



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 2 “TELECOMUNICACIONES Y SEGURIDAD INFORMÁTICA”**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**



Título: *“Sistema de Gestión del Movimiento de Alumnos Ayudantes”*

Autores:

Yusdani Hinojosa Domínguez.

Rangel Hurtados Díaz.

Tutor:

Ing. Yuliet Pérez Ananías.

Msc. Maura Berta Hidalgo.

Consultor:

Ing. Deivis Ricardo Álvarez.

Ciudad de La Habana, Junio del 2010.

“Año del 52 Aniversario del Triunfo de la Revolución Cubana

DECLARACIÓN DE AUTORÍA

Declaración de Autoría

Declaramos que Rangel Hurtados Díaz y YUSDANI HINOJOSA DOMÍNGUEZ somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (2) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de julio del 2009.

Firma del Autor
Rangel Hurtados Díaz

Firma del Autor
YUSDANI HINOJOSA DOMÍNGUEZ

Firma del Tutor
Ing. Yuliet Pérez Ananías

Firma del Tutor
Msc. Maura Berta Hidalgo



*“La revolución no se lleva en los labios para vivir de ella,
se lleva en el corazón para morir por ella”.*

DEDICATORIA



DEDICATORIA

Este trabajo de diploma va especialmente dedicado a mis padres y mi hermano, mis sobrinos y a toda mi familia, en especial a mi tías Midiala, Tuly, Cary, Lola, Agustina, mis primos Dagnys, Idito, Yangel, Manolo, Yuni y Agustinsito.

Rangel

Dedico este trabajo de diploma a mis padres, mis hermanos, mis tíos, mis primos, especialmente a la persona más importante en mi vida, mi mamá.

Yusdani

AGRADECIMIENTOS

“He vivido una vida repleta de problemas, pero no son nada comparado con los problemas que tuvieron que afrontar mi padres para lograr que mi vida empezase.” He empezado con esta frase porque mis primeros agradecimientos son para mis padres que toda una vida han estado conmigo en los momentos buenos y malos , guiándome siempre por el buen camino, siendo la máxima autoridad que ponían límites a mis deseos.

A mi hermano y sobrinos.

A mis tíos y tías en especial a mi tío Jorge Alberto que para mí es mi ejemplo a seguir, mi tía Midiala que me apoyo hasta el último momento de la tesis.

A toda mi familia en especial que todos aportaron su granito de arena para que todo lo que está sucediendo hoy en día sucediera.

A mi compañero de tesis que siempre nos ayudamos en todo este curso.

A mis tutores y en especial a mi consultor Deivis el cual me dio la mano, y de no ser por el esta tesis no hubiera sido. Y por último agradecer a la revolución cubana.

Rangel

A mis padres, en especial a mi madre, ya que gracias a su dedicación, apoyo y su cariño hoy se hace realidad uno de nuestros sueños.

A mis hermanos, mis tíos, mis primos gracias por su confianza en mi... bro, gracias por todo.

Al tribunal por el apoyo y la colaboración en el perfeccionamiento de este trabajo.

A nuestras tutoras por su dedicación y su apoyo, y a nuestro consultor por brindarnos su ayuda y su tiempo.

A todos nuestros profesores por formarnos con su ejemplo.

A mi compañero de tesis, por ser más que un amigo.

A mis amigos y amigas que nos apoyaron continuamente, sin ustedes, este día no hubiera sido posible.

A los que están, a los que estuvieron y a los que estarán.

Gracias a todos.

Yusdani

RESUMEN

RESUMEN

En la Facultad 2, desarrollar el proceso de gestión de Alumnos Ayudantes (AA) se torna engorroso, debido a los problemas existentes como retraso en la selección, aprobación y evaluación de AA, manejo de la información referente a los mismos en documentos físicos y de forma manual, pérdida y deterioro de documentos, entre otros. Por esta razón se ha determinado informatizar la gestión de la información que genera dicho proceso, a través de la construcción de una aplicación Web para minimizar los esfuerzos y problemas que se presentan actualmente.

En el presente trabajo se documentan las características de las herramientas y lenguajes utilizados en el desarrollo de la propuesta de solución. Siendo especificados cada uno de los procesos de trabajo a informatizar, así como las funcionalidades que brinda el sistema. Se realiza además la estimación del esfuerzo necesario por concepto de funcionalidades, obteniendo así un acercamiento al tiempo de desarrollo de la aplicación. Finalmente, se procede tanto a la implementación de la solución propuesta como al proceso de calidad referente a las pruebas de aceptación del mismo.

INDICE

ÍNDICE

INTRODUCCIÓN	9
1.1 INTRODUCCIÓN	13
1.2 SISTEMAS DE GESTIÓN	13
1.3 PROCESO DE GESTIÓN DEL MAA	14
1.4 ESTADO DEL ARTE	14
1.5 NECESIDAD DE UN SISTEMA DE GESTIÓN DEL MOVIMIENTO DE ALUMNOS AYUDANTES.	16
1.6 HERRAMIENTAS, Y METODOLOGÍAS	17
1.6.1 Metodología de desarrollo	17
1.6.2 XP (Programación Extrema)	17
1.6.3 Lenguaje de Modelado Unificado (UML).	18
1.6.4 Notación para el Modelado del Negocio	19
1.6.5 Herramienta CASE	20
1.6.6 Gestores de Base de Datos	21
1.6.7 Herramienta de Desarrollo (IDE)	22
1.6.8 Servidor Web	23
1.6.9 Las aplicaciones Web	24
1.7 LENGUAJES DE PROGRAMACIÓN	25
1.7.1 Modelo Cliente-Servidor	25
1.8 FRAMEWORKS	28
1.8.1 CodeIgniter	29
1.8.2 Dojo Toolkit	30
1.9 PATRÓN DE ARQUITECTURA (MVC)	30
1.10 CONCLUSIONES	31
2.1 INTRODUCCIÓN	32
2.2 OBJETO DE ESTUDIO	32
2.2.1 Problema y situación problemática	32
2.2.2 Objeto de automatización	32
2.3 DESCRIPCIÓN DE LOS PROCESO DEL NEGOCIO	33
2.3.1 Proceso Solicitud de Ayudantía	33
2.3.2 Proceso Dar Baja a un AA	35
2.3.3 Proceso Evaluar a un AA	35
2.4 INFORMACIÓN QUE SE MANEJA	36
2.5 PROPUESTA DE SISTEMA	38
2.5.1 Requerimientos Funcionales	38
2.5.1 Requerimientos No Funcionales	39
2.6 CONCLUSIONES.	40
3.1 INTRODUCCIÓN.	41
3.2 FASE DE EXPLORACIÓN	41
3.2.1 Historias de Usuario (HU)	41
3.3 FASE DE PLANIFICACIÓN	47
3.3.1 Estimación de esfuerzo por Historia de Usuario	47
3.3.2 Plan de iteraciones	48
3.3.3 Plan de duración de iteraciones	49

INDICE

3.3.4 <i>Plan de Entregas</i>	50
3.4 CONCLUSIONES	50
4.1 INTRODUCCIÓN	51
4.2 FASE DE IMPLEMENTACIÓN	51
4.3 FASE DE PRUEBA	63
4.3.1 <i>Pruebas de aceptación</i>	64
4.4 CONCLUSIONES	69
CONCLUSIONES	70
RECOMENDACIONES.....	71
BIBLIOGRAFÍA.....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXOS.....	73
GLOSARIO DE TÉRMINOS	80

INTRODUCCIÓN

INTRODUCCIÓN

El desarrollo de toda sociedad siempre ha estado vinculado estrechamente a las tecnologías con que cuenta la misma, las cuales van evolucionando cada vez más y en la medida en que crecen, evoluciona tecnológicamente la sociedad. Es por ello que en el mundo se incrementa la informatización de todos los sectores de la sociedad. De esta manera, las empresas, instituciones y otras organizaciones pueden automatizar la gran mayoría de sus procesos de trabajo de forma más eficaz.

En Cuba a partir de los años 1996 el país comenzó a recuperarse de la crisis producida por la desintegración de la Unión Soviética, se dan los primeros pasos para el ordenamiento del trabajo continuo, destinado a impulsar el uso y el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC). Así en 1997, la Resolución Económica del V Congreso del Partido Comunista de Cuba, refleja orientaciones precisas para trabajar en ese sentido y el gobierno aprueba por primera vez los lineamientos generales para la informatización de la sociedad. En consecuencia se produjeron avances que condujeron en enero del 2000 a la creación del Ministerio de la Informática y las Comunicaciones (MIC) en Cuba, con la misión fundamental de fomentar el uso masivo de las TIC en la economía nacional, la sociedad y el servicio del ciudadano. El gobierno decidió entonces analizar los lineamientos generales de 1997 y la forma de implementarlos, tomando como sustento los cuatro pilares fundamentales del proceso revolucionario cubano: la educación, la salud, la seguridad social y la cultura.

Conceptualmente la informatización de la sociedad se define en Cuba como el proceso de utilización ordenada y masiva de las TIC para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad, logrando consigo más eficacia y eficiencia, que permitan una mayor generación de riquezas y hagan sustentable el aumento sistemático de la calidad de vida de los ciudadanos, contribuyendo a su crecimiento y a su desarrollo tecnológico.

Cuba ha estado inmersa en el profundo y novedoso proceso de transformaciones educacionales y sociales como el programa de la Batalla de Ideas, a partir del cual se emprendieron nuevos programas destinados a elevar el nivel cultural de la población y su calidad de vida. En estas circunstancias surge la idea de utilizar el territorio que ocupaba la base militar Lourdes para construir la UCI, siguiendo la tradición de la revolución cubana de convertir cuarteles en escuelas, con el objetivo de lograr grandes avances en

INTRODUCCIÓN

la rama de la informática, en el desarrollo de la Industria Cubana del Software y en la materialización de los proyectos asociados al programa cubano de informatización.

La misión de la UCI es desarrollar software y servicios informáticos a partir del modelo de formación centrado en la producción. La infraestructura de la UCI coordina la actividad productiva y brinda servicios para asegurar el correcto desarrollo y terminación de proyectos, productos y servicios. Entre los servicios más importantes que brindase encuentran los de calidad del software, arquitectura y tecnología, servicios legales y diseño de comunicación visual. La producción se concentra en el desarrollo de proyectos y se destacan en las esferas de la salud, educación, software libre, teleformación, sistemas legales, realidad virtual, automatización, bioinformática y procesamiento de imágenes y señales. Los estudiantes de la UCI son el eslabón fundamental de todo lo antes mencionado, conjuntamente con los profesores que son los guías para todas las actividades desarrolladas, pero la labor de educandos es muy ardua y necesita ayuda. Es por eso que a partir del curso 2006-2007 surge en la universidad el Movimiento de Alumnos Ayudantes (MAA) “Dr. Orlando Ramos” para apoyar el proceso docente, sus integrantes son estudiantes con ciertos avances en la docencia, que están capacitados para impartir clases a otros estudiantes y ayudarlos en su aprendizaje. Un AA tiene la posibilidad de hacer llegar sus ideas de forma más comprensible a sus compañeros, porque existe una mejor relación y comunicación **estudiante – estudiante** que comparten intereses comunes, desencadenando esto cierta motivación e interés ante el estudio, llegando a convertirse en muchas ocasiones en un paradigma positivo para el aprendiz.

Hoy en día la UCI se encuentra inmersa en un proceso de crecimiento y extensión de AA, lo que trae consigo un aumento considerable del volumen de datos que se manejan en las secretarías. Por otra parte, el proceso que se emplea para manipular los datos de solicitud, aceptación y evaluación de ayudantía resulta engorroso y de moderada calidad. Esto se debe al uso de información en documentos físicos y digitales de forma manual que implica demora en el proceso, al deterioro de documentación legal, a la pérdida de información docente relativa a los estudiantes y a la probable introducción de errores que disminuye la efectividad del proceso de gestión del MAA.

Además, el MAA es un proceso donde se mueven recursos humanos. A cada AA le corresponde un estipendio que asciende en \$50; cuando un estudiante deja de ser AA en el período en que recursos humanos no ha actualizado sus nóminas, al mismo se le sigue pagando. Esto genera una pérdida

INTRODUCCIÓN

económica debido a la actualización tardía de las nóminas de recursos humanos con respecto a la actualización de plantillas de los AA.

De ahí surge la necesidad de crear una **aplicación** que sirva de apoyo a la facultad, para realizar el proceso de Gestión del MAA de forma más eficiente.

El **problema científico** se puede formular de la siguiente manera: ¿Cómo facilitar la Gestión Sistemática del MAA?

Por tanto, el **objeto de estudio** de este trabajo es el Proceso de Gestión del Movimiento de Alumnos Ayudantes.

Derivándose como **campo de acción** el proceso de Gestión del MAA en la Facultad 2.

Y el **objetivo general** es desarrollar un sistema para la Informatización de la Gestión del MAA.

De esta forma, se pueden definir las siguientes **tareas de investigación**:

1. Consultar la documentación existente sobre las tareas que realiza actualmente la Facultad 2 de forma manual para desarrollar los procesos de Gestión de los AA.
2. Realizar el estudio del estado del arte de los sistemas automatizados similares para mejorar el proceso.
3. Describir las características y ventajas de las aplicaciones Web.
4. Definir herramientas y lenguajes adecuados para la implementación del Sistema de Gestión de AA.
5. Modelar los Procesos de Negocio de la Gestión de AA.
6. Identificar y describir las Historias de Usuario del proceso de Gestión de AA.
7. Implementar las funcionalidades del sistema a realizar.

INTRODUCCIÓN

A continuación se muestra una breve descripción de cada uno de los 4 capítulos por los que está estructurado el presente documento.

Capítulo 1 “*Fundamentación teórica*”.

Se analiza todo lo relacionado con las metodologías y herramientas que van a ser usadas para desarrollar el sistema, así como un estudio de los software similares existentes en el ámbito nacional e internacional.

Capítulo 2 “*Características del Sistema*”

Incluye la descripción de los procesos del negocio que van a ser informatizados, la información que se maneja en el sistema a desarrollar, así como la descripción del mismo.

Capítulo 3 “*Exploración y Planificación*”

Se identifican las Historias de Usuario y se le establece una prioridad a cada una, a continuación se realiza la estimación del esfuerzo necesario para implementar cada una de ellas, se procede a la planificación de la etapa de implementación donde se genera el plan de iteraciones, el plan de duración de las mismas y el plan de entregas.

Capítulo 4 “*Implementación y Prueba*”

Se procede con la implementación de cada Historia de Usuario previamente descrita, y a la realización de pruebas de aceptación de las mismas una vez que se han transformado en funcionalidades.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza una descripción de los conceptos que son utilizados en la investigación para una mejor comprensión. Se realiza un estudio del estado del arte de los sistemas de gestión de la información existente y la nueva necesidad. También se hace referencia a las tecnologías actuales y se definen las principales herramientas que pudieran ser utilizadas para el desarrollo del sistema.

1.2 Sistemas de Gestión.

Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, procedimientos y procesos de la organización. Ayuda a lograr los objetivos de la organización mediante una serie de estrategias que incluyen la optimización de procesos, enfoques centrados en la gestión y el pensamiento disciplinado. (1) Se utilizan para gestionar información generada en escuelas, empresas, institutos de investigación, órganos de seguridad interior, entre otros, lo que posibilita aprovechar y desarrollar el potencial existente en cualquiera de estos sectores.

La implementación de un sistema de gestión eficaz puede ayudar a gestionar los riesgos sociales, mejorar la efectividad operativa, reducir costos, aumentar la satisfacción de los clientes y partes interesadas, lograr mejoras continuas y aportar claridad al mercado.

Entre los diferentes tipos de sistemas de gestión existentes podemos hacer énfasis en:

- ✓ **Sistema de Gestión de Información:** En la actualidad, es común escuchar acerca de la importancia de la información, conjunto de datos organizados y que tienen un significado sobre un determinado proceso o fenómeno, permitiendo así resolver problemas y tomar decisiones, convirtiéndose en un elemento importante en el desarrollo de la sociedad. Un Sistema de Gestión de Información incluye el proceso de realizar operaciones ya sean de extracción, manipulación, depuración y conservación con el objetivo de obtener una información adecuada en un tiempo determinado y obteniendo con ello los resultados esperados. Estos sistemas de gestión surgen con el objetivo de controlar grandes volúmenes de información de forma fácil, sencilla y confiable.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.3 Proceso de Gestión del MAA

Para hacer la solicitud de ayudantía el estudiante debe ir a la secretaría. Una vez allí la persona encargada de realizar el proceso, debe buscar información sobre el estudiante para ver si cumple los requisitos necesarios para ingresar al movimiento. Para ello visita el sistema de gestión académica “Akademos” y además buscar información en documentos físicos. Una vez que se sabe si el estudiante está apto para ejercer como AA, de acuerdo con los requisitos académicos según la asignatura que quiere ejercer, se debe consultar con el departamento (Dpto.) de esa asignatura; y de acuerdo con la cantidad de alumnos que necesiten para ayudantía y la consideración de los profesores, será aceptado o no, y se le dará conocimiento al estudiante unos días después. Además, con los estudiantes que quieren continuar en el movimiento, muchas veces no se les puede verificar que siguen cumpliendo los requisitos necesarios, o si los han perdido porque esa información se registra de forma física, todo el proceso antes descrito resulta engorroso a la hora de hacerlo, porque se maneja gran cantidad de información y se realiza de forma manual.

Dentro de la información de los AA se encuentran la Planilla de Solicitud, las evaluaciones tanto docentes como en las residencias, el Plan de Trabajo, Expediente del Estudiante, entre otros. Para su registro y procesamiento existen procedimientos manuales, entrando en contradicción con el nivel de informatización alcanzado por otros procesos que se llevan a cabo en la universidad, generándose baja calidad y entorpeciendo el desarrollo de una adecuada estrategia para su gestión y desarrollo.

1.4 Estudio del Estado del Arte

En el plano internacional podemos citar el **Sistema de Control de Bienes Web-UG**, el mismo es un módulo que forma parte del Sistema Integral de Información Administrativa (SIIA) creado en la universidad de Guanajuato (México), este módulo fue desarrollado con el propósito de apoyar a las unidades académicas y administrativas de la universidad en el control de los bienes materiales que tienen bajo su cuidado, permitiendo el registro oportuno, brindando una información veraz, garantizando el resguardo de los bienes, así como la identificación, registro y etiquetado de los bienes, además la custodia de los documentos, también la aplicación permite al usuario realizar su propio registro de bienes, toda la

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

información que el usuario captura queda disponible para el departamento de control de bienes, guardándose la marca, modelo, descripción y el número de la serie.

El **SIU** (Sistema de Información Universitaria)-Guaraní es un sistema de gestión de alumnos que registra y administra todas las actividades académicas de la universidad, desde que los alumnos ingresan como aspirantes hasta que obtienen el diploma. Fue concebido para administrar la gestión de alumnos en forma segura, con la finalidad de obtener información consistente para los niveles operativos y directivos. El sistema brinda servicios para alumnos, docentes, usuarios administrativos y autoridades, ya que pueden explorar los datos y obtener información como soporte para toma de decisiones. Brinda los siguientes servicios a los docentes tales como consultas de agendas y notas de los alumnos, alta y baja de evaluaciones parciales, Envío de mensajes a estudiantes y al área de alumnos de la Universidad. Los alumnos por su parte pueden hacer inscripciones a exámenes, reinscripción a carrera, realizar consultas de plan de estudios, de historia académica, de notas de evaluaciones, de materiales regulares, entre otras. (2)

Sistema de Gestión de la Información de un Departamento Docente es un sistema para realizar la gestión de la información en un Dpto. docente, teniendo en cuenta la necesidad del intercambio de información, rápido acceso y nivel de actualización. Está constituido por 4 Módulos:

- ✓ Módulo de Capital Humano.
- ✓ Módulo de Formación del Profesional.
- ✓ Módulo de Educación de Postgrado.
- ✓ Módulo de Ciencia y Técnica.

Estos módulos facilitan la gestión de la información del Capital Humano de un Dpto. Docente, dentro de ellos quedan registrados los datos personales de los trabajadores del Dpto., cumplimiento del Plan de Trabajo, los objetivos del profesor, su evaluación, etc. Se controlan los datos referentes a sus asesoramientos a otros profesores. También se desarrolla todo el trabajo de planificación de la carga docente de los profesores.

Por otro lado, se gestiona la documentación de la Educación Posgraduada: cursos, maestrías, diplomados, doctorados que se ofertan o son recibidos por profesores del Dpto., obteniendo de reportes como el Plan de Postgrados del Dpto., el estado de la superación del claustro, su planificación y control. Permite la inscripción en las diferentes opciones del sistema de postgrado. Además de lo anterior, también

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

se trabaja con la documentación de Ciencia y Técnica a través de proyectos, líneas de investigación, publicaciones, etc. De los profesores de un Dpto. Docente.

En Cuba podemos citar el ejemplo del sistema de gestión **BABEL**, este es un sistema que integra tecnologías de la información de las solicitudes de los servicios de traducción e interpretación del Centro de Información. Mediante una interfaz de comunicación amigable los usuarios pueden realizar el intercambio de datos entre todas las funciones implicadas en este proceso y este sistema le permite al usuario tener la información precisa sobre el estado de su solicitud y las competencias del traductor, al aumentar el valor añadido de cada recurso que interviene en el proceso. Esta herramienta de trabajo permite la organización, clasificación de la información, la recuperación de documentos con oportunas normas de seguridad.

UCI Reservaciones, disponible en <https://transportacion.uci.cu/>, es un sistema capaz de agilizar la gestión del proceso de reservaciones de pases masivos que se realizan en la Universidad de las Ciencias Informáticas (UCI), este es de acceso rápido, confiable, seguro y garantiza un flujo adecuado de información. El mismo brinda varias funcionalidades entre las que se encuentra la reservación para la reservación de transporte, obtención de reportes de las reservaciones y distribuciones que se generan a partir de un viaje o salida, búsqueda para visualizar usuarios registrados o no en el sistema, el envío automático o personal de correos en modo de aviso o información a los usuarios.

1.5 Necesidad de un Sistema de Gestión del Movimiento de Alumnos Ayudantes.

Como resultado del análisis realizado a los sistemas automatizados estudiados a nivel internacional se detecta que estos se acercan a la propuesta pero no pueden ser aplicadas directamente en el objeto. En la República de Cuba no se encontró ninguna aplicación que informaticase el proceso de Gestión de AA. Por lo que se propone desarrollar un sistema informático, capaz de aprovechar las ventajas de las tecnologías de la información y las comunicaciones para la captación de los AA.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.6 Herramientas, y metodologías

1.6.1 Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de procedimientos y técnicas utilizadas para ordenar, planificar y controlar el proceso de desarrollo en sistemas de información. Su objetivo es desarrollar un software con mayor eficiencia y calidad, dándole solución a los problemas existentes durante su producción, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

1.6.2 XP (Programación Extrema)

Es una de las metodologías de desarrollo de software más apropiadas para proyectos a corto plazo. Como clave para su éxito en el desarrollo, se basa en reforzar las relaciones interpersonales, siendo muy importante el aprendizaje de los desarrolladores y promoviendo a su vez el trabajo en equipo, tiene la particularidad de que el cliente forma parte del equipo de desarrollo, existiendo así una comunicación fluida entre ellos, lo que es uno de los requisitos para obtener un producto con la calidad deseada. Consiste en una programación rápida y extrema, y está definida principalmente para proyectos con requisitos imprecisos y muy cambiantes, por ello esta metodología presenta los mecanismos necesarios para adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto.

El ciclo de desarrollo consiste en los siguientes pasos:

- ✓ El cliente define el valor de negocio a implementar.
- ✓ El programador estima el esfuerzo necesario para su implementación.
- ✓ El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- ✓ El programador construye ese valor de negocio.
- ✓ Se regresa al paso 1.

Se basa en tres características fundamentales:

- ✓ Pruebas Unitarias: son las pruebas realizadas a los principales procesos durante el desarrollo, lo que permite detectar las fallas en cualquier momento del desarrollo del software.
- ✓ Re-fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio y permitiendo un continuo perfeccionamiento.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- ✓ Programación en pares: consiste en la participación de dos desarrolladores en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está desarrollando.

XP propone que los proyectos comienzan con cosas pequeñas y se añadan poco a poco las demás funcionalidades, que el manejo del cambio se convierta en parte sustantiva del proceso. El costo de los cambios no es dependiente de la fase en la que se encuentre en proyecto y no se introduce ninguna funcionalidad antes de que sea necesaria. Al cliente le permite decidir qué es lo que se desarrollará, el estado real en que se encuentra el desarrollo; cambiar, incluir o eliminar requerimientos en cualquier momento.

Esta metodología promueve la comunicación entre los usuarios y los desarrolladores; la simplicidad al desarrollar y codificar los módulos del sistema; y la retroalimentación concreta y frecuente del equipo de desarrollo, del cliente y los usuarios finales.

Se optó por elegir XP como metodología para el desarrollo de nuestro sistema debido a que se dispone de poco tiempo para el desarrollo del software, el mismo es realizado por dos programadores, de esta manera, se logra una menor tasa de errores, y un mejor diseño, permitiendo cambiar cualquier parte del código en el momento que fuese necesario. El cliente es parte del equipo de desarrollo y se tiene muy buena comunicación con el mismo, es muy importante mostrar continuamente el progreso del software al cliente, verificando que sus requerimientos estén siendo cumplidos.

1.6.3 Lenguaje de Modelado Unificado (UML).

Para la fundamentación del análisis del sistema a desarrollar, se utilizará un lenguaje de modelado visual para la realización de los diagramas necesarios para el proceso de desarrollo, se optó por utilizar UML ya que es usado para especificar, construir y documentar esquemas de sistemas de software, el mismo divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto. El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos.

Los principales beneficios de UML son:

- ✓ Mejores tiempos totales de desarrollo (de 50 % o más).

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- ✓ Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- ✓ Establecer conceptos y artefactos ejecutables.
- ✓ Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- ✓ Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- ✓ Mejor soporte a la planeación y al control de proyectos.
- ✓ Alta reutilización y minimización de costos. (3)

1.6.4 Notación para el Modelado del Negocio

IDEF (Integrated Definition for Function Modeling) es una notación para representar flujos de trabajo, usada para representar diseños de circuitos eléctricos, bases de datos, entre otros problemas. Se utilizará esta notación para la modelación de los procesos del negocio. Por la estructura jerárquica que posee para representar un proceso, permite ordenarlos de un nivel más alto de abstracción hasta el nivel de detalle que sea necesario, además se puede analizar, documentar, mejorar los procesos de producción y facilitar la comunicación y captura de información. IDEF está conformado por actividades, entradas, salidas, mecanismos de control y sujetos que realizan las actividades.

Explicando la notación a seguir, se pueden dibujar los procesos del negocio y disminuir el tiempo de modelación.

Ventajas de IDEF:

- ✓ Mejor conocimiento y control de los procesos, permitiendo de esta forma identificar posibles procesos redundantes o defectuosos.
- ✓ Poder minimizar el tiempo de identificación y reparación de averías así como el tiempo de producción.
- ✓ Mejor organización de los recursos humanos y materiales.
- ✓ Más flexibilidad ante los cambios.
- ✓ Facilitar un análisis profundo de las entradas y salidas, así como los elementos de control y recursos de cada nivel.

Cada actividad se representa con un rectángulo, deben tener obligatoriamente entradas y salidas y de manera opcional los mecanismos de control y sujetos que ejecutan las actividades. Una entrada es

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

representada como una conexión que entra a la actividad por la izquierda y una salida como una conexión que sale por la derecha. Un mecanismo de control se representa como una conexión que entra por la parte superior de la actividad y un sujeto se representa como una conexión que entra a la actividad por la parte inferior. Las entradas y salidas son datos o elementos de información. Una entrada a una actividad es cualquier dato necesario a la misma. Una salida es un dato que la actividad tiene como responsabilidad producir. Un mecanismo de control es la información que regula la realización de la actividad. A diferencia de una entrada, cuyo contenido es dinámico, un mecanismo es una colección de información que tiene una estabilidad relativa. Los sujetos son quienes realizan las actividades en el proceso.

1.6.5 Herramienta CASE

Las herramientas CASE también permiten a los analistas tener más tiempo para el análisis y diseño y minimizar el tiempo para codificar y probar. La principal ventaja de la utilización de una herramienta CASE, es la mejora de la calidad de los desarrollos realizados y, en segundo término, el aumento de la productividad. (4)

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo de software, dígame análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Esta herramienta fue usada ya que permite dibujar todos los tipos de diagramas de clases, realizar ingeniería tanto directa como inversa, generar código desde diagramas y generar documentación automáticamente en formatos como Web o PDF. Es libre y multiplataforma¹.

Microsoft Office Visio 2007 es una herramienta desarrollada para brindar una manera más descriptiva y eficiente de presentar información. Posibilita la visualización, el análisis y la comunicación de información que puede ser organizada en diferentes plantillas, diagramas de flujos de datos, diagramas de redes, modelos de bases de datos, diagramas de software entre otros. Los esquemas que se generan son utilizados para optimizar procesos empresariales, dar seguimiento a proyectos y recursos, es decir, para mostrar de manera eficaz la información de un trabajo realizado.

¹ Multiplataforma es un término para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software que puedan funcionar en diversas plataformas.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.6.6 Gestores de base de datos

Un sistema gestor de bases de datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (5)

Un SGBD debe permitir:

- ✓ Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- ✓ Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.

Para el desarrollo de nuestra aplicación se hará uso del SGBD libre **PostgreSQL**, el cual posee las características siguientes:

- ✓ Altamente Extensible: soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario.
- ✓ Soporte SQL Comprensivo: soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- ✓ Integridad Referencial: soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- ✓ API Flexible: La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS² PostgreSQL.
- ✓ Lenguajes Procedurales: soporta lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

² Sistema administrador de bases de datos relacionales.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- ✓ Cliente/Servidor: usa una arquitectura proceso por usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL. (6)

Además, PostgreSQL corre en casi todos los sistemas operativos (LINUX, UNIX, MacOS, Windows). Soporta todas las características de una base de datos profesional (triggers (disparadores), store procedures (procedimientos almacenados), secuencias, relaciones, reglas y tipos de datos definidos por usuarios. (7)

1.6.7 Herramienta de Desarrollo (IDE)

Las herramientas de programación son aquellas en las que se desarrollan los programas, las mismas pueden soportar diferentes lenguajes de programación, en dependencia de para cual o cuales están destinadas. Pueden o no presentar una interfaz gráfica, pero independientemente de ello, permiten crear programas, rutinas, aplicaciones Desktop, aplicaciones Web, entre otras.

Para el desarrollo de este proyecto se seleccionaron como herramientas de desarrollo Dreamweaver en su versión 8 y NetBeans IDE en su versión 6.7.1 de las cuales se exponen las principales características a continuación.

Dreamweaver 8 es la opción profesional para crear sitios Web y aplicaciones, dado que proporciona una potente combinación de herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código, características todas ellas que permiten a los desarrolladores y diseñadores más expertos o menos expertos crear rápidamente sitios Web y aplicaciones basados en estándares. Desde el avanzado soporte de diseño basado en CSS a las funciones de codificación manual, Dreamweaver proporciona las herramientas profesionales que requiere un entorno integrado y agilizado. Los desarrolladores pueden utilizar Dreamweaver con su tecnología de servidor preferida para crear potentes aplicaciones en Internet destinadas a conectar a los usuarios a las bases de datos, las fuentes de datos dinámicos y los sistemas heredados. (8)

Además, incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios web, actualizando el sitio web en el servidor sin salir del programa. (9)

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

NetBeans IDE 6.7.1 (IDE Integrated Development Environment) es una aplicación de código abierto diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java.

NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones Web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de paquetes. (10)

Es un software de código abierto que soporta el desarrollo de todos los tipos de aplicaciones en Java como son J2SE, WEB, EJB 3 y JAX-WS, así mismo permite incluir paquetes para el desarrollo en lenguajes como C/C++, PHP entre otros.

1.6.8 Servidor Web

Un servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. (11)

Existen varios ejemplos de servidores Webs como Internet Information Server (IIS), AOL Server, Java Server, Hawkeye, Apache Web Server entre otros.

Se utilizará de todos los servidores mencionados el **Apache** Web Server. Este es un servidor Web de tecnología Open Source (código abierto) sólido, su licencia es descendiente de BSD³, dentro de las características de este servidor se pueden destacar:

- ✓ Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- ✓ Apache es una tecnología gratuita de Open Source.
- ✓ Apache es un servidor altamente configurable de diseño modular.
- ✓ Es muy sencillo ampliar las capacidades del servidor.
- ✓ Apache trabaja con lenguajes como Perl, PHP, Java y otros lenguajes de script⁴.

³ Licencia de software libre permisiva, permite el uso de código fuente en software libre.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- ✓ Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- ✓ Tiene una alta configurabilidad en la creación y gestión de logs. Permite la creación de ficheros de log⁵ a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

1.6.9 Las aplicaciones Web

A medida que ha avanzado el mundo de la informática, las aplicaciones Webs han ido haciéndose cada vez más prácticas, esto está dado por las facilidades que ofrecen:

- ✓ Pueden ejecutarse en cualquier navegador web mínimamente actualizado.
- ✓ Permiten una comunicación activa entre el usuario y la información; esto hace que el usuario acceda a los datos de manera interactiva, porque la página responderá a cada una de sus acciones.
- ✓ No dependen de ningún sistema operativo ni de configuración de hardware específica.
- ✓ Su disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones.
- ✓ Sus actualizaciones se hacen de una manera muy sencilla, sin necesidad de hacer descargas, instalaciones o comprar físicamente el producto.
- ✓ Las tareas que realiza el software no consumen recursos del usuario ya que la aplicación no se encuentra en su ordenador.
- ✓ Pueden ser usadas por múltiples usuarios a la vez.
- ✓ Flexibilidad de determinar niveles de acceso según la confidencialidad de los datos.

⁴ Un script es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Es muy utilizado para la administración de sistemas UNIX. Son ejecutados por un intérprete de línea de comandos. Usualmente son archivos de texto. (19)

⁵ Registro de todos los hits que un servidor ha recibido en un período de tiempo dado, que puede ser utilizado por auditores externos para registrar el uso de los sitios.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.7 Lenguajes de programación

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. Son ejemplos de lenguajes de programación: PHP, Prolog, ASP, Action Script, Ada, Python, Pascal, c, Basic, JAVA, Java Script, entre otros. (12)

A continuación se hace referencia a distintos tipos de lenguajes de programación web teniendo en cuenta que los mismos se dividen, de acuerdo con la propia arquitectura cliente- servidor, en lenguajes del lado del cliente y lenguajes del lado del servidor.

1.7.1 Modelo Cliente-Servidor

En el mundo de TCP/IP las comunicaciones entre computadoras se rigen por lo que se llama modelo cliente servidor, Modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad, y escalabilidad en las comunicaciones. Desde el punto de vista funcional, se puede definir la computación Cliente\Servidor, como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. (Ver Figura 1)

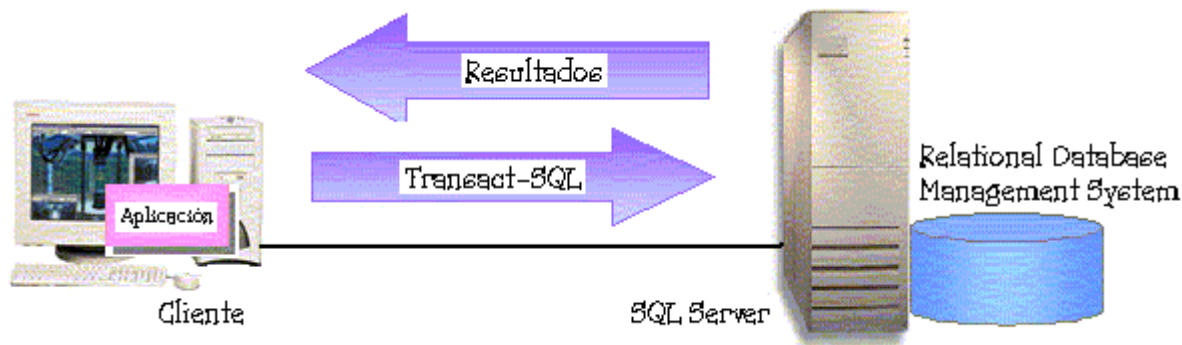


Figura 1. Modelo Cliente-Servidor.

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio).

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Permite distribuir físicamente los procesos y los datos en forma más eficiente lo que en computación distribuida afecta directamente el tráfico de la red.

1.7.1.1 Tecnologías del lado del cliente

El cliente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- ✓ Administrar la interfaz de usuario.
- ✓ Interactuar con el usuario.
- ✓ Procesar la lógica de la aplicación y hacer validaciones locales.
- ✓ Generar requerimientos de bases de datos.
- ✓ Recibir resultados del servidor.
- ✓ Formatear los resultados.

1.7.1.1.1 HTML

HTML, siglas de **Hypertext Markup Language** es un lenguaje para la creación de páginas web. Usado para la descripción de la estructura y el contenido en forma de texto. Es el lenguaje en que se escriben los millones de documentos que hoy existen en la WWW (World Wide Web). Cuando accedemos a unos de estos documentos, el cliente los interpreta y los muestra.

Son ficheros de texto plano (también conocidos como ASCII⁶) que pueden ser editados en cualquier editor de texto (Bloc de Notas). Suele tener la extensión HTML. Se basa en la sintaxis SGML⁷ (Standard Generalized Markup Language) (13). Tiene como ventaja que lo soportan una gran variedad de navegadores y exploradores, además su compatibilidad con los navegadores más antiguos y también

⁶ ASCII (American Standard Code for Information Interchange): Este código se basa en un conjunto de caracteres del alfabeto latino usando tanto el inglés como otras lenguas occidentales.

⁷ SGML (Standard Generalized Markup Language): Es un sistema para organizar y etiquetar Documentos. Fue Definido por la norma ISO 8879 en 1986.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

disminuye el costo a las personas que empiezan a aprender a leer y a escribir HTML, siendo consigo el lenguaje más fácil de interpretar y de aprender.

1.7.1.1.2 JavaScript

JavaScript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. Es un lenguaje basado en objetos que no requiere compilación, usado para acceder a objetos en aplicaciones. Es utilizado en las páginas web para el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Los programas JavaScript tienden a ser pequeños y compactos, no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red.

1.7.1.1.3 CSS

CSS (Cascading Style Sheets): Las hojas de estilo en cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML. Es una tecnología que nos permite crear páginas web de una manera más exacta. Gracias a las CSS somos mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc. (14) Mediante las CSS tenemos el control de presentación de muchos documentos desde una única hoja de estilos, si en algún momento se modificase cualquier estilo, automáticamente se actualizará el estilo de todos los documentos vinculados a la hoja de estilos. Empleando CSS, el archivo de HTML tiene un tamaño menor y los navegadores pueden leer de forma más sencilla el sitio Web.

1.7.1.2 Tecnologías del lado del servidor

Servidor es la máquina desde la que se suministran servicios y que está a la espera del requerimiento del cliente. Una vez hecho, busca la información solicitada y le envía la respuesta al cliente; incluso puede enviar varios servicios a la vez, lo que es posible porque entre ellos están conectados mediante redes.

(15)

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Existen diversos lenguajes para el desarrollo en las tecnologías del lado del servidor como ASP, ASP.NET, JSP, PHP, etc. Para el desarrollo de nuestra aplicación se hará uso del PHP 5 por las características que serán descritas a continuación.

1.7.1.2.1 PHP

PHP, es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios web con los cuales se puede programar las páginas HTML y los códigos de fuente. Se trata de un “lenguaje interpretado” usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web, además: (16)

- ✓ Es un lenguaje multiplataforma, con capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- ✓ Lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- ✓ Posee capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- ✓ Permite las técnicas de Programación Orientada a Objetos y permite crear los formularios para la web.
- ✓ Es sencillo de utilizar, no se requiere tener muchos conocimientos en programación.

1.8 Frameworks

Es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son:

- ✓ Acelerar el proceso de desarrollo.
- ✓ Reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

La mayoría de los frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC⁸, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación. (17)

Teniendo en cuenta el tiempo disponible para realizar la aplicación, se decidió utilizar **CodeIgniter** por ser más liviano, rápido y fácil de aprender, además, cumple perfectamente el fin de cualquier framework, una estructura bien definida que da soporte a nuestro proyecto web y que ayuda a que este nuevo proyecto sea organizado y desarrollado. Para desarrollar la interfaz gráfica se hará uso del **Dojo Toolkit**.

1.8.1 CodeIgniter

CodeIgniter es un framework que permite crear webs dinámicas con PHP, creado para ofrecer un alto rendimiento, es bastante ligero y muy sencillo de configurar. Algunas características son: (18)

- ✓ Compatible tanto para PHP4 como para PHP5
- ✓ Magnífica documentación y enorme comunidad de desarrolladores.
- ✓ Facilidad, puesto que no requiere más que una media hora de estudio para empezar a hacer cosas interesantes.
- ✓ A la hora de desarrollar un proyecto no es necesario crear toda la estructura desde cero.
- ✓ Usa el patrón Modelo Vista Controlador (MVC) como arquitectura de paradigma de desarrollo. Separando la vista, el acceso a datos y el controlador.
- ✓ Provee un rico juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y estructura lógica para acceder a esas librerías
- ✓ Permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada.

⁸ MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz del usuario y la lógica de control en tres componentes distintos, o sea, todo el proceso se encuentra dividido en tres capas (Modelo, Vista y Controlador).

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- ✓ El sistema puede ser fácilmente extendido a través del uso de plugins y librerías asistentes, o a través de extensión de clases o ganchos del sistema.
- ✓ Genera URLs⁹ limpias.

1.8.2 Dojo Toolkit

Dojo Toolkit conocido como “la navaja suiza del ejército de las bibliotecas del JavaScript“, es un framework JavaScript que permite el desarrollo de aplicaciones web enriquecidas en el cliente y Ajax. Es popular porque está integrada en numerosos IDEs y otros frameworks para desarrollo de webs.

Características del Dojo Toolkit:

- ✓ Contiene APIs y widgets (controles) para soportar el desarrollo de aplicaciones Web.
- ✓ Contiene un sistema de empaquetado inteligente, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX.
- ✓ Resuelve asuntos de usabilidad comunes como ser la navegación y detección del browser, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking).
- ✓ Múltiples puntos de entrada
- ✓ Independencia del intérprete

1.9 Patrón de Arquitectura (MVC)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El Modelo, las Vistas y los Controladores se tratan como entidades separadas. Cualquier cambio producido en el modelo se refleja en cada una de las Vistas.

Componentes:

- ✓ El **Modelo** administra el comportamiento y los datos de dominio de aplicación. Es el encargado del acceso a datos.

⁹ Algo muy útil que permite el CodeIgniter es utilizar "URLs limpias", lo que hace esto es quitar cualquier nombre de archivo y extinción al final de la URL.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

- ✓ La **Vista** Maneja la representación visual de los datos representados por el Modelo. Define la interfaz de usuario.
- ✓ El **Controlador** interpreta las acciones del usuario, ya sea por del ratón o el teclado, haciendo que el modelo y la vista cambien según resulte apropiado.



Figura 2. Modelo Vista Controlador

1.10 Conclusiones

En este capítulo se realizó un análisis exhaustivo sobre la importancia de los sistemas de gestión. Además, se realizó una investigación sobre sistemas automatizados similares al nuestro para tener una guía a la hora de desarrollar la aplicación y con el objetivo de agilizar nuestro trabajo se escogieron las herramientas y tecnologías que tuvieran mayores ventajas.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se tratará el objeto de la investigación, así como la situación del problema. Se especifica el objeto de automatización, la información que se maneja, la propuesta del sistema y la descripción de los procesos del negocio del sistema a desarrollar.

2.2 Objeto de estudio

2.2.1 Problema y situación problemática

Limitaciones y fallas en el proceso de selección de AA: esto se debe a la demora de 20 a 30 días de proceso, por parte de los involucrados. Características negativas de los estudiantes como son las sanciones, participación en un consejo disciplinario, etc. muchas veces son desconocidas; trayendo consigo la selección incorrecta de un estudiante para integrar el MAA y propiciando que el proceso de selección sea ineficiente y de baja calidad.

También surgen problemas con los AA como el poco conocimiento de su desempeño. Esta situación desencadena un trabajo de mala calidad e incumplimiento de las actividades fundamentales por parte del AA. Otro aspecto es el pago ascendente en \$50 a estudiantes que ya han dejado de integrar el movimiento; dado en este caso porque el método usado para manejar esa información es ineficaz. No se lleva el control de las evaluaciones de los AA, existe la pérdida de documentos físicos, demoras en actualizaciones de las nóminas de recursos humanos. Este proceso no se puede realizar de forma eficiente en los momentos actuales porque no existen las condiciones adecuadas.

2.2.2 Objeto de automatización

Se analizó el problema y fueron identificados los procesos principales que desarrollan el negocio en la actividad de gestión de los AA, que es la tarea fundamental que debe cumplir el sistema. Se determinó

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

que el proceso de informatización estará enfocado a la gestión de la información de los AA, y del proceso de selección, evaluación y baja de un AA, para dar solución a la problemática que existe actualmente.

2.3 Descripción de los Proceso del Negocio

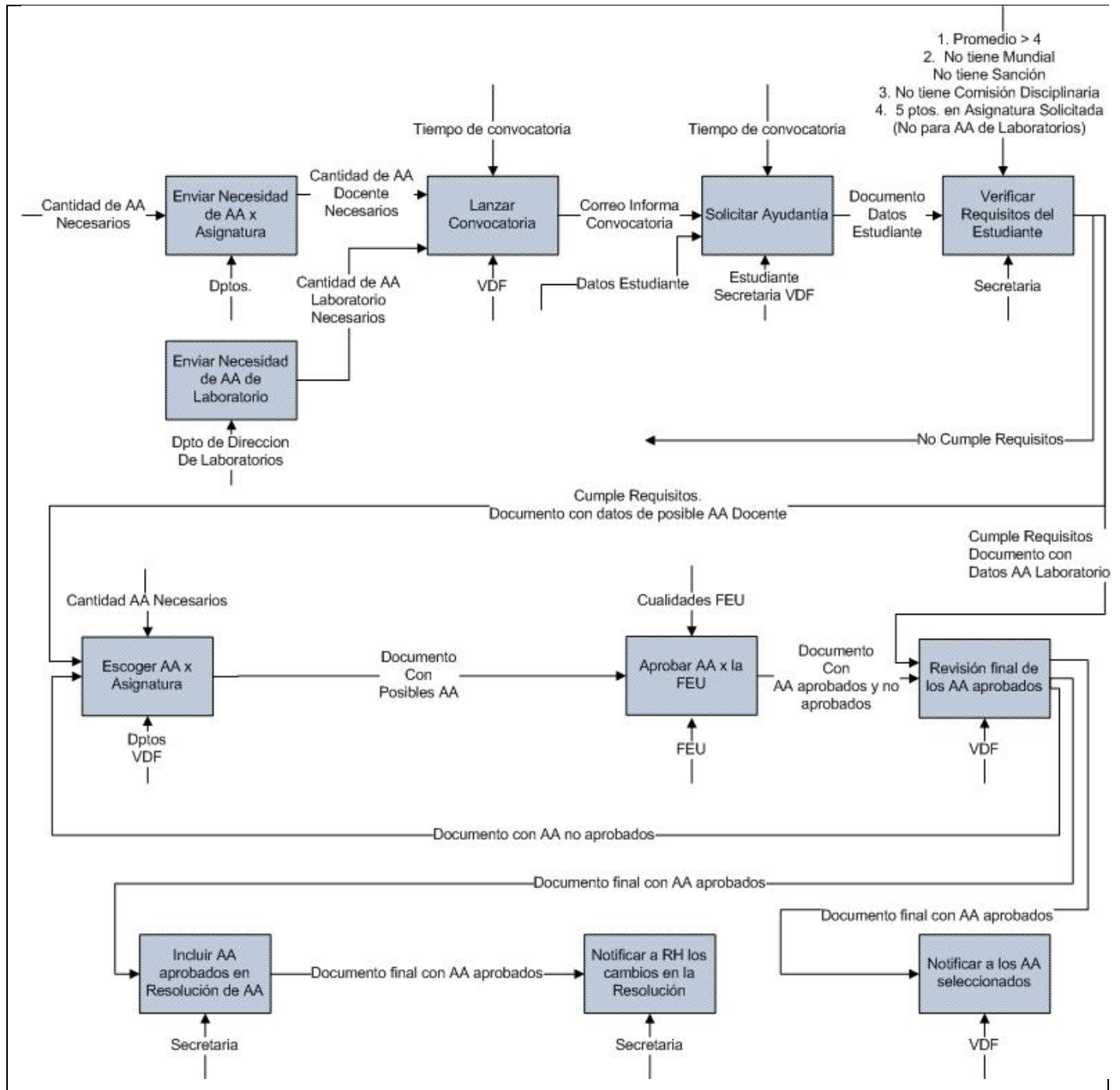
Para la descripción de los procesos del negocio del presente trabajo, se utilizará la modelación mediante la notación IDEF con el objetivo de cumplir con las exigencias del cliente en el tiempo establecido, y agilizar la identificación, definición y producción de los procesos del negocio. Seguidamente se describen los procesos del negocio que sustentan el sistema que se desarrolla.

2.3.1 Proceso Solicitud de Ayudantía

Los Departamentos (Dptos) Docentes y de Laboratorio de la Facultad envían al Vice Decano de Formación (VDF) las necesidades de ayudantía. El VDF lanza una convocatoria, vía correo, con las necesidades antes mencionadas a todos los estudiantes de la facultad. El estudiante interesado en ser AA, se presenta en el Vice-Decanato en el período en que se establece la convocatoria, donde la Secretaria de Formación archiva sus datos en el Documento de Solicitudes. Al cerrar el plazo de admisión de solicitudes, se envía el Documento de Solicitudes a la Secretaría Docente, quien verifica cuales estudiantes cumplen con los requisitos necesarios para integrar el MAA, actualizando el Documento de Solicitudes. Seguidamente se envía este último al VDF, quien organiza una reunión por Dpto. para escoger los estudiantes que van a ser AA. Una vez seleccionados los AA, el VDF da la tarea a la FEU de aprobar o no a los AA. A continuación se le entrega al VDF el Documento de Alumnos Ayudantes aprobados por la FEU, así como aquellos que no fueron aprobados para informar a los Dptos. la nueva necesidad. Finalmente, la Secretaria Docente incluye a los AA en la Resolución de AA y notifica a Recursos Humanos los cambios. Luego el VDF notifica a los estudiantes seleccionados su ingreso a las filas del MAA.

Diagrama IDEF del Proceso de Solicitar Ayudantía

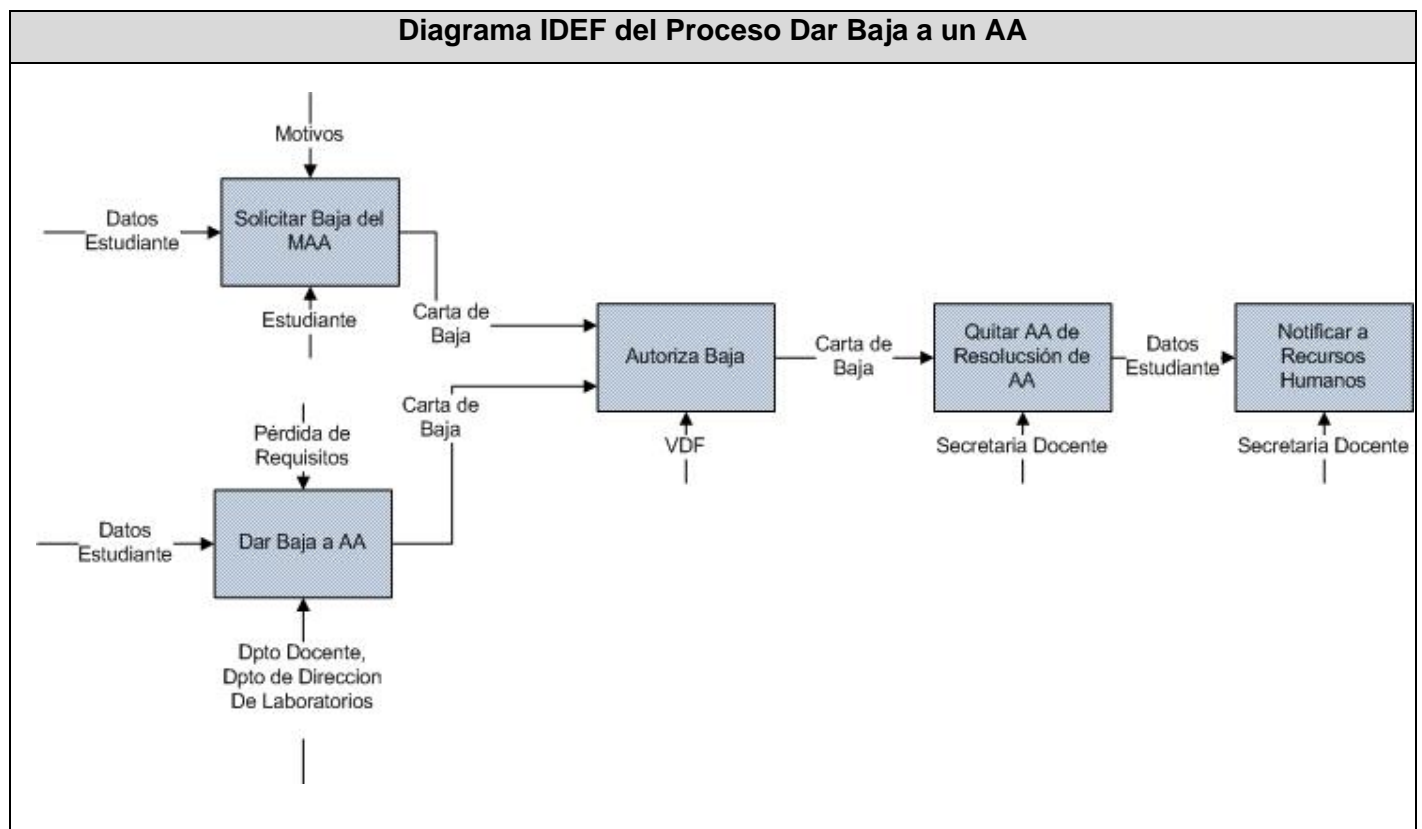
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA



CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

2.3.2 Proceso Dar Baja a un AA

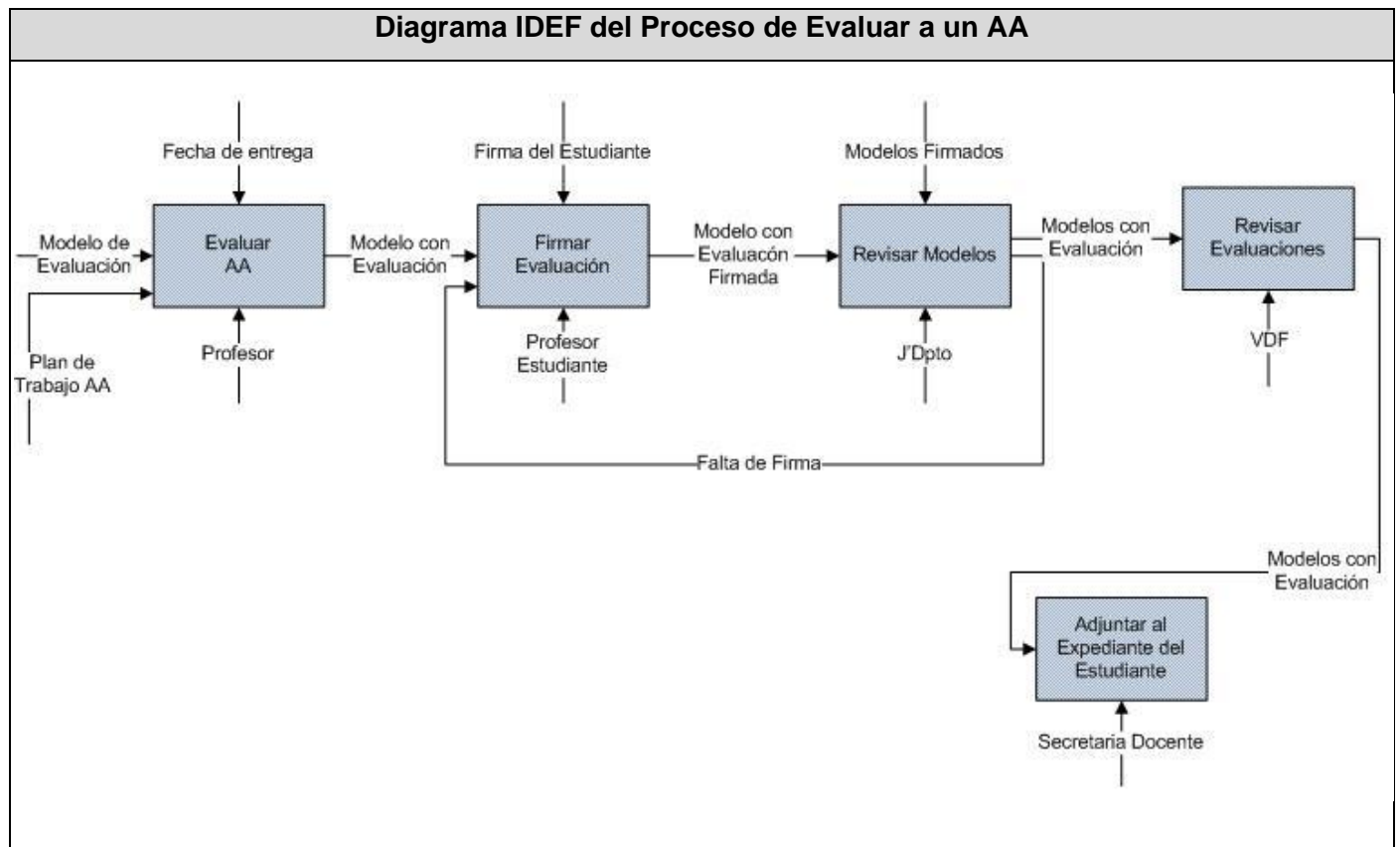
El Dpto. presenta al VDF la Carta de Baja para el AA por pérdida de requisitos; o el estudiante hace entrega al VDF de una Carta de Baja expresando los motivos de la misma. El VDF autoriza a la Secretaría Docente a proceder con la baja. Seguidamente la Secretaria Docente borra de la Resolución de AA al estudiante, dirigiéndose después a Recursos Humanos a notificar dicha acción.



2.3.3 Proceso Evaluar a un AA

El profesor evalúa al AA llenando el Modelo de evaluación de AA. Una vez que este modelo está lleno, se convoca al estudiante para que lo firme. Se entrega dicho documento al Jefe de Dpto. el cual verifica las evaluaciones y que estén firmadas. Luego envía los modelos al VDF, quien revisa las evaluaciones y luego da la actividad a la Secretaria Docente de adjuntar esta evaluación al expediente del estudiante.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA



2.4 Información que se maneja

A continuación se exponen los documentos e información que se manipulan en los procesos de gestión de los AA de nuestra Facultad, así como el sistema que se consulta para verificar datos del estudiante.

Información sobre necesidad de AA por Dpto.

- ✓ Cantidad necesaria de AA por Asignatura.

Planilla de Solicitud

- ✓ Nombre
- ✓ Apellidos
- ✓ Grupo

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

- ✓ Asignatura solicitada (no para el caso de AA de laboratorio).

Sistema de Gestión de Información Akademos

- ✓ Notas del estudiante.

Expediente

- ✓ Sanción.
- ✓ Comisión disciplinaria.
- ✓ Ha sido AA
- ✓ Tuvo Mundial.

Resolución de AA Docente

- ✓ Nombre
- ✓ Apellidos

Resolución de AA de Laboratorio

- ✓ Nombre
- ✓ Apellidos

Modelo de Evaluación de AA Docente

- ✓ Nombre
- ✓ Apellidos
- ✓ Año académico
- ✓ Evaluación del trabajo docente
- ✓ Evaluación del trabajo científico-productivo
- ✓ Evaluación integral.
- ✓ Firma profesor
- ✓ Firma estudiante

Carta de Baja

- ✓ Nombre
- ✓ Apellidos

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

- ✓ Motivos

2.5 Propuesta de sistema

2.5.1 Requerimientos Funcionales

RF-1 Autenticar Usuario

RF-2 Solicitar Ayudantía

- ✓ RF-2.1 Verificar Requisitos para ser AA

RF-3 Aprobar AA

RF-4 Gestionar Usuarios

- ✓ RF-4.1 Adicionar Usuario
- ✓ RF-4.2 Modificar Usuario
- ✓ RF-4.3 Eliminar Usuario
- ✓ RF-4.4 Buscar Usuario

RF-5 Evaluar AA

RF-6 Gestionar Solicitud

- ✓ RF-5.1 Adicionar Solicitud
- ✓ RF-5.2 Modificar Solicitud
- ✓ RF-5.3 Eliminar Solicitud

RF-7 Administrar Curso

- ✓ RF-6.1 Adicionar Curso
- ✓ RF-6.2 Eliminar Curso

RF-8 Administrar Semestre

- ✓ RF-7.1 Adicionar Semestre
- ✓ RF-7.2 Eliminar Semestre

RF-9 Administrar Asignatura

- ✓ 8.1 Adicionar Asignatura
- ✓ 8.2 Eliminar Asignatura

RF-10 Notificar

RF-11 Publicar

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

RF-12 Generar Reportes

2.5.1 Requerimientos No Funcionales

Requerimientos de Software

Los usuarios sólo necesitarán un navegador web Mozilla Firefox en versiones superiores al 3.0, Internet Explorer en versiones superiores al 6.0 u otros, bajo cualquier sistema operativo Windows 2000 en adelante o cualquier distribución de Linux. En el servidor se requiere Windows 2000 en adelante o cualquier distribución de Linux, Servidor Apache 2.x con módulo PHP disponible y como gestor de base de datos PostgreSQL 7.1.x.

Requerimientos de Hardware

El cliente requiere de una computadora con 256 Mb de RAM como mínimo, El servidor Web conjuntamente con el servidor de base de datos debe tener 512 Mb de RAM o superior para un mejor funcionamiento y 80 Gb de disco duro como mínimo, todas las computadoras implicadas en el funcionamiento de la aplicación deben estar conectadas a la red.

Requerimientos de Apariencia o interfaz externa

La aplicación propuesta tendrá una interfaz de usuario sencilla y amigable, de forma que no se les dificulte la navegación a los usuarios por la misma.

Requerimientos de Seguridad

Confidencialidad: La información que se maneja en el sistema sólo puede ser accedida por los usuarios autorizados.

Integridad: La información que se maneja en el sistema se mantiene de forma íntegra en la base de datos del mismo. La comunicación con la base de datos automatizada, validando todas las consultas y filtrando los datos variables, usando comillas simples para delimitar las variables que se usan en las consultas a la base de datos; en el framework CodeIgniter tanto las librerías de base de datos, como ActiveRecord contemplan la sanitización de parámetros para disminuir el riesgo de inyección SQL.

Disponibilidad: Se utiliza el servicio de autenticación para dar acceso a los usuarios que interactúan con el sistema. Sólo los usuarios autenticados tendrán acceso a la información a la que tienen permisos. Se podrá tener acceso a la aplicación las 24 horas del día.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

Requerimientos de Usabilidad

La aplicación propuesta será usada por personas con un conocimiento medio de informática, al administrador de la aplicación se le dará un adiestramiento básico en el uso del sistema. Estas personas tendrán un nivel de acceso amplio en la aplicación, y será utilizado sin un gran esfuerzo por las personas para las que fue diseñado.

Requerimientos de Soporte

- ✓ Multiplataforma.
- ✓ Fácil instalación.
- ✓ Mantenimiento cuando sea necesario.

2.6 Conclusiones.

En este capítulo se planteó la propuesta de solución al problema, se detalló la arquitectura del sistema propuesto y se describieron los procesos del negocio del sistema a desarrollar. Se considera ya están las bases creadas para empezar a construir el sistema que constituye la propuesta de solución.

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

3.1 Introducción.

En el presente capítulo se aborda el trabajo realizado en las fases de exploración y planificación de la metodología de desarrollo XP utilizada en el desarrollo del sistema propuesto. Además de exponer los artefactos generados en el transcurso de dichas fases.

3.2 Fase de Exploración.

Esta es la primera fase de desarrollo de la metodología XP. En la misma se realiza la identificación de las Historias de Usuario y el equipo de desarrollo se familiariza con las herramientas y tecnologías seleccionadas para la construcción del proyecto para lograr la primera entrega del producto.

3.2.1 Historias de Usuario (HU).

Las HU son representaciones escritas de requerimientos de software, utilizando el lenguaje común del usuario. Estas describen lo que el sistema debe realizar y son escritas desde la perspectiva del cliente aunque los desarrolladores ayudan en la identificación de las mismas. Cada HU debe ser lo suficientemente comprensible con el contenido concreto, sencillo, dinámico y flexible, para que los programadores puedan implementarla en pocas semanas.

Se identificaron las HU que se detallan a continuación:

Tabla Usuarios y su Rol en el sistema.

Usuarios que interactúan con el sistema	Rol del usuario en el sistema
Administrador del sistema	Representa el Vice-Decano de Formación de la facultad, tiene total dominio del sistema y de los usuarios del mismo.
Profesor	Representa un profesor de la facultad, puede evaluar a su AA y publicarle el Plan de Trabajo.

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

Jefe de Dpto.	Los profesores que son jefes de Dpto., además de las acciones que puede realizar el usuario profesor, aprobarán AA y adicionarán a la aplicación los profesores de su Dpto.
Estudiante	Representa un estudiante de la facultad, la acción que puede realizar en el sistema es solicitar Ayudantía.
AA	Representa un AA de la facultad, que podrá revisar sus evaluaciones y consultar su Plan de Trabajo.

Tabla HU Autenticar Usuario

Historia de Usuario	
No.: 1	Nombre: Autenticar Usuario
Usuario: Todos	
Prioridad en el Negocio: Alta	Riesgo de Desarrollo: Medio
Iteración Asignada: 1	
<p>Descripción: Se realiza la acción de autenticar al usuario que se registre en el sistema. Se verificará que los datos entrados sean correctos otorgándole los permisos que le correspondan en dependencia del tipo de usuario que sea (Estudiante, Estudiante AA, Profesor, Administrador), en caso contrario el sistema lanzará un error.</p>	
<p>Observación: Para la autenticación son necesarios los datos: usuario y contraseña.</p>	

Tabla HU Solicitar Ayudantía

Historia de Usuario	
No.: 2	Nombre: Solicitar Ayudantía
Usuario: Estudiante	

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

Prioridad en el Negocio: Alta	Riesgo de Desarrollo: Medio
Iteración Asignada: 1	
Descripción: Se realiza la acción de solicitar ayudantía. Se brinda la posibilidad al usuario estudiante de realizar la solicitud de ayudantía de la asignatura que le interesa y de saber si cumple los requisitos esenciales para ser AA, esto dentro del plazo que comprende la convocatoria de ayudantía.	
Observación: Solo estará activo dentro del plazo de convocatoria.	

Tabla HU Aprobar AA

Historia de Usuario	
No.: 3	Nombre: Aprobar AA
Usuario: Jefe de Dpto.	
Prioridad en el Negocio: Alta	Riesgo de Desarrollo: Medio
Iteración Asignada: 1	
Descripción: Se realiza la acción de aprobar los estudiantes que solicitaron ayudantía. El usuario Profesor (Jefe de Dpto.) tiene la posibilidad de seleccionar y aprobar los estudiantes que necesitan en su Dpto. para AA.	
Observación: No aplica	

Tabla HU Gestionar Usuario

Historia de Usuario	
No.: 4	Nombre: Gestionar Usuario
Usuario: Administrador del Sistema	
Prioridad en el Negocio: Alta	Riesgo de Desarrollo: Medio
Iteración Asignada: 1	

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

<p>Descripción: Se brinda la posibilidad al administrador del sistema (VDF) de crear, modificar y eliminar cuentas de usuarios del sistema, así como realizar búsquedas de los usuarios registrados.</p>
<p>Observación: No aplica</p>

Tabla HU Evaluar AA

Historia de Usuario	
No.: 5	Nombre: Evaluar AA
Usuario: Profesor y Jefe Dpto.	
Prioridad en el Negocio: Media	Riesgo de Desarrollo: Medio
Iteración Asignada: 1	
<p>Descripción: Se brinda la posibilidad de Evaluar los AA. El usuario con el rol Profesor tiene la posibilidad de evaluar el desempeño del AA.</p>	
<p>Observación: El profesor debe guiarse por el Modelo de Evaluación de los AA para dar la evaluación.</p>	

Tabla HU Gestionar Solicitud

Historia de Usuario	
No.: 6	Nombre: Gestionar Solicitud
Usuario: Administrador del Sistema	
Prioridad en el Negocio: Alta	Riesgo de Desarrollo: Medio
Iteración Asignada: 1	
<p>Descripción: Se brinda la posibilidad al administrador del sistema (VDF) de adicionar, modificar y eliminar solicitudes del sistema.</p>	
<p>Observación: No aplica</p>	

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

Tabla HU Administrar Curso

Historia de Usuario	
No.: 7	Nombre: Administrar Curso
Usuario: Administrador del Sistema	
Prioridad en el Negocio: Media	Riesgo de Desarrollo: Medio
Iteración Asignada: 2	
Descripción: Se brinda la posibilidad al administrador del sistema (VDF) de adicionar y eliminar cursos del sistema.	
Observación: No aplica	

Tabla HU Administrar Semestre

Historia de Usuario	
No.: 8	Nombre: Administrar Semestre
Usuario: Administrador del Sistema	
Prioridad en el Negocio: Media	Riesgo de Desarrollo: Medio
Iteración Asignada: 2	
Descripción: Se brinda la posibilidad al administrador del sistema (VDF) de adicionar y eliminar semestres del sistema.	
Observación: No aplica	

Tabla HU Administrar Asignatura

Historia de Usuario	
No.: 9	Nombre: Administrar Asignatura
Usuario: Administrador del Sistema	
Prioridad en el Negocio: Media	Riesgo de Desarrollo: Medio

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

Iteración Asignada: 2
Descripción: Se brinda la posibilidad al administrador del sistema (VDF) de adicionar y eliminar asignaturas a los Dptos. docentes.
Observación: No aplica

Tabla HU Notificar

Historia de Usuario	
No.: 10	Nombre: Notificar
Usuario: Sistema	
Prioridad en el Negocio: Media	Riesgo de Desarrollo: Medio
Iteración Asignada: 2	
Descripción: Se realiza la acción notificar automáticamente, mediante el correo electrónico a los estudiantes, los cambios importantes que ocurran desde que los mismos realizan la solicitud de ayudantía, así como a los profesores cuando se les asigna un AA.	
Observación: No aplica	

Tabla HU Publicar

Historia de Usuario	
No.: 11	Nombre: Publicar
Usuario: Profesor	
Prioridad en el Negocio: Media	Riesgo de Desarrollo: Medio
Iteración Asignada: 2	
Descripción: Se realizará la acción de publicar la convocatoria de ayudantía (lanzar la convocatoria) para los estudiantes, así como la publicación del plan de trabajo de los AA y las evaluaciones de los	

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

mismos.
Observación: No aplica

Tabla HU Generar Reportes

Historia de Usuario	
No.: 12	Nombre: Generar Reportes
Usuario: Administrador del sistema	
Prioridad en el Negocio: Media	Riesgo de Desarrollo: Medio
Iteración Asignada: 2	
Descripción: Se brinda la posibilidad obtener el listado de los AA existentes. El Administrador del sistema podrá obtener listados de los AA existentes en el sistema.	
Observación: Los reportes podrán ser generados cuando sea necesario.	

3.3 Fase de Planificación

En esta fase el cliente establece la prioridad de cada HU y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. En XP las métricas son libres, pudiendo utilizarse cualquier criterio para medir el desarrollo del proyecto. Una métrica muy común es la que utiliza como medida el punto. Las historias generalmente valen de 1 a 3 puntos. Un punto equivale a una semana ideal de programación, donde los miembros del equipo de desarrollo trabajan con las condiciones ideales, es decir, cuando se trabaja sin distracciones y con una dedicación total a la codificación en el tiempo completo planeado.

3.3.1 Estimación de esfuerzo por Historia de Usuario

Para el desarrollo del sistema propuesto en este trabajo se realizó una estimación del esfuerzo necesario para cada una de las HU identificadas. Se muestran los resultados en la siguiente tabla:

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

Estimación de Esfuerzo por HU

Historia de Usuario	Puntos Estimados
Autenticar Usuario	1.5
Solicitar Ayudantía	1.5
Aprobar AA	1
Gestionar Usuario	1.5
Evaluar AA	1
Gestionar Solicitud	1
Administrar Curso	1
Administrar Semestre	0.5
Administrador Asignatura por Dpto.	0.5
Notificar	0.5
Publicar	1
Generar Reportes	1

3.3.2 Plan de iteraciones

Una vez identificadas las HU y estimado el esfuerzo dedicado a la realización de cada una de estas, se procede a la planificación de la etapa de implementación de la aplicación. Para la implementación del sistema se decidió realizar dos iteraciones las cuales se describen a continuación.

Iteración 1

En esta iteración se implementarán las HU con más prioridad respecto a las demás. Al concluir esta iteración se podrá contar con las funcionalidades descritas en las primeras seis HU, que hacen referencia a la autenticación, gestión de los usuarios que van a interactuar con el sistema, solicitud de ayudantía, gestión y aprobación de la misma y la evaluación de los AA. Al concluir esta iteración se contará con la primera versión (0.1) del producto, la que tiene como objetivo mostrar al cliente como va quedando la aplicación y verificar que el producto que se está obteniendo es lo que realmente el cliente desea.

Iteración 2

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

Esta es la última iteración en la cual quedarán implementadas las restantes HU descritas anteriormente, es decir, la administración de cursos, semestres y asignaturas; las publicaciones, que incluyen publicación de ayudantía (lanzar la convocatoria), de evaluaciones así como del Plan de Trabajo de AA y por último generar reportes de las plantillas de AA existentes. Al terminar esta iteración quedará terminada la aplicación y se contará con una nueva versión (1.0) del producto

3.3.3 Plan de duración de iteraciones

Usando XP se crea el plan de duración de las iteraciones que forma parte del ciclo de vida del proyecto. En este plan se muestran las HU a desarrollar en cada iteración y el tiempo de duración estimada de cada una, así como el orden en que se implementarán.

Plan de duración de Iteraciones

Iteración	Orden de las Historias de Usuario	Duración de las iteraciones
Iteración 1	Autenticar Usuario Solicitar Ayudantía Aprobar AA Gestionar Usuario Evaluar AA Gestionar Solicitud	7.5 semanas
Iteración 2	Administrar Curso Administrar Semestre Administrar Asignatura Notificar Publicar Generar Reportes	4.5 semanas

CAPÍTULO 3 EXPLORACIÓN Y PLANIFICACIÓN

3.3.4 Plan de Entregas

A continuación se presenta el Plan de Entregas ideado para la fase de Implementación. Al finalizar cada iteración se realizarán pruebas al producto obtenido en la fecha aproximada que aparece en la siguiente tabla:

Plan de entregas

Sistema	Final 1ra Iteración 2da semana de Abril	Final 2da Iteración 3ra semana de Mayo
Sistema de Gestión del MAA	0.1	1.0

3.4 Conclusiones

En el presente capítulo se abordó todo lo referente a las fases de exploración y planificación de la entrega de iteraciones del proyecto, donde fueron descritos todos los artefactos generados en las mismas durante su desarrollo. Fueron identificadas ocho HU, haciendo una descripción detallada de las mismas. También se creó el plan de duración de las iteraciones para hacer entrega del producto al final de las mismas en el tiempo requerido y el plan de entregas del sistema basado en la duración de las iteraciones.

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, donde al final de cada iteración se obtendrá un producto funcional que será probado y mostrado al cliente, para incrementar la visión de los desarrolladores e incrementar la retroalimentación cliente-desarrollador. En el presente capítulo se detallan las dos iteraciones llevadas a cabo durante la etapa de construcción del sistema, así como las pruebas de aceptación efectuadas sobre el mismo. Es válido aclarar que para el diseño y construcción de las aplicaciones, la metodología XP no requiere la representación del sistema mediante notación UML, aunque puede implementarse su uso siempre que sea necesario, para mantener una buena comunicación entre el cliente y los desarrolladores. Con este fin se crearon los diagramas de clases, despliegue y componentes, así como la representación del modelo de datos, los mismos pueden observarse en los anexos.

4.2 Fase de Implementación

En la fase de Implementación en XP se lleva a cabo el desarrollo de las HU que fueron definidas en la fase de Planificación, siempre teniendo en cuenta realizar una revisión del plan de iteraciones, modificando el mismo en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de desarrollo. Dichas tareas, son usadas únicamente por los programadores, por lo que pueden ser descritas con un lenguaje técnico que no va a ser necesariamente entendible por el cliente. Cada vez que se realice una prueba a una HU implementada, el cliente debe estar presente, para verificar que la funcionalidad cumple su objetivo.

De acuerdo con la planificación realizada en el capítulo anterior, se llevaron a cabo dos iteraciones de desarrollo sobre el sistema, obteniéndose un producto con las características deseadas que satisfizo las necesidades del cliente.

Iteración 1

En esta iteración se implementarán las HU de mayor importancia para el cliente, por lo que tienen mayor prioridad respecto a las demás.

Historias de Usuario desarrolladas en la iteración 1:

Historia de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Autenticar Usuario	1.5	1.5
Solicitar Ayudantía	1.5	1.5
Aprobar AA	1	1
Gestionar Usuario	1.5	1.5
Evaluar AA	1	0.5
Gestionar Solicitud	1	1

Tareas de Historias de Usuario desarrolladas en la iteración 1

Tarea Configuración de la interfaz de usuario autenticarse

Tarea: 1	
Número de tarea:	Número de historia (Nro. 1): Autenticar Usuario
Nombre de tarea: Configuración de la interfaz de usuario autenticarse	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 22/2/2010	Fecha Fin: 26/2/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: El usuario teclea la dirección URL del sistema en su navegador y le es mostrada inmediatamente la interfaz de autenticación.	

Tarea Captura y Comprobación de Datos de autenticación

Tarea: 2	
Número de tarea:	Número de historia (Nro. 1): Autenticar Usuario
Nombre de tarea: Captura y Comprobación de Datos de autenticación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 24/2/2010	Fecha Fin: 27/3/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se capturan los datos de autenticación del usuario y se comprueba si estos son válidos o no	

Tarea Acceso a la aplicación con los permisos correspondientes

Tarea: 3	
Número de tarea:	Número de historia (Nro. 1): Autenticar Usuario
Nombre de tarea: Acceso a la aplicación con los permisos correspondientes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.7
Fecha Inicio: 27/3/2010	Fecha Fin: 3/3/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Se le da acceso a la aplicación al usuario previamente comprobado, con los permisos asignados de acuerdo con el rol que juega en el sistema.	

Tarea Configuración de la interfaz de usuario Solicitar Ayudantía

Tarea: 1	
Número de tarea:	Número de historia (Nro. 2): Solicitar Ayudantía
Nombre de tarea: Configuración de la interfaz de usuario Solicitar Ayudantía	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 3/3/2010	Fecha Fin: 6/3/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Se configura la interfaz de usuario Solicitar Ayudantía para que los estudiantes puedan realizar esta acción.	

Tarea Captura y Comprobación de datos de solicitud

Tarea: 2	
Número de tarea:	Número de historia (Nro. 2): Solicitar Ayudantía
Nombre de tarea: Captura y Comprobación de datos de solicitud	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 5/3/2010	Fecha Fin: 8/3/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Se capturarán los datos de la asignatura a solicitar por el estudiante, luego se comprueba que el estudiante que realizó la solicitud cumpla con las características que debe tener un estudiante para poder ser AA de la asignatura que demanda, de no cumplirlas se le mostrará un mensaje, "Ud. no cumple las	

características necesarias para ser AA", en caso contrario se le mostrará un mensaje, "Ud. será notificado en caso de ser aceptado".

Tarea Insertar solicitud de estudiante en la base de datos

Tarea: 3	
Número de tarea:	Número de historia (Nro. 2): Solicitar Ayudantía
Nombre de tarea: Insertar solicitud de estudiante en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.7
Fecha Inicio: 8/3/2010	Fecha Fin: 13/3/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: En caso de que el estudiante haya cumplido con las características para ser AA, los datos del mismo se insertarán en la base de datos para su posterior aprobación.	

Tarea Configuración de la interfaz de usuario Aprobar AA

Tarea: 1	
Número de tarea:	Número de historia (Nro. 3): Aprobar AA
Nombre de tarea: Configuración de la interfaz de usuario Aprobar AA	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 13/3/2010	Fecha Fin: 15/3/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Se configura la interfaz de usuario Aprobar AA para realizar la acción correspondiente.	

Tarea Captura y Comprobación de datos de Aprobar AA

Tarea: 2	
Número de tarea:	Número de historia (Nro. 3): Aprobar AA
Nombre de tarea: Captura y Comprobación de datos de Aprobar AA	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Fecha Inicio: 15/3/2010	Fecha Fin: 17/3/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Se capturarán los datos los estudiantes seleccionados y luego se	

verifica en la base de datos que el estudiante no haya sido previamente seleccionado por otro Dpto.

Tarea Aprobar AA

Tarea: 3	
Número de tarea:	Número de historia (Nro. 3): Aprobar AA
Nombre de tarea: Aprobar AA	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 18/3/2010	Fecha Fin: 20/3/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se aprueba el estudiante y es insertado en la base de datos.	

Tarea Configuración de las Interfaces de usuario Gestionar Usuario

Tarea: 1	
Número de tarea:	Número de historia (Nro. 4): Gestionar Usuario
Nombre de tarea: Configuración de las Interfaces de Gestionar Usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 22/3/2010	Fecha Fin: 25/3/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se configuran las interfaces de usuario correspondientes para crear, modificar y eliminar cuentas de los usuarios, necesarias para el administrador del sistema.	

Tarea Captura y Comprobación de los datos del Usuario

Tarea: 2	
Número de tarea:	Número de historia (Nro. 4): Gestionar Usuario
Nombre de tarea: Captura de los datos del Usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 24/3/2010	Fecha Fin: 27/3/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se capturan los datos introducidos en caso de crear un nuevo usuario	

para posterior comprobación o se muestran los usuarios para capturar los datos de un usuario específico y realizar la acción deseada posteriormente.

Tarea Gestión de datos del usuario en la base de datos

Tarea: 4	
Número de tarea:	Número de historia (Nro. 4): Gestionar Usuario
Nombre de tarea: Gestión de datos del usuario en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.7
Fecha Inicio: 27/3/2010	Fecha Fin: 31/3/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Para ingresar el nuevo usuario, si este no existe en la base de datos sus datos se almacenan en la misma, en el caso de que exista ya se pueden realizar todas las acciones de gestionar de acuerdo con la opción escogida.	

Tarea Configuración de la interfaz de usuario de usuario Evaluar AA

Tarea: 1	
Número de tarea:	Número de historia (Nro. 5): Evaluar AA
Nombre de tarea: Configuración de la interfaz de usuario Evaluar AA	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 1/4/2010	Fecha Fin: 2/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se configura la interfaz de usuario Evaluar AA para realizar la acción correspondiente.	

Tarea Captura de datos de Evaluar AA

Tarea: 2	
Número de tarea:	Número de historia (Nro. 5): Evaluar AA
Nombre de tarea: Configuración de la interfaz de usuario Evaluar AA	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Fecha Inicio: 2/4/2010	Fecha Fin: 3/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se capturan los datos de evaluación del AA.	

Tarea Actualizar Evaluación del AA

Tarea: 3	
Número de tarea:	Número de historia (Nro. 5): Evaluar AA
Nombre de tarea: Evaluar AA	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 3/4/2010	Fecha Fin: 5/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se actualizan los datos del AA en la base de datos.	

Tarea Configuración de las Interfaces de usuario Gestionar Solicitud

Tarea: 1	
Número de tarea:	Número de historia (Nro. 6): Gestionar Solicitud
Nombre de tarea: Configuración de las Interfaces de Gestionar Solicitud	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 6/4/2010	Fecha Fin: 8/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se configuran las interfaces de usuario correspondientes para crear, modificar y eliminar solicitudes.	

Tarea Captura de los datos de Gestionar Solicitud

Tarea: 2	
Número de tarea:	Número de historia (Nro. 6): Gestionar Solicitud
Nombre de tarea: Captura de los datos de Gestionar Solicitud	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 7/4/2010	Fecha Fin: 9/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se capturan los datos introducidos en caso de adicionar una solicitud para su posterior comprobación o se muestran las solicitudes para capturar los datos de las mismas y realizar la acción deseada posteriormente.	

Tarea Gestión de datos solicitud en la base de datos

Tarea: 3	
Número de tarea:	Número de historia (Nro. 6): Gestionar Solicitud
Nombre de tarea: Gestión de datos solicitud en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 8/4/2010	Fecha Fin: 11/3/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Para insertar una nueva solicitud, si esta no existe en la base de datos sus datos se almacenan en la misma, en el caso de que exista ya se pueden realizar todas las acciones de gestionar de acuerdo con la opción escogida.	

Iteración 2

Una vez terminada esta iteración se tendrá una idea de cómo quedará la aplicación con todas sus funcionalidades.

Historias de Usuario desarrolladas en la Iteración 2:

Historia de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Administrar Curso	1	
Administrar Semestre	0.5	
Administrar Asignatura	0.5	
Notificar	0.5	
Publicar	1	
Generar Reportes	1	

Historias de Usuario desarrolladas en la iteración 2:

Tarea Configuración de las Interfaces de usuario Administrar Curso

Tarea: 1	
Número de tarea:	Número de historia (Nro. 7): Administrar Curso
Nombre de tarea: Configuración de las Interfaces de Administrar Curso	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2

Fecha Inicio: 12/4/2010	Fecha Fin: 12/4/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Se configuran las interfaces de usuario correspondientes para adicionar y eliminar cursos.	

Tarea Captura de los datos de Administrar Curso

Tarea: 2	
Número de tarea:	Número de historia (Nro. 7): Administrar Curso
Nombre de tarea: Captura de los datos de Administrar Curso	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 13/4/2010	Fecha Fin: 13/4/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Se muestran los cursos para capturar los datos del que se quiera administrar.	

Tarea Administrar datos de curso en la base de datos

Tarea: 3	
Número de tarea:	Número de historia (Nro. 7): Administrar Curso
Nombre de tarea: Administrar datos de curso en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 14/4/2010	Fecha Fin: 14/4/2010
Programador Responsable: Rangel Hurtado Díaz – YUSDANI HINOJOSA DOMÍNGUEZ	
Descripción: Para insertar un nuevo curso, si este no existe en la base de datos sus datos se almacenan en la misma, en el caso de que exista se puede eliminar si así se desea.	

Tarea Configuración de las Interfaces de usuario Adicionar Semestre

Tarea: 1	
Número de tarea:	Número de historia (Nro. 8): Administrar Semestre
Nombre de tarea: Configuración de las Interfaces de Administrar Semestre	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 15/4/2010	Fecha Fin: 15/4/2010

Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez
Descripción: Se configuran las interfaces de usuario correspondientes para adicionar y eliminar semestres.

Tarea Captura de los datos de Administrar Semestre

Tarea: 2	
Número de tarea:	Número de historia (Nro. 8): Administrar Semestre
Nombre de tarea: Captura de los datos de Administrar Semestre	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 17/4/2010	Fecha Fin: 18/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se muestran los semestres para capturar los datos del que se quiera administrar.	

Tarea Administrar datos de semestre en la base de datos

Tarea: 3	
Número de tarea:	Número de historia (Nro. 8): Administrar Semestre
Nombre de tarea: Gestión de datos de curso en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 18/4/2010	Fecha Fin: 19/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Para insertar un nuevo semestre, si este no existe en la base de datos sus datos se almacenan en la misma, en el caso de que exista se puede eliminar si así se desea.	

Tarea Configuración de las Interfaces de usuario Administrar Asignatura por Dpto.

Tarea: 1	
Número de tarea:	Número de historia (Nro. 8): Administrar Asignatura.
Nombre de tarea: Configuración de las Interfaces de Administrar Asignatura por	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 19/4/2010	Fecha Fin: 20/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	

Descripción: Se configuran las interfaces de usuario correspondientes para adicionar y eliminar asignaturas a un determinado Dpto.

Tarea Captura de los datos de Administrar Asignatura por Dpto.

Tarea: 2	
Número de tarea:	Número de historia (Nro. 8): Administrar Asignatura.
Nombre de tarea: Captura de los datos de Administrar Asignatura por Dpto.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha Inicio: 20/4/2010	Fecha Fin: 21/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se muestran las asignaturas para capturar los datos de la que se quiera administrar.	

Tarea Gestión de datos de asignatura en la base de datos

Tarea: 3	
Número de tarea:	Número de historia (Nro. 8): Administrar Asignatura.
Nombre de tarea: Gestión de datos de asignatura en la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha Inicio: 22/4/2010	Fecha Fin: 23/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Para insertar una nueva asignatura, si este no existe en la base de datos sus datos se almacenan en la misma, en el caso de que exista se puede eliminar si así se desea.	

Tarea Enviar Notificación

Tarea: 1	
Número de tarea:	Número de historia (Nro. 9): Notificar.
Nombre de tarea: Enviar Notificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 23/4/2010	Fecha Fin: 25/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se enviará una notificación tanto a estudiantes como a profesores	

después de haber realizado actividades referentes a solicitud y aceptación de ayudantía, así como cuando se realicen publicaciones.

Tarea Configuración de la interfaz de usuario Publicar

Tarea: 1	
Número de tarea:	Número de historia (Nro. 10): Publicar
Nombre de tarea: Mostrar Interfaces de usuario Publicar Plan de Trabajo AA	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 28/4/2010	Fecha Fin: 28/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Al seleccionar la opción "Publicar " mostrará la interfaz de usuario de publicación.	

Tarea Captura de datos de la publicación

Tarea: 2	
Número de tarea:	Número de historia (Nro. 10): Publicar Plan de Trabajo AA
Nombre de tarea: Captura de datos de la publicación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Fecha Inicio: 28/4/2010	Fecha Fin: 29/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Se capturan los datos de la publicación deseada.	

Tarea Publicar

Tarea: 3	
Número de tarea:	Número de historia (Nro. 10): Publicar Plan de Trabajo AA
Nombre de tarea: Publicar	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 29/4/2010	Fecha Fin: 30/4/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Presentación del plan de trabajo en la interfaz de usuario del AA y en la interfaz de usuario Profesor, así como la evaluación de los AA.	

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

Tarea Configuración de la interfaz de usuario Generar Reportes

Tarea: 1	
Número de tarea:	Número de historia (Nro. 11): Generar Reportes
Nombre de tarea: Configuración de la interfaz de usuario Generar Reportes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 3/5/2010	Fecha Fin: 5/5/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Al seleccionar la opción "Generar Reporte" se mostrará la interfaz correspondiente.	

Tarea Mostrar Datos del Reporte

Tarea: 1	
Número de tarea:	Número de historia (Nro. 11): Generar Reportes
Nombre de tarea: Captura de Datos de los reportes	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 5/5/2010	Fecha Fin: 9/5/2010
Programador Responsable: Rangel Hurtado Díaz – Yusdani Hinojosa Domínguez	
Descripción: Al seleccionar la opción "Generar Reporte" se mostrará la interfaz correspondiente.	

4.3 Fase de Prueba

Una de las características fundamentales de la metodología XP es el seguimiento del desarrollo guiado por pruebas para comprobar el funcionamiento de las funcionalidades a medida que se vayan implementando, y a su vez mostrar al cliente los resultados obtenidos durante el desarrollo del producto. Gracias a ello existe menos posibilidad de cometer errores a la hora de hacer algún cambio o modificación en el código de la aplicación, disminuyendo así tanto el número de errores no detectados como el tiempo de su detección.

XP divide las pruebas en dos grupos, los cuales son conocidos como pruebas unitarias y pruebas de aceptación.

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

Las pruebas unitarias son realizadas por los programadores y se encargan de revisar el código de forma automática y las pruebas de aceptación encargadas de evaluar si se obtuvo la funcionalidad requerida por el cliente al final de cada iteración.

Las pruebas de aceptación tienen más peso que las unitarias ya que constituyen un indicador de la satisfacción del cliente con la solución. La realización de este tipo de pruebas debe ser lo más rápido posible, para aunque los desarrolladores puedan realizar con mayor rapidez los cambios que se necesiten.

4.3.1 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las HU. En cada una el cliente especifica los aspectos a probar una vez que la HU ha sido implementada. El cliente es el máximo responsable de verificar cada una de las pruebas y de priorizar la corrección de pruebas fallidas. Una HU puede tener tantas pruebas de aceptación como sean necesarias para garantizar un funcionamiento correcto. El objetivo final de éstas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Una historia de usuario no se considera completa hasta que no ha pasado por sus pruebas.

Prueba de aceptación Nro. 1 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: Autenticar usuario
Nombre: Insertar datos de usuario	
Descripción: Probar que se autentique correctamente un usuario	
Condiciones de ejecución: Se insertarán datos válidos	
Entrada / Pasos de ejecución: Se introducen los datos de autenticación del usuario del dominio UCI.	
Resultado esperado: Se accede a la aplicación mostrando la interfaz de usuario correspondiente al rol que desempeña.	
Evaluación de la Prueba: Satisfactoria.	

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

Prueba de aceptación Nro. 2 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: Solicitar ayudantía
Nombre: Hacer solicitud	
Descripción: Probar que la solicitud de ayudantía se hace correctamente.	
Condiciones de ejecución: El usuario debe estar autenticado por el rol de estudiante. El usuario que va a solicitar ayudantía cumple con los requisitos para ser AA. El usuario no ha hecho una solicitud en esta convocatoria. Debe seleccionarse al menos una asignatura.	
Entrada / Pasos de ejecución: Se selecciona la asignatura deseada.	
Resultado esperado: La solicitud es enviada y se muestra el siguiente mensaje: "Su solicitud fue enviada con éxito".	
Evaluación de la Prueba: Satisfactoria.	

Prueba de aceptación Nro. 3 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: Solicitar ayudantía
Nombre: Hacer solicitud	
Descripción: Probar que la solicitud de ayudantía se hace correctamente.	
Condiciones de ejecución: El usuario debe estar autenticado por el rol de estudiante. El usuario que va a solicitar ayudantía no cumple con los requisitos para ser AA. El usuario no ha hecho una solicitud en esta convocatoria.	
Entrada / Pasos de ejecución: Se selecciona la opción Solicitar Ayudantía.	
Resultado esperado: No se visualiza la interfaz para seleccionar asignatura y se muestra el siguiente mensaje: "Ud. no cumple con las condiciones para ser AA".	

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

Evaluación de la Prueba: Satisfactoria.

Prueba de aceptación Nro. 4 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: Aprobar AA
Nombre: Seleccionar estudiantes.	
Descripción: Prueba para seleccionar los estudiantes que serán AA.	
Condiciones de ejecución: El usuario debe estar autenticado por el rol de Profesor. Debe haberse cerrado el plazo de convocatoria de ayudantía. Existen solicitudes de asignaturas que pertenecen al Dpto del usuario. Debe asignársele un grupo al estudiante.	
Entrada / Pasos de ejecución: Se selecciona la asignatura de la cual se aprobarán los AA. Se selecciona el estudiante y se le asigna un grupo.	
Resultado esperado: Se aprobara el estudiante y mostrará el mensaje: "(Poner aquí el mensaje pertinente)".	
Evaluación de la Prueba: Satisfactoria.	

Prueba de aceptación Nro. 5 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: Gestionar usuario
Nombre: Adicionar un usuario	
Descripción: Probar que se insertan los datos del usuario en la base de datos.	
Condiciones de ejecución: La aplicación debe ser ejecutada por el Administrador del Sistema. Se insertarán datos válidos del usuario. Deben ser llenados todos los campos para la inserción. El usuario no debe estar registrado en la base de datos.	

Entrada / Pasos de ejecución: Se inserta un usuario con datos válidos.
Resultado esperado: El usuario es insertado en la base de datos.
Evaluación de la Prueba: Satisfactoria.

Prueba de aceptación Nro. 6 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: Gestionar usuario
Nombre: Editar un usuario del sistema	
Descripción: Probar que se modifiquen los datos del usuario en la base de datos.	
Condiciones de ejecución: La aplicación debe ser ejecutada por el Administrador del Sistema. Existencia del usuario a modificar en la base de datos. Deben ser llenados todos los campos para la modificación.	
Entrada / Pasos de ejecución: Se selecciona un usuario para modificar sus datos.	
Resultado esperado: Los datos del usuario son modificados.	
Evaluación de la Prueba: Satisfactoria.	

Prueba de aceptación Nro. 7 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU4_P3	Historia de Usuario: Gestionar usuario
Nombre: Eliminar un usuario del sistema	
Descripción: Probar que se elimina un usuario de la base de datos.	
Condiciones de ejecución: La aplicación debe ser ejecutada por el Administrador del Sistema. Existencia del usuario a eliminar en la base de datos.	
Entrada / Pasos de ejecución: Se selecciona un usuario a eliminar.	

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

Resultado esperado: El usuario es eliminado de la base de datos.
Evaluación de la Prueba: Satisfactoria.

Prueba de aceptación Nro. 8 (Iteración 1)

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: Evaluar AA
Nombre: Generar documento de evaluación.	
Descripción: Prueba para dar una evaluación a un AA.	
Condiciones de ejecución: La aplicación debe ser ejecutada por un Profesor. El profesor debe tener al menos un AA.	
Entrada / Pasos de ejecución: Se selecciona el AA a evaluar y luego se llena el documento de evaluación.	
Resultado esperado: Se genera el documento de evaluación del estudiante.	
Evaluación de la Prueba: Satisfactoria.	

Prueba de aceptación Nro. 9 (Iteración 2)

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: Publicar
Nombre: Publicar plan de trabajo AA	
Descripción: Probar que los AA tengan acceso al plan de trabajo AA	
Condiciones de ejecución: La aplicación debe ser ejecutada por un AA. El AA debe tener un tutor.	
Entrada / Pasos de ejecución: El AA selecciona la opción Ver el plan de trabajo	
Resultado esperado: El AA puede abrir el documento plan de trabajo AA.	
Evaluación de la Prueba: Satisfactoria.	

Prueba de aceptación Nro. 10 (Iteración 1)

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA

Caso de Prueba de Aceptación	
Código: HU11_P2	Historia de Usuario: Publicar
Nombre: Prueba para comprobar la publicación de la ayudantía.	
Descripción: Prueba para dar una evaluación a un AA.	
Condiciones de ejecución: La aplicación debe ser ejecutada por el Administrador del sistema. El profesor debe tener al menos un AA.	
Entrada / Pasos de ejecución: Se selecciona la opción Publicar Ayudantía.	
Resultado esperado: En la interfaz de los estudiantes aparece la opción para solicitar ayudantía, y en la interfaz del profesor se muestra la opción de aprobar ayudantía.	
Evaluación de la Prueba: Satisfactoria.	

4.4 Conclusiones

En el presente capítulo se detallan las dos iteraciones llevadas desarrolladas durante la etapa de implementación, así como las pruebas realizadas a las funcionalidades del sistema. Para ello se realizó una descripción de cada uno de los artefactos que fueron generados.

CONCLUSIONES

CONCLUSIONES

Luego de concluir el desarrollo de este trabajo se arribó a las siguientes conclusiones:

- ✓ Se logró desarrollar una herramienta capaz de informatizar el proceso de Gestión de AA de la Facultad 2, cumpliéndose satisfactoriamente con todas las especificaciones requeridas por el cliente.
- ✓ El sistema fue desarrollado siguiendo la metodología de desarrollo XP, permitiendo la modelación del sistema en menos tiempo del estimado, de esta forma, se hicieron en tiempo las entregas al cliente.
- ✓ Se obtuvo una herramienta que da la posibilidad a las personas que intervienen en el proceso de Gestión de AA de lograr un proceso de mayor calidad y eficiencia, así como llevar el control de la información de todos los AA de manera actualizada.

RECOMENDACIONES

Durante el desarrollo de este trabajo de diploma se ha cumplido con los objetivos trazados en el mismo, no obstante, se proponen las siguientes recomendaciones para su posterior análisis:

- ✓ Estudiar la posibilidad de realizar la informatización del proceso de Gestión de AA en todas las Facultades.
- ✓ Aplicar el software lo antes posible en la Facultad 2 para desarrollar un proceso de Gestión de AA con mayor calidad y eficiencia.

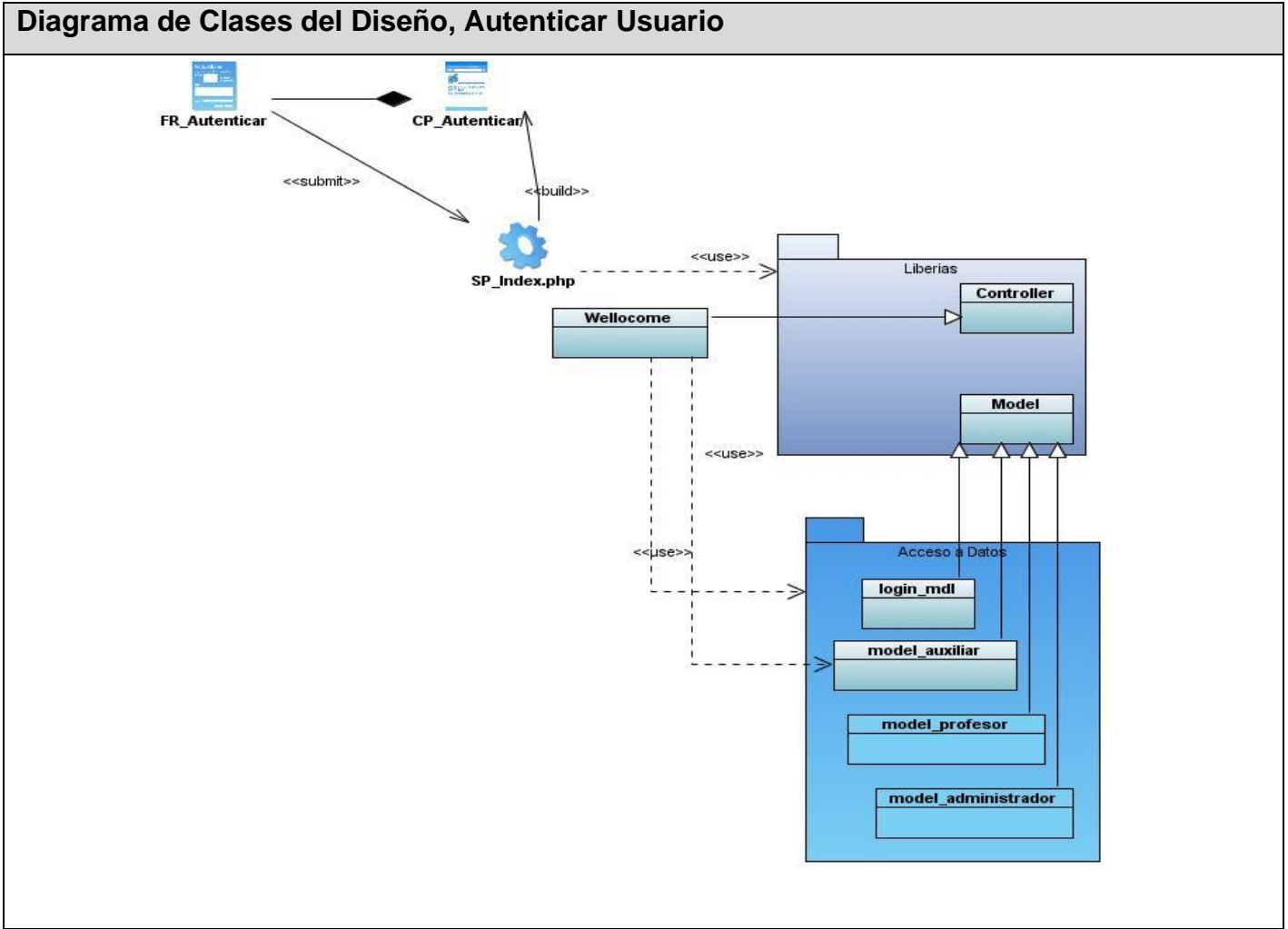
BIBLIOGRAFÍA

1. MASOFT. *MASOFT*. [En línea] [Citado el: 28 de Enero de 2010.] <http://www.sistema-de-gestion.com.ar/>.
2. SIU Guaraní. *SIU Guaraní*. [En línea] [Citado el: 28 de Enero de 2010.] <http://alumnos.unsa.edu.ar/>.
3. El lenguaje de modelado Unificado. *El lenguaje de modelado Unificado*. [En línea] [Citado el: 3 de Febrero de 2010.] <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>.
4. Herramientas CASE. *Herramientas CASE*. [En línea] [Citado el: 4 de Febrero de 2010.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
5. CAVSI. *CAVSI*. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
6. ManualesdeAyuda.com. *ManualesdeAyuda.com*. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/caracteristicas-de-postgresql-01844.html>.
7. **A., Ernesto Quiñones**. EQSOFT. *EQSOFT*. [En línea] [Citado el: 5 de Febrero de 2010.] http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf.
8. Adobe developer connection. *Adobe developer connection*. [En línea] [Citado el: 8 de Febrero de 2010.] http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures.html.
9. Aula Clic. *Aula Clic*. [En línea] [Citado el: 8 de Febrero de 2010.] http://www.aulaclic.es/dreamweaver8/t_1_1.htm.
10. Softonic. *Softonic*. [En línea] [Citado el: 8 de Febrero de 2010.] <http://netbeans-ide.softonic.com/>.
11. Departamento de Informatica (DI). *Departamento de Informatica (DI)*. [En línea] [Citado el: 8 de Febrero de 2010.] <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.
12. Tecnológico. *Tecnológico*. [En línea] [Citado el: 9 de Febrero de 2010.] <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>.
13. JUNTA DE ANDALUCIA. *JUNTA DE ANDALUCIA*. [En línea] [Citado el: 9 de Febrero de 2010.] <http://www.juntadeandalucia.es/averroes/iesgaviota/informatica/html.html>.
14. Desarrolloweb . *Desarrolloweb* . [En línea] [Citado el: 9 de Febrero de 2010.] <http://www.desarrolloweb.com/articulos/26.php> .
15. MasterMagazine . *MasterMagazine* . [En línea] [Citado el: 9 de Febrero de 2010.] <http://www.mastermagazine.info/termino/4294.php> .
16. Lenguaje PHP. *Lenguaje PHP*. [En línea] [Citado el: 9 de Febrero de 2010.] <http://1sinfo.blogspot.com/2007/05/lenguaje-de-programacin-php.html>.
17. **Gutierrez, Javier J**. Lenguajes y sistemas informaticos. *Lenguajes y sistemas informaticos*. [En línea] [Citado el: 9 de Febrero de 2010.] http://www.lsi.us.es/~javier/investigacion_ficheros/Framework.pdf.
18. **Lozano, Viktor**. Code Igniter. *Code Igniter*. [En línea] [Citado el: 10 de Febrero de 2010.] http://codeigniter.com/user_guide/.
19. Pergaminovirtual. *Pergaminovirtual*. [En línea] [Citado el: 8 de Febrero de 2010.] <http://www.pergaminovirtual.com.ar/definicion/Script.html>.

ANEXOS

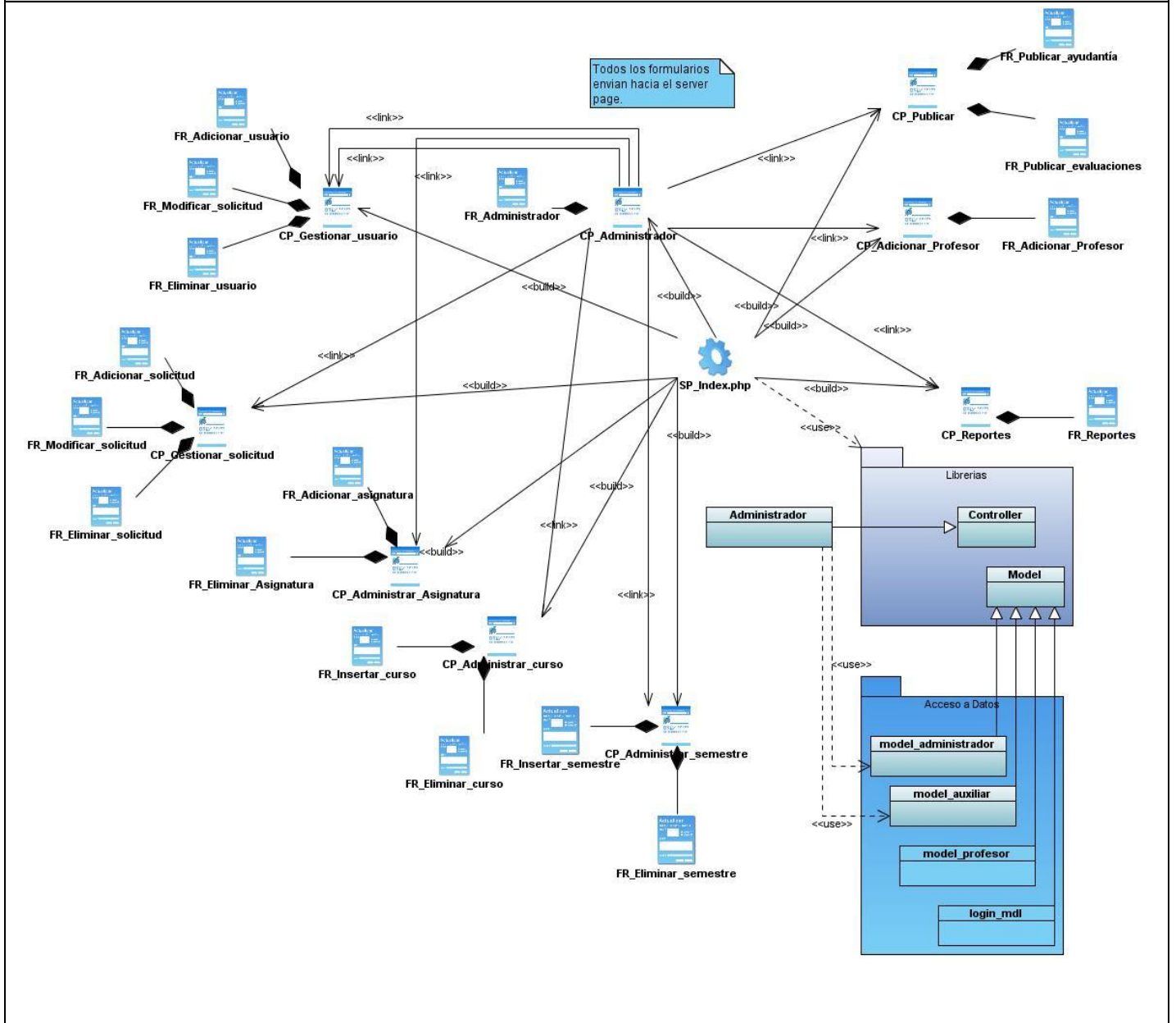
Anexo 1.

Diagrama de Clases del Diseño, Autenticar Usuario



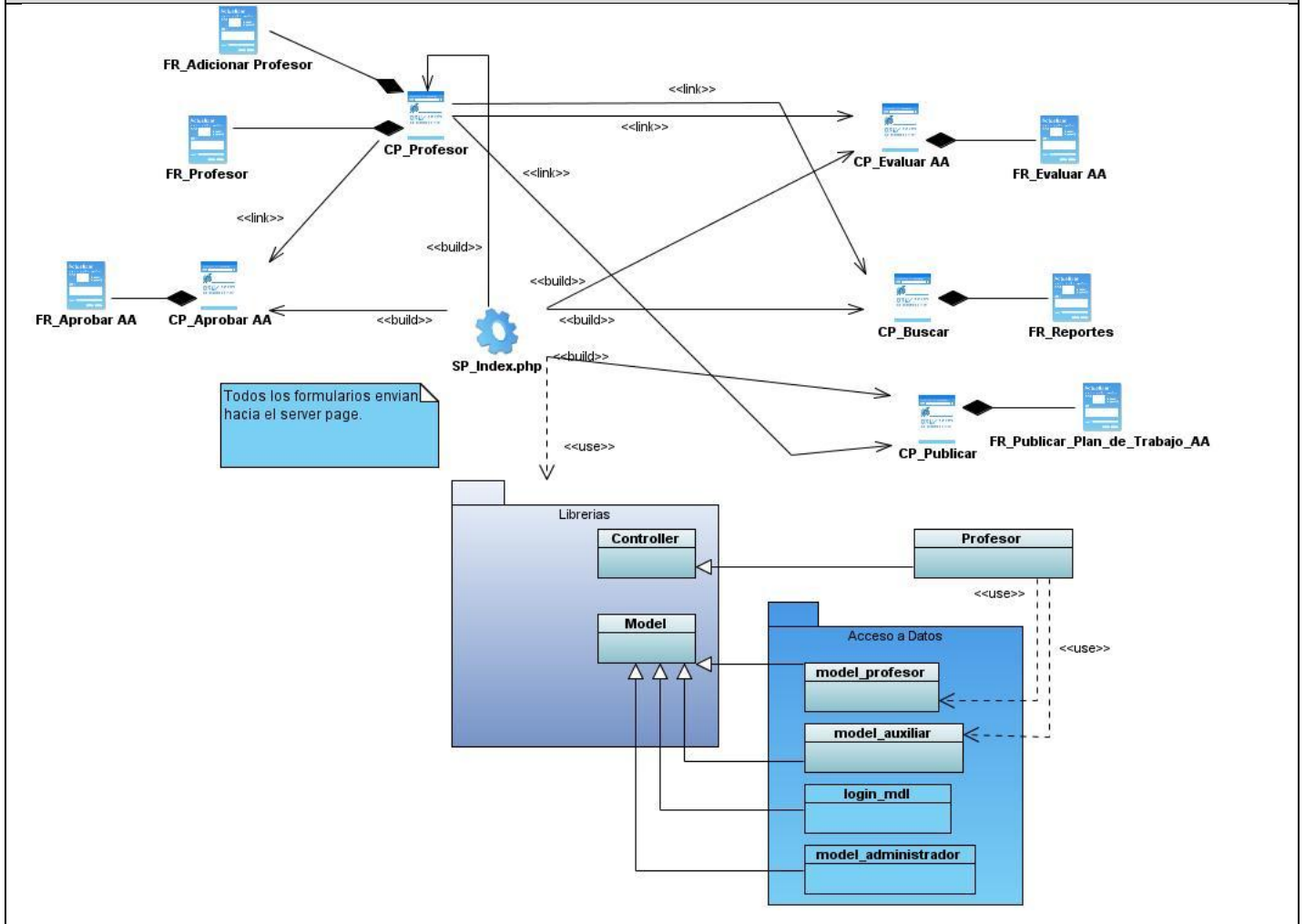
Anexo 2.

Diagrama de Clases del Diseño, Módulo Administrador



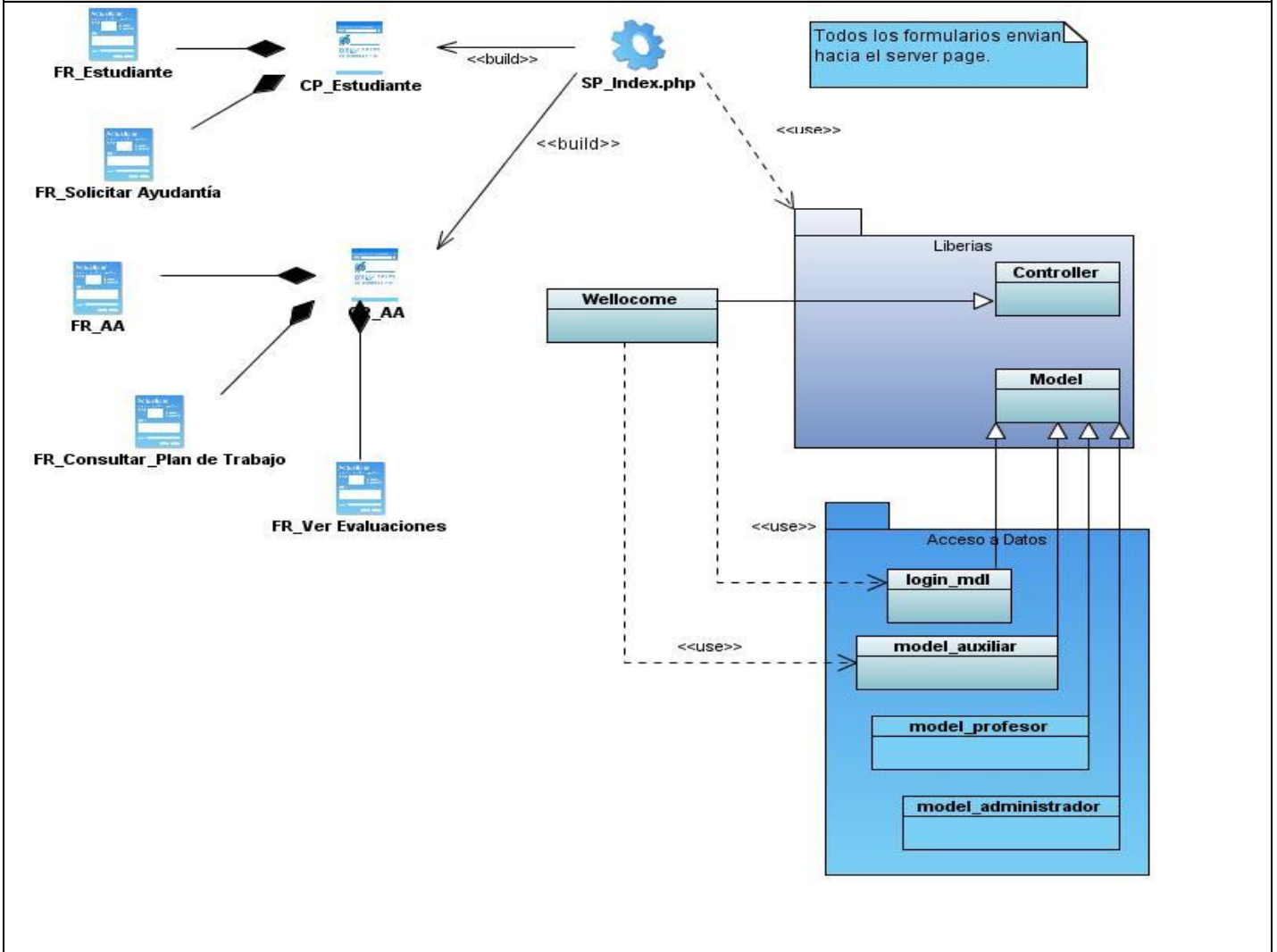
Anexo 3.

Diagrama de Clases del Diseño, Módulo Profesor

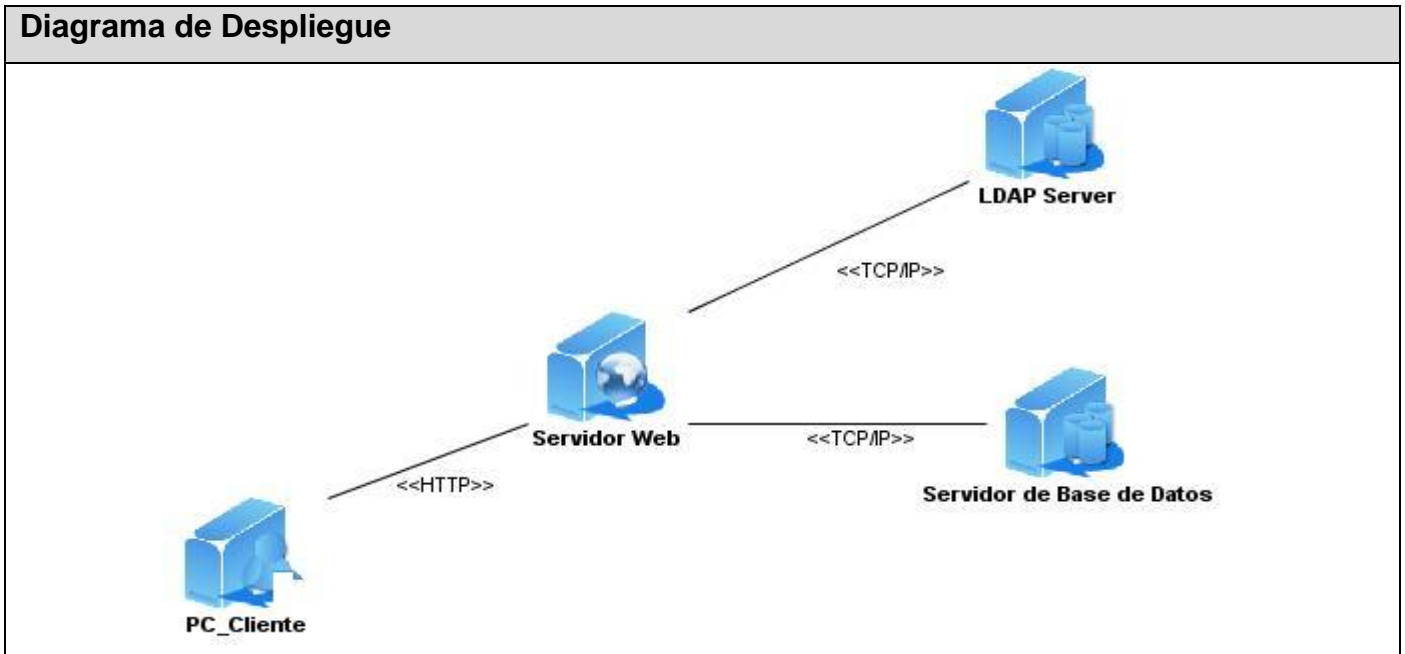


Anexo 4.

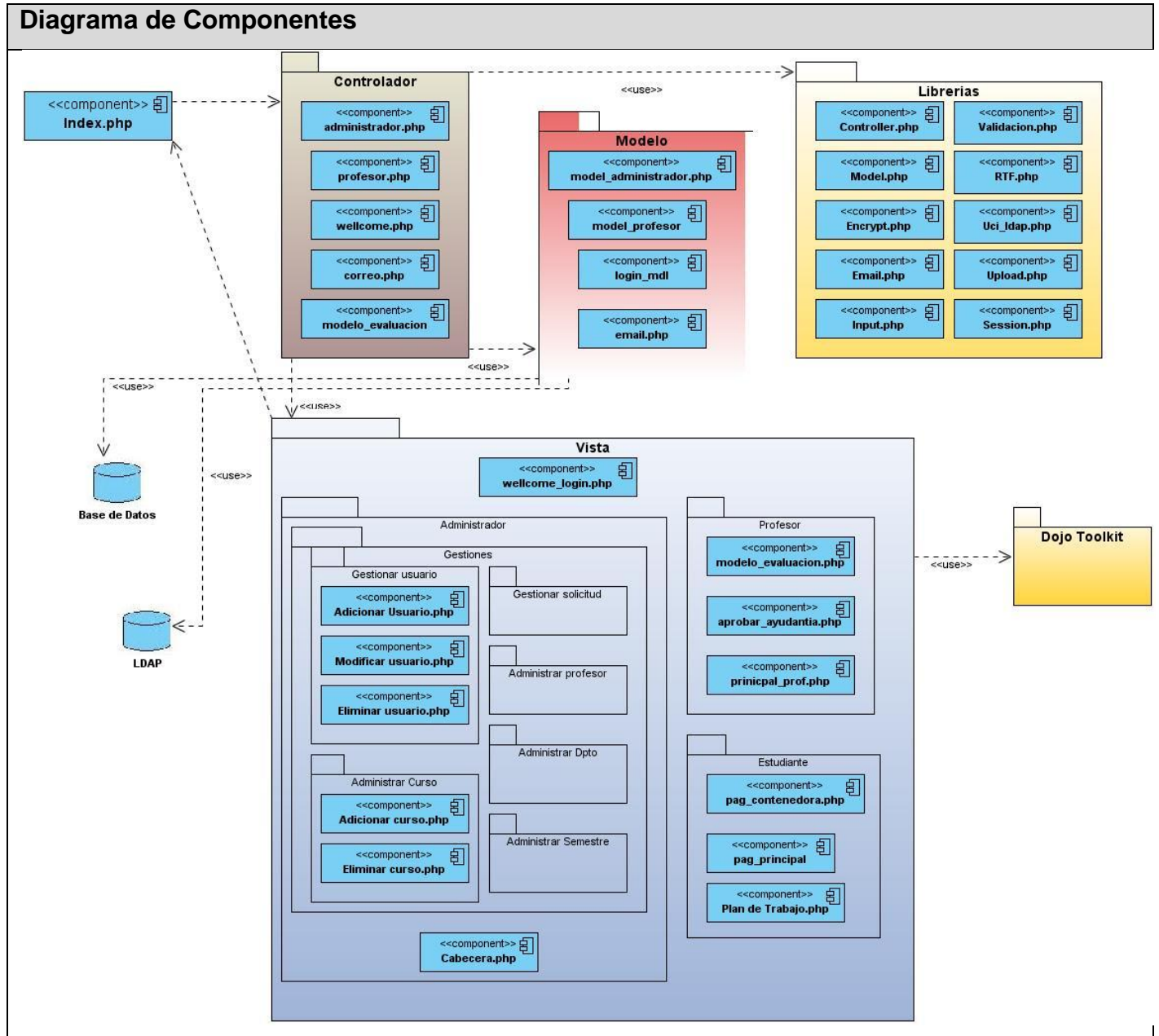
Diagrama de Clases del Diseño, Módulo Estudiante



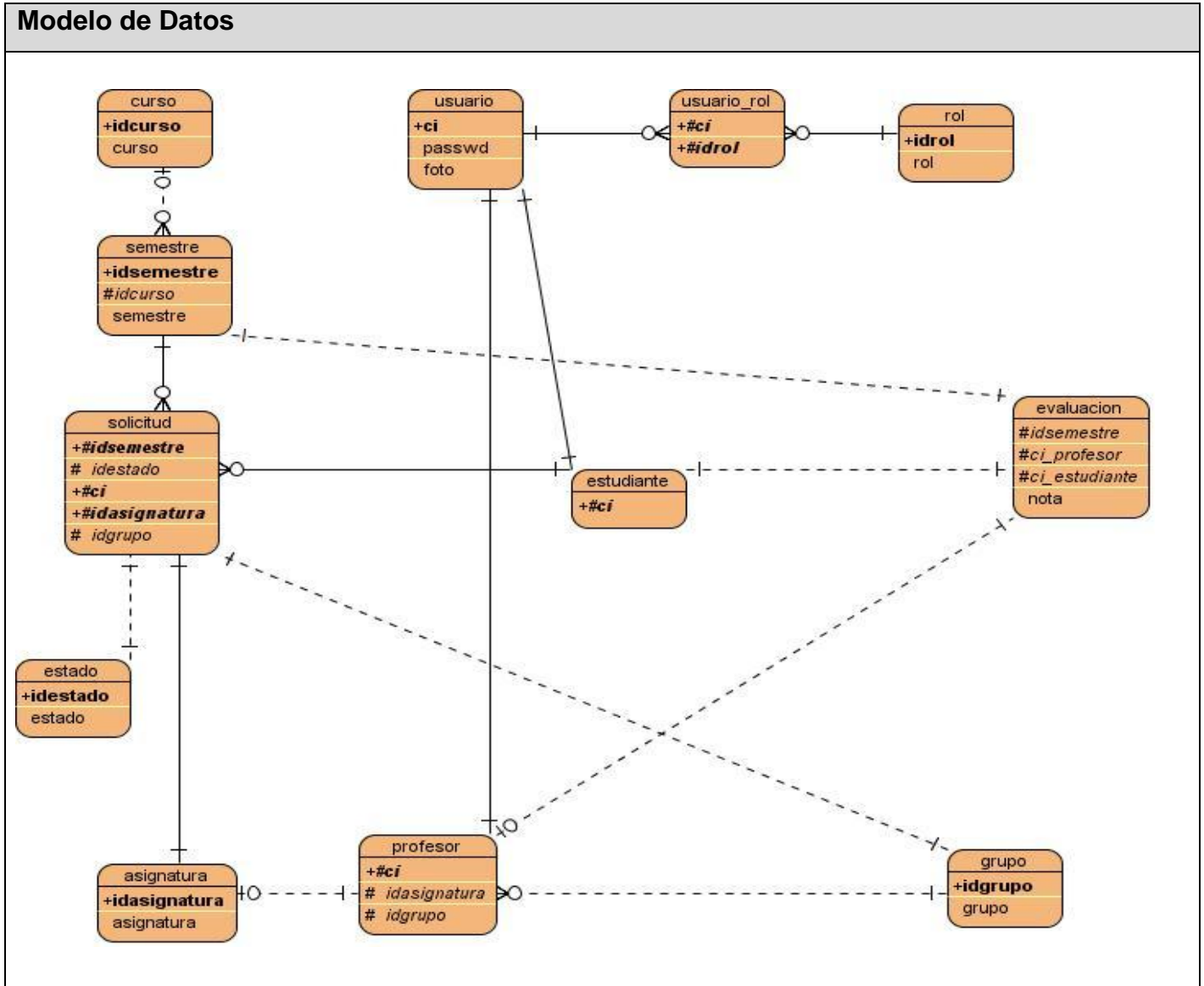
Anexo 5.



Anexo 6.



Anexo 7.



GLOSARIO DE TÉRMINOS

CASE (Computer Aided Systems Engineering): Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

IDE: Es un entorno de desarrollo integrado, programa compuesto por un conjunto de herramientas para un programador desde el que se pueden editar programas, compilarlos y depurarlos.

AJAX (Asynchronous JavaScript And XML): Es una técnica de desarrollo web para crear aplicaciones interactivas.

ASP (Active Server Pages): Es una tecnología del lado del servidor de Microsoft para las páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida.

JSP (Java Server Pages): Es una tecnología orientada a crear páginas web con programación en Java.

API (Application Programming Interface): Es un conjunto de rutinas que provee acceso a funciones de un determinado software.

TCP/IP: Es un conjunto de protocolos de red en los que se basa internet y que permiten la transmisión de datos entre redes de computadoras.

SQL: Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

XML (Extensible Markup Language): Es un metalenguaje extensible de etiquetas.

URL (Uniform Resource Locator): Es una secuencia de caracteres de acuerdo con un formato estándar para nombrar determinados recursos de red, para su localización o identificación.