	34
1.9.4 PRINCIPALES GESTORES DE BASES DE DATOS	35
1.9.5 SERVIDORES WEB.	36
1.10 OTRAS HERRAMIENTAS ÚTILES EN EL DESARROLLO Y PRODUCCIÓN DE SOFTWARE.	37
1.11 CONCLUSIONES.	38
2.1 INTRODUCCIÓN	39
2.2 MODELO DE NEGOCIO	39
2.2.1 PROCESOS DEL NEGOCIO	39
2.2.2 ACTORES Y TRABAJADORES DEL NEGOCIO	41
2.2.3 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	42
2.2.4 DESCRIPCIÓN DE LOS CASOS DE USO DE NEGOCIO	43
2.3 REQUERIMIENTOS FUNCIONALES	46
2.4 REQUERIMIENTOS NO FUNCIONALES	51
2.5 DESCRIPCIÓN DEL SISTEMA PROPUESTO	52
2.5.1 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.	52
2.5.2 DIAGRAMA DE CASOS DE USOS DEL SISTEMA.	54
2.5.3 DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.	54
2.7 CONCLUSIONES	59
CAPÍTULO 3 “CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA”	60
3.1 INTRODUCCIÓN	60
3.2 DIAGRAMAS DE CLASES.	60
3.3 PRINCIPIOS DEL DISEÑO.	64
3.3.1 APLICACIONES DE LOS PRINCIPIOS DE DISEÑO A LA INTERFAZ DE LA APLICACIÓN.	65
3.3.2 CONCEPCIÓN GENERAL DE LA AYUDA	66
3.4 DISEÑO DE LA BASE DE DATOS	66
3.5 MODELO DE IMPLEMENTACIÓN	69
3.6.1 DIAGRAMA DE COMPONENTE	69
3.6.2 DIAGRAMA DE DESPLIEGUE	70

3.7 PRUEBA DEL SISTEMA PROPUESTO Y VALIDACIÓN DE LA SOLUCIÓN.	71
3.9 CONCLUSIONES	73
ANEXOS.	82
GLOSARIO DE TÉRMINOS.	84

Introducción

En el mundo, las actividades fundamentales de la sociedad están enfocadas al desarrollo sostenible de su economía. Constituye un reto para cualquier organización tener ganancia y productividad.

En la era de la información, uno de los principales problemas es su exceso. Desde un comienzo, las empresas han manejado pequeños volúmenes de información debido al estrecho terreno donde vender sus mercancías. Con el paso del tiempo se ha ampliado el mercado y la cantidad de información ha crecido, por lo que su manejo se ha vuelto complejo, tornándose difícil la toma de decisiones.

La información es un elemento fundamental para el desarrollo. Con el paso de los años, la gestión de la información ocupa, cada vez más, un espacio mayor en la economía de los países a escala mundial.

La aplicación de las nuevas Tecnología de la Información y las Comunicaciones (TICs) se presenta imprescindible para procesar grandes cantidades de datos. Las TICs son un medio irrefutable para transmitir y gestionar datos, información y conocimiento.

El manejo de información requiere desarrollar un conjunto de habilidades que permiten definir la información necesaria, obtenerla y aprovecharla. Con el surgimiento de las computadoras, se desarrollaron herramientas informáticas que permiten mayor rapidez, reducir el esfuerzo, representar y comunicar la información.

En la actualidad, la tendencia de soporte de la información es en formato digital. El aumento de los volúmenes de información y el acortamiento de los tiempos de procesamiento, hacen indispensable el desarrollo de software para los procesos en las organizaciones. La gestión de información es caracterizada por la reducción de etapas y recursos, brindando la oportunidad de disminuir el tiempo de algunos procesos para invertirlo en la realización de otros. Otra característica es la simplificación, debido a que va intentando reducir el número de personas y departamentos implicados, posibilitando que el personal encargado de atender un proceso pueda diseminarse y encargarse de otras tareas o simplemente se logra aligerar laboralmente a las personas que tengan mucha carga de trabajo.

A través de la observación y con la puesta en práctica de sistemas gestores de información, se traslucen varias ventajas tales como: hacer mucho más efectivo el manejo de la información para la toma de decisiones, contar con archivos de información de gran volumen en muy poco espacio facilitando la disponibilidad para posteriores análisis, se reduce el personal que tiene que manejar la información, es decir, se gestiona mejor y con menos personas.

En Cuba se hace uso de las TICs para el desarrollo y auge de la economía. Gracias a los avances que se han experimentado en el camino al desarrollo de la industria del software, se han estado desarrollando software para los distintos sectores de la economía entre los que se encuentra los softwares de gestión.

Una de las instituciones impulsoras de la industria del software en Cuba es la Universidad de las Ciencias Informáticas (UCI), que constituida por diez facultades centrales y tres facultades regionales desarrollan soluciones informáticas para clientes internacionales, nacionales y su funcionamiento propio. Las diez facultades centrales llevan a cabo el proceso de atención, asesoría y control de los trabajos de diploma de los estudiantes que entran en su quinto curso académico (las facultades regionales no cuentan aún con estudiantes en 5to año).

El TD es un trabajo donde el egresado universitario tiene la oportunidad de reconocer la necesidad de desarrollar respuestas ante una situación problemática existente. Estos trabajos se caracterizan por resaltar la capacidad de razonamiento científico que desarrolla el estudiante haciendo uso de los conocimientos, teorías y experiencias que fue adquiriendo en su formación.

A todo este proceso se le debe realizar un seguimiento continuo, controlar los avances y evaluar lo trabajado. Profesores y tesis interactúan en cortes de tesis, que definen señalamientos, recomendaciones y notas; y por último en una predefensa y una defensa, evaluaciones finales del desempeño total en el desarrollo de la tesis.

El problema científico de esta investigación se enmarca en la facultad 9, la cual a partir del curso 2006 - 2007 se enfrenta a este proceso con la inexistencia de una solución informática que automatice el proceso de gestión de los trabajos de diploma. El objetivo general es el desarrollo de un sistema de gestión que automatice dicho proceso. El objeto de estudio que se establece es el proceso

de gestión de los trabajos de diploma de dicha facultad y como campo de acción los sistemas de gestión de información en procesos de realización, seguimiento, control y evaluación de trabajos de diploma.

Para el cumplimiento del objetivo se determinaron las siguientes tareas:

1. Analizar el proceso de gestión de los trabajos de diploma en la facultad.
2. Identificar los subprocesos automatizables.
3. Diseñar el nuevo sistema que apoye la gestión y control del proceso.
4. Implementar el sistema gestor de tesis.

Los métodos científicos teóricos que se emplean para darle solución a estas tareas son:

- ✓ Análisis histórico - lógico: Se evidencia en el análisis de cómo los sistemas de gestión han venido comportándose históricamente, sus características, objetivos, importancia, entre otros aspectos. Además, se utilizó en el análisis del comportamiento del proceso de tesis que se ha realizado en la facultad 9.
- ✓ Analítico - sintético: Se emplea en las tres primeras tareas antes mencionadas y en el análisis y síntesis de la documentación utilizada para la bibliografía.

Los métodos científicos empíricos que se emplean para darle solución a estas tareas son:

- ✓ Observación: Se emplea para identificar algunas características en el proceso de gestión de tesis como la forma de realización, quienes intervienen, que utilizan, entre otras.
- ✓ Entrevista: Se realizaron entrevistas a profesores que interactúan directamente en el proceso de gestión de tesis, para ayudar a identificar algunos de los subprocesos automatizables, y conocer el negocio.

El documento de presentación de resultados de esta investigación está fundamentado en 3 capítulos, los cuales tienen las siguientes responsabilidades:

- En el Capítulo 1, titulado “Fundamentación teórica y referencial”, se realiza una descripción del objeto de estudio y se relacionan conceptos asociados al dominio del problema y del campo de acción. Además se analizan las tendencias de las tecnologías y sistemas relacionados al tema investigado, así como a herramientas y tecnologías útiles para el desarrollo del sistema que se propone.
- El Capítulo 2: “Presentación de la solución propuesta”, comprenderá la presentación del Modelo del negocio, los requerimientos y la descripción del sistema propuesto.
- Ya en el Capítulo 3: “Construcción de la solución propuesta” se hace énfasis en la fase de análisis y diseño, implementación, posteriormente en la de prueba del sistema.

Por último, el contenido complementario se estructura en forma de anexos, y se define un glosario de términos que incluye una explicación de determinadas palabras y siglas utilizadas en este informe de resultados.

Capítulo 1 “Fundamentación teórica y referencial”

1.1 Introducción

Las actividades asociadas a la realización y obtención de resultados, y al seguimiento, control y evaluación a cada trabajo de diploma (TD) tienen un objetivo fundamental dentro del conjunto de todas estas. Las tareas se organizan y se ejecutan paulatinamente, conformando el proceso en toda su amplitud del control de gestión de los TD.

La automatización de este proceso, lleva implícito una serie de conceptos que resultan de vital importancia destacar. Conociéndolos se puede lograr un mejor entendimiento y dominio del campo de acción.

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

1.2 Descripción general del objeto de estudio.

En el curso académico 2008/2009 se aprueba la Resolución Decanal, que plantea los lineamientos para la concepción y el seguimiento de los trabajos de diploma de la facultad 9 desde los polos productivos. Tomando este documento como material de apoyo principal, se realiza la descripción y análisis del objeto de estudio de esta investigación.

El proceso de realización de los trabajos de diploma de la facultad 9 comienza con la definición y aprobación de los perfiles. Los temas investigativos que definen los perfiles son propuestos por los jefes de polos productivos, el comité de tesis es el único órgano facultado para la aprobación y modificación de los perfiles de tesis.

Posteriormente, en caso de que los tesisistas no hayan hecho ninguna solicitud de asesoría y estas no hayan sido aprobadas por el comité de tesis, se asignan los perfiles a cada tesisista y a los profesores que asesorarán el TD. Cada TD es desarrollado por uno o dos tesisistas a lo sumo y asesorado por un grupo de profesores y analistas funcionales externos a la UCI. Estas figuras son los encargados de guiar el trabajo, medir el avance de los tesisistas y son nombradas: tutor(es), cotutor(es), consultor(es), asesor y oponente. Todas

ellas son definidas oficialmente por el Decano de la facultad bajo resolución decanal.

Todo el proceso está dividido en evaluaciones (cortes de tesis), una predefensa y una defensa. Para la realización de los cortes de tesis el comité define los tribunales de tesis. Un tribunal está compuesto por un presidente, un secretario y un vocal y son conformados balanceando la carga de trabajo, para lograr una mayor atención personalizada a cada TD.

Los cortes de tesis derivan una serie de documentos oficiales como vigencia del avance del TD. El tribunal entrega al comité de tesis al finalizar cada corte los siguientes documentos debidamente confeccionados y firmados por los correspondientes:

1. Acta del corte de tesis.
2. Opinión del tutor.
3. Opinión del oponente.
4. Cualquier otro documento que acredite resultados o avales de la investigación que se defiende por el diplomante.

Para la realización de cada corte de tesis el tribunal debe consultar actas de cortes de tesis anteriores y opiniones de los profesores (tutor y oponente) para analizar y evaluar el avance del trabajo y dejar constancia en un acta nueva del corte realizado. Estas evaluaciones son planificadas por el Comité de la facultad. A lo largo de todo el proceso y con el objetivo de lograr un mayor control se necesitan reportes de datos estadísticos del proceso de forma general y específica. Todo termina con el subproceso de predefensa-defensa, donde se define una nota final en un acta oficial de predefensa y de defensa del TD.

Todo lo que se ha mencionado antes que compone el proceso en general: la recopilación y obtención de información, la planificación de las evaluaciones, la realización de los reportes y todas las demás tareas y actividades, se realizan manualmente. La solución informática que se propone para la situación problemática concerniente al objeto de estudio de esta investigación, es precisamente un sistema gestor de toda la información de los TD y las personas que los conforman o lo evalúan. Este sistema cuenta con una aplicación escritorio encaminada a los procesos de la parte administrativa, una

aplicación Web cuyo principal objetivo es informativo y una base de datos para almacenar toda la información manejada.

1.3 Conceptos asociados al dominio del problema.

Son varios los conceptos que se deben especificar para lograr un completo entendimiento de lo que se propone desarrollar en esta investigación.

El principal objetivo es lograr la automatización del proceso de gestión de los TD con un sistema informático. Tomando como apoyo una metodología, se realiza el modelado del software que sirve para visualizar lo que posteriormente se va a construir, empleando lenguajes de programación de alto nivel sobre plataformas y ambientes de desarrollo, amigables y flexibles.

En el análisis del primero de todos los conceptos: *la automatización*, coincidimos con Paz Jiménez cuando plantea que automatizar es: “*Compatibilizar objetivos de eficiencia y flexibilidad en los sistemas productivos mediante la integración de ordenadores y máquinas.*” (Jiménez Sainz; 2002)

El desarrollo de la automatización, ha sido un proceso paulatino que comenzó a finales de la década de los 60 e inicios de los 70 con la aparición de los microordenadores. Desde los 90 hasta la actualidad, se ha alcanzado una madurez en esta disciplina debido a la necesidad cada vez mayor de manejar y controlar grandes volúmenes de información, así como agilizar procesos y optimizar los recursos. “*La automatización es la aplicación de procedimientos automáticos a un proceso.*” (Diccionario de la Lengua Española; 2005) Cuando se habla de la automatización del proceso de *gestión* de tesis se refiere a la ejecución automática mediante la aplicación de herramientas informáticas, de tareas referentes al desarrollo, control, seguimiento y evaluación de los TD.

La *gestión*, propiamente dicha, es un proceso que se lleva a cabo en busca de mejorar la eficacia de las actividades que se desarrollan entorno al objeto de estudio. Es cierto que: “*Gestionar es emplear adecuadamente los recursos disponibles para lograr el objetivo que se haya trazado, es alcanzar los propósitos a través de la acción coordinada de personas.*” (Gestión Institucional. 2006) La organización, ejecución, control y evaluación de la información, constituyen las fases de la gestión. Con la puesta en marcha de cada una de ellas se puede lograr una gestión efectiva.

Los autores coinciden en que: *“La gestión permite aumentar progresivamente la efectividad, la equidad y la eficiencia de los procesos y alcanzar propósitos utilizando una menor cantidad de recursos, incluyendo el tiempo.”* (Restrepo; 2002)

En todo proceso de desarrollo de software es necesario el uso de alguna *metodología* de desarrollo, pues toda organización debe tener un plano por el cual apoyarse. Una decisión muy importante a tomar sería entonces cuál de todas las metodologías puede ser usada.

Es cierto que: *“Una metodología para el desarrollo de un proceso de software no es más que un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos.”* (Maddison. 1983).

Además, en las metodologías se especifica:

1. ¿Cómo dividir un proyecto en etapas?
2. ¿Qué tareas se llevarán a cabo en cada etapa?
3. ¿Qué salidas se producen y cuándo deben producirse?
4. ¿Qué restricciones se aplican?
5. ¿Qué herramientas van a ser utilizadas?
6. ¿Cómo se gestiona y controla el proyecto?

Empleando una buena metodología se logra un mejor modelado del software que se pretende desarrollar. El *modelado de software* permite al ingeniero visualizar el sistema que se va a construir. El Lenguaje Unificado de Modelado de Software (UML), fue adoptado en noviembre de 1997 por OMG (Object Management Group). Los autores coinciden con García y Moreira cuando afirman que: *“UML se ha convertido en un estándar para visualizar, especificar y documentar los modelos que se crean durante la aplicación de un proceso de software.”* (García; Moreira; 2005)

Otro concepto válido a aclarar es el de *aplicación web*. *“Una aplicación web es un sistema que permite a un usuario final acceder a una parcela de información contenida en el universo al que hace referencia la definición del W3C. (Word Wide Web Consortium)”* (Colado Rodríguez, 2003)

Teniendo en cuenta el anterior concepto, los autores definen que las aplicaciones web son fragmentos de código que se ejecutan en el servidor Web cuando se realizan ciertas peticiones en el cliente web. El cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Java Script. Mientras que el servidor Web es quien proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts).

La aplicación Web interactúa además con una *base de datos* que integra también el sistema que se propone.

“Una base de datos es el lugar donde se guardan los datos en reposo y al cual acceden las diferentes aplicaciones de una organización dada.” (Rozic; 2004)

Basándose en este concepto, los autores definen a una base de datos o banco de datos como el conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Existen unos programas denominados sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Se puede decir que: *“Un SGBD es un software que proporciona servicios para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en la base de datos de forma segura y eficiente y actúa como un intermediario entre las aplicaciones y los datos.”* (Sicilia; 2008)

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos.

Es importante destacar que: *“Los SGBD ayudan a realizar las siguientes acciones:*

- *Definición de los datos.*
- *Mantenimiento de la integridad de los datos dentro de la base de datos.*
- *Control de la seguridad y privacidad de los datos.*
- *Manipulación de los datos”* (Álvarez; 2007)

Por otra parte se encuentran los servidores de bases de datos, *“los servidores de bases de datos surgen con motivo de la necesidad de manejar grandes y complejos volúmenes de datos”* (Menéndez Barzanallana; 2009) al tiempo que requieren compartir la información con un conjunto de clientes (que pueden ser tanto aplicaciones como usuarios) de una manera segura.

Para crear cualquier sistema se emplean los *lenguajes de programación*. Estos son conocidos también como código de máquinas o lenguaje de máquinas y se consideran herramientas que permiten construir programas.

Los lenguajes pueden ser de alto o bajo nivel. En los de bajo nivel las instrucciones son simples y cercanas al funcionamiento de la máquina, como por ejemplo el código máquina y el ensamblador. En los lenguajes de alto nivel hay un alto grado de abstracción y el lenguaje es más próximo a los humanos, como por ejemplo LÉXICO, PASCAL, COBOL o JAVA.

Estos lenguajes se utilizan sobre ambientes de desarrollo de software que constituyen herramientas que ayudan a los programadores a desarrollar software sobre entornos más amigables. Es decir, aquellos en los que el programador puede acceder con el menor esfuerzo a diferentes recursos como editores, compiladores, herramientas de análisis, etc.

Para lograr mayor claridad de la relación entre los conceptos analizados, se puede consultar el mapa conceptual (anexo 1).

1.4 Análisis de otras soluciones existentes.

Luego de hacer un estudio preliminar del negocio se decidió indagar por soluciones informáticas que se ajustaran a la solución del problema. En internet se utilizaron los criterios de búsqueda “Sistema de Gestión”, “Gestión de Tesis”, “Management Systems”, “Information systems”, en algunos casos se encontraron soluciones pero no presentaban los requerimientos necesarios.

En la universidad se detectaron 3 soluciones anteriores, todas en otras facultades. Las soluciones son las siguientes:

1. Propuesta de diseño de una aplicación Web para la gestión de perfiles de los trabajos de diploma. (Carcassés; Estrada. 2007)
2. Aplicación Web para el control del proceso de tesis de grado en la facultad 8. (Heredia; Paz. 2008)

3. Automatización de la gestión y reproducción de las tesis en la UCI. (Núñez. 2008)

La solución 1 presenta los requerimientos para la gestión y revisión de los perfiles, la conformación de tribunales y comisiones de tutores, el registro y verificación de los usuarios y la elaboración de reportes. No se está utilizando en el proceso de tesis.

El solución 2 presenta los requerimientos para la autenticación y gestión de usuarios, la propuesta, revisión y rechazo de perfiles de tesis, la gestión de evaluaciones de las tesis y algunos reportes. El tutor del trabajo de diploma es el único que permanece en la universidad y no tiene consigo la solución.

El solución 3 presenta requerimientos para gestionar el banco de problemas, gestionar los perfiles, gestionar notas de cortes, impresión del documento, publicación del cronograma de discusión y el establecimiento del acceso a la aplicación. No se está usando en el proceso de tesis.

En ninguno de los casos se trabajó en requerimientos para facilitar planificación de evaluaciones y solo en la solución 2 y 3 le dan seguimiento a las mismas. Tampoco los presenta para que las figuras de tesis autorizadas den su opinión respecto al trabajo de diploma. Además en ninguno de los casos se puede continuar o extender el desarrollo porque la razón requerimientos – documentación - implementación no lo favorece.

Se concluye que la solución más apropiada es el desarrollo de una solución propia para la facultad 9. El diseño de la misma debe permitir en su momento la generalización para el uso en el resto de las facultades de la Universidad.

1.5 Las TICs. Metodologías de desarrollo de software.

El fin del siglo pasado estuvo marcado por una convergencia entre la electrónica, la informática y las telecomunicaciones, que constituye el núcleo central de la transformación multidimensional que experimenta la economía y la sociedad, imponiéndole al ser humano modificar no sólo sus hábitos y patrones de conducta, sino incluso, su forma de pensar.

“Diversos estudios previeron ya desde la década del setenta, incluso antes, que las tecnologías microelectrónicas y de telecomunicación, conocidas como Nuevas Tecnologías de la Información y las Comunicaciones (TICs), transformarían a corto plazo las estructuras y sistemas de producción y

servicios en los países industrializados, agudizando al mismo tiempo la desigualdad y dependencia del Tercer Mundo.” (Revista Cuba Socialista. 2006)

Esto constituye una dura realidad prevista por nuestro Partido Comunista de Cuba en la revista Cuba Socialista. Las tecnologías se desarrollan a gran velocidad, y a la misma velocidad crecen la desigualdad, la pobreza y el subdesarrollo en los países pobres.

Coincidimos con que las TICs son un término que se utiliza actualmente para hacer referencia a una gama amplia de servicios, aplicaciones, y tecnologías, que utilizan diversos tipos de equipos y programas informáticos, y que a menudo se transmiten a través de las redes de telecomunicaciones.

La Universidad de las Ciencias Informáticas surge al calor de la batalla de ideas con el principal objetivo de formar especialistas en esta rama, promulgando el desarrollo y auge de la informática y el desarrollo del software en Cuba; y sentando las bases para un no tan lejano mercado nacional e internacional de productos informáticos enteramente cubanos. Son precisamente estas tecnologías, las TICs, las herramientas disponibles para la labor de informáticos que se debe realizar.

Las metodologías de desarrollo de software surgieron por la necesidad de controlar y documentar proyectos cada vez más complejos. Los autores de esta investigación definen a una metodología como el conjunto de procedimientos y herramientas que apoyan el desarrollo de un software.

Las características de cada proyecto exigen que el proceso sea adaptable. No existe una metodología de software universal, se conocen muchas metodologías con variadas características y ventajas propias. Según la norma 1074 de IEEE toda metodología de desarrollo de software debe incluir la forma en que se va a realizar la captura de requisitos, el diseño, la implementación y prueba. Esto con el propósito de contar con un marco de trabajo bien definido que permita obtener productos con calidad, que satisfagan los requisitos del cliente y se desarrollen en el tiempo estimado y con los costos presupuestados. En la actualidad se cuenta con una buena cantidad de metodologías ágiles como Extreme Programming, Scrum, Cristal Methods y Feature Driven Development y con metodologías tradicionales o robustas como Rational Unified Process (RUP), Microsoft Solutions Framework (MSF) y Métrica.

Una de las más destacadas dentro de las ágiles es la Programación Extrema (Extreme Programming, XP). XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Además se define para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Craig Larman señala como factores negativos de XP la ausencia de énfasis en la arquitectura durante las primeras iteraciones (no hay arquitectos en XP) y la consiguiente falta de métodos de diseño arquitectónico.

Dentro de las metodologías fuertes la más destacada es el Proceso Unificado de Modelado (RUP). RUP sugiere su uso para proyectos nuevos o actualizaciones de sistemas existentes, y se recomienda adoptarlo en forma gradual. Es un proceso de desarrollo de software configurable que se adapta a variados proyectos teniendo en cuenta tamaños y complejidades. Por sus significativas características y ventajas expuestas a continuación, es que se decide tomar RUP como metodología de desarrollo para el sistema que se propone.

1.6.1 El Proceso Unificado de Desarrollo de Software (RUP) y el Lenguaje Unificado de Modelado (UML).

El Proceso Unificado de Desarrollo de Software es uno de los procesos más generales de los existentes actualmente, está pensado para adaptarse a cualquier proyecto y contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un software.

“RUP se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental”. (Jacobson; Booch; Rumbaugh. 2000). Estas cualidades son válidas para el desarrollo del sistema que se propone en esta investigación. Además incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento a lo largo del proceso).

Aunque RUP es un proceso de desarrollo de software genérico, se concibió en gran medida para el desarrollo de sistemas basados en programación orientada a objetos. Por ejemplo se suele emplear RUP en proyectos de programación en lenguajes como Java o .NET. Este proceso define en cada momento del ciclo de vida del proyecto, qué artefactos, con qué nivel de detalle, y por qué rol, se deben crear.

Estas actividades se han organizado en grupos lógicos, definiéndose 6 flujos de trabajo principales:

1. *“Modelamiento del negocio*
2. *Requerimientos*
3. *Análisis y diseño*
4. *Implementación*
5. *Prueba (Testeo)*
6. *Instalación.”* (Pressman; 2005)

Entre las ventajas del RUP podemos encontrar:

1. Mitigación temprana de posibles riesgos altos.
2. Progreso visible en las primeras etapas.
3. Temprana retroalimentación que se ajuste a las necesidades reales
4. Gestión de la complejidad.
5. Conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.

Es necesario modelar Software debido a la dimensión y complejidad de los sistemas, a que no son hechos por una sola persona, para un mayor entendimiento del desarrollo y para lograr representaciones simples como medio de manejar la complejidad.

Los modelos ayudan a visualizar cómo es o cómo queremos que sea un sistema. Además permiten especificar su estructura o comportamiento, proporcionan plantillas que nos guían en la construcción y documentan las decisiones que se adoptan en el transcurso del desarrollo del software.

Es cierto que: *“UML es un lenguaje visual de modelado para visualizar, especificar, construir y documentar los artefactos de un sistema software”* (González; Fuentes; Llorens; 2004). Puede además considerarse un lenguaje

gráfico que estandariza la forma de crear diagramas, el significado preciso de los mismos, y las relaciones existentes entre ellos. Este lenguaje facilita la tarea de dibujar diagramas correctos y coherentes.

Aprovechando toda esta gama de posibilidades y ventajas que nos brinda RUP y a la vez UML, se seleccionan para el desarrollo del sistema que se propone. La característica iterativa e incremental de RUP permite diseñar e implementar un sistema en iteraciones y versiones, sobre la marcha del proceso de desarrollo de los TD, y así lograr una verdadera retroalimentación que se ajuste a las necesidades de los usuarios mediante la observación de lo que va aconteciendo en pleno desarrollo del proceso de control de gestión de los TD. Además de que podemos minimizar la complejidad y costo de desarrollo y emplear la estructura consistente que brinda RUP, que abarca todas las fases de vida de un producto (Inicio, elaboración, construcción, transición).

1.6.2 Visual Parading como herramienta computacional de modelado.

La introducción de las herramientas CASE ha permitido que los diagramas puedan ser fácilmente creados y modificados, mejorando la calidad de los diseños de software. Entre las herramientas de este tipo, las más destacadas y que presentan amplias utilidades para lo que se necesita modelar se pueden mencionar al Rational Rose Enterprise y al Visual Parading for UML.

“Rational Rose Enterprise” es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Esta herramienta ofrece un lenguaje de modelado común que agiliza la creación del software. Además se tiene en cuenta que: *“Para su uso se pueden obtener licencias de usuario autorizado (usuario único) o con licencias flotantes (usuarios simultáneos).”* (Rational Rose. 2005)

Con la evolución del UML a través de los años y ahora en su serie de versiones 2.x., *“Visual Parading for UML”* se ha actualizado rápidamente en sincronía con el nuevo desarrollo del UML 2.1, proporcionando un entorno de modelado visual en el que se reúne hoy el software, la tecnología y las necesidades de comunicación a la misma vez.

Visual Paradigm (VP), tiene una extensión de funcionalidades exquisita, un tratado del modelo UML muy claro, y una parte dedicada para la gestión de requisitos no funcionales. A pesar de que este programa no es de código libre,

tiene una versión gratuita para uso no comercial la denominada Community. Es una plataforma de modelado diseñada para el modelado UML con la que desarrolladores pueden crear diagramas mucho más rápidos y más intuitivos que cualquier otra en el mercado. Es cierto que: *“VP facilita la organización de los diagramas con un solo clic del ratón utilizando la distribución automática y permite incorporar formas personalizadas mediante el uso de símbolos o icono de imagen de importación”* (Hernán. 2006)

Trabajando con esta herramienta se puede conseguir una integración para Eclipse/IBM WebSphere, Borland JBuilder, NetBeans IDE/Sun ONE, IntelliJ IDEA, Oracle JDeveloper y BEA Weblogic workshop. Cabe destacar el número considerable de lenguajes que soporta esta plataforma de modelado tales como: Java, C ++, CORBA IDL, PHP, XML Schema, Ada y Python. Además, apoya la generación de código C #, VB. NET, Object Definition Language (ODL), Flash ActionScript, Delphi, Perl, Objective-C, y Ruby.

VP soporta la importación y exportación de XMI de versiones 1.0, 1.2 y 2.1 y permite manejar grandes y complicadas estructuras de un proyecto. (González H. 2008)

Por estas características y amplias ventajas que ofrece esta herramienta, su uso amplio vinculado a plataformas y a lenguajes de programación tenidos en consideración para usarse en el sistema propuesto y por el grado de experiencia que tienen los desarrolladores en el trabajo con el VP se decide seleccionarlo para realizar el modelado del sistema.

1.7 Las aplicaciones Web

Las aplicaciones Web como parte de las TIC están cada vez más presentes en Internet, evolucionando y haciéndose cada vez más dinámicas y ajustables a las necesidades de sus usuarios, no solo como simples maquetas de exposición de información sino como verdaderos elementos integradores de las potencialidades de las aplicaciones de escritorios, multimedia, hipermedia y los conceptos positivos de la Web de las generaciones anteriores.

Definitivamente podemos afirmar que: *“La web es hoy un medio extraordinariamente flexible y económico para la comunicación, el comercio y los negocios, ocio y entretenimiento, acceso a información y servicios, difusión de cultura, etc.”*(Castell P. 2008)

Las aplicaciones Web ofrecen grandes ventajas que pueden ser aprovechadas por muchas organizaciones, sobre todo ahora que la globalización es una realidad. Entre las ventajas, se pueden mencionar las siguientes:

Actualización: Están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación.

Inmediatez de acceso: No necesitan ser descargadas, instaladas y configuradas, simplemente el cliente accede a una cuenta a través de Internet y están listas para trabajar sin importar cuál es la configuración o el hardware y a cualquier hora.

Menos requerimientos de memoria: Tienen más demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, dejando más espacio para correr múltiples aplicaciones al mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.

Múltiples usuarios concurrentes: Pueden realmente ser utilizada por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta.

Los datos son más seguros: Los usuarios tienen mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora.

Toda esta gama de facilidades que nos brinda la tecnología web permite mejorar y lograr una mayor fluidez en el proceso de desarrollo y evaluación de los TD. Una aplicación web ofrece funcionalidades a tesis, profesores de tesis, tribunales y miembros de comités de tesis. Además viabiliza el manejo de la información derivada de todo el proceso, facilitando el acceso a la misma por un mayor número de usuario a la vez y socializando resultados y planificaciones de evaluaciones. Estas características definieron la decisión de llevar el proceso a la web.

1.8 Arquitectura de una aplicación Web

Las aplicaciones web se han convertido en pocos años en complejos sistemas con interfaces de usuario cada vez más parecidas a las aplicaciones de escritorio, dando servicio a procesos de negocio de considerable envergadura y estableciéndose sobre ellas requisitos estrictos de accesibilidad y respuesta. Esto ha exigido reflexiones sobre la mejor arquitectura y las técnicas de diseño más adecuadas.

La arquitectura de una aplicación es la vista conceptual de la estructura de esta. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código. (González C. 2001)

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. En este tipo de arquitectura la lógica del negocio reside en la PC cliente pero tiene que estar implementada en el servidor mediante los llamados disparadores o procedimientos almacenados. Los ambientes de dos capas requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se les hacen cambios. Esto se debe al hecho de que la mayoría de la aplicación lógica existe en la estación de trabajo del cliente. (Duque M; 2008)

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución. Dentro de las soluciones estudiadas los patrones mayormente utilizados son el Modelo Vista Controlador (MVC) y la arquitectura por capas.

Los autores coinciden con que: *“El MVC tiene como características principales que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.”* (Patrón MVC; 2008)

Es un patrón arquitectónico usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente.

Entre sus ventajas cabe destacar el soporte de vistas múltiples. Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los datos de manera simultánea. Algunas de las desventajas observadas es que la cantidad de archivos a mantener y desarrollar se incrementa considerablemente; la curva de aprendizaje del patrón de diseño es más alta que usando otros modelos más sencillos y la separación de conceptos en capas agrega complejidad al sistema.

Por otra parte se encuentra la arquitectura de tres capas, la cual introduce una capa intermedia al proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. Se instala una interfaz de usuario en la computadora del cliente. La tercera capa generalmente es el sistema de administración de la base de datos que se localiza en un servidor separado, donde son almacenados los datos requeridos por la capa intermedia. En esta última se implementa la lógica del negocio y proporciona un ambiente donde miles de usuarios pueden estar conectados simultáneamente.

Entre las ventajas en la arquitectura de tres capas se destacan las siguientes:

- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación sólo necesita transferir parámetros a la capa intermedia.
- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, la estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular, esto puede reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- Separando la aplicación de la base de datos, hace más fácil utilizar nuevas tecnologías de agrupamiento y balance de cargas.
- Separando la interfaz del usuario de la aplicación, libera de gran procesamiento a la estación de trabajo y permite que las actualizaciones de la aplicación sean centralizadas en el servidor de aplicaciones.

Como principales desventajas hay que aceptar que los ambientes de tres capas pueden incrementar el tráfico en la red y requerir más balance de carga y tolerancia a las fallas. Además se conoce que los exploradores actuales no son todos iguales, ha sido muy lenta la estandarización entre diferentes proveedores y sucede que muchas organizaciones son forzadas a escoger uno en lugar de otro, mientras que cada uno ofrece sus propias y distintas ventajas. En este trabajo se propone utilizar la arquitectura de tres capas para la realización de la aplicación que dará solución al problema ya que supera en muchos aspectos a la de dos capas, la arquitectura de tres capas soluciona las crecientes necesidades de los clientes, incrementa la habilidad para responder a los cambios, posibilita la reutilización de código, simplifica el mantenimiento y hace más fácil la migración a nuevas plataformas, ofreciendo soluciones viables. Sin duda alguna esta arquitectura cumple con las expectativas de crecimiento de la flexibilidad y de la escalabilidad; así como de una disminución del tiempo de desarrollo.

1.9 Principales lenguajes y tecnologías de la Web.

Son muchos y muy variados los lenguajes de programación para Web que nos brindan las nuevas tecnologías. Cada uno de ellos, con sus particularidades, ventajas y desventajas, se vislumbran en la amplia lista de recursos y herramientas que están al alcance de los desarrolladores de software y más específicamente de aplicaciones Web en todo el mundo. Gestores de bases de datos y servidores web también se insinúan en el amplio desarrollo de sistemas, para el manejo, seguridad y soporte de la información almacenada.

1.9.1 Lenguajes de Programación para la Web

Entre los principales lenguajes y tecnologías de la Web están incluidos el lenguaje de marcas de hipertexto o lenguaje de enlaces hipertexto (HTML), el lenguaje de marcas extensible (XML), las hojas de estilo en cascadas (CSS), java, java script, java server pages, PHP, C# y ASP.NET

“PHP, también denominado PHP Hypertext Preprocessor, es un lenguaje de programación interpretado de alto nivel para Internet, muy similar en su sintaxis al Lenguaje C, Java o Perl. Este lenguaje no compila como el lenguaje C, pues es un intérprete, por lo tanto cada vez que debe ejecutar un programa, lo interpreta verificando toda su sintaxis.” (Ramos M. 2004) Indiscutiblemente

PHP es un poderoso lenguaje intérprete, incluido como parte de un servidor Web en forma de módulo, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier comando que sea ejecutado en un servidor Web sea inseguro por naturaleza.

C#, sin embargo resulta un lenguaje mucho más seguro. Para empezar: *“C# es un lenguaje con una estricta comprobación de tipos, en el que no se efectúan automáticamente conversiones potencialmente peligrosas”* (Charte O. 2002), y que además da un valor inicial por defecto a todas las variables, independientemente de su ámbito.

C# es una versión avanzada de C y de C++ y se ha diseñado especialmente para el entorno .NET. Es cierto que: *“C# es un nuevo lenguaje orientado a objetos empleado por programadores de todo el mundo para desarrollar aplicaciones que se ejecuten en la plataforma .NET”*. (Alarcón A. 2003)

A continuación se recoge de manera resumida las principales características de C#:

- ✓ Dispone de todas las características propias de cualquier lenguaje orientado a objetos: encapsulación, herencia y polimorfismo.
- ✓ Ofrece un modelo de programación orientada a objetos homogéneos, en el que todo el código se escribe dentro de clases y todos los tipos de datos, incluso los básicos, son clases que heredan de System.Object.
- ✓ Permite definir estructuras, que son clases un tanto especiales: sus objetos se almacenan en pila, por lo que se trabaja con ellos directamente y no referencian al montículo, lo que permite accederlos más rápido.
- ✓ Es un lenguaje fuertemente tipado, lo que significa que se controla que todas las conversiones entre tipos se realicen de forma compatible, lo que asegura que nunca se acceda fuera del espacio de memoria ocupado por un objeto.
- ✓ Tiene a su disposición un recolector de basura que libera al programador de la tarea de tener que eliminar las referencias a objetos que dejen de ser útiles.
- ✓ Incluye soporte nativo para eventos y delegados.

- ✓ Incorpora propiedades, que son un mecanismo que permite el acceso controlado a miembros de una clase tal y como si de campos públicos se tratasen.
- ✓ Permite la definición del significado de los operadores básicos del lenguaje (+, -, *, &, ==, etc.) para nuestros propios tipos de datos, lo que facilita enormemente tanto la legibilidad de las aplicaciones como el esfuerzo necesario para escribirlas.

Los autores coinciden con que: *“Si se conoce C++, Java o incluso C, empezar a programar en C# resulta fácil. C++ es el lenguaje orientado a objetos más usado en la actualidad, de ahí que parezca la opción más lógica como punto de partida para la evolución hacia otros lenguajes más avanzados.”* (Charte O.; 2002)

Con estas características y ventajas, C# se convierte en un buen candidato para la implementación del sistema que se propone. Se pretende que el Sistema gestor de TD, comience a usarse en el curso académico 2008-2009, los desarrolladores han estudiado y poseen mayor experiencia en el lenguaje C#, por tanto se gana en el costo beneficio/tiempo al seleccionarlo para la construcción del sistema.

1.9.2 La plataforma ASP.NET como soporte de la implementación

“El lenguaje de programación C# jugó un papel fundamental en el desarrollo de .NET Frameworks”. (Mayo; 2002) Esta afirmación es válida tenerla en cuenta.

ASP.NET es una nueva estructura de programación que permite el desarrollo de aplicaciones Web y fue diseñado con el objetivo de que dichas aplicaciones respondan rápidamente a las solicitudes de los usuarios, sin importar la cantidad de datos que se estén procesando en el servidor. Cabe destacar que: *“ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció el 20 de enero del 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP)”.* (Requisitos de la plataforma ASP.NET. 2006)

ASP.NET es compatible con Windows 2000 (Professional, Server y Advanced Server), con Windows XP Professional y con la familia de Windows Server 2003, tanto para las aplicaciones de cliente como de servidor. Es la nueva versión de la tecnología de Microsoft para la creación dinámica de páginas web conocida como ASP (Active Server Pages). Es totalmente orientado a objetos y puede ser escrito en cualquier lenguaje soportado por el .net Framework, es decir: VB.net; C# y JScript.net.

Podemos mencionar diversas ventajas, entre las cuales se destacan:

- Rendimiento: la aplicación se compila una sola vez en el lenguaje nativo, y luego, en cada petición tiene una compilación Just In Time, es decir se compila desde el código nativo, lo que permite mucho mejor rendimiento. También permite el almacenamiento del caché en el servidor.
- Rapidez en programación: mediante diversos controles, podemos con unas pocas líneas y en menos de 5 minutos mostrar toda una base de datos y hacer rutinas complejas.
- Servicios Web: trae herramientas para compartir datos e información entre distintos sitios.
- Seguridad: tiene diversas herramientas que garantizan la seguridad de nuestras aplicaciones.

Resumiendo, existe mayor velocidad, mayor potencia, mayor seguridad, mayor facilidad de mantenimiento y herramientas de trabajo, pero las ventajas no paran aquí. A continuación se enumeran algunas otras que no tienen ASP, PHP o JSP:

- Caché: se puede almacenar en la caché del servidor tanto páginas enteras, como controles personalizados o simples variables. En páginas críticas con mucha carga de base de datos es muy útil almacenar datos de la base de datos en la caché, reduciendo enormemente el consumo de recursos.
- Carpetas especializadas, como por ejemplo app_code que compila automáticamente las clases que se alojan en él, o la carpeta app_theme que alojan ficheros que marcan los temas de estilos de la Web.

- Los archivos de configuración Web.config y Machine.config permiten realizar operación de configuración en ficheros que hasta ahora había que realizar en el servidor.
- La adaptación automática del código devuelto a los dispositivos que le acceden. Una misma página puede servirnos para el Internet Explorer, para el Pocket Internet Explorer desde una PDA o para un navegador de un móvil cualquiera.
- La eliminación total de la necesidad de frames (marcos) con la introducción de las masterpages (páginas maestras).
- La multitud de controles Web que permiten mucha funcionalidad con poco código. Desde enlace con las bases de datos o enseñar fácilmente todos los datos, hasta simples etiquetas, hiperenlaces o generadores de imágenes.
- Se puede utilizar hasta cuarenta lenguajes distintos para el desarrollo en ASP.NET, aunque en el 95% de las aplicaciones se usa C#, VB.NET o J#. (Vázquez Z. 2008)

Las razones de elegir la versión 2.0 en lugar de la ya madura versión 1.1 es una gran mejora en todos los aspectos en general, pero se destaca la comunicación con base de datos, ADO.NET 2.0, C# 2.0 y el incremento de los controles Web.

1.9.3 VisualStudio.Net como plataforma de desarrollo.

Aunque en principio sería posible desarrollar una aplicación de ventanas con un simple editor de texto y utilizando las clases de los espacios de nombres adecuados (principalmente System.WinForms), mediante la observación y la práctica se puede llegar a la conclusión de que ello implicaría invertir mucho tiempo en la escritura del código encargado de generar la interfaz de nuestra aplicación, tiempo que podríamos estar aprovechando para centrarnos en resolver los problemas relativos a su lógica y no a su aspecto.

Por tanto se decide utilizar Visual Studio.Net para realizar la implementación del sistema pues esta herramienta permite diseñar la interfaz de la aplicación de manera visual, sin más que arrastrar con el ratón los elementos que necesitemos (botones, lista de selección, etc.) sobre las posiciones adecuadas

en la ventana de nuestra aplicación. También incluye otras facilidades para el desarrollo, como una ventana de propiedades desde la que se puede modificar los valores de las propiedades de cada objeto sin tener que escribir código, un depurador de código gráfico, un editor de códigos inteligente que puede detectar nuestros errores de sintaxis instantáneamente, etc.

1.9.4 Principales gestores de bases de datos

Oracle, Microsoft SQL Server, MYSQL y PostgreSQL, son destacados gestores de bases de datos que se utilizan en la actualidad.

PostgreSQL es un servidor de base de datos relacional y libre, tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, y almacenamiento de objetos de gran tamaño. Los autores unen su criterio al de Medina cuando afirma que: *“PostgreSQL es un potente sistema de base de datos relacional libre (open source, su código fuente está disponible) liberado bajo licencia BSD. Tiene más de 15 años de activo desarrollo y arquitectura probada que se ha ganado una muy buena reputación por su confiabilidad e integridad de datos”*. (Medina. 2008)

Este gestor se destaca en ejecutar consultas complejas sobre vistas, subconsultas y joins (saltos) de gran tamaño. Además permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. PostgreSQL tiene una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y algo muy importante es que dicha herramienta es multiplataforma. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial y tiene mejor soporte para vistas y procedimientos almacenados en el servidor.

Entre las principales mejoras que PostgreSQL manifiesta por encima de los demás gestores de bases de datos, están que los bloqueos de tabla han sido sustituidos por el control de concurrencia versiones, el cual permite a los accesos de sólo lectura, continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad mientras la base de datos permanece disponible para consultas. Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones valores en los campos (constraints) y disparadores (triggers).

Actualmente la utilización de este gestor de base de datos en el mundo continúa siendo una de las de mayor demanda por el gran alcance que tiene y las innumerables ventajas que proporciona, entre ellas está que opera en una arquitectura cliente/servidor, y trabaja en múltiples plataformas, pues existen versiones para Windows y Linux. Es rápido, confiable y posee un mayor rendimiento, mejores utilidades de administración y mejor control de acceso a los usuarios. Por todas estas razones se decide utilizar como gestor de base de datos al PostgreSQL.

1.9.5 Servidores Web.

Un servidor Web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP).

Apache

Apache es un servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. El proyecto del servidor del HTTP de Apache es un esfuerzo de desarrollar y de mantener un servidor del HTTP de código abierto para los sistemas operativos modernos incluyendo UNIX y Windows NT.

Internet Information Server (IIS)

Es una buena solución a las necesidades de las empresas y usuarios de enviar y recibir la información no sólo de sus clientes si no también en el entorno de la empresa. IIS engloba una serie de herramientas administrativas que le permitirán controlar sitios Web, FTP, SMTP (correo saliente) y servicio de noticias. Dispone también del soporte necesario para crear páginas dinámicas (ASP), lenguaje de aplicaciones para Internet bastante extendido y que permite la conexión y acceso a bases de datos consiguiendo aplicaciones Web dinámica, escalables y dotando a los sitios Web de una mayor complejidad y rendimiento.

Estas condiciones resultan necesarias para nuestro sistema, por lo que resulta IIS idóneo para montar el servidor de la aplicación Web.

1.10 Otras herramientas útiles en el desarrollo y producción de software.

Existen muchas herramientas necesarias y útiles para desarrollar un software con calidad, efectividad y eficacia. Muchas veces se pierden recursos valiosos en el desarrollo de sistemas por falta de coordinación entre los integrantes del equipo, por poca reutilización de contenidos o por pérdidas irrecuperables de información.

Subversion es precisamente una de estas herramientas muy útiles para desarrolladores. Es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular Concurrent Versions System (CVS).

Tal como señala la muy usada Wiki: *“Subversion es un software libre bajo una licencia de tipo Apache/BSD y se le conoce también como SVN, por ser ese el nombre de la herramienta de línea de comandos que utiliza.”* (Wikipedia. 2008)

Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

El repositorio Subversion es un sistema centralizado para compartir información. En su núcleo está un repositorio, que es un almacén central de datos. El repositorio almacena información en forma de un árbol de ficheros, una jerarquía típica de ficheros y directorios. Cualquier número de clientes se conectan al repositorio, y luego leen o escriben esos ficheros. Al escribir datos, el cliente hace que la información esté disponible para los otros; al leer los datos, el cliente recibe la información de los demás.

Una de las características más significativas es la siguiente expresada por Guerra y Tula: *“Lo que hace al repositorio de Subversion especial es que recuerda todos los cambios que alguna vez se hayan escrito en él: cada cambio en cada fichero, e incluso los cambios en el propio árbol de directorios, como el añadir, borrar o reorganizar ficheros y directorios”.* (Guerra R.; Tula M. 2008)

TortoiseSVN es un cliente Subversion, implementado como una extensión al Shell de Windows. Es un software libre liberado bajo la licencia GNU GPL y maneja ficheros y directorios a lo largo del tiempo. Estos ficheros se almacenan

en un repositorio central, el cual es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. Esta es la razón por la que mucha gente piensa que Subversion, y los sistemas de control de versiones en general, son una especie de “máquinas del tiempo”. (Guerra R.; Tula M. 2008)

Evidentemente la aplicación de estas tecnologías facilita el desarrollo de esta investigación, nos ahorra tiempo y recursos.

1.11 Conclusiones.

En este capítulo, después de haber realizado una investigación teórica para respaldar la labor a realizar, se realiza un análisis del objeto de estudio y se puntualizan los conceptos necesarios referentes al campo de acción de manera detallada para entender la temática de este trabajo .

Además se determina que el avance de los medios y tecnologías enfocados a la Web se encuentran en un desarrollo vertiginoso y acelerado, lo que permite que se puedan seleccionar de una gran gama de productos y tecnologías, los más adaptables al sistema que se pretende diseñar. Metodologías, ambientes de desarrollo y lenguajes de programación, son definidos y seleccionados atendiendo a los requerimientos del sistema .

Capítulo 2 “Presentación de la solución propuesta”

2.1 Introducción

Antes de comenzar a desarrollar un sistema es necesario comprender el entorno de trabajo donde se va aplicar la herramienta informática, con el estudio de los procesos que en él se desarrollan.

Primeramente se realiza la modelación del negocio, durante esta etapa se definen y se describen los actores y trabajadores del negocio; se analizan los requerimientos funcionales, los cuales serían las funcionalidades a implementar en el sistema. En esta temática es donde entra todo lo referente al sistema, definición de los actores del sistema, diagrama de casos de usos del sistema así como las especificaciones de cada uno de estos casos de usos para tener una idea de cómo interactúan actor-sistema. Son precisamente tales elementos los que se desarrollan en este capítulo.

2.2 Modelo de Negocio

Cuando se comienza a desarrollar un sistema para una organización, empresa e institución es necesario comprender todos los procesos que en estas tienen lugar. Cada uno de estos procesos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado.

Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio que determinan la estructura de la información y las políticas de la organización. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades, roles y reglas de negocio.

El modelo del negocio se realiza además con el propósito de comprender las características y actividades que se llevan a cabo en el contexto a automatizar. Cada uno de los epígrafes que siguen a continuación describe textualmente y gráficamente etapas del modelo del negocio.

2.2.1 Procesos del negocio

Un proceso del negocio es el conjunto estructurado de las actividades que han sido diseñadas para producir un resultado específico para un cliente. Estos, son de gran importancia ya que definen los límites del proceso de modelado posterior.

Del proceso de desarrollo, control, seguimiento y evaluación de los TD se pueden definir los siguientes subprocesos (procesos del negocio):

1. Recopilación de información.

En este primer proceso se realizan las solicitudes de asesoría (tutorías, cotutorías, consultorías, oponencias y asesorías). Además que se solicitan los perfiles. El actor principal de este proceso es el tesista, quien debe realizar las solicitudes pertinentes y entregarlas a los responsables de estas (líder del proyecto al que pertenece el tesista o al jefe del polo productivo).

2. Oficialización del proceso de Tesis.

En la oficialización del proceso interviene el órgano responsable a nivel de base, o sea el Comité de tesis. Esta oficialización consiste en la aprobación de las solicitudes antes realizadas (las solicitudes de asesoría y las solicitudes de perfiles)

3. Estructuración de tribunales y grupos de tesis.

Luego de haber sido oficializado perfiles, profesores y tesistas asociados entre sí, lo siguiente es precisamente la conformación de tribunales de tesis. Estos se estructuran con tres miembros ya antes mencionados: presidente, secretario y vocal. Las tesis se agrupan por polos productivos y se asignan a los tribunales.

4. Realización de cortes de tesis.

Los cortes de tesis serán las evaluaciones que definen el avance de los TD y una nota del periodo que se evalúa. Generan documentos oficiales tales como: actas de cortes de tesis, opiniones de tutores y oponente.

5. Desarrollo de las predefensas - defensas.

Para entrar en el subproceso de predefensas – defensas, las tesis deben de haber sido aprobadas por el comité de tesis. Estas últimas evaluaciones definirán una nota final al TD. Se recopilan igual que en los cortes, un acta de predefensa y acta de defensa que definen una nota. Además se requieren las opiniones del tutor y del oponente acerca del trabajo realizado de forma general.

Una vez identificados los procesos del negocio, es necesario encontrar los actores que están involucrados en su realización.

2.2.2 Actores y trabajadores del negocio

“Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos que interactúan con el negocio y se benefician de sus resultados”. (Jacobson; Booch; Rumbaugh; 2000) Además debe de tenerse en cuenta que el nombre de un actor debe expresar su rol dentro del negocio.

Después del análisis detallado del negocio, se obtiene un listado de actores y trabajadores del negocio como se muestra a continuación:

Actores del negocio y su justificación:

Actores	Justificación
Tesista	Inicia el proceso de entrega de solicitudes de asesoría (tutorías, cotutorías, oponencia, consultorías y asesoría). Además intervienen en la solicitud de los perfiles a desarrollar.
Tiempo	Inicia el proceso de revisión de las solicitudes (de asesoría y de perfiles). Indica el momento de conformar los tribunales de tesis, planificar los cortes de tesis y reunir la información referente a cortes de tesis. También comienza el proceso de asignación de tesis a tesistas, de profesores a las tesis y de tesis a tribunales
Comité	Inicia el proceso de ejecución de los cortes de tesis, pre defensas y defensas de las tesis.

Los autores definen a un trabajador del negocio como la representación de personas, o sistemas (software) dentro del negocio que realizan las actividades que están comprendidas dentro de un caso de uso. Son los candidatos a futuros usuarios del sistema que se quiere construir.

Trabajadores del negocio y su justificación:

Trabajadores	Justificación
--------------	---------------

Jefe de Polo	Interviene en el proceso de entrega de solicitudes de tutorías, cotutorías, oponencia, consultorías y en la entrega de los perfiles de tesis.
Comité	Interviene en el proceso de revisión de las solicitudes y de los perfiles y en el informe de los resultados de estas revisiones.
Tribunal	Interviene en el proceso de realización de los cortes de tesis, predefensas y defensas de las tesis.

2.2.3 Diagrama de Casos de Uso del Negocio

Un diagrama de casos de uso del negocio representa gráficamente los casos de uso del negocio y su interacción con los actores que corresponden con los procesos del negocio. Además es el encargado de presentar un sistema (en este caso el negocio) desde la perspectiva de su uso, y esquematiza cómo proporciona valor a sus usuarios.

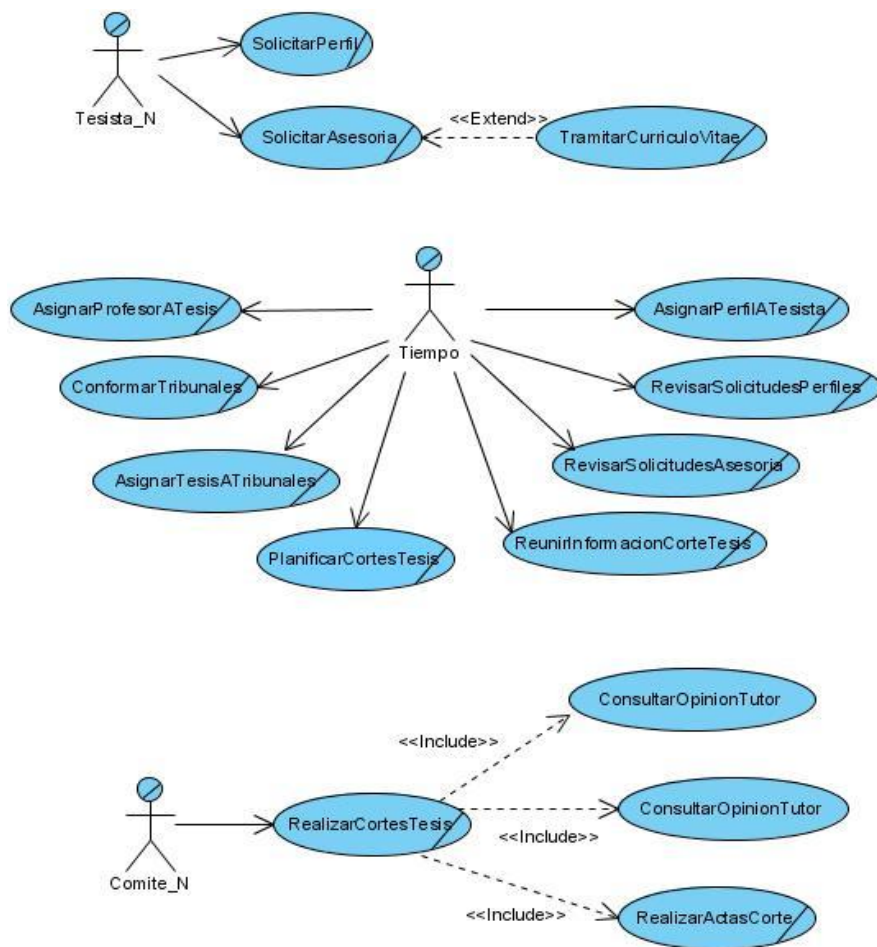


Fig. 1 Diagrama de Casos de Uso del Negocio.

2.2.4 Descripción de los Casos de Uso de Negocio

La descripción textual de los casos de uso del negocio, ayuda a comprender la lógica del mismo, mostrando además como ocurre la relación actor-negocio.

A continuación se reflejan 3 de estas descripciones, las demás se pueden consultar en el expediente de proyecto de este informe.

CUN: SolicitarPerfil.

Nombre del caso de uso:	SolicitarPerfil.	
Actores:	Tesisista_N	
Trabajadores:	Jefe de Polo o Líder de Proyecto (responsables).	
Propósito:	Realizar la solicitud del perfil de tesis.	
Resumen:	El caso de uso inicia cuando en un tiempo determinado el tesisista solicita a los responsables el documento llamado: perfil de tesis.	
Pre-condiciones	No aplica	
Casos de uso asociados:	No aplica	
Flujo de trabajo		
	Acción del actor	Acción del negocio
	1) El Tesisista solicita el perfil de tesis.	2) El responsable del perfil en presencia del tesisista entrega al Comité la solicitud del perfil de tesis.
		3) El caso de uso termina.
Mejoras:	No aplica.	

CUN: RevisarSolicitudesPerfiles.

Nombre del caso de uso:	RevisarSolicitudesPerfiles.
Actores:	Tiempo.
Trabajadores:	Comité.
Propósito:	Revisar los perfiles de tesis presentados por el

	responsable del perfil de tesis.
Resumen:	El caso de uso inicia cuando el actor Tiempo le indica al comité de tesis que es momento de efectuar la revisión de los perfiles de tesis.
Casos de uso asociados:	SolicitarPerfil.
Pre-condiciones:	Que se hayan entregado las solicitudes de perfiles de tesis.
Flujo de trabajo	
Acción del actor	Acción del negocio
1) El actor Tiempo indica que es el momento de realizar la revisión de los perfiles de tesis.	2) El comité de tesis realiza la revisión de los perfiles de tesis.
	3) El comité dictamina la aprobación o desaprobarción de los perfiles.
	4) El caso de uso termina.
Flujo Alterno	
Acción del actor	Acción del negocio
3) En caso de que el perfil halla sido desaprobado por primera vez o segunda vez.	3.1) Primera vez: El comité brinda los elementos que debe mejorar para su aprobación. 3.1) Segunda vez: El comité suspende en el período académico de un año el tema de tesis presentado por no cumplir con los requisitos mínimos establecidos.
Mejoras:	No aplica.

CUN: AsignarPerfilATesista.

Nombre del caso de uso:	AsignarPerfilATesista.
Actores:	Tiempo.
Trabajadores:	Comité.

Propósito:	Asignar perfil de tesis al tesista.	
Resumen:	El caso de uso inicia cuando el actor Tiempo le indica al comité de tesis que es el momento de asignar un perfil a un tesista.	
Casos de uso asociados:	RevisarSolicitudesPerfiles.	
Pre-condiciones:	Que se haya ejecutado la revisión de las solicitudes de perfiles de tesis.	
Flujo de trabajo		
	Acción del actor	Acción del negocio
	1) El actor Tiempo le indica que es el momento de asignar un perfil de tesis a un tesista.	2) El comité de tesis busca si existe una solicitud de perfil aprobada para ese tesista.
		3) El comité encuentra una solicitud de perfil aprobada y le asigna el perfil al tesista.
		4) El caso de uso finaliza.
Flujo de trabajo alternativo		
	Acción del actor	Acción del negocio
	3) El comité no encuentra una solicitud de perfil aprobada para ese tesista.	3.1) El Comité consulta con el jefe de polo o líder de proyecto del tesista para llegar a un consenso de qué perfil asignarle.
		3.2) El comité de tesis, en conjunto con el jefe de polo o líder de proyecto del tesista le asigna un perfil de tesis al tesista.
Mejoras:	No aplica.	

2.3 Requerimientos Funcionales

La especificación de requerimientos de software (ERS) define de forma precisa el producto de software que se va a construir. Las decisiones hechas escribiendo la ERS están basadas en información de los documentos de la propuesta del proyecto y requerimientos del usuario. Definitivamente *“el conjunto de requerimientos deben ser satisfechos en el diseño del sistema.”* (Requisitos de Software. 2003)

Para lograr un sistema automatizado con eficiencia y calidad se debe tener en cuenta los requisitos funcionales y no funcionales, la captura de requisitos constituye el paso fundamental para desarrollar cualquier sistema.

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por tanto: *“Los requerimientos funcionales son aquellos requisitos del sistema (condiciones o capacidades) que desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.”* (Jacobson, g.b. 2003)

A continuación aparecen los requisitos funcionales:

R1: Autenticar usuario.

Este requisito se produce cuando el usuario realiza la petición de conexión a la interfaz principal.

1.1 Para acceder a cualquier interfaz del sistema, el usuario debe haberse autenticado con su usuario y contraseña del dominio UCI y además estar registrado en la base de datos del sistema.

1.2 Para autenticar el sistema verifica que el usuario esté en el dominio UCI, luego verifica los permisos en la base de datos del sistema.

1.3 Se debe registrar los datos del usuario que se haya autenticado pero no esté registrado en el sistema.

R2: Gestionar datos del tesista.

Este requisito se produce cuando un usuario autenticado desea obtener información de un tesista determinado o cuando un tesista desea modificar sus datos personales.

2.1: Obtener los datos del tesista.

2.2: Modificar los datos del tesista.

R3: Gestionar datos de profesores.

Este requisito se produce cuando un usuario autenticado desea obtener información de un profesor determinado o cuando un profesor desea modificar sus datos personales.

3.1: Obtener los datos del profesor.

3.2: Modificar los datos del profesor.

R4: Gestionar datos del comité de tesis.

Este requisito se produce cuando un usuario autenticado con permisos de administración desea registrar, obtener o modificar los datos de comité de tesis.

4.1: Registrar los datos del comité de tesis.

4.2: Obtener los datos del comité de tesis.

4.3: Modificar los datos del comité de tesis.

R5: Gestionar miembros del comité de tesis.

Este requisito se produce cuando un usuario autenticado con permisos de administración desea adicionar, obtener o modificar los miembros del comité de tesis.

5.1: Adicionar miembros al comité de tesis.

5.2: Obtener miembros del comité de tesis.

5.3: Modificar miembros del comité de tesis.

R6: Gestionar datos del perfil de tesis.

Este requisito se produce cuando un usuario autenticado desea registrar, obtener o modificar los datos del perfil de tesis.

6.1: Registrar los datos del perfil de tesis.

6.2: Obtener los datos del perfil de tesis.

6.3: Modificar los datos del perfil de tesis.

R7: Revisar perfil de tesis

Este requisito se produce cuando un usuario autenticado con permisos de administración desea revisar el perfil de tesis.

7.1 El sistema debe facilitar realizar comentarios de mejoras a los datos del perfil de tesis.

R8: Mostrar resultado de revisión del perfil de tesis.

Este requisito se produce cuando un usuario autenticado desea consultar el resultado de la revisión del perfil de tesis.

8.1 El sistema debe permitir consultar el resultado de la revisión del perfil de tesis.

R9: Gestionar solicitudes de perfiles de tesis.

Este requisito se produce cuando un usuario autenticado desea registrar, obtener o modificar los datos de solicitudes de perfiles de tesis.

9.1: Registrar datos de solicitud de perfiles de tesis.

9.2: Obtener datos de solicitud de perfiles de tesis.

9.3: Modificar datos de solicitud de perfiles de tesis.

R10: Gestionar solicitudes de las figuras de tesis.

Este requisito se produce cuando un usuario autenticado desea registrar, obtener o modificar los datos de solicitudes de figuras de tesis.

10.1: Registrar datos de solicitud de las figuras de tesis.

10.2: Obtener datos de solicitud de las figuras de tesis.

10.3: Modificar datos de solicitud de las figuras de tesis.

R11: Revisión a solicitudes.

Este requisito se produce cuando un usuario autenticado con permisos de administración desea revisar las solicitudes de perfiles de tesis o de asesorías u otro usuario autenticado sin permisos particulares desea consultar el resultado de la revisión a solicitudes.

11.1: El sistema debe facilitar emitir el fallo de aprobada o denegada a la solicitud.

11.2: El sistema debe permitir consultar el resultado de la revisión de cualquier solicitud.

R12: Gestionar currículum vitae de una figura de tesis externa a la UCI.

Este requisito se produce cuando se procede con una solicitud a una figura de tesis que es externa a la UCI.

12.1: Registrar currículum vitae de una figura de tesis externa a la UCI.

12.2: Obtener currículum vitae de una figura de tesis externa a la UCI.

12.3: Modificar currículum vitae de una figura de tesis externa a la UCI.

R13: Gestionar perfil de tesis a tesista.

Este requisito se produce cuando un usuario autenticado con determinados permisos procede a vincular un perfil de tesis a un tesista.

14.3: Facilitar la asignación del perfil de tesis al tesista.

14.4: Obtener el perfil asignado a un tesista.

14.6: Facilitar la modificación de la asignación de un perfil a un tesista.

R14: Gestionar profesor a tesis.

Este requisito se produce cuando un usuario autenticado con determinados permisos procede a vincular profesores a tesis.

14.3: Facilitar la asignación de profesor a la tesis.

14.4: Obtener el profesor asignado a una tesis determinada.

14.5: Obtener todas las tesis asignadas a un profesor determinado.

14.6: Facilitar la modificación de la asignación de un profesor a una tesis.

R15: Gestionar tribunal de tesis.

Este requisito se produce cuando un usuario autenticado con determinados permisos procede a conformar tribunales de tesis.

15.1: Conformar tribunales de tesis.

15.2: Facilitar la adición de un miembro al tribunal (tribunal ya existente o crear un nuevo tribunal y adicionarle un miembro).

15.3: Facilitar la modificación de un miembro al tribunal

R16: Gestionar perfil de tesis a tribunal de tesis.

Este requisito se produce cuando un usuario autenticado con determinados permisos procede a vincular tribunales de tesis con perfiles.

15.4: Asignar las tesis a los tribunales.

15.5: Facilitar la asignación de tesis a los tribunales.

15.6: Obtener el tribunal asignado a una tesis determinado.

15.7: Obtener todas las tesis asignadas a un tribunal determinado.

15.8: Facilitar la modificación de la asignación de la tesis al tribunal.

R17: Gestionar opinión del tutor.

Este requisito se produce cuando un usuario autenticado relacionado como tutor a alguna tesis, proceda a registrar, obtener o modificar su opinión.

17.1: Registrar opinión del tutor.

17.2: Obtener opinión del tutor.

17.3: Modificar opinión del tutor.

R18: Gestionar opinión del oponente.

Este requisito se produce cuando un usuario autenticado relacionado como oponente a alguna tesis, proceda a registrar, obtener o modificar su opinión.

18.1: Registrar opinión del oponente.

18.2: Obtener opinión del oponente.

18.3: Modificar opinión del oponente.

R19: Gestionar preguntas del oponente.

Este requisito se produce cuando un usuario autenticado relacionado como oponente a alguna tesis, proceda a registrar, obtener o modificar sus preguntas.

El sistema debe permitir:

19.1: Registrar preguntas del oponente.

19.2: Obtener preguntas del oponente.

19.3: Modificar preguntas del oponente.

R20: Gestionar datos de acta de corte de tesis.

Este requisito se produce cuando un usuario autenticado relacionado como miembro de tribunal de tesis a alguna tesis, proceda a registrar, obtener o modificar el acta del corte de tesis.

20.1: Registrar datos de acta de corte de tesis.

20.2: Obtener datos de acta de corte de tesis.

20.3: Modificar datos de acta de corte de tesis.

R21: Planificación de evaluaciones.

Este requisito se produce cuando un usuario autenticado con permisos procede a planificar una evaluación (corte de tesis, pre defensa o defensa) de perfiles de tesis.

21.1 Realizar la planificación evaluaciones.

21.2 Facilitar la planificación manual de las evaluaciones.

21.3 Proponer una planificación automática de las evaluaciones.

R22: Generar y mostrar reporte del proceso.

Este requisito se produce cuando un usuario autenticado desea obtener un reporte determinado.

22.1 Obtener el listado de todos los perfiles involucrado en el proceso, con los tesisistas y profesores asociados.

22.2 Obtener el listado de tesisistas y profesores por polo, con las tesis asociadas.

22.3 Obtener el listado de tribunales con los miembros, tesis y tesisistas asociado.

22.4 Obtener cantidad de tesis en proceso de desarrollo, cantidad de tesisistas, cantidad de profesores (asesorando tesis e integrados a tribunales).

22.5 Obtener el estado general del proceso en cuanto a cantidad de cortes realizados, tesis evaluadas en cada corte, promedio de notas por tesis y por cortes.

22.6 Obtener el estado del proceso: en cuanto a tesis listas para pre defender, defender, el listado de tesis ya pre defendidas, defendidas, notas de predefensas y defensas.

R23: Eliminar usuario.

Este requisito se produce cuando un usuario autenticado con permisos de administración procede a eliminar un usuario del sistema.

23.1: Eliminar usuario del sistema.

R24: Eliminar perfil.

Este requisito se produce cuando un usuario autenticado con permisos de administración procede a eliminar un perfil del sistema.

24.1: Eliminar perfil del sistema.

2.4 Requerimientos no funcionales

Los requerimientos no funcionales responden a cualidades que el producto debe tener y las características para que este sea atractivo, confiable, usable y seguro. Los que se evidencian en este caso son los siguientes:

20. Apariencia o interfaz externa: Este tipo de requerimiento determina que la interfaz debe ser sencilla, intuitiva, amigable y mantener el formato en vistas similares. Debe poseer una apariencia profesional, legible y simple de usar.

21. Requerimientos de software: Se utiliza Internet Information Server (IIS) v6.0 como servidor, .NET Framework v2.0.50727 como plataforma de desarrollo y PostgreSQL 8.2 como Gestor de Base de Datos.

22. Requerimientos de hardware: Para que los usuarios del sistema puedan utilizarlo es necesario que cada uno tenga una computadora conectada a la red. El sistema requiere de una computadora que haga función de servidor, la cual debe cumplir con las características mínimas siguientes: como sistema operativo puede tener instalado Windows, la Memoria RAM debe ser de 256 MB y la capacidad del disco duro de 20 GB.

23. Requerimientos de seguridad.

Confiabilidad: La información estará protegida contra accesos no autorizados utilizando para esto mecanismos de autenticación y encriptación que garanticen su confiabilidad.

Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, por lo que será de vital importancia el uso de políticas de salvadas de la base de datos.

Disponibilidad: Los usuarios autorizados tendrán acceso a la información contenida en el sistema, siempre que esté publicado.

24. Requerimientos de rendimiento: La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente / Servidor y la velocidad de las consultas en la Base de Datos. El sistema debe implementar varias transacciones por segundo para dar una respuesta rápida al usuario y evitar demoras innecesarias.

25. Restricciones de diseño: Se utiliza el Visual Paradigm como herramienta para la modelación del sistema y el estilo arquitectónico tres capas para el diseño de la aplicación.

26. Requerimiento de ayuda: Se deberá brindar una ayuda que facilite a los usuarios el trabajo con la aplicación.

2.5 Descripción del sistema propuesto

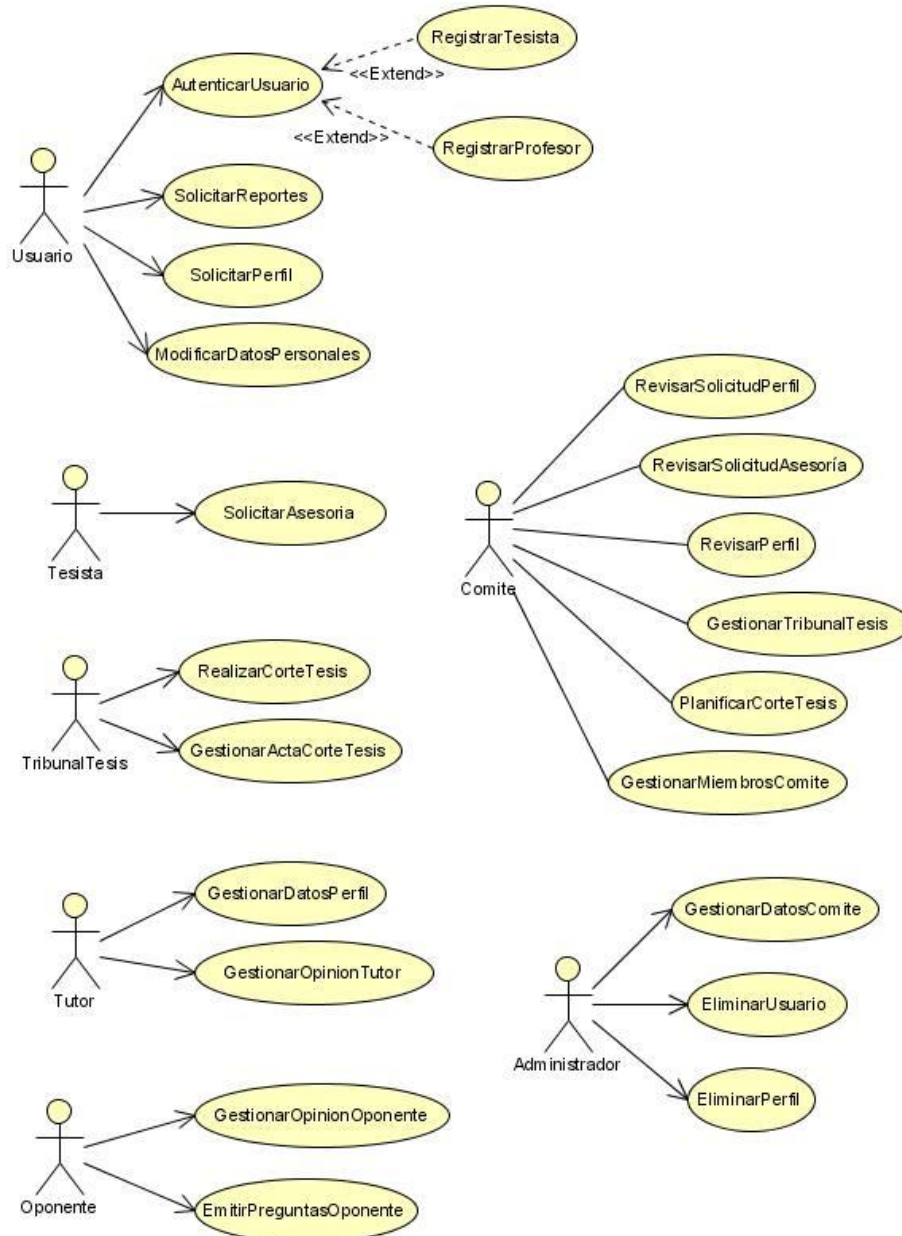
Los casos de uso del sistema representan las funcionalidades que definieron los requerimientos funcionales para el sistema que se propone. A continuación se detallan las características del sistema: actores, diagrama de casos de uso y descripciones de casos de uso.

2.5.1 Descripción de los actores del sistema.

Actores	Justificación
---------	---------------

Usuario	<p>Inicia la autenticación en el sistema para acceder a él. Si no está registrado en la base de datos del sistema, según la especialización del usuario (profesor o tesista) accede a registrar sus datos.</p> <p>El Usuario, con la pre-condición de estar autenticado, hace uso de las operaciones de obtener información referente a los fallos del comité respecto a las solicitudes de perfiles y asesorías, a la información de los miembros del comité, tribunal de tesis, tesistas y profesores; además, sobre la planificación de cortes de tesis, las actas de cortes de tesis correspondientes a los perfiles asociados al usuario, así como sus opiniones del tutor y oponente y las preguntas del oponente. También cada usuario autenticado podrá modificar sus datos personales y realizar la solicitud de un perfil.</p>
Tesista	<p>Hace uso de las operaciones de solicitud de asesoría al perfil.</p>
Tutor	<p>Hace uso de las operaciones de gestionar los datos del perfil y emitir la opinión de tutoría.</p>
Oponente	<p>Hace uso de las operaciones de emitir la opinión del oponente y sus preguntas.</p>
TribunalTesis	<p>Hace uso de las operaciones de realizar cortes de tesis y gestionar los datos del acta del corte. Además de consultar las opiniones de tutor (es), consultor (es) y asesor y las actas.</p>
Comité	<p>Hace uso de las operaciones de consultar las solicitudes de perfiles y asesoría (tutor, cotutor, consultor, oponente) para emitir su aprobación. Además, hace uso de las operaciones de gestionar los miembros del comité y las de planificar las evaluaciones (cortes de tesis, pre-defensas y defensas).</p>

2.5.2 Diagrama de Casos de Usos del Sistema.



2.5.3 Descripción de los casos de uso del sistema.

En la descripción de los casos de uso del sistema se puede apreciar la interacción del actor del sistema con el sistema, así como conocer las especificidades propias de cada caso de uso o funcionalidad. Como muestra de estas descripciones se especifican los siguientes:

CUS: RegistrarTesista.

Caso de Uso:	RegistrarTesista.
---------------------	-------------------

Actores:	No aplica.
Resumen:	El caso de uso inicia cuando el CUS AutenticarUsuario requiere del registro de los datos del tesista en la BD.
Precondiciones:	Que en el CUS AutenticarUsuario, el sistema no encuentre al usuario registrado en la BD y este usuario haya sido reconocido en el directorio activo como tesista (estudiante de 5to año).
Referencias	R1; R25; R28.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El Usuario procede al registro de sus datos como tesista.	1.1. El sistema le muestra en una tabla los siguientes datos del tesista: <i>nombre(s) y apellidos, sexo, teléfonos, residencia, área, polo productivo, grupo.</i> Además de un botón tipo link llamado "Registrar".
2. El tesista registra los datos deseados y acepta.	2.1 El sistema guarda la información en la base de datos.
	2.2 El caso de uso finaliza.

Prototipo de Interfaz

Registrar Datos de Tesista

Usuario UCI:
 Nombre:
 Apellidos:
 Sexo:
 Telefonos:
 Residencia:
 Grupo:
 Area del proceso: [Mostrar sus Polos](#)
 Polo Productivo:

Flujos Alternos

Acción del Actor	Respuesta del Sistema
2.1 El sistema lanza una excepción durante el registro de los datos en la base de datos.	2.1.1. El sistema no guarda la información y se mantiene en el mismo formulario hasta que el usuario procede nuevamente al paso 2.
Pos condiciones	El usuario se registra en el BD como tesista y el sistema lo reconoce.

CUS: RegistrarProfesor

Caso de Uso:	RegistrarProfesor.
Actores:	No aplica.
Resumen:	El caso de uso inicia cuando el CUS AutenticarUsuario requiere del registro de los datos del profesor en la BD.
Precondiciones:	Que en el CUS AutenticarUsuario, el sistema no encuentre al usuario registrado en la BD y este usuario haya sido reconocido en el directorio activo como profesor.

Referencias:	R1; R25; R28.
---------------------	---------------

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El Usuario procede al registro de sus datos como profesor.	1.1. El sistema le muestra en una tabla los datos del profesor: <i>nombre(s) y apellido, sexo, teléfonos, residencia, área, polo productivo, categoría docente, grado científico, título de Especialidad de graduado, instituto de Graduado, año de graduación, cargo laboral.</i> Además un botón tipo link llamado "Registrar".
2. El Profesor registra los datos deseados y acepta.	2.1 El sistema guarda la información en la base de datos.
	2.2 El caso de uso finaliza.

Prototipo de Interfaz

Flujos Alternos

Acción del Actor	Respuesta del Sistema
-------------------------	------------------------------

2.1 El sistema lanza una excepción durante el registro de la información en la BD.	3.1.1. El sistema no guarda la información y se mantiene en el formulario hasta que el usuario proceda nuevamente al paso 2.
Pos condiciones	El usuario se registra en el BD como profesor y el sistema lo reconoce.

Las restantes descripciones se pueden consultar en el expediente de proyecto de este informe.

2.6 Vista de Gestión del Modelo.

La vista de gestión del modelo (VGM), modela la organización del modelo en sí mismo. Un modelo abarca un conjunto de paquetes que contienen sus elementos, tales como clases, máquinas de estados, y casos de uso.

La siguiente figura muestra la descomposición de la totalidad del sistema de gestión de tesis en paquetes y sus relaciones de dependencia.

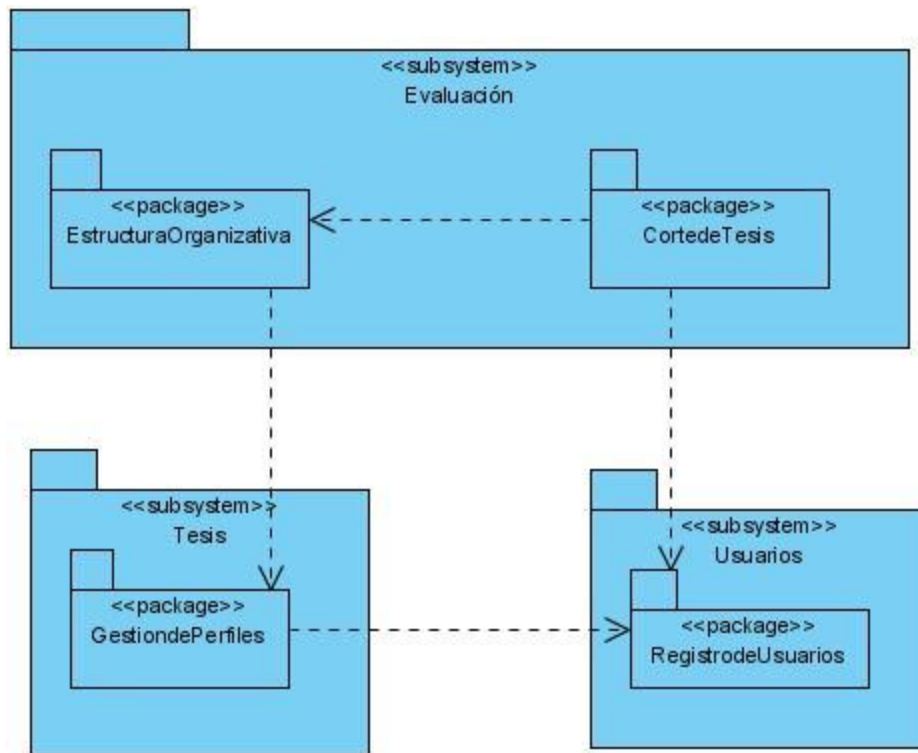


Diagrama de Paquetes

2.7 Conclusiones

Este capítulo ha brindado una clara definición de los procesos del negocio, el modelo del negocio y los requisitos que deberá cumplir el sistema. Además se hizo una representación de las funcionalidades a construir mediante el diagrama de casos de usos del sistema. Con lo antes mencionado, se ganó claridad en cuanto a la concepción del sistema a construir y se sentaron las bases para las restantes fases del proceso de análisis, diseño e implementación del sistema. Los resultados serán utilizados como entrada en la siguiente etapa de trabajo.

Capítulo 3 “Construcción de la solución propuesta”

3.1 Introducción

La etapa o fase de trabajo de la construcción del producto es de vital importancia. En esta, se debaten los principios de diseño que han de tenerse en cuenta y se generan artefactos tales como el diagrama de despliegue del sistema y el diagrama de componente.

Además, se aborda todo lo referente al proceso de pruebas del sistema y a la validación de la solución propuesta para el problema científico de esta investigación. En este capítulo se debaten estos aspectos con mayor profundidad y especificidad.

3.2 Diagramas de clases.

Los diagramas de clases del análisis y del diseño concretan los casos de uso en clases, métodos, atributos y relaciones entre clases. Para la realización de ambos, se toman como entradas los casos de uso del sistema y los diagramas de interacción, respectivamente. A continuación se muestran tres de los diagramas, en el expediente se encuentran los restantes.

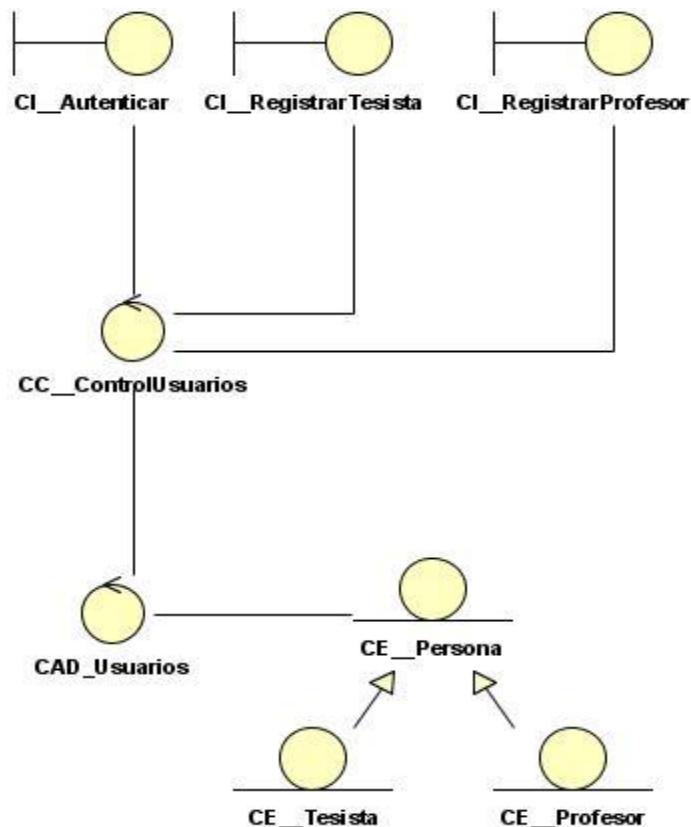


Fig.2 DCA_AutenticarUsuario

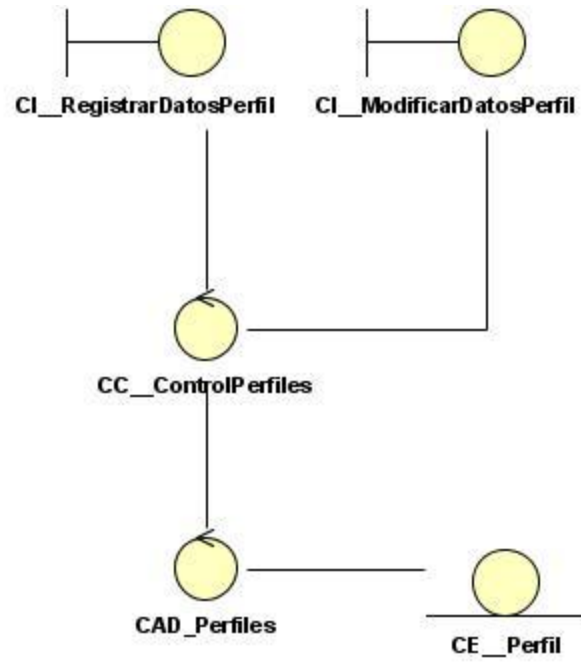


Fig.3 DCA_GestionarDatosPerfil

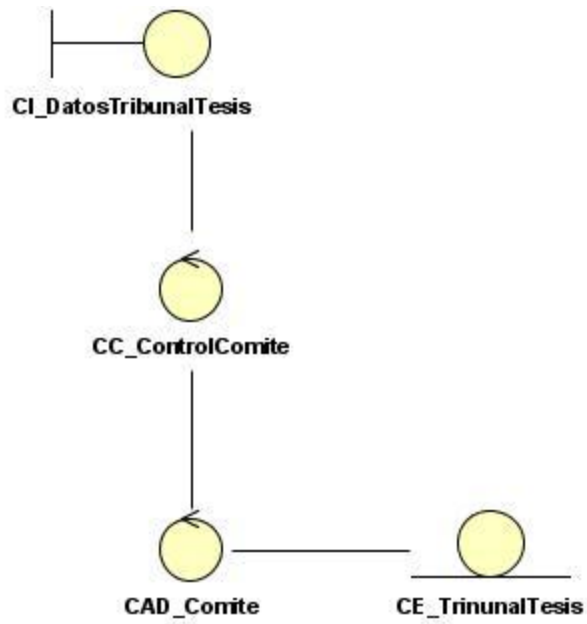


Fig.4 DCA_GestionarTribunalTesis

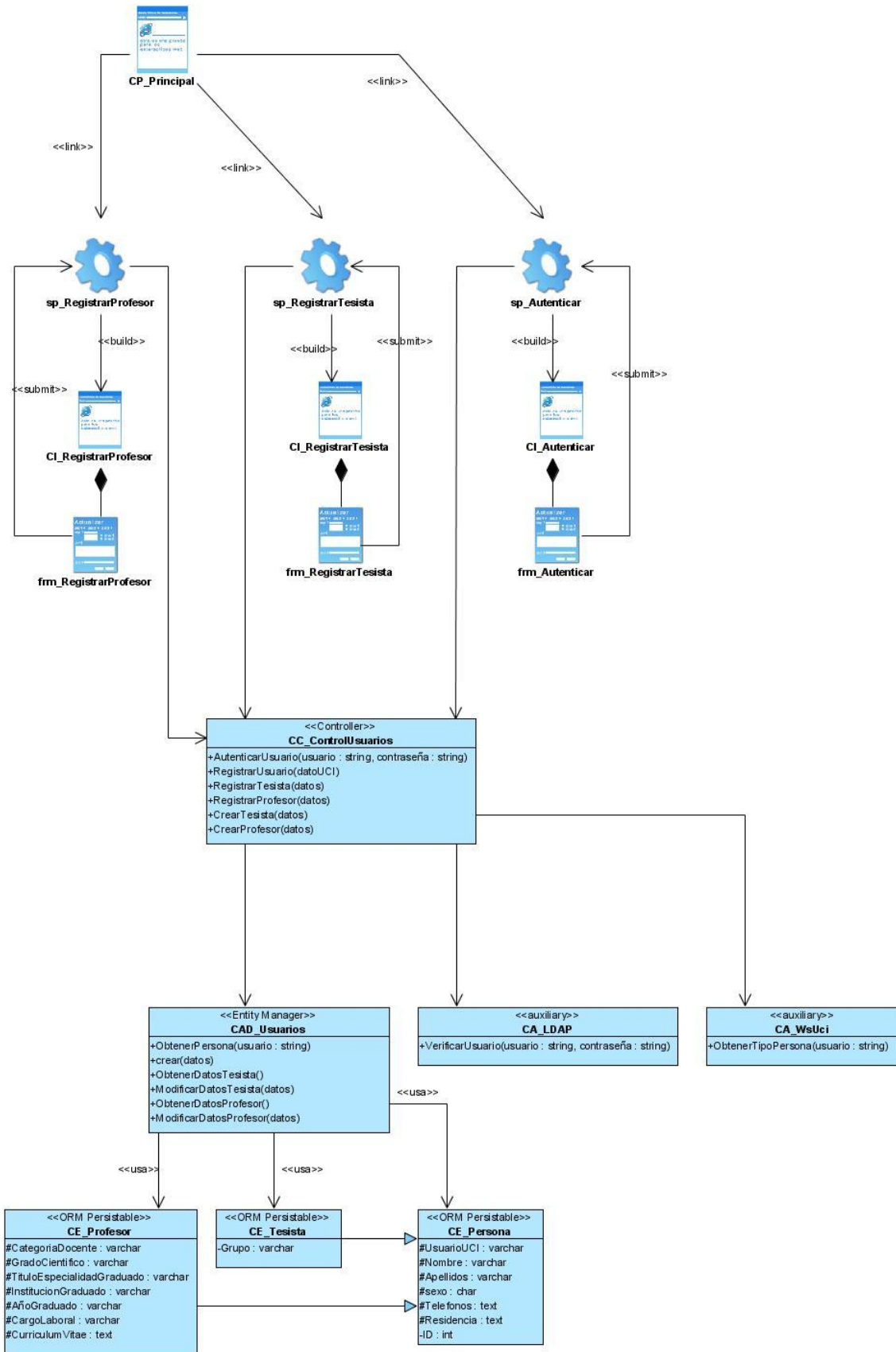


Fig.5 DCD_AutenticarUsuario

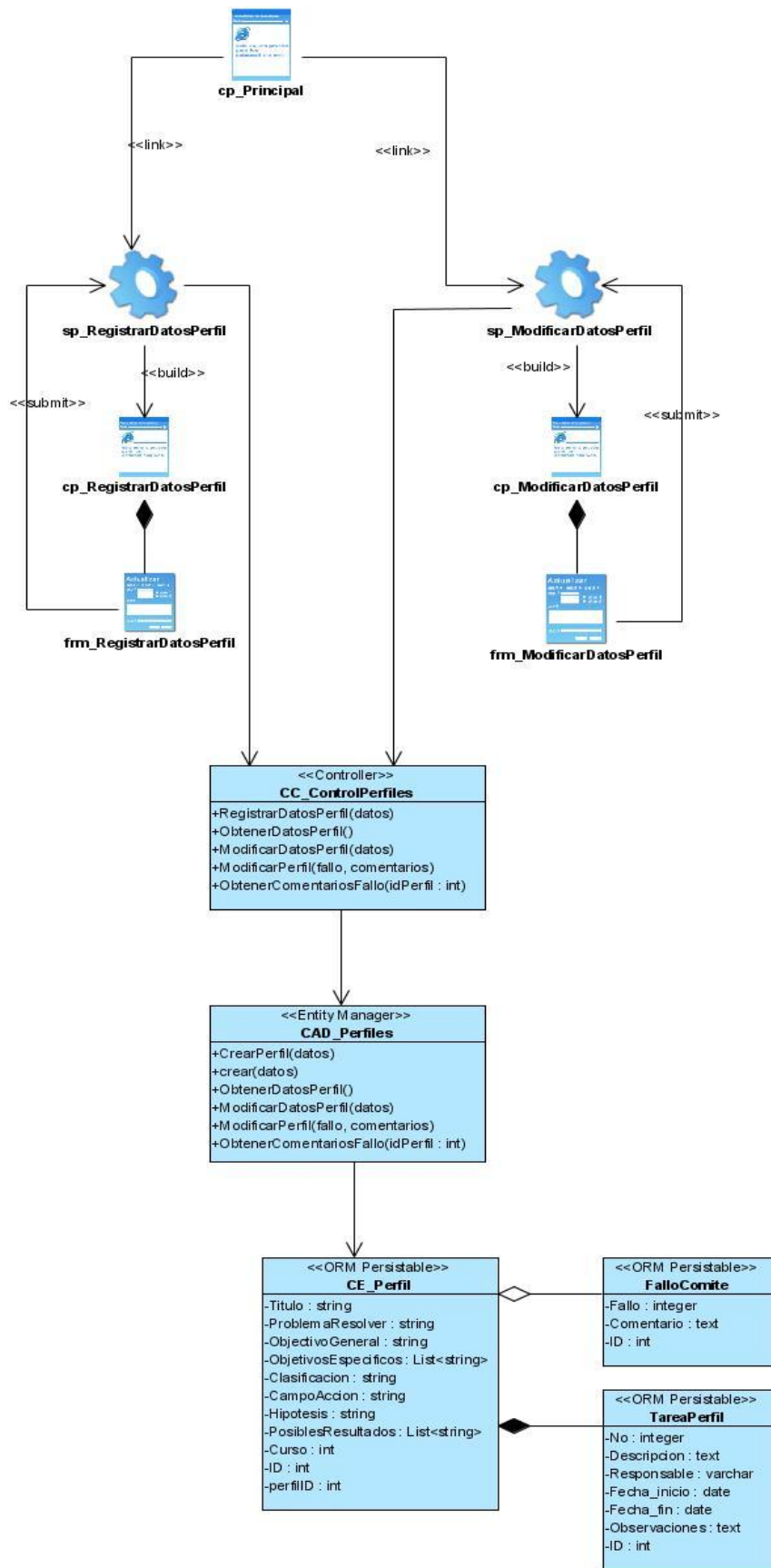


Fig.6 GestionarDatosPerfil

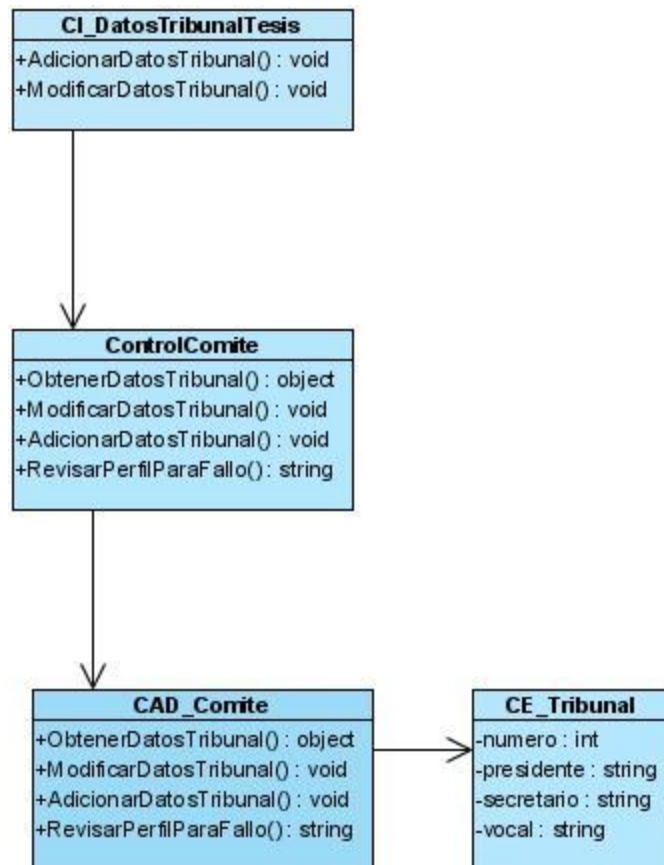


Fig.7 GestionarTribunalTesis

3.3 Principios del diseño.

Los 7 principios del diseño universal, se centran en el diseño utilizable por todos. Intervienen también otros aspectos tales como el costo, la cultura y ambiente en el que será usado. Estos principios son aplicables en la arquitectura, la ingeniería y por supuesto, en las páginas y aplicaciones Web. Se mencionan brevemente a continuación:

1er Principio: Uso equiparable. El diseño es útil y vendible a personas con diversas capacidades.

2do Principio: Uso flexible. El diseño se acomoda a un amplio rango de preferencias y habilidades individuales.

3er Principio: Simple e intuitivo. El uso del diseño es fácil de entender, atendiendo a la experiencia, conocimientos, habilidades lingüísticas o grado de concentración actual del usuario.

4to Principio: Información perceptible. El diseño comunica de manera eficaz la información necesaria para el usuario, atendiendo a las condiciones ambientales o a las capacidades sensoriales del usuario.

5to Principio: Con tolerancia al error. El diseño minimiza los riesgos y las consecuencias adversas de acciones involuntarias o accidentales.

6to Principio: Que exija poco esfuerzo físico. El diseño puede ser usado eficaz y confortablemente y con un mínimo de fatiga.

7mo Principio: Tamaño y espacio para el acceso y uso. Que proporcione un tamaño y espacio apropiados para el acceso, alcance, manipulación y uso. (Rose Connell, Betty y otros. 1997)

El diseño tiene que basarse en la mezcla de las ideas, deseos y necesidades del usuario y en los materiales de que dispone el programador. En este caso se trata de usuarios que son expertos: profesores y estudiantes de 5to año; todos cuentan con una preparación en las cuestiones de la informática, de diseño, navegación y manejabilidad de la información.

No obstante, el sistema utiliza ciertos principios que garantizan la usabilidad en el diseño para de la aplicación:

1. La interfaz no sobrecarga al usuario con complejidades innecesarias ni lo distrae de su labor.
2. Se facilita un ambiente flexible para que pueda familiarizarse rápidamente a usar la aplicación.
3. La interfaz es consistente.
4. Se debe considerar la productividad del usuario antes que la productividad de la máquina. Si el usuario debe esperar la respuesta del sistema por un período prolongado, estas pérdidas de tiempo se pueden convertir en pérdidas económicas para la organización.
5. La ayuda es sencilla y provee respuestas a los problemas que puedan presentarse.
6. Los menús y etiquetas de botones toman las palabras claves del proceso.

3.3.1 Aplicaciones de los principios de diseño a la interfaz de la aplicación.

En el desarrollo de la interfaz se tuvo en cuenta los principios anteriores de la siguiente manera:

- Se usó colores que resultaran agradables, logrando una mayor y adecuada comunicación entre la aplicación y el usuario.

- Se mantuvieron las alineaciones de textos así como la alineación de los botones navegables.
- Se utilizaron elementos característicos para la descripción de acciones.
- Se evidencia en la aplicación el cumplimiento de las exigencias fisiológicas con respecto al ser humano a la hora de recibir información digital, para que no se viera afectada su salud por un uso inadecuado, por ejemplo: tamaño de letra, espaciamiento entre caracteres, tipografía, contraste figura – fondo de la información mayoritaria, estandarización de márgenes y distribución de información a lo largo de todo el software.

3.3.2 Concepción general de la ayuda

Teniendo en cuenta que todos los usuarios que trabajarán con la aplicación son prácticamente profesionales pero que pueden tener poca o ninguna experiencia con la aplicación se deriva la necesidad de mostrarle una guía.

La ayuda del sistema permite conocer el funcionamiento de cada una de las opciones del sistema, y permite a los usuarios estar informados y orientados en todo momento. Para lograr este objetivo se confecciona un documento con todas las especificaciones necesarias. Este documento está incluido en el expediente de proyecto GesTes.

3.4 Diseño de la Base de Datos

La base de datos representa la información persistente presente en el proceso de tesis de la facultad 9. Consiste en los datos personales de los usuarios tesista y profesor, los datos específicos de los perfiles, actas de corte de tesis, opiniones de profesores y datos de tribunales. Se presentan los modelos lógico y físico de datos, conocidos como diagrama de clases persistentes y diagrama entidad – relación respectivamente. La figura 8 muestra el Diagrama de clases persistentes, mientras la figura 9 muestra Diagrama entidad – relación.

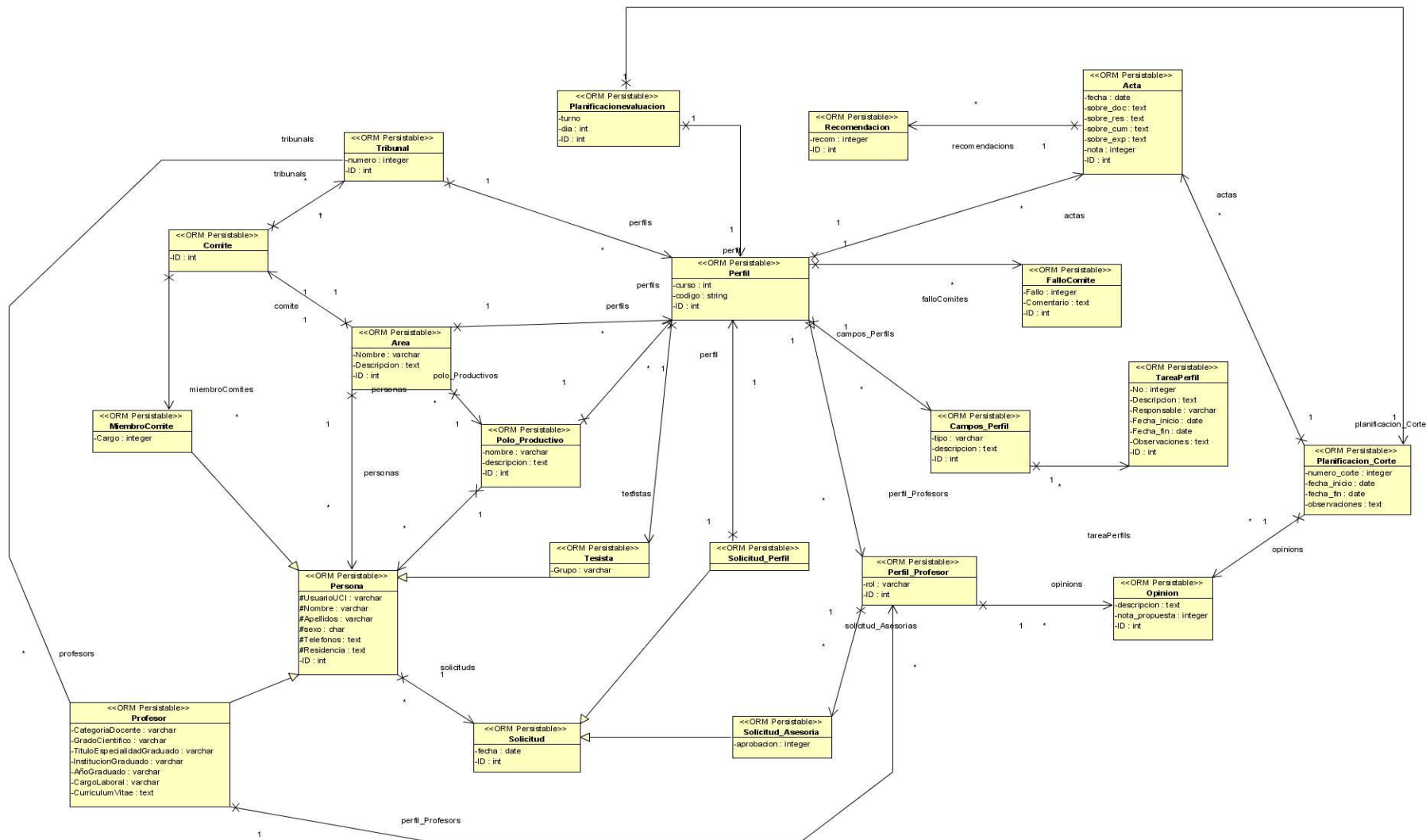


Fig8: Diagrama de clases persistentes.

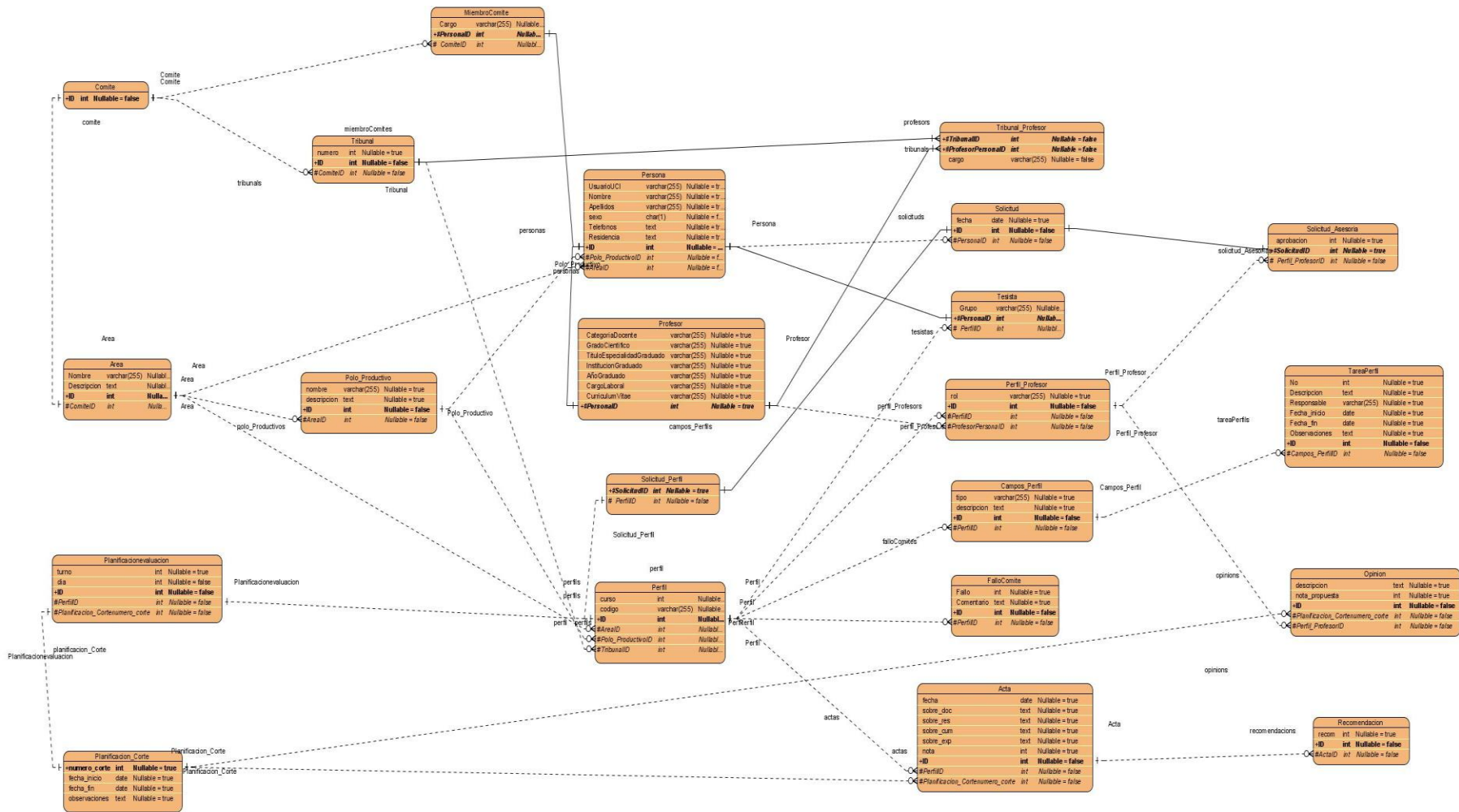


Fig. 9: Diagrama entidad – relación

3.5 Modelo de Implementación

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema y el posterior despliegue del mismo.

Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

3.6.1 Diagrama de componente

El diagrama de componente ilustra los componentes del software que serán usados para conformar el sistema. Estos pueden ser construidos para el modelo de clase y escritos para satisfacer los requisitos del nuevo sistema. Un componente puede ser siempre considerado como una unidad autónoma dentro de un sistema o subsistema. UML define cinco estereotipos estándar que se aplican a los componentes: ejecutable, librería, tabla, archivo y documento. Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación.

Es cierto que: *“Los componente son paquetes estereotipados en subsistemas. Estos organizan la vista de realización de un sistema y pueden contener componentes y otros subsistemas.”* (Fernández Vilas; 2001) Como se muestra en la figura 10, el diagrama de componentes queda estructurado de esta forma.

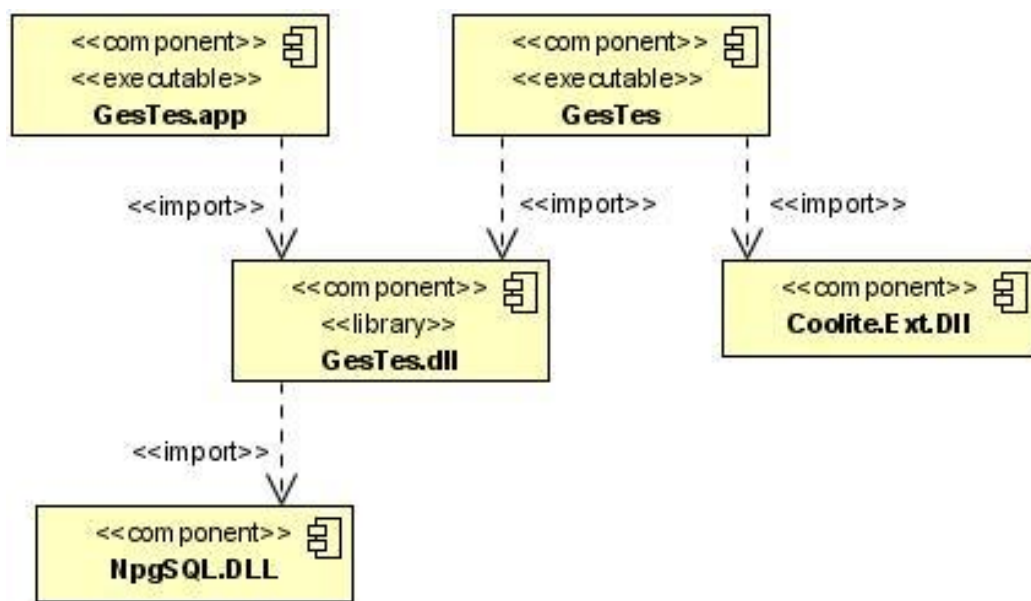


Fig.10 Diagrama de Componentes.

3.6.2 Diagrama de despliegue

El diagrama de despliegue describe como una aplicación se despliega a través de una infraestructura. La intención del modelo de despliegue no es para describir la infraestructura, es el camino en el cual los componentes específicos deben corresponder a una aplicación que despliega a través de él. El modelo de despliegue muestra la configuración (relaciones físicas) de los nodos que participan en la ejecución y de los componentes hardware y software que residen en ellos. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene algo de memoria y, a menudo, capacidad de almacenamiento. Son los elementos donde se ejecutan los componentes.

“El diagrama de distribución como también se conoce como diagrama de despliegue contiene: Nodos (Servidores o Procesadores y Dispositivos) y Relaciones de dependencia y asociación”. (Fernández Vilas; 2001) Los autores con esta afirmación, además agregan que el diagrama de despliegue muestra las relaciones entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software.

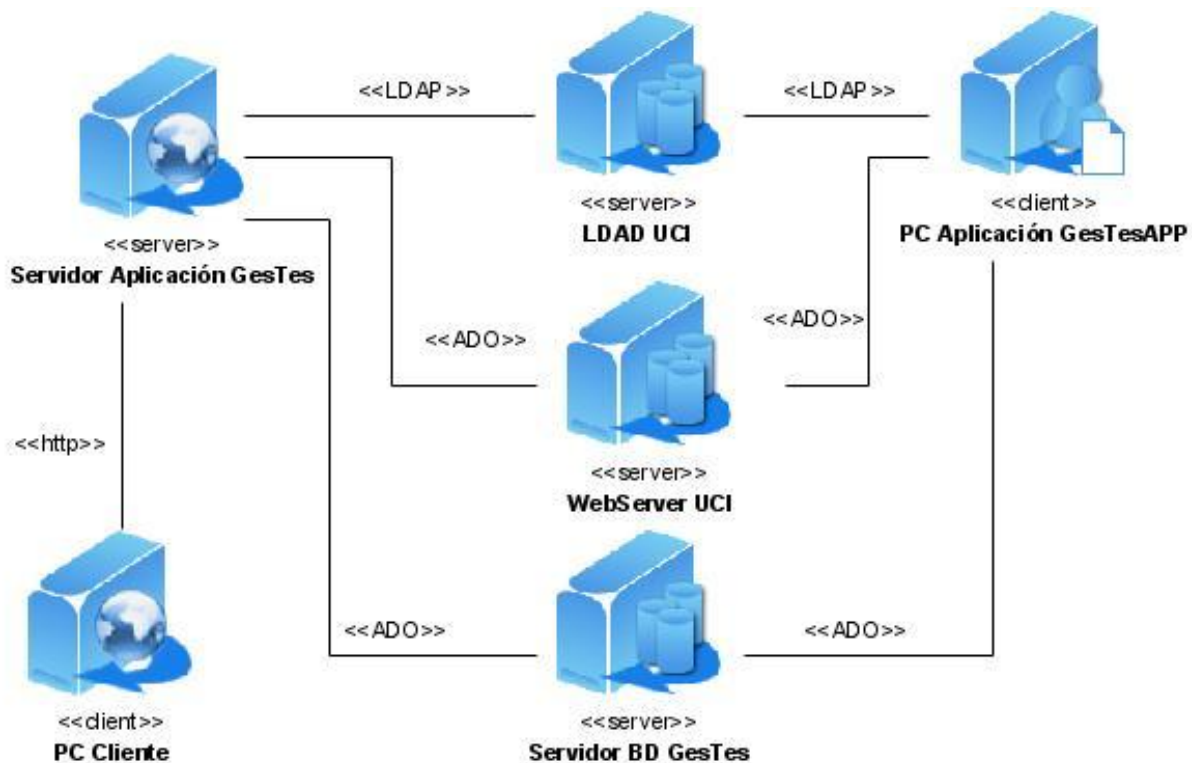


Fig.11 Diagrama de Despliegue.

3.7 Prueba del sistema propuesto y validación de la solución.

Los Casos de Prueba son el conjunto de entradas con datos de prueba, unas condiciones de ejecución, y unos resultados esperados con el propósito de identificar y comunicar las condiciones que se llevarán a cabo en la prueba.

El método que se propone para el diseñar los casos de pruebas es una variante del de particiones equivalentes, propuesto por RUP y al que se le han añadido algunas consideraciones, este método consta de 3 pasos fundamentales y se basa totalmente y usa como artefacto de entrada los casos de usos.

1. Para cada caso del uso, generar un sistema completo de escenarios.

Lo primero que se hace es leer la descripción textual del caso de uso e identificar todas las combinaciones posibles de caminos de ejecución del caso de uso, es decir todas las combinaciones entre el camino principal y los caminos alternativos y se le asigna un nombre. Cada combinación será un escenario de uso. Esta información se representa en una matriz, que se denomina Matriz Parcial de escenarios, donde se tiene en cuenta el nombre del escenario, el flujo donde comienza y si contiene algún flujo alternativo.

Nombre del escenario	Flujo donde empieza	Alternativo
Escenario 1 “Registro exitoso”	Flujo Básico	
Escenario 2 “Faltan datos obligatorios”	Flujo Básico	A1
Escenario 3 “Ya existe el usuario”	Flujo Básico	A2

Fig.12 Matriz Parcial de escenarios

2. Identificar los Casos de Prueba.

Luego de identificados los diferentes escenarios de prueba, se identifican las variables (la información necesaria para que el escenario se ejecute) que hacen que dicho escenario

se comporte de una forma o de otra, es decir como un flujo básico o alternativo y nuevamente esta información se representa en una matriz, llamada Matriz de Casos de Prueba, donde se representa además el comportamiento esperado del sistema.

Id del escenario	Escenario	Código	Nombre	Cantidad	Respuesta del Sistema
EC 1	Escenario 1: "Registro exitoso"	V	V	V	El sistema muestra un mensaje informándole al administrador que ya ha sido efectuado el registro del libro.
EC 2	Escenario 2: "Faltan datos obligatorios"	Alguno de los datos toma valor I.			El sistema emite un mensaje para que llene los campos obligatorios
EC 3	Escenario 3: "Ya existe el usuario"	V	V	V	El sistema muestra un mensaje informativo.

Fig. 13 Matriz de Casos de Prueba

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

3. Identificar Valores de Datos para las Pruebas.

Una vez que todos los casos de prueba se han identificado, todos ellos deben repasarse y validarse para asegurar exactitud e identificar los casos de prueba redundantes o que

faltan. Entonces, una vez que sean aprobados, el paso final es sustituir los valores reales de los datos para las I's y V's, porque la idea es probar con datos reales.

Los casos de prueba realizados pueden consultarse en el expediente de proyecto de este informe. Ya diseñados los casos de prueba faltaría probar contra la aplicación, y verificar si se cumplieron los requisitos establecidos en los casos de usos. El nivel de las pruebas a realizar en este caso podría ser de liberación o de aceptación por parte del cliente.

Las pruebas realizadas al sistema alcanzaron resultados satisfactorios en más del 95% de los casos. Se garantizó la robustez y a la vez flexibilidad del sistema, además de la seguridad, su fácil uso y la gran utilidad en el desarrollo del proceso en general.

3.9 Conclusiones

En este capítulo se definieron artefactos importantes de la construcción del sistema. Se construyó el diagrama de despliegue en donde se detalló la relación hardware y software del sistema final, así como el diagrama de componentes para observar la relación existente entre los componentes físicos del sistema.

Al finalizar este capítulo ya se tiene la solución construida, probada y funcionando correctamente.

Conclusiones generales

Con el diseño e implementación del Sistema Gestor de Trabajos de Diploma, se resuelven las limitantes presentadas en el problema y se pone a disposición de la facultad 9, una herramienta que facilita la automatización de la gestión de la información generada en el proceso de organización, evaluación, control y planificación de los trabajos de diploma.

Se realizó el modelado del negocio, generándose los artefactos y la documentación correspondientes.

Se diseñó e implementó el sistema compuesto por un módulo Web y un módulo escritorio, ambos de interfaz amigable y con alta usabilidad, que muestran información referente a todo el proceso y permiten la gestión de usuarios, la gestión de solicitudes de perfiles, la gestión de tribunales, comité de tesis; así como la planificación de evaluaciones y la generación de reportes.

Fueron satisfechos los requisitos de seguridad, mediante el control de los permisos de usuarios.

Concluyendo que fueron cumplidos los objetivos planteados y alcanzados los resultados esperados.

Recomendaciones

- Agregar al sistema funcionalidades para la gestión de ficheros .
- Implementar las funcionalidades de planificación automática de las evaluaciones .
- Implementar las funcionalidades de generación automática de tribunales .
- Implementar un sistema de control de cambios para los datos manejados en la aplicación.

Referencias Bibliográficas

1. **Resolución Decanal; 2008.** *Lineamientos para la concepción y el seguimiento de los trabajos de diploma de la facultad desde los polos productivos.* Facultad 9. Universidad de las Ciencias Informáticas. Ciudad de la Habana.
2. **Jiménez Sainz, P.; 2002.** *La automatización en la empresa actual.* Consultado [7 de enero del 2009]. Disponible en: www.eui.upm.es/~wdoe/automatizacion.ppt
3. **Diccionario de la lengua española; 2005.** *Automatización.* Consultado [7 de enero del 2009]. Disponible en: <http://www.wordreference.com/definicion/automatización>
4. **Restrepo González, Guillermo; 2002.** *El concepto y alcance de la gestión tecnológica.* Consultado [7 de enero del 2009]. Disponible en: http://ingenieria.udea.edu.co/producciones/guillermo_r/concepto.html
5. **El portal educativo de Estrado Argentino; 2006.** *Gestión Institucional: Conceptos introductorios.* Consultado [7 de enero del 2009]. Disponible en: <http://www.educ.ar/educar/site/educar/Gesti%F3n%20institucional%20%20conceptos%20introductorios.html?uri=urn:kbee:90191430-5192-11dc-a181-00163e000024&page-uri=urn:kbee:ff9221c0-13a9-11dc-b8c4-0013d43e5fae>
6. **García Molina J., Moreira, Ana; Rossi, Gustavo; 2004.** *UML: el lenguaje estándar para el modelado de software.* Ediciones SA.2004.
7. **Colado Rodríguez, César; 2003.** *Diseño y desarrollo de aplicaciones web multidispositivo.* Consultado [7 de enero del 2009]. Disponible en: [http://www.germinus.com/sala_prensa/articulos/Diseno_desarr_aplicaciones_web_multidispo%20\(Febrero%202003\).pdf](http://www.germinus.com/sala_prensa/articulos/Diseno_desarr_aplicaciones_web_multidispo%20(Febrero%202003).pdf)
8. **Ezequiel Rozic, Sergio; 2004.** *Bases de Datos y su aplicación con SQL.* Consultado [7 de enero del 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg03438.pdf>
9. **Angel Sicilia, Miguel; 2008.** *Funciones de los Sistemas Gestores de Bases de Datos.* Consultado [7 de enero del 2009]. Disponible en: <http://cnx.org/content/m17543/latest/>
10. **Álvarez, Sara ; 2007.** *Sistemas gestores de bases de datos.* Consultado [7 de enero del 2009]. Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>

11. **Menéndez Barzanallana, Rafael; 2009.** *Informática Bases de datos Servidores de bases de datos.* Consultado [7 de enero del 2009]. Disponible en: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sghbd.html>
12. **Carcassés Díaz, Lisbety; Abad Estrada, Suanny; 2007.** *Propuesta de diseño de una aplicación Web para la gestión de perfiles de los trabajos de diploma.* Tesis de Grado inédita. Universidad de las Ciencias Informáticas. Ciudad de la Habana.
13. **Heredia Rodríguez, Reynaldo; Salcerio Paz, Emilio; 2008.** *Aplicación Web para el control del proceso de tesis de grado en la facultad 8.* Tesis de Grado inédita. Universidad de las Ciencias Informáticas. Ciudad de la Habana.
14. **Núñez García, Dunel; 2008.** *Automatización de la gestión y reproducción de las tesis en la UCI.* Tesis de Grado inédita. Universidad de las Ciencias Informáticas. Ciudad de la Habana.
15. **Pérez Salomón, Omar; 2006.** *Impacto de las tecnologías de la información y las comunicaciones en los países del Tercer Mundo.* Consultado [7 de enero del 2009]. Disponible en: <http://www.cubasocialista.cu/texto/cs0231.htm>
16. **Rosario, Jimmy; 2005.** *La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual.* Consultado [7 de enero del 2009]. Disponible en: <http://www.cibersociedad.net/archivo/articulo.php?art=218>
17. **Partido Comunista de Cuba; 1997.** *Resolución Económica del V Congreso del Partido Comunista de Cuba.* Consultado [8 de enero del 2009]. Disponible en: http://www.sld.cu/galerias/doc/sitios/infodir/21_res_econom.docm
18. **Jacobson, I; G. Booch, et al.; 2000.** *El proceso unificado de desarrollo de software.* La Habana, Addison Wesley Longman.
19. **Pressman, R.S.; 2005.** *Ingeniería del Software, Un enfoque práctico.* 5a. ed. La Habana, Mc Graw Hill.
20. **Rational Rose; 2005.** *Definition* Consultado [25 de marzo del 2009]. Disponible en: http://searchcio-midmarket.techtarget.com/sDefinition/0,,sid183_gci516025,00.html
21. **Gonzalo Génova, José M.; Llorens, Juan; 2004.** *Evaluación de herramientas CASE para UML.* Consultado [10 de enero del 2009]. Disponible en: <http://www.procuno.com/users/taller/Presentaciones/PresentacionIEG-UC3.ppt>

22. **Hernan Zájara, Javier; 2009.** *Tecnología Mac. Novedades y actualizaciones.* Consultado [10 de enero del 2009]. Disponible en: <http://personales.ya.com/javierhz/mactec.htm>
23. **González Hernández, Daimara; 2008.** *Sistema Informatizado para la Gestión de la Información de los Pacientes en Rehabilitación.* Tesis de Grado inédita. Universidad de Ciencias Informáticas. Ciudad de la Habana.
24. **CASTELLS, P.; 2008.** *La Web semántica.* Consultado [7 de enero del 2009]. Disponible en: <http://arantxa.ii.uam.es/~castells/publications/castells-uclm03.pdf>
25. **González Cornejo, J. E.; 2001.** *Arquitectura en Capas. Un camino hacia los procesos distribuidos.* Consultado [7 de enero del 2009]. Disponible en: http://www.docirs.cl/arquitectura_tres_capas.htm
26. **Duque Méndez, N. D.; 2008.** *Conceptos de arquitectura Cliente/Servidor.* Consultado [7 de enero del 2009]. Disponible en: http://www.it.uc3m.es/mcfp/docencia/si/material/1_cli-ser_mcfp.pdf
29. **Patrón MVC; 2008** Consultado [7 de enero del 2009]. Disponible en: <http://www.proactiva-calidad.com/java/patrones/mvc.htm>.
27. **Ramos Monso, Martin; 2004.** *Manual de programación en PHP. Sitios Web dinámicos e interactivos.* Consultado [7 de enero del 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg03729.pdf>
28. **Charte Ojeda, Francisco; 2002.** *Programación Visual C#.NET.* Consultado [7 de enero del 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg03242.pdf>
29. **Mayo, Joseph; 2002.** *C# al descubierto.* Consultado [7 de enero del 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg04041.pdf>
30. **Requisitos de la plataforma ASP.NET; 2006** *Manual del programador de .NET Framework.* Consultado [14 de enero del 2009]. Disponible en: [http://msdn.microsoft.com/es-es/library/aa719564\(VS.71\).aspx](http://msdn.microsoft.com/es-es/library/aa719564(VS.71).aspx)
31. **Vázquez Zambrano, Donel; 2008.** *Sistema de Ventas Mayoristas.* Tesis de Grado inédita. Universidad de Ciencias Informáticas. Ciudad de la Habana.
32. **WEBEXPERTO; 2002.** sitio web: www.elguruprogramador.com.a . Consultado [7 de enero del 2009]. Disponible en: <http://www.elguruprogramador.com.ar/articulos/instalacion-de-mysql-para-windows.htm>

33. **Medina, J. M.; 2008.** *Administración SGBD PostgreSQL*. Consultado [7 de enero del 2009]. Disponible en: <http://www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/SITARGES/manual.pdf>.
34. **MATEU, C; 2008.** *Desarrollo de aplicaciones Web*. Consultado [7 de enero del 2009]. Disponible en: http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/desarrollo-de-aplicaciones-web/desarrollo-de-aplicaciones-web/XP06_M2108_01497.pdf.
35. **Barrapunto; 2008.** *El servidor de web Apache*. Consultado [7 de enero del 2009]. Disponible en: <http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/c20.html>.
37. **Wikipedia; 2008.** *Subversion*. Consultado [7 de enero del 2009]. Disponible en: www.es.wikipedia.org/wiki/Subversion.
36. **Guerra Rodríguez, G.; Tula Mederos, R.; 2008.** *Sistema de Gestión de Información en el Proceso de Formación Posgraduada en Salud Módulo de Doctorados y Maestrías*. Tesis de Grado inédita. Universidad de Ciencias Informáticas. Ciudad de la Habana.
37. **Rose Connell, Betty; Jones, Mike; Mace, Ron; Mueller, Jim; Mullick, Abir; Ostroff Elaine, Sanford Jon, Steinfeld Ed, Story Molly, Vanderheiden Gregg; 1997.** *Los principios del diseño universal..* Consultado [7 de enero del 2009]. Disponible en: <http://www.abc-discapacidad.com/archivos/pud-spanishv2.pdf>
38. **Fernández Vilas, Ana; 2001.** *Diagrama de Despliegue*. Consultado [7 de enero del 2009]. Disponible en: <http://tvdι.det.uvigo.es/~avilas/UML/node50.html>
39. **Fernández Vilas Ana; 2001.** *Diagrama de Componentes*. Consultado [7 de enero del 2009]. Disponible en: <http://tvdι.det.uvigo.es/~avilas/UML/node49.html>
40. **British Standards Institution; 2009.** *Qué son los sistemas de gestión*. Consultado [12 de marzo del 2009]. Disponible en: <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>

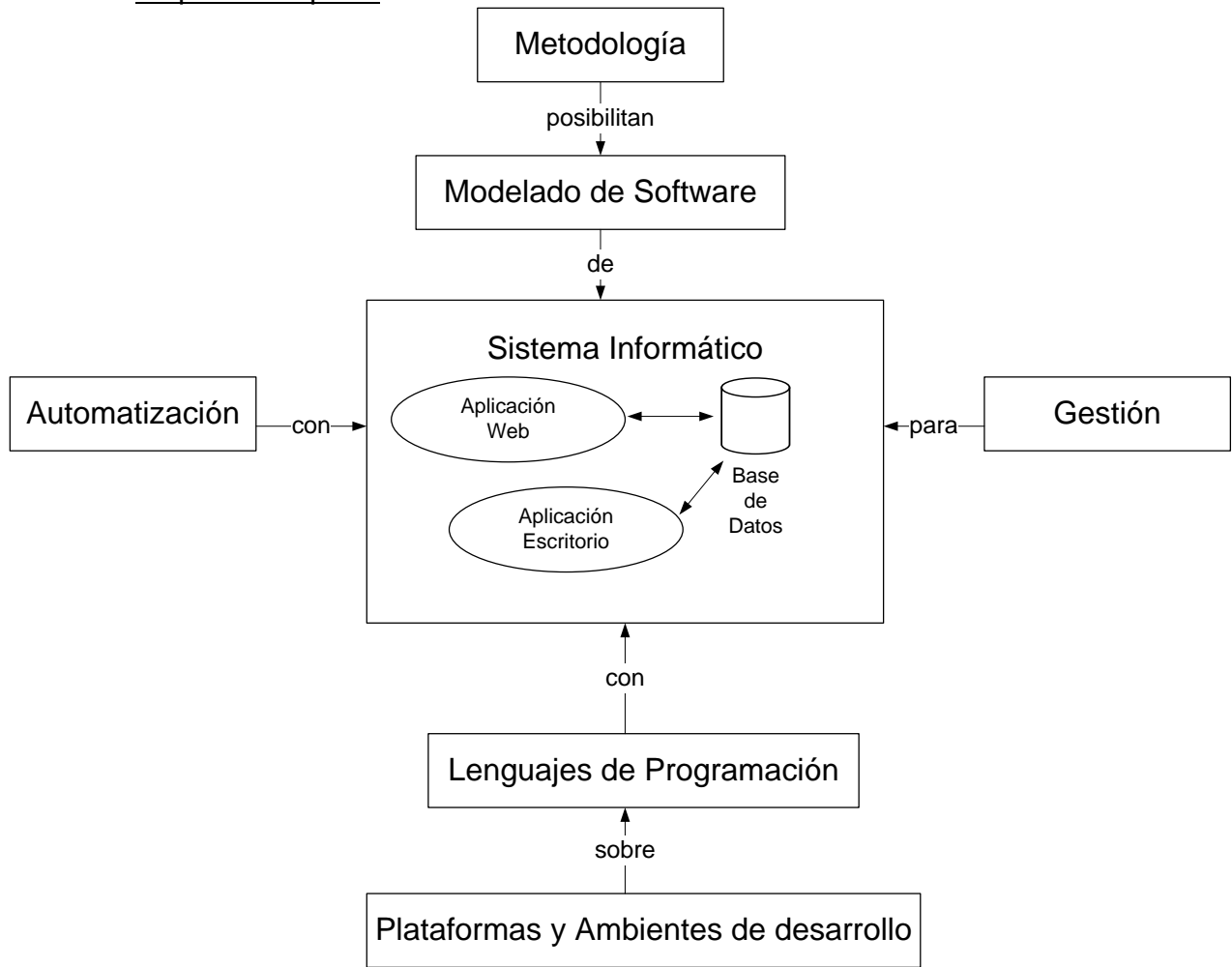
Bibliografía consultada.

1. **Pons, Juan P.; 2006.** *Las tecnologías de la información y la comunicación: un punto de vista educativo.* Consultado [7 de enero del 2009]. Disponible en: <http://www.ucm.es/info/multidoc/multidoc/revista/num8/jpablos.html>
2. **MIC; 2008.** *Tecnología de la información.* Consultado [7 de enero del 2009]. Disponible en: <http://www.mic.gov.cu/HThemEmp.aspx?1>
3. **Izquierdo, José M.; 2007.** *Las tecnologías de la información y las comunicaciones (TICs) en la gestión académica del proceso docente educativo en la educación superior universidad de oriente.* Consultado [7 de enero del 2009]. Disponible en: <http://revistas.mes.edu.cu/Pedagogia-Universitaria/articulos/2007/1/189407105.pdf/view>
4. **Pérez Ramos, Yandy; Gustabello Cogle, Robby; 2009.** *Las tecnologías de la información y las comunicaciones con enfoque a la educación.* Consultado [7 de enero del 2009]. Disponible en: <http://www.cujae.edu.cu/eventos/convencion/Sitios/Efing/trabajos/60C%20EF.%20Y.P%C3%A9rez,%20CUB.pdf>
5. **Martínez M., Andrés Y.; 2004.** *Una metodología para el diseño de sistemas de información, basada en el estudio de sistemas blandos.* Consultado [7 de enero del 2009]. Disponible en: http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0798-10152004000200005&lng=es&nrm=iso.
6. **Vela Dávila, José A.; 2008.** *Ambientes de Desarrollo de Software Basado en Componentes.* Consultado [7 de enero del 2009]. Disponible en: <http://www.cimat.mx/Eventos/seminariotecnologias08/javd.pdf>
7. **Carcassés Díaz, Lisbety; Abad Estrada, Suanny; 2007.** *Propuesta de diseño de una aplicación Web para la gestión de perfiles de los trabajos de diploma.* Tesis de Grado inédita. Universidad de las Ciencias Informáticas. Ciudad de la Habana.
8. **Heredia Rodríguez, Reynaldo; Salcerio Paz, Emilio; 2008.** *Aplicación Web para el control del proceso de tesis de grado en la facultad 8.* Tesis de Grado inédita. Universidad de las Ciencias Informáticas. Ciudad de la Habana.
9. **Núñez García, Dunel; 2008.** *Automatización de la gestión y reproducción de las tesis en la UCI.* Tesis de Grado inédita. Universidad de las Ciencias Informáticas. Ciudad de la Habana.

10. **Pressman, Roger S.; 2005.** *Ingeniería del Software: un enfoque práctico*. La Habana, Editorial Félix Varela, quinta edición. **Formato:** Impreso.
11. **Larman, C.; 1999.** *UML y Patrones: Introducción al análisis y programación orientada a objetos*. Consultado [10 de enero del 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg00062.pdf>
12. **Schmuller, J.; 2000.** *Aprendiendo UML en 24 horas*. Consultado [10 de enero del 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg00004.pdf>
13. **Rumbaugh, James; 2000.** *El Lenguaje unificado de modelado. Manual de referencia*. Consultado [10 de enero del 2009]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>
14. **Letelier Torres, Patricio; 2005.** *Desarrollo de Software Orientado a Objeto usando UML*. Consultado [10 de enero del 2009]. Disponible en: http://teleformacion.uci.cu/mod/resource/view.php?id=4599&subdir=/Materiales_Complementarios_Conf_1
15. **Guía de referencia del lenguaje C#; 2008.** *Documento con la especificación del lenguaje C#*. Consultado [14 de enero del 2009]. Disponible en: <http://www.msdn.microsoft.com/net/ecma>
16. **Portal de C# en español; 2009.** Consultado [14 de enero del 2009]. Disponible en: <http://tdg.lsi.us.es/~csharp>
17. **Portal sobre .NET de Microsoft; 2009.** Consultado [14 de enero del 2009]. Disponible en: <http://www.msdn.microsoft.net>
18. **S. Platt, David; 2008.** *Web Services: Building Reusable Web Components with SOAP and ASP.NET*. Consultado [14 de enero del 2009]. Disponible en: <http://www.msdn.microsoft.com/msdnmag/issues/01/02/webcomp/print.asp>
19. **Howard, Rob; 2009.** *Web Services in ASP.NET*. Consultado [14 de enero del 2009]. Disponible en: <http://www.msdn.microsoft.com/voices/asp02222001.asp>

Anexos.

Anexo 1. Mapa Conceptual



Anexo 2. Entrevistas.

Estas entrevistas se hicieron mediante 2 vías. Personal y por correo.

Las fechas en que se realizaron fueron entre el 13 y 20 de febrero del 2009 en algunas de las facultades de la UCI a una población escogida concientemente.

La población se caracterizó por estar involucrada en el proceso de atención de los trabajos de diploma en sus facultades o por ser autores de tesis realizadas sobre sistemas para la gestión de los trabajos de diplomas o por estar vinculado a la producción y haber desarrollado en su proyecto algún modulo o sistema para lo mismo.

Preguntas.

- ¿Se ha desarrollado en su proyecto algún modulo para la gestión de los trabajos de diploma?
- ¿Cuáles los resultados obtenidos del modulo?
- ¿Esta siendo utilizado en algún proceso de tesis de alguna facultad?

A personas involucradas en el proceso de atención a las tesis de su facultad.

En este caso al Jefe al Dpto. de PP de la Facultad 5 Yuniesky Coca.

- ¿Ha tenido algún resultado el trabajo de diploma “*Propuesta de diseño de una aplicación Web para la gestión de perfiles de los trabajos de diploma*” realizado en su facultad?
- ¿Este sistema desarrollado se utilizó en algún proceso de atención de tesis de su facultad realizado anteriormente o en el actual?
- ¿En estos momentos tienen algún sistema que le estén ayudando en la realización del proceso de atención a las tesis de su facultad?

A los autores de trabajos de diploma realizados sobre este tema:

- ¿Han tenido resultados de su trabajo?
- ¿Esta siendo utilizado en el proceso de atención a las tesis de su facultad?

Glosario de términos.

- 1 **Artefacto:** Producto tangible del proyecto que es producido, modificado y usado por las actividades.
- 2 **ASP:** Active Server Page, tecnología creada por Microsoft destinada a la creación de sitios Web.
- 3 **CC:** Prefijo para las clases controladoras.
- 4 **CE:** Prefijo para las clases entidades.
- 5 **CI:** Prefijo para las clases interfaces.
- 6 **Componente:** Es la unidad de construcción elemental del diseño físico.
- 7 **CP:** Prefijo para las páginas clientes o que corren del lado del cliente.
- 8 **CUN:** Caso de Uso del Negocio.
- 9 **CUS:** Caso de Uso del Sistema.
- 10 **HTML:** Lenguaje de Marcas de Hipertexto, es un lenguaje para elaborar páginas Web.
- 11 **HTTP:** Protocolo de Transferencia de Hipertexto, es el método más común de intercambio de información en Internet, el método mediante el cual se transfieren las páginas Web a un ordenador.
- 12 **IDE:** Entorno integrado de desarrollo.
- 13 **Interfaz:** Es uno de los componentes más importantes de cualquier sistema computacional, funciona como el vínculo o comunicación entre el humano y la máquina.
- 14 **Negocio:** Cualquier ambiente o entorno en cual esta enmarcado el problema.
- 15 **Requerimientos:** Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- 16 **TD:** trabajo de diploma.