

Universidad de las Ciencias Informáticas



Título:

Gestión de Pases No Reglamentados en la Facultad 7.

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

Autores:

Lisandra Arazo Aroche.

Yoelkis Hernández Víctor.

Tutores:

Ing. David Barreto Medina.

Ing. Alexander Paneque Meschenkov.

Asesora:

Meybel Rivero Rivadeneira

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 26 días del mes de Junio del año 2009.

Lisandra Arazo Aroche

Yoelkis Hernández Víctor

Ing. Alexander Paneque

Ing. David Barreto Medina

Datos de Contacto

Ing. Alexander Paneque: Trabaja como Profesor Adiestrado de la disciplina Práctica Profesional la cual pertenece al Dpto. Práctica Profesional de la Facultad 7. Donde se ha desempeñado como Jefe de la asignatura Práctica Profesional 1. Ha puesto en práctica sus conocimientos en el proyecto Registro de Áreas de Salud para la Atención Primaria de la Salud, donde se ha obtenido resultados satisfactorios. Participó en el evento Uciencia. Como desarrollador de software tiene experiencias en los lenguajes de programación, PHP y lenguajes script (Java Script). Su dirección de correo es apaneque@uci.cu.

Ing. David Barreto Medina: Profesor graduado de Ing. Industrial en la UHO en el año 2004. Ha impartido asignaturas como Administración de Empresa, Contabilidad, Comercio Electrónico y Metodología de la Investigación, ha presentado trabajos en el Fórum, UCIENCIA y en las jornadas científicas. Se desempeña como asesor de estrategia, jefe de disciplina y líder del proyecto APS. Su dirección de correo es dbarreto@uci.cu.

Exordio





Aunque el panorama puede ser sombrío, el cielo siempre resplandece.

Lisandra

La única fuerza y la única verdad que hay en esta vida es el amor. El patriotismo no es más que amor, la amistad no es más que amor.

Yoelkjs

Agradecimientos



A Dios, por haberme dado la vida y permitirme vivir estos momentos.

A mi madre, por su sacrificio durante toda su vida, por luchar y esforzarse para darme más de lo que ha podido, por su amor, por su comprensión y su apoyo durante estos años. Gracias mamita.

A mis abuelos, por haber consagrado su vida a sus nietos, por haberme criado y cuidado siempre.

A mi BB, por estar siempre a mi lado y cuidarme tanto, por su paciencia, por sus consejos, por la fuerza que me da en los momentos difíciles, por ser siempre mi apoyo, mi confianza, mi ayuda y mi amor incondicional. Te amo.

A mis tías (Yanet, Luisita, Mayda, Lourdes), por ser partes también de éste logro, por cada palabra de aliento, por cada ayuda, por cada uno de sus consejos, por tan sólo dejarme saber que estaban ahí.

A mi hermano, por los momentos en que estuvo presente, por el largo camino que aún nos queda por recorrer.

A mi familia, a mis tíos Margueury, Merci, Pedro, a Rosito, Gelo, Marci, Mari, a todos mil gracias de corazón por la más mínima ayuda.

A mi nueva familia, Tere, Ronald, Osky, Merci, Lucía, María. Gracias por ser tan especiales, que bendición tan linda me ha dado Dios al tenerlos también a ustedes.

A Tere, por recibir de ella tanto cariño y tanta fuerza en momentos bien difíciles, por su apoyo en tan poco tiempo, por su amor, delicadeza y comprensión en todo momento. Por ser tan especial.

A Ronald, por enseñarme a modificar los encabezados de páginas. Eso también es importante.

A mi amiga Diany, por haber estado ahí siempre que la necesite en estos cinco años, por escucharme y haberme ayudado en todo momento, por su fidelidad.

A los profesores Yandy, Yoney, Yurio, por su apoyo, por estar ahí siempre que los necesité.

A Yoelkis, mi compañero de tesis, por su paciencia, por ayudarme a vencer esta batalla.

A todos, mis más sinceros agradecimientos.

Lisandra Arazo Aroche.

Muchas son las personas a las cuales debo parte de este triunfo, de lograr alcanzar mi culminación académica, lo cual es el anhelo de todos los que así lo deseen:

Definitivamente:

A mis hermanos, mis tíos, mis padres, por darme la estabilidad emocional, económica, sentimental, para poder llegar y cumplir mis sueños. Ya que definitivamente no hubiese sido realidad sin ustedes. Gracias por darme la enseñanza de amar y cuidar esa palabra que tanto valoramos llamada FAMILIA.

*A mi madre que siempre será mi inspiración de lucha para alcanzar mis metas.
A mi padre, que siempre será mi faro y guía en todo lo que se avecina, gracias por estar ahí siempre que lo he necesitado, gracias por tu apoyo incondicional, gracias por enseñarme a valorar lo todo hasta ahora.*

A mis tíos por soportarme en todo momento, dándome su apoyo y confianza, sabiendo que ellos son mis 2^{dos} padres y madres.

A Gualberto, Ruben amigos inseparables que siempre estaban en los momentos buenas y malos.

*A chacón por aceptarme completo, cuidarme, motivarme, por su amor
A Karel, Aramis, Luis Jose, Filiberto, Leonelbis, Yordanis, que fueron mi apoyo durante mi estancia en la universidad, gracias por ser mis amigos.*

A mis tutores, a la presidencia ya que varios formaron parte de mi formación.

A los chicos de la FEU ya que junto a ellos me forme, aprendí y reafirme mis valores revolucionarios.

A la Revolución por darme la posibilidad de poder hoy hacerme ingeniero.

A todos en general que de una forma u otra hicieron posible este resultado

Yoelkis Hernández Víctor

Resumen



Resumen

El desarrollo de la informática y las comunicaciones en el mundo está en constante cambio y búsqueda de soluciones óptimas de acuerdo a las necesidades del hombre actual, donde son cada vez más las instituciones u organizaciones que optan por incorporar aplicaciones que gestionen su información.

La Universidad de las Ciencias Informáticas (UCI) no está ajena a estas necesidades. Por ello, el siguiente trabajo se enmarca en la elaboración de una Aplicación Web que permita la gestión de la información del proceso de solicitud y aprobación de pases no reglamentados a los estudiantes de la Facultad 7 en dicha Universidad.

Para el desarrollo de la solución se utilizan una serie de tecnologías orientadas al software libre, así como otras definidas en el Documento de Arquitectura de la Facultad 7. Se utilizó como lenguaje de programación PHP 5.2.5 y como gestor de base de datos PostgreSQL.

Con la utilización de esta aplicación, se logrará agilizar el proceso de solicitud y aprobación de los pases no reglamentados, permitiendo obtener de esta forma un mayor control sobre el flujo de información que se genera a partir de éste proceso.

PALABRAS CLAVE

Aplicación Web, Pases no Reglamentados, Gestor de Base Datos.

Indice



DATOS DE CONTACTO.....	III
RESUMEN.....	XI
INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Sistemas de Reservaciones.....	6
1.2 Marco de Trabajo y Herramientas Utilizadas.....	9
1.2.1 Aplicaciones Web.	10
1.2.2 Lenguajes de Programación para la Web.	12
1.2.2.1 Lenguaje ASP.	12
1.2.2.2 Lenguaje ASP.NET.....	13
1.2.2.3 Lenguaje PHP.	13
1.2.2.4 Lenguaje HTML.	15
1.2.2.5 Lenguaje Java Script.	16
1.2.3 Servicios Web (Web Services).	16
1.2.4 Sistema de Gestión de Base de Datos (SGBD).....	17
1.2.4.1 SGBD MySQL.....	18
1.2.4.2 SGBD PostGreSQL.	19
1.2.5 Patrones de Arquitectura y Diseño.	21
1.2.5.1 Modelo Cliente/Servidor.	21
1.2.5.2 Arquitectura Orientada a Servicios (SOA).....	22
1.2.5.3 Modelo Vista Controlador (MVC).	22
1.2.5.4 Alta Cohesión y Bajo Acoplamiento.	24
1.2.6 Proceso de Desarrollo de Software.	24
1.2.6.1 Metodología RUP.	24
1.2.6.2 Lenguaje de Modelado: UML 2.0.....	26
1.2.7 Herramientas a Utilizar.....	27
1.2.7.1 Herramientas CASE.....	27
1.2.7.2 Herramienta de Diseño.....	30

1.2.7.3	Zend Studio.	33
1.2.8	Frameworks en PHP.	33
CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA.....		40
2.1	Objeto de Estudio.	40
2.1.1	Situación Problemática.	40
2.2	Flujo Actual de los Procesos.	41
2.3	Objeto de Automatización.....	42
2.3.1	Información que se maneja.	42
2.3	Modelo del Negocio.....	42
2.3.1	Actores del Negocio.	43
2.3.2	Trabajadores del Negocio.	43
2.3.3	Diagrama de Casos de Uso del Negocio.	44
2.3.4	Descripción de los Casos de Uso del Negocio.....	44
2.3.4.1	Especificación Textual del Caso de Uso del Negocio: Solicitar _Pase.	45
2.3.4.2	Diagrama de Actividad del Caso de Uso del Negocio: Solicitar _Pase.	47
2.3.4.3	Especificación Textual del Caso de Uso del Negocio: Solicitar _Opinión _Profesor.	48
2.3.4.4	Diagrama de Actividad del Caso de Uso del Negocio: Solicitar _Opinión _Profesor.	49
2.3.5	Diagrama de Clases del Modelo de Objetos.	49
2.4	Especificación de los Requisitos de Software.	50
2.4.1	Requerimientos Funcionales.	50
2.4.2	Requerimientos No Funcionales.	52
2.5	Descripción del Sistema Propuesto.	54
2.6	Modelo de Casos de Usos del Sistema.	55
2.6.1	Definición de los Actores.	55
2.7	Diagrama de Casos de Uso del Sistema.	56
2.8	Especificación de los Casos de Uso del Sistema.	57
2.8.1	Descripción del Caso de Uso del Sistema: Solicitar _Pase.	57
2.8.2	Descripción del Caso de Uso del Sistema: Notificar _Llegada _Pase.	58

2.8.3	Descripción del Caso de Uso del Sistema: Autenticar _Usuario.	59
2.8.4	Descripción del Caso de Uso del Sistema: Dar _Opinión.	61
2.8.5	Descripción del Caso de Uso del Sistema: Imprimir _Pase.....	62
2.8.6	Descripción del Caso de Uso del Sistema: Otorgar _Pase _Urgente.	63
2.8.7	Descripción del Caso de Uso del Sistema: Otorgar _Pase.	65
2.8.8	Descripción del Caso de Uso del Sistema: Ver _Reportes.	66
2.8.9	Descripción del Caso de Uso del Sistema: Configurar _Rol.	68
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA.....		70
3.1	Modelo de Análisis.	70
3.1.1	Diagrama de Clases del Análisis.	70
3.1.1.1	Diagrama de Clases del CU: Solicitar Pase.....	71
3.1.1.3	Diagrama de Clases del CU: Dar Opinión.	71
3.1.1.2	Diagrama de Clases del CU: Notificar Llegada Pase.	72
3.1.1.4	Diagrama de Clases del CU: Imprimir Pase.....	72
3.1.1.5	Diagrama de Clases del CU: Otorgar Pase Urgente.	73
3.1.1.8	Diagrama de Clases del CU: Autenticar Usuario.	73
3.1.1.6	Diagrama de Clases del CU: Otorgar Pase.....	74
3.1.1.7	Diagrama de Clases del CU: Ver Reportes.	75
3.1.1.9	Diagrama de Clases del CU: Configurar Rol.	76
3.2	Modelo de Diseño.....	76
3.2.1	Diagrama de Clases del Diseño Web.	77
3.3.2.1	Diagrama de Clases del CU: Solicitar Pase.....	78
3.3.2.2	Diagrama de Clases del CU: Notificar Llegada Pase.	79
3.3.2.3	Diagrama de Clases del CU: Dar Opinión.	80
3.3.2.4	Diagrama de Clases del CU: Imprimir Pase.....	81
3.3.2.5	Diagrama de Clases del CU: Otorgar Pase Urgente.	82
3.3.2.6	Diagrama de Clases del CU: Otorgar Pase.....	83
3.3.2.7	Diagrama de Clases del CU: Ver Reportes.	84

3.3.2.8	Diagrama de Clases del CU: Autenticar Usuario.	85
3.3.2.9	Diagrama de Clases del CU: Configurar Rol.	86
3.2.2	Diagramas de Secuencia del Diseño.	86
3.3	Diseño de la Base de Datos.	87
3.3.1	Diagrama de Clases Persistentes.	87
3.3.2	Modelo de Datos.	88
3.4	Principios de Diseño Empleados.	88
3.4.1	Estándares en la Interfaz de la Aplicación.	89
3.4.3	Formatos de Reportes.	90
CAPÍTULO IV: IMPLEMENTACIÓN.....		92
4.1	Diagrama de Despliegue.	92
4.2	Diagrama de Componentes.	93
ANEXOS.....		111

Introducción



Introducción

La Universidad de las Ciencias Informáticas (UCI) surge a partir del profundo y novedoso proceso de transformaciones educacionales y sociales en el que ha estado inmerso el país como parte del programa de la Batalla de Ideas. Convirtiendo así, el territorio que ocupaba la antigua base rusa (Centro de Exploración y Radio Escucha Electrónica), en lo que hoy es, dicha Universidad.

La misma consta de 10 facultades en la sede central y 3 facultades regionales en diferentes provincias del país: Artemisa, Ciego de Ávila y Manzanillo; donde cada una se identifica por tener un perfil de estudio diferente y en las cuales actúan personas formando el capital humano especializado, investigando y produciendo software y servicios informáticos para la sociedad cubana y para el mundo, como aporte y contribución al desarrollo socio-económico del país.

Por sus características académicas y productivas es la única de su tipo en Cuba, con una matrícula promedio de 10 000 estudiantes en la sede central que provienen de todos los poblados, municipios y provincias del país. Por lo que constituye en la actualidad, la Universidad más grande del mismo, razones que conllevan a que todos sus estudiantes y gran parte de sus profesores sean internos.

Esto implica que al mismo tiempo que se desarrollan estas actividades académicas y productivas, se enfrenta en la Universidad una gigantesca obra constructiva para crear las condiciones de vida y de trabajo necesarias para el desarrollo y ampliación de sus capacidades. Para ello se cuenta con el apoyo de toda su comunidad, ya que casi todos sus residentes, viven en la misma durante todo el período escolar y se les da la posibilidad de salir de pase cada fin de semana y los fines de año, siendo estos los únicos pases oficialmente reglamentados por la Universidad.

A pesar de ello, estos sistemas de pase que brinda la Universidad no abarcan todas las necesidades que tienen los estudiantes al vivir diariamente en la misma, y en varias ocasiones se les hace necesario salir de ésta en horarios no establecidos dentro de estos pases previamente ya programados, o sea, durante el transcurso de la semana. En estos casos, los estudiantes deben solicitar éste pase a los directivos de su Facultad, puesto que su horario docente corresponde con el mismo, por lo que se consideran pases no reglamentados.

Hoy en día, en la Facultad 7 de la UCI esta solicitud se realiza de forma manual y se lleva a cabo sólo si el estudiante se entrevista personalmente con los autorizados a dar éste pase, ya sea el Decano o algún Vicedecano de la misma. Trayendo como consecuencia los siguientes problemas:

1. Pérdida de tiempo en el proceso de solicitud y aprobación de estos pases no reglamentados donde el más afectado en este caso es el estudiante, éste, primeramente debe dirigirse a algún directivo de su Facultad para poder realizar la solicitud personalmente y puede que en ese momento , se encuentre con la situación de no hallar a ninguno de ellos en su local de trabajo, o que precisamente en esa ocasión no puedan ser atendidos por estos, ya que los mismos, por la dinámica y responsabilidad de su trabajo, no siempre se encuentran disponibles en un determinado lugar, o con tiempo suficiente en cualquier momento del día para atender estas solicitudes.

De igual forma sucede si el estudiante tiene que ir a ver a algún profesor para pedirle su opinión porque tenga que faltar a uno o varios de sus turnos de clase, en caso de que su solicitud haya sido aprobada, lo que implica una pérdida de tiempo aún mayor porque el estudiante debe volver a dirigirse al directivo para que éste consulte estas opiniones y así poder aprobar o no dicha solicitud sin que afecte la docencia del estudiante.

2. Se interrumpe consecutivamente el trabajo de todos los involucrados en el proceso de solicitud y aprobación de estos pases no reglamentados. Pues la matrícula actual de la Facultad es de cerca de 1000 estudiantes. Por lo que se hace imposible predecir el número de solicitudes que serán realizadas durante el transcurso del día y con qué frecuencia.
3. Después de aprobada la solicitud, no se notifica a los profesores la salida de pase del estudiante provocando que los mismos no tengan forma de comprobar o saber con certeza, si uno o varios de sus estudiantes se encuentran oficialmente autorizados a faltar a sus clases.
4. Inconsistencia de los datos archivados en los Controles da la Facultad sobre toda la información referente a estos pases no reglamentados que fueron otorgados, pues la misma se guarda en hojas de papel y éstas corren el riesgo de sufrir deterioro e incluso pueden llegar a extraviarse, siendo estos datos de valiosa importancia en caso de que se necesite saber en cualquier momento si un determinado estudiante estuvo o está fuera de la Universidad.

5. No se chequea el cumplimiento del período de tiempo otorgado a un estudiante de pase, provocando que en muchas ocasiones, la fecha de entrada a la Universidad se realice de forma tardía y sin control alguno.
6. Escasa posibilidad de realizar reportes, puesto que se hace muy engorrosa la búsqueda y obtención de datos ya antiguamente archivados

Actualmente, al realizarse éste proceso de forma manual y teniendo en cuenta las dificultades anteriores y la importancia que posee su mejoramiento para el funcionamiento de la Facultad, se tiene en cuenta el siguiente **problema a resolver**: ¿Cómo gestionar la información que se genera a partir de la solicitud de pases no reglamentados por parte de los estudiantes de la Facultad 7 en la Universidad de las Ciencias Informáticas?

Por lo que se define como **objetivo general**: Desarrollar una aplicación web que gestione la información que se genera a partir de la solicitud de pases no reglamentados por parte de los estudiantes de la Facultad 7 en la Universidad de las Ciencias Informáticas. Para ello se tiene como **objeto de estudio**: el proceso de gestión de la información que se genera a partir de la solicitud de pases no reglamentados por parte de los estudiantes en la Universidad de las Ciencias Informáticas, y el **campo de acción** se enmarca en el proceso de gestión de la información que se genera a partir de la solicitud de pases no reglamentados por parte de los estudiantes de la Facultad 7 en la Universidad de las Ciencias Informáticas.

Se propone, luego de haber realizado un profundo estudio y análisis del objetivo general, las siguientes **tareas de la investigación**:

1. Realizar un análisis de los antecedentes sobre el tema de pases a nivel de Universidad.
2. Asimilar el Documento de Arquitectura definido por la Facultad.
3. Describir las herramientas y tecnologías informáticas para la implementación de la aplicación.
4. Modelar los Flujos de Trabajo propuesto por RUP con el Enterprise Architect 7.
5. Implementar la aplicación web utilizando estándares de codificación, tratamiento de errores o excepciones definidos por la Facultad en su Documento de Arquitectura.

El documento está estructurado en cuatro capítulos, en los cuales se brinda toda la información referente al trabajo investigativo realizado, el análisis y diseño del sistema a desarrollar. Además incluye una serie de anexos que complementan la información.

Capítulo 1. Fundamentación Teórica.

Se presenta la información necesaria para la realización del trabajo, se realiza una descripción detallada del objeto de estudio, se analizan otros sistemas existentes vinculados al campo de acción, se enuncian las tendencias y tecnologías actuales seleccionadas a emplear en el desarrollo de la propuesta y el por qué de su utilización.

Capítulo 2. Características del Sistema.

Se presentan las características generales del problema, describe los procesos mediante un modelo del negocio, identificando actores, trabajadores y casos de uso del negocio. Se definen además las funcionalidades del sistema, a través de los requerimientos funcionales y no funcionales, y se describen detalladamente. Se realiza la modelación de los casos de usos del sistema, así como una descripción detallada de cada uno de ellos.

Capítulo 3. Análisis y Diseño del Sistema.

Se presenta el análisis del sistema, trata sobre la construcción de la solución. Se modelan diagramas de clases, de datos, y se plantean los principios de diseño seguidos en la implementación del sistema. Descripción de la arquitectura, fundamentación. Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración.

Capítulo 4. Implementación y Prueba del Sistema.

Son modelados los diagramas de despliegue y componentes, los cuales muestran la topología del hardware sobre la que es ejecutado el sistema; y muestra la organización y dependencias entre un conjunto de componentes respectivamente.

Desarrollo



CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

Capítulo I: Fundamentación Teórica

El presente capítulo contiene un enfoque general de sistemas creados a nivel internacional y nacional vinculados al campo de acción y al análisis comparativo de las soluciones existentes con la propuesta dada en éste trabajo. Se describen además las tecnologías actuales de desarrollo utilizadas para el análisis, diseño e implementación del sistema sobre las cuales se apoya la propuesta.

1.1 Sistemas de Reservaciones.

En la actualidad, el mercado de aplicaciones informáticas es tremendamente variado y abarca un sin fin de utilidades para desarrollar tareas en todos los campos activos de la sociedad, donde la informática juega un papel fundamental en todos los ámbitos. Hoy en día, el desarrollo de las organizaciones demanda una enorme cantidad de información, de ahí que las empresas están obligadas a tomar decisiones cada vez más precisas y con mayor rapidez.

En el caso particular de la educación se está utilizando la computadora como una herramienta que permite tanto el proceso interactivo de información, como la comunicación a distancia; motivo por el cual, existen en la actualidad, instituciones educacionales que tienen desarrollados diferentes sistemas informáticos que les proporcionan grandes ventajas. Ejemplo de ellas son:

A nivel Internacional:

Infoalumno, aplicación que se usó durante el año 2007 en la Universidad de Concepción, Chile para la reservación masiva del Pase Escolar. Su objetivo fundamental era actualizar los datos personales de los alumnos como dirección y teléfono, no su nombre o matrícula. Donde cada cual debía revisar en la aplicación si estaban actualizados los datos de dirección y si no lo estaban debían modificarlos en la misma plataforma que se consideraba como una secretaria virtual.

Existen también otras aplicaciones en diferentes Distritos Escolares Unificados del mundo que brindan grandes beneficios a sus estudiantes proporcionándoles mediante éstas, una forma más rápida y segura de obtener estos pases. Generalmente estas aplicaciones no son más que el llenado de un formulario que se encuentra en los sitios Web de las escuelas y que los estudiantes tienen que enviar para ver si su solicitud es aprobada o no.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

En el Distrito Escolar Unificado de Capistrano cada estudiante debe tener su propia aplicación donde cada uno de ellos debe poseer un pase de autobús para poder subir al autobús desde el primer día de escuela. Estos pases son vendidos basados en el primer-recibido, primer-servido y una vez que una ruta ha alcanzado la capacidad, ningún pase estará disponible y los estudiantes serán colocados en una lista de espera, por lo que los interesados en obtenerlo deben ser ágiles y precisos en el uso de estas aplicaciones, debido a que las mismas, si están incompletas serán devueltas. Para ello los estudiantes deben dirigirse al sitio web del distrito encargado de todo éste proceso: www.capousd.org/transport.

Además de ellas, en la Universidad del Mar sede Santiago en Chile, se realiza la inscripción y entrega del pase escolar mediante el sistema informático de JUNAEB (Junta Nacional de Auxilio Escolar y Becas www.junaeb.cl) donde los estudiantes primero que nada deben registrar sus datos en éste sistema e imprimir el documento “Estado de pase escolar” y después cancelar en Finanzas una suma de dinero establecida por el derecho a éste pase. En el caso de alumnos antiguos deben realizar los mismos pasos pero con la extracción de una cantidad menor de dinero. Después de esto el estudiante debe realizar una serie de pasos que son los que le garantizan que dicho pase tenga validez y finalmente éste debe firmar la “Planilla de entrega de pases” y Asuntos Estudiantiles de la escuela archiva la planilla una vez entregados todos los pases que ésta contiene.

A nivel Nacional:

Particularmente en la UCI, existen también otros sistemas de reservación, entre ellos se tienen los dirigidos especialmente a la reservación del transporte para las salidas de pase que proporciona la Universidad cada fin de semana y de forma masiva en tres ocasiones durante el período de clase. Estos sistemas son:

- **Sistema de Reservación de Transporte (versión inicial).**

Éste sistema fue desarrollado en el área de informatización de la Universidad con el objetivo de proporcionarles a los estudiantes una forma de poder realizar la reservación de transporte para el pase de fin de semana desde cualquier ordenador que estuviese conectado a la red interna. También posibilitaba que los usuarios pudiesen hacer modificaciones sobre una reservación hecha con anterioridad, en algunos casos los mismos tenían la necesidad de volver a formular y especificar sus datos de lugares para donde

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

realizarían la reservación del transporte. Brindaba además la posibilidad de cancelar una reservación previamente realizada, en caso de que el usuario se hubiese arrepentido de haberla hecho.

Aun así, el hecho de que esta aplicación resolvía las funcionalidades básicas que se requerían en ese momento, no establecía una pauta final a las necesidades existentes; por lo que la dirección de Informatización de la Universidad, además de ello requería que el sistema brindara la posibilidad de hacer una programación de estas reservaciones de transporte para el pase de los estudiantes.

Es así como surge el **Sistema de Reservación de Transporte, UCI (pase.uci.cu)**, que hoy en día le da la posibilidad al estudiante de hacer una programación de su pase, dígame reservar para varios fines de semana, para todos o para alguno en específico durante el período académico; puede configurar sus salidas/entradas de forma periódica, aunque no esté en el transcurso de la semana hacer esta reservación. Por otra parte éste sistema también contiene una documentación que puede ser útil para su posterior mantenimiento, el mismo es bastante extenso y abarcador.

- **Sistema de Reservación de Pase Masivo.**

Creado con el objetivo de brindarle soporte técnico a los procesos de la dirección de transporte, incluyendo también una serie de funcionalidades de administración del sistema, que permita gestionar toda la información con que éste trabaja y que proporcione la mayor flexibilidad posible con vista a su adaptación a cualquier tipo de situación que pueda presentarse. Para realizar todas sus acciones, éste consulta la información contenida en la base de datos central que es la que le proporciona los datos de todas las personas que reservarán. Además de ello, da la posibilidad de ajustarse a cualquier disposición diferente que exista en cuanto a quiénes pueden ser las personas que reserven y hacia dónde.

En éste momento no existe en la Facultad 7 de la Universidad de las Ciencias Informáticas un sistema que se encargue de gestionar toda la información relacionada con la solicitud y aprobación de los pases no reglamentados a sus estudiantes; es por ello que se plantea la necesidad de desarrollar una Aplicación Web que de solución a la problemática existente.

Para ello se han tenido en cuenta los sistemas antes mencionados, pero se ha decidido no utilizar ninguno de los mismos ya que cada cual ha sido implementado con la finalidad de dar solución a problemas específicos que presentaron las instituciones para las cuales fueron creados y por tanto no son afines a la solución propuesta.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.2 Marco de Trabajo y Herramientas Utilizadas.

Para la elaboración de esta aplicación se usarán una serie de herramientas de código abierto dada la importancia que le confieren estas al país viéndolo desde tres ámbitos diferentes:

Político:

Desde un primer punto de vista representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación. En el presente Cuba se encuentra a merced de la empresa norteamericana Microsoft, que tiene la capacidad legal de reclamar a Cuba que no siga utilizando un sistema operativo de su propiedad, basada en leyes de propiedad industrial por las cuales también Cuba se rige; esto provocaría una interrupción inmediata del programa de informatización de la sociedad que como parte de la batalla de ideas está desarrollando el país, además pudiera implementarse una campaña de descrédito a la isla, abogando el uso de la piratería informática por parte de las instituciones estatales cubanas.

Desde un segundo punto de vista, el software libre representa la alternativa para los países pobres, y es por concepción, propiedad social, si se tiene en cuenta que una vez que comienza a circular rápidamente se encuentra disponible para todos los interesados sin costo alguno o en su defecto a muy bajo costo.

En tercer lugar es desarrollado de forma colectiva y cooperativa, tanto en su creación como en su desarrollo, tanto cuantitativa como cualitativamente mostrando su carácter público y sus objetivos de beneficiar a toda la comunidad.

Económico:

Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país.

La adquisición de cualquiera de sus distributivas puede hacerse de forma gratuita, descargándolas directamente de Internet o en algunos casos a muy bajos precios, se garantiza su explotación con un mínimo de recursos, por cuanto no hay que pagar absolutamente nada por su utilización (no requiere de licencia de uso, las cuales son generalmente muy caras), distribución y/o modificación.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

Tecnológico:

Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor por ciento de efectividad, además la corrección de sus errores de programación y obtención de las actualizaciones y nuevas versiones.

Dado que Cuba apuesta por el software libre y por las razones antes expuestas de las ventajas que concede su uso, se ha pensado en la utilización de las siguientes herramientas soportadas por el software libre y que tienen iguales o mejores prestaciones que las usadas en el sistema operativo Windows. Sin dejar de mencionar que también se utilizarán dos herramientas con software propietario por la importancia que le confiere la Facultad al uso de las mismas y que serán abordadas posteriormente.

1.2.1 Aplicaciones Web.

Las aplicaciones Web son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web (HTML, Java Script, Java, etc.) en la que se confía la ejecución al navegador. Son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizarlas y mantenerlas sin distribuir e instalar software a miles de usuarios potenciales. [1]

Utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Por tanto, las aplicaciones web son una especialización y concreción de las aplicaciones cliente-servidor, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (el HTTP: HyperText Transfer Protocol) son un estándar, y no han de ser creados por el desarrollador. [2]

La parte del cliente en estas aplicaciones está formada por el código HTML (HyperText Markup Language) que forma la página Web, con opción a código ejecutable mediante los lenguajes script de los navegadores (Java Script, VBScript, Perl Script) o mediante pequeños programas (applets) en Java y la parte del servidor está formada por un script que es ejecutado por el servidor Web, y cuya salida se envía al navegador del cliente. [3]

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

Una ventaja significativa en la construcción de aplicaciones Web que soporten las características de los browsers estándar es que deben funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, MacOS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada casi en todos lados. Sin embargo, aplicaciones inconsistentes de HTML, CSS, DOM y otras especificaciones de browsers pueden causar problemas en el desarrollo y soporte de aplicaciones Web. [4]

Entre otras de sus numerosas ventajas se destacan: [5]

- **Multiplataforma:** con un solo programa, un único ejecutable, las aplicaciones web pueden ser utilizadas a través de múltiples plataformas, tanto de hardware como de software.
- **Actualización instantánea:** debido que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.
- **Fácil de integrar con otros sistemas:** debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- **Acceso móvil:** el usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas establecidas por dicha organización; puede hacerlo desde una computadora de escritorio, una LAPTOP, una agenda electrónica, un celular, desde la oficina, la casa o cualquier otra parte del mundo.

No obstante a la serie de ventajas que presenta tiene además algunas desventajas, las cuales se plantean a continuación:

- **Acceso limitado:** la necesidad de conexión permanente y rápida a Internet hace que el acceso a estas aplicaciones no esté al alcance de todos.
- **La interactividad no se produce en tiempo real:** en las aplicaciones web cada acción del usuario conlleva a un tiempo de espera que en algunas ocasiones es extremadamente grande hasta que sea obtenida la reacción del sistema.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- **Los elementos de interacción muy limitados:** en comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas. Diferencias de presentación entre plataformas y navegadores, la falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

1.2.2 Lenguajes de Programación para la Web.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, éstos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet, fueron apareciendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y con ellas nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para las web dinámicas, que permitieran interactuar con los usuarios y utilizaran sistemas de base de datos. A continuación se dará un panorama de los diferentes lenguajes de programación para la web, utilizando de ellos, sólo los tres finales.

1.2.2.1 Lenguaje ASP.

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitios web dinámicos. En inglés significa (Active Server Pages) y fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo éste lenguaje necesitan tener instalado Internet Information Server (IIS).

No necesita ser compilado para ejecutarse y existen varios lenguajes que se pueden utilizar para crear estas páginas ASP. El más utilizado es VBScript, nativo de Microsoft. También ASP se puede hacer en Perl and Jscript (no Java Script) y su código puede ser insertado junto con el código HTML. Sus archivos cuentan con la extensión (asp).

Ventajas:

- Comunicación óptima con SQL Server.
- Soporta el lenguaje Jscript (Java script de Microsoft).
- Usa Visual Basic Script, siendo fácil para los usuarios.

Desventajas:

- Código desorganizado.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Tecnología propietaria.
- Hospedaje de sitios web costosos.
- Se necesita escribir mucho código para realizar funciones sencillas.

1.2.2.2 Lenguaje ASP.NET.

Éste es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP y fue lanzado al mercado mediante una estrategia de mercado denominada .NET.

El ASP.NET fue desarrollado para resolver las limitantes que brindaba su antecesor ASP y creado para desarrollar web sencillas o grandes aplicaciones. Para su desarrollo se puede utilizar C#, VB.NET o J# y para el funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net. Sus archivos cuentan con la extensión (aspx).

Ventajas:

- Mayor seguridad en su ejecución.
- Completamente orientado a objetos.
- Controles de usuario personalizados.
- Facilita el mantenimiento de grandes aplicaciones.
- Incremento de velocidad de respuesta del servidor.
- División entre la capa de aplicación o diseño y el código.

Desventajas:

- Mayor consumo de recursos.

1.2.2.3 Lenguaje PHP.

Es un lenguaje de programación de código libre utilizado para la creación de sitios web. Es un acrónimo recursivo que significa “PHP HyperText Pre-processor”, (inicialmente se llamó Personal Home Page) y surgió en 1995, desarrollado por PHP Group.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. No necesita ser compilado para ejecutarse y para su funcionamiento necesita tener instalado Apache o IIS con las bibliotecas de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Sus archivos cuentan con la extensión (php).

Ventajas:

- Muy fácil de aprender.
- Incluye gran cantidad de funciones.
- Se caracteriza por ser un lenguaje muy rápido.
- Capacidad de expandir su potencial utilizando módulos.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.

Desventajas:

- Dificulta la modularización.
- Se necesita instalar un servidor web.
- Dificulta la organización por capas de la aplicación.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.

Ante todo lo expuesto anteriormente y por sus ventajas entre las que sobre salen que es software libre (OpenSource) y lenguaje multiplataforma son las que inclinan la balanza para usar éste lenguaje de programación.

1.2.2.4 Lenguaje HTML.

Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de Hypertext Markup Lenguaje, en español Lenguaje de Marcas Híper textuales), desarrollado por el Word Wide Web Consortium y utilizado desde el surgimiento de Internet para la publicación de sitios web. Sus archivos pueden tener las extensiones (htm, html).

Ventajas:

- Despliegue rápido.
- Archivos pequeños.
- Permite describir hipertexto.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG (What You See Is What You Get, en español, "lo que ves es lo que obtienes").

Desventajas:

- Lenguaje estático.
- El diseño es más lento.
- Las etiquetas son muy limitadas.
- La interpretación de cada navegador puede ser diferente.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.

1.2.2.5 Lenguaje Java Script.

Éste es un lenguaje interpretado que no requiere compilación, creado por Brendan Eich en la empresa Netscape Communications y utilizado principalmente en páginas web. No dispone de herencias, y la mayoría de los navegadores en sus últimas versiones interpretan código Java script.

El código Java script puede ser integrado dentro de páginas web, por lo que el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento) que evitara las incompatibilidades entre ellos.

Ventajas:

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código Java script se ejecuta en el cliente.

Desventajas:

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

1.2.3 Servicios Web (Web Services).

Los servicios Web tienen el objetivo de lograr la interoperabilidad entre sistemas dispares para que puedan comunicarse y compartir información. Un servicio Web es una entidad programable que proporciona alguna funcionalidad determinada, y es accesible a cualquier número de sistemas que usen las normas de Internet XML y HTTP. Permiten la ejecución de sus funcionalidades sin importar la plataforma, sistema operativo, o lenguaje en el cual estén implementados. [6]. Para conseguir la interoperabilidad lo hace mediante la adopción de estándares abiertos.

Estándares empleados: [7]

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML: Es el formato estándar para los datos que se vayan a intercambiar.
- SOAP o XML-RPC: Protocolos sobre los que se establece el intercambio.

Los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP, FTP, o SMTP.

- WSDL: Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- UDDI: Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones comprobar qué servicios web están disponibles.
- WS-Security: Protocolo de seguridad aceptado como estándar por OASIS. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

Con los servicios Web se logra que sistemas heterogéneos trabajen conjuntamente como una sola aplicación computacional.

1.2.4 Sistema de Gestión de Base de Datos (SGBD).

Hoy en día existen muchas empresas y sitios web que necesitan mantener de forma eficiente un gran volumen de datos. Muchos de ellos optan por soluciones comerciales, aunque muchas otras confían en el software libre optando por una solución como PostgreSQL o MySQL.

Estos Sistemas de Gestión de Bases de Datos (SGBD) pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Se componen de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta y tienen como propósito general manejar de manera clara, sencilla y ordenada un conjunto de información.

Por lo que las principales funciones que deben cumplir estos SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

De forma general se destacan entre ellos las siguientes ventajas :

- Gran velocidad de procesamiento en muy poco tiempo.
- Integridad referencial al terminar los registros.
- Independencia del tratamiento de información.
- Facilidad de manejo de grandes volúmenes de información.
- No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.

1.2.4.1 SGBD MySQL.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también el de la marca.

Éste gestor de bases de datos es, probablemente, el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta aceptación es debida, en parte, a que existen infinidad de bibliotecas y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Características:

- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Soporta gran cantidad de tipos de datos para las columnas.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Dispone de API's en cantidad de lenguajes (C, C++, Java, PHP,).
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

Ventajas:

- Gran velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de éste gestor son envidiables para muchos de los gestores comerciales existentes, debido a su facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
- El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).

Desventajas:

- Carece de soporte para transacciones, rollback's y subconsultas.
- No es viable para su uso con grandes bases de datos a las que se acceda continuamente, ya que no implementa una buena escalabilidad.
- El hecho de que no maneje la integridad referencial, hace de éste gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí poseen esta característica.

1.2.4.2 SGBD PostGreSQL.

PostGreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

comerciales. PostgreSQL incluye características de la orientación a objetos, como son la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Pero a pesar de ello, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Características:

- Incorpora una estructura de datos array.
- Soporta el uso de índices, reglas y vistas.
- Implementación del estándar SQL92/SQL99.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes entre otras.
- Incluye herencia entre tablas aunque no entre objetos, porque no existen, por lo que a éste gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.

Ventajas:

- Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.
- Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).

Desventajas:

- Consume gran cantidad de recursos de procesamiento y memoria.
- Es de 2 a 3 veces más lento que MySQL.
- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.

Por todo lo antes expuesto y por ser PostgreSQL ampliamente considerado como el sistema de bases de datos de fuente abierta más avanzado del mundo y poseer un gran número de características que históricamente sólo pueden ser vistas en productos comerciales de alta confiabilidad, se seleccionó como SGBD .

1.2.5 Patrones de Arquitectura y Diseño.

La utilización de los patrones ha tomado un gran auge a partir del desarrollo del modelo orientado a objetos. Un patrón describe un problema que ocurre una y otra vez en un entorno y describe también el núcleo de la solución al problema, de forma que pueden reutilizarse continuamente. Los patrones capturan la experiencia existente y probada para promover buenas prácticas. [8]

1.2.5.1 Modelo Cliente/Servidor.

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. [9]

En éste modelo se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde las solicitudes. Las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios y en el cliente permanece sólo lo particular de cada usuario.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

Principales características: [10]

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

Ventajas: [11]

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se comunica con el servidor utilizando un protocolo de alto nivel de abstracción como por ejemplo hace un servidor de SQL

La tecnología cliente/servidor es ampliamente difundida y muy utilizada por las aplicaciones Web.

1.2.5.2 Arquitectura Orientada a Servicios (SOA).

La Arquitectura Orientada a Servicios (en inglés Service-Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. Proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación. Así como provoca una reducción considerable de tiempo y costo del desarrollo de aplicaciones al utilizar servicios disponibles que ya han sido desarrollados, en la solución de problemas comunes. El sistema que reutiliza componentes ya probados aumenta su robustez, disminuye la complejidad en la integración, pues existe una abstracción de la tecnología y ubicación de los servicios.

1.2.5.3 Modelo Vista Controlador (MVC).

El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y el funcionamiento de la lógica de control en tres componentes muy diferentes. Éste patrón se

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

observa con mucha frecuencia en las aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a las páginas.

En la actualidad, numerosas empresas deciden pasar sus aplicaciones a la arquitectura MVC para documentar mejor el código, ahorrar espacio y en caso de no disponer de diseñadores Web, poder auxiliarse en los servicios de un diseñador quien a su vez no sepa mucho de programación pero que les construya las vistas.

El **Modelo** es completamente acceso a datos, y las funciones que llevan a lo que comúnmente se llama “lógica del negocio”, más bien sean datos y reglas del negocio. El modelo es quien lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función particular y mediante esta forma, si por alguna razón decide cambiarse el gestor de bases de datos este cambio sólo afecta a estas funciones, y nunca al resto de la aplicación. Tener un modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo modelo.

La **Vista**, en una aplicación Web, es HTML y lo necesario para convertir datos en HTML, o sea, muestra la información del modelo al usuario. Las vistas son relativamente sencillas; se encargan de la entrada y salida, sin que conserven los datos ni ofrezcan directamente la funcionalidad de las aplicaciones.

El **Controlador** es lo que une la vista y el modelo. Es un objeto de interfaz no destinado al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. Un controlador podría ser por ejemplo las funciones que toman los valores de un formulario, consultan la base de datos (a través del modelo) y producen valores, que la vista recepcionará para posteriormente convertirlos a HTML.

Es el encargado de gestionar las entradas del usuario. Recibe los eventos de entrada, por ejemplo: un clic, un cambio de campo de texto, la opresión de una tecla, un tiempo prefijado. Contiene reglas de gestión de eventos, del tipo “Si Evento A, entonces Acción B”. Estas acciones pueden suponer peticiones al modelo o a las vistas. Mediante esta forma bien estructurada el código encargado de “realizar algo” está completa y perfectamente separado del código dedicado a crear HTML.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.2.5.4 Alta Cohesión y Bajo Acoplamiento.

Se pueden separar, aunque están íntimamente ligados, de hecho si se esfuerza en aumentar mucho la cohesión del sistema o software, se tiene un alto acoplamiento entre las clases, y por el contrario si se reduce mucho el acoplamiento, se verá mermada la cohesión:

Alta cohesión: La información que almacena una clase debe ser coherente y estar en la mayor medida posible relacionada con la clase, donde cada elemento del diseño debe realizar una labor única dentro del sistema no desempeñada por el resto de los elementos. Un ejemplo de baja cohesión son servicios que realizan demasiadas tareas.

Bajo acoplamiento: Es la idea de tener las clases lo menos ligadas entre sí posible. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando así, su reutilización y disminuyendo la dependencia entre ellas.

1.2.6 Proceso de Desarrollo de Software.

Hoy en día no existe una metodología universal para el desarrollo del software. La tendencia al crecimiento del volumen y complejidad de los productos en la Industria del Software y la búsqueda de adecuarse más a las necesidades del usuario; exigen cada día mayor calidad y productividad de los mismos. Por lo que para enfrentar esta situación las empresas requieren desarrollar o adquirir una disciplina en el desarrollo del software y controlar que los ingenieros usen de forma consistente los nuevos métodos, esto sólo es posible si se sigue una metodología específica, que se encargará de establecer los principios necesarios para la obtención de un software económico, fiable y que funcione eficientemente.

Por tanto se decide utilizar como metodología de Ingeniería de Software para el desarrollo de la propuesta de solución planteada, el Proceso Unificado de Desarrollo de Software (RUP).

1.2.6.1 Metodología RUP.

RUP es un proceso para el desarrollo de un proyecto de software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto.

Como 3 características esenciales se presentan:

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- **Dirigido por los Casos de Uso:** Orientan el proyecto a la importancia para el usuario y lo que éste quiere.
- **Centrado en la arquitectura:** Relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.
- **Iterativo e incremental:** Divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Como filosofía RUP maneja 6 principios clave:

- **Adaptación del proceso.**

El proceso deberá adaptarse a las características propias de la organización. El tamaño del mismo, así como las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

- **Balancear prioridades.**

Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.

- **Colaboración entre equipos.**

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.

- **Demostrar valor iterativamente.**

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

- **Elevar el nivel de abstracción.**

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software o esquemas (frameworks) por nombrar algunos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- **Enfocarse en la calidad.**

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

Ciclo de vida de RUP.

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades.

- **Inicio:**

Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.

- **Elaboración:**

Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.

- **Construcción:**

Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

- **Transición:**

Se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

1.2.6.2 Lenguaje de Modelado: UML 2.0.

UML (Unified Modeling Lenguaje) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Mediante su creación se pretendió unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar.

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en forma convencional y fácil de comprender para comunicar a todas las personas que estén vinculadas en el desarrollo del software, por lo que es llevado a cabo mediante un conjunto de símbolos y diagramas.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todas aquellas personas que de una u otra forma estén involucrados en el proceso de desarrollo. Un modelo UML es lo que supuestamente hará el sistema pero no como lo hará.

La versión que se utilizará para el futuro desarrollo de esta aplicación es la 2.0, ésta ofrece nuevas ventajas sobre la versión anterior, pues en ella se definen una serie de diagramas adicionales a los ya establecidos. UML 2.0 es la mayor revisión que se le ha hecho a UML desde la versión 1.0, por lo que el modelo conceptual ha sido reestructurado completamente y nuevos diagramas han sido incorporados.

Los diagramas tradicionales también han sido mejorados. La nueva versión permitirá a los fabricantes de herramientas CASE, proporcionar a los analistas, arquitectos y desarrolladores, herramientas cada vez más potentes; que les permitan aprovechar mejor los modelos y como consecuencia generar una mayor cantidad de código; reduciendo significativamente el ciclo de desarrollo de sus aplicaciones.

1.2.7 Herramientas a Utilizar.

En la confección de éste trabajo se hace indispensable tener en cuenta la utilización de algunas herramientas necesarias en el diseño de interfaz, el trabajo con las imágenes y el modelado del proyecto. Estas herramientas son:

1.2.7.1 Herramientas CASE.

Las herramientas CASE (en inglés Computer Aided Software Engineering; Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. Entre sus principales objetivos se encuentran:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software

1.2.7.1.1 Rational Rose.

Es la herramienta Case desarrollada por los creadores de UML que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Permite una trazabilidad real entre modelo (análisis y diseño) y código ejecutable.

Rational Rose domina el mercado de herramientas para el análisis, modelado, diseño y construcción orientado a objeto, tiene todas las características que los desarrolladores, analistas y arquitectos exigen, soporte UML incomparable, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración óptima.

Rose constituye una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de sus grandes ventajas es que utiliza la notación estándar en la arquitectura del software UML, lo cual permite a los arquitectos del software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modular sus componentes e interfaces en forma individual y luego unirlos en otros componentes del proyecto.

1.2.7.1.2 Enterprise Architect (EA).

Es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. El manual de usuario está disponible en línea.

El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta éste proceso en un ambiente fácil de usar, rápido y flexible.

Provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Combinados con la ubicación de recursos y tareas incorporados, los equipos de Administradores de Proyectos y Calidad están equipados con la información que ellos necesitan para ayudarles a entregar proyectos en tiempo.

Ventajas:

- Intuitivo y simple de usar.
- Documentación de alta calidad compatible con MS Word.
- EA ayuda a administrar la complejidad con herramientas para rastrear las dependencias, soporte para modelos muy grandes, control de versiones para diferentes proveedores, líneas base por cada punto del tiempo.
- Ofrece un completo panel de herramientas básicas tales como diagramas de CU, diagramas de secuencia, diagramas de clases entre otras, también ofrece complejas herramientas tales como ingeniería inversa o control de versionamiento.
- Contiene una interfaz gráfica muy amigable, es fácil de usar, y proporciona rapidez una vez que el usuario logra amigarse con sus opciones y elementos.
- Genera reportes detallados y complejos de EA con la información que se necesita en el formato que una determinada compañía o cliente demanda.
- EA permite navegar y explorar un modelo de código fuente en el mismo ambiente.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

Por todo lo antes expuesto y a pesar de ser una herramienta de carácter propietario, se ha seleccionado el mismo como la herramienta para el desarrollo del presente trabajo por la importancia que le confiere la Facultad.

1.2.7.2 Herramienta de Diseño.

En la actualidad existen muchos entornos para el desarrollo Web, pero son las aplicaciones de Adobe y Macromedia las que tienen un mayor número de seguidores. Esto es así porque ambos superan en opciones y calidad al resto de sus competidores, además de ofrecer mayor integración con productos tan conocidos como Macromedia Flash en el caso de Dreamweaver o Adobe Photoshop en el de GoLive, agilizando de este modo el proceso de trabajo.

Ventajas:

GoLive y Dreamweaver ofrecen la posibilidad de crear páginas en un entorno visual gráfico, como si de un programa de maquetación se tratara, añadiendo funcionalidad e interactividad sin necesidad de escribir código HTML.

Ambos programas permiten crear páginas mediante la escritura o modificación de código, tarea que se facilita entre otras utilidades con las funciones y herramientas integradas de depuración (inserción de Java Script o la inclusión de estilos CSS). Entre otras ventajas se tienen:

- **Disponibilidad:**

Macromedia en su política de venta ha optado por ofrecer lo que denominan “Studios”, en los que se incluyen varios programas destinados a un mismo fin. En el caso de Dreamweaver existe la posibilidad de comprarlo como producto independiente o con el software para el tratamiento de imágenes Fireworks como complemento del editor.

En el caso de Adobe se ofrece todo el paquete para la web, denominado Adobe Web Collection, en el que se incluye GoLive, Photoshop, Illustrator y LiveMotion.

- **Área de trabajo:**

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

GoLive ha sido criticado por el excesivo número de paletas y opciones con las que cuenta y que a veces, sobre todo cuando se empieza a usar el programa, puede resultar un tanto confuso. Pero a medida que se va usando demuestra que en comparación con Dreamweaver no posee muchas más Paletas.

- **Enlaces con programas:**

Uno de los puntos fuertes de ambas aplicaciones es la posibilidad de conectarse con otras aplicaciones gráficas sin tener que cerrar el editor o abrir el programa de edición gráfica. GoLive aporta más opciones a la hora de importar una imagen y poder editarla mediante un doble clic. Gracias a la incorporación en la última versión de los objetos inteligentes. La única opción disponible en Dreamweaver es editar una imagen en Fireworks y una vez modificada, volver al programa con la imagen ya variada.

Dreamweaver aventaja a GoLive gracias al Macromedia Extensión Manager que a través de él es posible añadir nuevas funciones. Estas extensiones son descargables gratuitamente desde varias páginas de la red, siendo la web de Macromedia la que dispone de mayor número de ellas. GoLive también permite añadir nuevas funciones a su programa, llamadas acciones; pero la cantidad y variedad de las que dispone Dreamweaver no es ni comparable con las que cuenta Adobe.

- **Importación de elementos:**

Dreamweaver da la opción de insertar una página web creada desde Fireworks, manteniendo toda la interactividad que se haya incluido: mapas de imagen e incluso divisiones.

GoLive en este sentido permite importar archivos en formato PSD con capas, cuyo contenido es distribuido en diferentes cajas flotantes, colocando cada capa en una de estas cajas y animarlas añadiendo acciones DHTML en el editor de línea de tiempo.

A la hora de importar en Dreamweaver documentos HTML creados en un editor externo apenas se verá afectada la estructura original gracias a Roundtrip HTML, el cual, entre otras características, permite incorporar una página con etiquetas de ASP o CFML sin que sean modificados.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

GoLive ha mejorado en la importación de código. También es posible incorporar código XML o PHP sin sufrir modificaciones.

La mayoría de editores web permiten editar botones interactivos sin necesidad de conocer lenguaje JavaScript, pero Dreamweaver ofrece una mayor flexibilidad en este apartado, también permite realizar Barras de Navegación, así como objetos de Flash en formato swf.

GoLive permite crear botones (las Imágenes de Sustitución de Dreamweaver), importar archivos ShockWave de Flash, y crear animaciones.

Otra de las funciones que disponen ambos editores es la posibilidad de crear páginas HTML dinámicas, o DHTML, capaces de cambiar el tamaño, la posición o las propiedades de estilo de las capas.

- **Edición de código y colaboración:**

Dreamweaver incorpora un panel Referencia de OReilly que constituye una herramienta de consulta rápida de etiquetas HTML, objetos Java Script y estilos CSS.

GoLive también puede integrar código ASP y conectar a bases de datos, ya sean de Microsoft Access, Oracle, Microsoft SQL Server o cualquier base de datos accesible mediante ODBC.

Ambos programas soportan CSS, teniendo opciones para importar hojas de estilo externas (archivos .css) e insertarlas en la cabecera de un documento HTML concreto, y también para exportar una hoja de estilo interna como archivo .css permitiendo así vincularlo a otras páginas web. GoLive es el único que soporta CSS-2.

Selección:

En la actualidad existen muchos entornos desarrollados para el manejo de aplicaciones como XHTML, CSS y Java Script, pero su éxito depende de su calidad, rendimiento, facilidades de edición y actualización. La desventaja de estos entornos es su costo al ser adquiridos mediante licencia, y no son gratuitos. Ambos son programas de edición de páginas web de un alto grado de profesionalización, pero sigue siendo Dreamweaver el que sobresale por contar con mayores aplicaciones para el desarrollo web y por tanto constituye la selección para el desarrollo de éste trabajo.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.2.7.3 Zend Studio.

Zend Studio es uno de los ambientes de desarrollo o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP. Mediante un gran conjunto de herramientas de edición, depuración, análisis, traceo, optimización y bases de datos, Zend Studio acelera la codificación por parte de los desarrolladores. [12]

Sirve de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código, lo que acelera los ciclos de desarrollo y simplifica los proyectos complejos.

Consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Ambas se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. Existen versiones del producto para Windows, Linux y MacOS. [13]

Los expertos en PHP consideran a Zend Studio como el entorno IDE más maduro y con más características útiles. Si se desea aumentar la productividad en desarrollos PHP no cabe duda que éste programa puede ayudar seriamente. [14]

1.2.8 Frameworks en PHP.

PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, y aunque esté dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño **Modelo Vista Controlador** (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

En la actualidad existen más de 40 frameworks disponibles en la comunidad de PHP por lo que se hace cada día más difícil decidir qué Framework funciona mejor, sobre todo porque cada uno de ellos ofrece diferentes funcionalidades. Entre los más populares se encuentran:

Zend Framework:

- Un poderoso framework MVC.
- Completa documentación y test de alta calidad.
- Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad para el futuro.
- Robustas clases para autenticación y filtrado de entrada.
- Un buscador compatible con Lucene.
- Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo).
- Clientes para servicios web, incluidos Google Data Apis y Strikelron.
- Muchas otras clases útiles para hacerlo tan productivo como sea posible.
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.

CakePHP:

- Componentes de seguridad y sesión.
- CRUD de la base de datos integrado.
- Sistema de plantillas rápido y flexible.
- Es principalmente el más avanzado framework MVC, con algunos módulos añadidos en la parte superior.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Se puede manejar la mayoría de material del proyecto a desarrollar, y que incluye el soporte a Ajax y validación de datos.
- También cuenta con un módulo de autenticación de usuario único llamado 'Access Lists', que se puede utilizar para dar acceso a los diferentes usuarios de diferentes partes del sitio web con CakePHP.

WACT - Web Application Toolkit:

- Tiene las cosas básicas necesarias de un framework, como MVC, objetos de la base de datos y un motor de plantillas.
- No tiene módulos adicionales, y definitivamente carece de algunas cosas importantes, como Ajax, almacenamiento en caché, un módulo de autenticación y más.
- Tiene uno de los mejores motores de la plantilla que le permite incluso, crear sus propios códigos de plantilla para añadir cierta funcionalidad.

Prado:

- Es un framework muy singular, ofrece algunas características muy interesantes aunque no viene con soporte incorporado para MVC, pero tiene todo tipo de componentes para manejar la mayoría de las tareas habituales de PHP.
- Toma acontecimiento impulsado por los scripts. Por ejemplo, es posible poner un evento OnClick sobre un botón en HTML que vincula a una función de PHP. Además contiene otros componentes, como GridView y Repeater Control.

PHP on Trax:

- Éste framework parece ser un puerto de PHP sobre Ruby on Rails, y es usado para llamarlo en PHP sobre rails.
- Incluye soporte incorporado para Ajax.
- Le faltan algunas cosas, como un módulo de autenticación y un módulo de almacenamiento en caché. No tiene módulos adicionales, por lo que es un todo en uno en éste momento.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

ZooP Framework:

- El ZooP Framework es realmente de características empacadas, que soporta casi cualquier cosa, salvo por una capa ORM.
- Ofrece controles GUI, lo que facilita la creación de formularios rápidamente. Soporta la mayoría de los elementos básicos de un formulario, y la ventaja de usar los controles de GUI de HTML es que puede integrarse perfectamente con la lógica de validación de los controles GUI.

CodeIgniter:

- CodeIgniter es relativamente un nuevo framework, por los fabricantes de ExpressionEngine, y parece muy prometedor.
- No maneja múltiples bases de datos.
- Está inspirado en Ruby on Rails, y ofrece una gran parte de la misma funcionalidad, como los scaffolding.
- Tiene una excelente documentación, y se han incluido vídeo tutoriales.

Symfony Project:

Framework diseñado para optimizar el desarrollo de las aplicaciones Web creado con PHP 5. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación y se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos.
- Sigue las mejores prácticas y patrones de diseño para la Web.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

- Preparado para aplicaciones empresariales y es adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

Éste framework toma lo mejor de la arquitectura MVC y la implementa de forma tal que el desarrollo de la aplicación sea rápido y sencillo. Su flujo de trabajo se muestra a continuación:

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

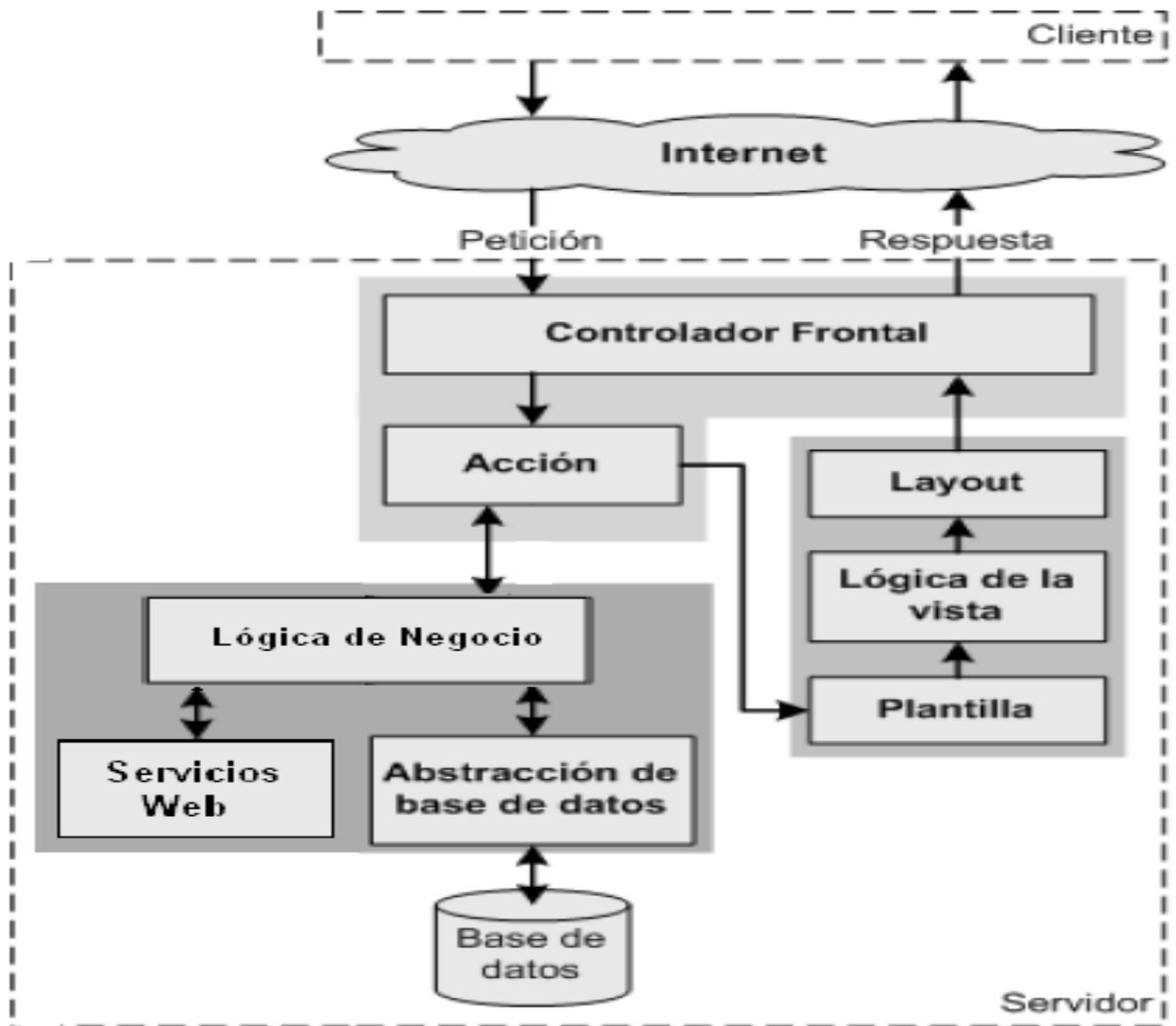


Fig. 1.1

El hecho de que éste framework utilice MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por éste framework, donde el código de la presentación se guarda en la vista, el código de manipulación de datos en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. Symfony facilita también el mantenimiento más sencillo de las aplicaciones, la encapsulación y mayor seguridad en el acceso a la base de datos.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

Además de las características mencionadas para cada framework, todos los frameworks antes mencionados son compatibles con PHP5. CakePHP, WACT, ZOOP y CodeIgniter pueden ejecutarse en PHP4. El framework Prado no soporta MVC. Todos los frameworks soportan múltiples bases de datos y objetos DB, además de que Prado no soporta objetos DB. No se admite el almacenamiento en caché por WACT y PHP on TRAX.

Cada uno de ellos ofrece diferentes funcionalidades por lo que su elección se basa en los intereses de quienes los vayan a utilizar. En éste caso, se propone éste último como framework a emplear.

Conclusiones

Se ha brindado de forma general un preámbulo de sistemas existen a nivel mundial y nacional vinculados al campo de acción. A partir del cual se concluyó, que dichos sistemas no cumplen con los requisitos necesarios que puedan satisfacer las necesidades encontradas, por lo que no se hará uso de ninguno de ellos. Además, se ha profundizado en el estudio y comparación de herramientas que son de gran ayuda para la elaboración de la aplicación en cuestión, seleccionando de ellas, las más acordes a la propuesta de solución.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Capítulo II. Características del Sistema.

En el presente capítulo se identifican los principales problemas que fundamentan la propuesta de solución, abordando aspectos fundamentales como el objeto de estudio y el flujo actual del proceso. Se realiza un Modelo del Negocio, donde se recoge toda la información necesaria para desarrollar un sistema que brinde todas las funcionalidades requeridas. Se describen los requisitos funcionales y no funcionales que se deben tener en cuenta para la elaboración de dicho sistema, permitiendo de esta forma, obtener una visión general del mismo.

2.1 Objeto de Estudio.

La Universidad de las Ciencias Informáticas (UCI), se encuentra en estos momentos inmersa en el proyecto de informatización de todos sus procesos, cuyo objetivo es crear un micro-sociedad completamente informatizado, que pudiera constituir el modelo de sociedad cubana del futuro.

Precisamente el objeto de estudio de éste trabajo lo constituye el proceso de gestión de la información de la solicitud y aprobación de pases no reglamentados a los estudiantes de la Facultad 7, proceso que se realiza de forma manual y personal, lo que conlleva a una pérdida sustancial de tiempo desde la solicitud hasta la autorización del mismo.

2.1.1 Situación Problemática.

Los sistemas de reservación de pase que existen en estos momentos en la Universidad resuelven los grandes problemas que existían con las salidas oficiales que da la misma cada fin de semana y cada fin de año y para los cuales garantiza el transporte; pero no le dan solución a la problemática que representa hoy en día para la Facultad 7, la solicitud y aprobación de un pase no reglamentado, que no son más que aquellos pases que no han sido previamente programados por la Universidad y que son solicitados personalmente por el estudiante a algún directivo de la Facultad.

En la Facultad 7 de dicha Universidad, éste proceso se realiza de forma manual y se lleva a cabo solo si el estudiante se entrevista personalmente con los autorizados a dar éste pase, ya sea el decano o algún Vicedecano de la misma. Esto trae como consecuencia que los mismos se vean interrumpidos en su trabajo en muchas ocasiones durante el día, debido a que el volumen de estudiantes residentes en la Facultad es un tanto numeroso, afectando así, de forma general, el funcionamiento del trabajo de cada

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

uno de ellos y como generalmente la solicitud de estos pases no reglamentados son de carácter urgente y sin previo aviso, su atención se hace necesaria en el menor tiempo posible.

Por otra parte, aquellos estudiantes que necesitan realizar esta solicitud, también se ven afectados con los métodos actuales para realizarla, puesto que el horario de atención a los estudiantes que necesiten analizar algún problema con el decano es después de las 5:00 pm salvo los casos de urgencia que si se atienden en el momento que se presentan si el Decano o algún Vicedecano están presentes, por lo que frecuentemente estos no pueden esperar a esa hora y recurren en otros horarios, en los cuales también pueden encontrarse con otras situaciones puesto que no siempre estos directivos se encuentran localizables por la dinámica de su trabajo.

Todo esto, aparejado al hecho de que la información de aquellos estudiantes que se encuentran de pase, pueda perderse, puesto que sólo se registra en una hoja de papel, traen como consecuencia un atraso considerable en la gestión de la información que implica una de estas solicitudes de pases no reglamentados por parte de los estudiantes de la Facultad 7 de la UCI.

2.2 Flujo Actual de los Procesos.

El análisis del flujo de procesos permite reconocer como funciona realmente el negocio para producir uno o varios resultados. El resultado puede ser un producto, un servicio, una información o combinaciones de ellos. Analizar el flujo de los procesos permite revelar problemas potenciales tales como: los cuellos de botella, los pasos innecesarios, la circulación doble de la información, la duplicación del trabajo, solo por citar algunos.

La Facultad 7 brinda a sus estudiantes la posibilidad de solicitar pases no reglamentados después de las 5:00 pm, a excepción de los casos de urgencia, en cuyo caso no pueden esperar a ese horario y son atendidos si los directivos se encuentran disponibles, siendo para los demás casos, la única solución para poder solicitar estos pases.

Para solicitar éste pase no reglamentado el estudiante puede dirigirse además de al Decano, a algún Vicedecano de la Facultad, los cuales también tienen la autorización para dar estos pases siempre y cuando sigan el mismo procedimiento del Decano.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Para realizar estas solicitudes, los estudiantes deben hacerlo en el horario establecido e informar sobre la causa de su problema. Si el mismo implica que el estudiante tenga que faltar a actividades docentes o productivas planificadas, ello requiere entonces que el estudiante entregue por escrito la autorización de los profesores responsables de cada una de estas actividades; y es cuando el directivo, después de tener en sus manos, el autorizo de dichos profesores, otorga o no el pase al estudiante. En caso que la solicitud sea aprobada, se registran los datos personales del estudiante, así como su grupo, y por cuánto tiempo fue autorizado a estar fuera de la Universidad.

2.3 Objeto de Automatización.

Con la aplicación se pretende automatizar el siguiente proceso:

Gestión de la información de la solicitud y aprobación de pases no reglamentados a los estudiantes de la Facultad 7.

2.3.1 Información que se maneja.

La Solicitud de pases no reglamentados contiene implícita datos personales del estudiante y datos propios de la reservación siendo esta información sensible. Se debe lograr, y es uno de los objetivos que persigue la implementación de la Aplicación Web, que la manipulación de la misma sea de una manera segura y confiable, que no atente contra la autenticidad e integridad de la misma.

2.3 Modelo del Negocio.

En esta etapa se definen cuales son los procesos y procedimientos que se tuvieron en el escenario para el cual se desarrolló la aplicación. Esto permite identificar los casos concretos que se automatizaron en el sistema, la relación que debe existir entre la ingeniería de software y el negocio, con el fin de aclarar el enfoque que quiere tener el cliente con el software.

El proceso de solicitud de pases no reglamentados de los estudiantes de la Facultad 7 de la Universidad de las Ciencias Informáticas comienza cuando un estudiante va a ver al Decano o algún Vicedecano para pedirle un pase, éste explica su problema y el período de tiempo por el que necesita el pase. Si el mismo implica que el estudiante tenga que faltar a actividades docentes o productivas planificadas, ello requiere entonces que el estudiante entregue por escrito la autorización de los profesores responsables de cada una de estas actividades, de ausentarse o no, según el criterio que tenga del mismo. Una vez recibida la

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

opinión de los profesores, el directivo, después de valorar estas, conjuntamente con las razones y el plazo de tiempo solicitado por dicho estudiante, decide si otorgar o no el pase y el, o la secretaria del Decano, elaboran el pase para el estudiante y dejan los datos registrados en los controles de la Facultad.

2.3.1 Actores del Negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [15]

Actores del Negocio	Descripción
Estudiante	El estudiante es quien inicia todas las acciones durante el proceso de solicitar un pase no reglamentado, por lo tanto, es el que se beneficia con tales acciones al finalizar el mismo.

Tabla 2.1 Justificación de los actores del negocio.

2.3.2 Trabajadores del Negocio.

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), un ordenador o un sistema de información; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.

Trabajadores del Negocio	Descripción
Directivo	Son los únicos autorizados a otorgar o denegar un pase no reglamentado, por lo que son los encargados de recibir todas las solicitudes realizadas y dar una respuesta al estudiante.
Profesor	Es el encargado de darle su autorización o no al estudiante para ausentarse a sus clases o actividades productivas.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Secretaria	Es la encargada de elaborar un pase no reglamentado y llevar un control en la Facultad de los datos de las solicitudes aprobadas.
------------	---

Tabla 2.2 Justificación de los trabajadores del negocio.

2.3.3 Diagrama de Casos de Uso del Negocio.

El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del negocio. [16] A continuación se muestra la figura 2.1 correspondiente al diagrama de casos de uso del negocio.

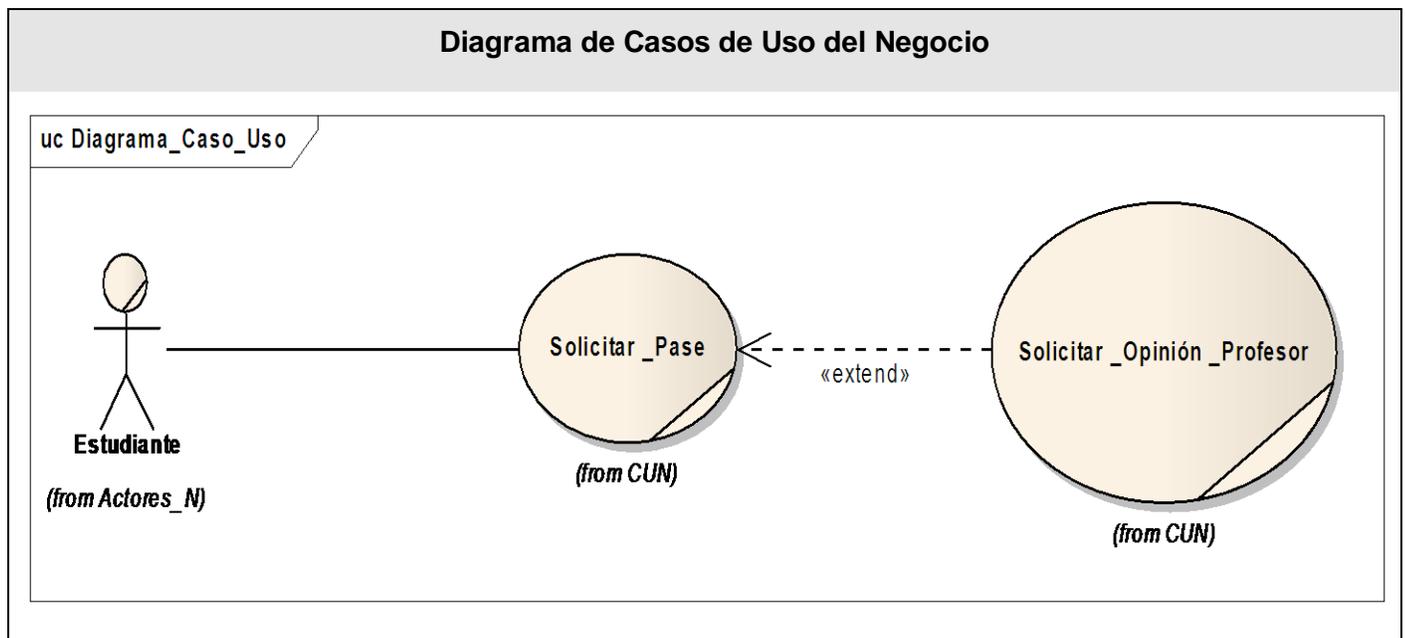


Figura 2.1 Diagrama de CUN.

2.3.4 Descripción de los Casos de Uso del Negocio.

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Para el caso de uso del negocio se tendrá una especificación textual con su correspondiente diagrama de actividades y el diagrama de clases del Modelo de Objeto.

2.3.4.1 Especificación Textual del Caso de Uso del Negocio: Solicitar _Pase.

Caso de Uso:	Solicitar _Pase.
Actor:	Estudiante.
Trabajadores:	Directivo, Secretaria.
Resumen:	El caso de uso se inicia cuando el estudiante va a pedirle pase a un directivo, éste valora la situación y decide si otorgar el pase o no, para ello cuenta con la opinión de los profesores de dicho estudiante, en caso de que éste tenga afectaciones docentes, y finaliza cuando el estudiante recibe una respuesta de su solicitud.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El estudiante se presenta ante el directivo para solicitar el pase y explica las razones y por cuánto tiempo lo necesita.	2. El directivo aprueba la solicitud.
	3. El directivo verifica si el estudiante tiene afectaciones.
	4. El directivo otorga el pase.
	5. El directivo le pide a la secretaria de la decana la elaboración del pase.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

	6. La secretaria elabora el pase y actualiza el documento de registro de datos.
	7. La secretaria entrega el pase al directivo.
	8. El directivo firma el pase y lo entrega al estudiante.
9. Se retira.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Línea 2	2.1 El directivo rechaza la solicitud. Retornar al paso 9.
Línea 3 3.3 Entrega autorización. Se retorna al paso 4.	3.1 El estudiante tiene afectaciones. 3.2 El directivo solicita autorización de profesores.
Línea 3.3 3.3.1 No tiene autorización. 3.3.3 Busca autorizo. Retorna al paso 3.3.	3.3.2 El directivo lo manda a buscar autorización de profesores.
Línea 4	4.1 El directivo no otorga pase. Retorna al paso 9.

Tabla 2.3 Descripción textual del CUN: Solicitar Pase.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.3.4.2 Diagrama de Actividad del Caso de Uso del Negocio: Solicitar _Pase.

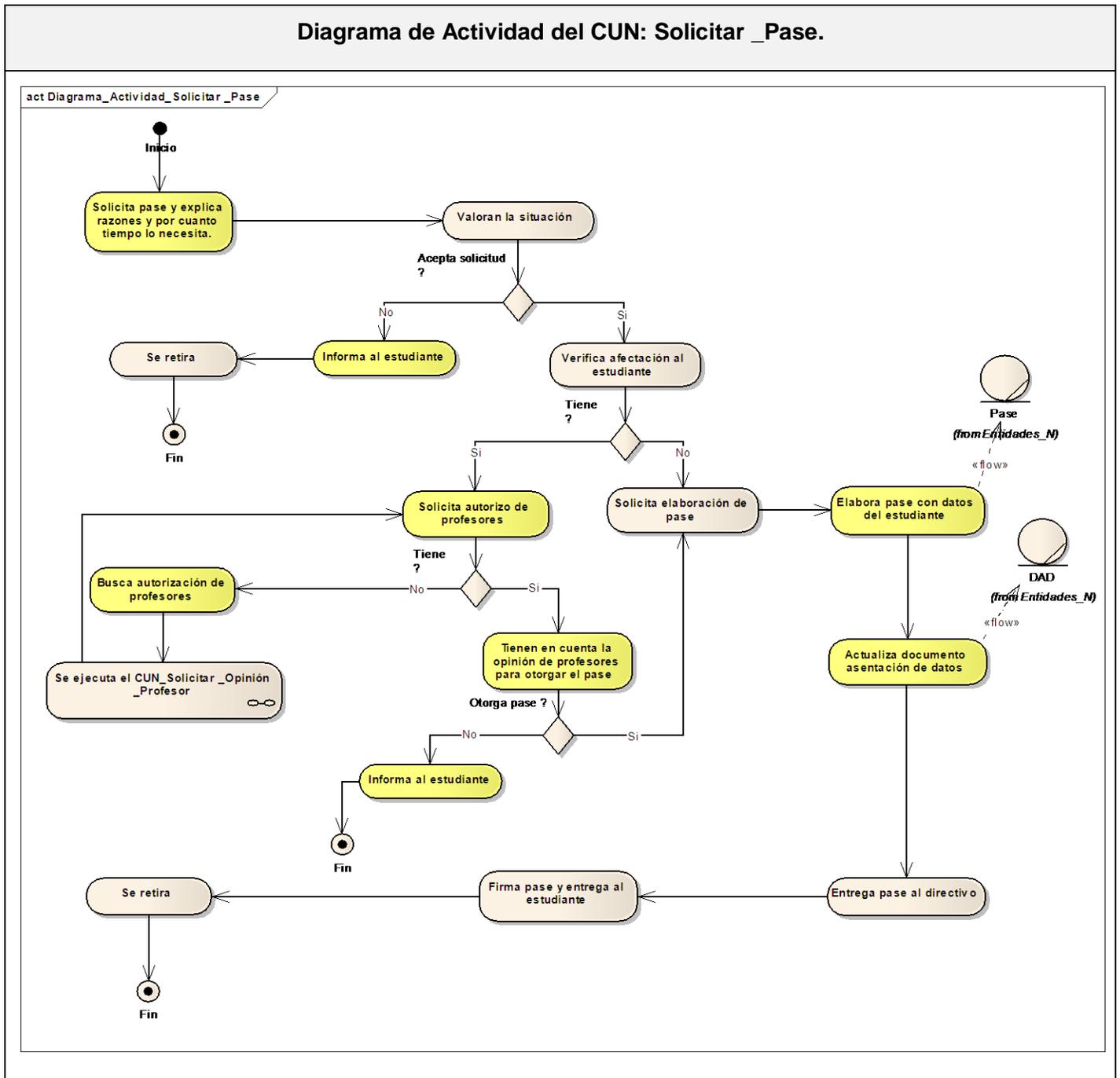


Figura 2.2 Diagrama de actividad del CUN: Solicitar _Pase.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.3.4.3 Especificación Textual del Caso de Uso del Negocio: Solicitar _Opinión _Profesor.

Caso de Uso:	Solicitar _Opinión _Profesor.
Actor:	Estudiante.
Trabajador:	Profesor.
Resumen:	El CU se inicia cuando el estudiante se dirige al profesor para pedirle autorización para ausentarse a sus clases según el tiempo que vaya a estar fuera de la Universidad, y finaliza cuando el profesor le da el autorizo al estudiante por escrito.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1 .El estudiante se presenta ante el profesor para solicitar el autorizo y le explica las razones y por cuánto tiempo necesita ausentarse.	2. El profesor elabora el autorizo y lo entrega al estudiante por escrito.
3. Se retira.	

Tabla 2.4 Descripción textual del CUN: Solicitar _Opinión _Profesor.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.3.4.4 Diagrama de Actividad del Caso de Uso del Negocio: Solicitar _Opinión _Profesor.

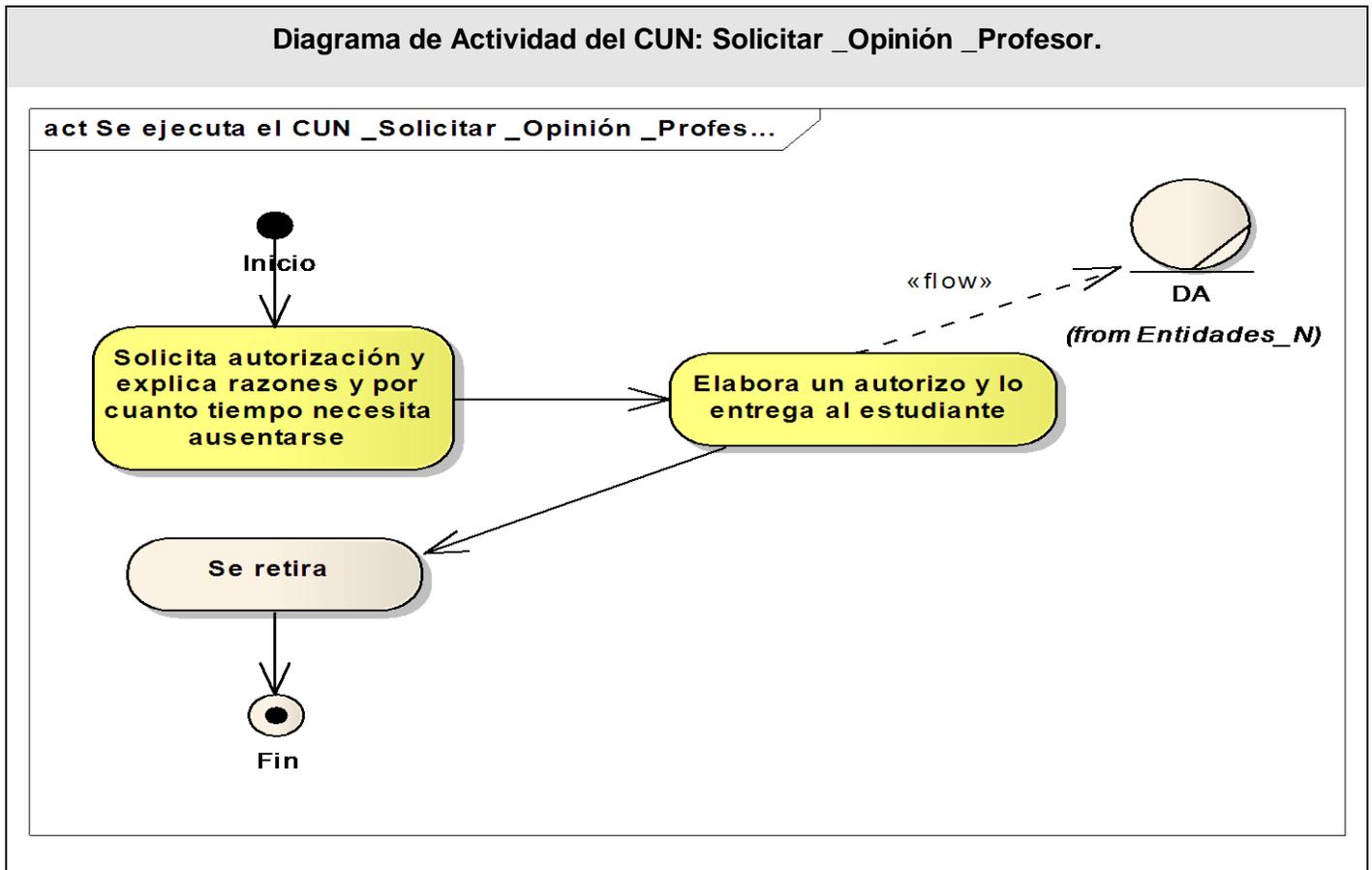


Figura 2.3 Diagrama de actividad del CUN: Solicitar _Opinión _Profesor.

2.3.5 Diagrama de Clases del Modelo de Objetos.

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe cómo colaboran los trabajadores y entidades del negocio dentro del flujo de trabajo del proceso del negocio.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

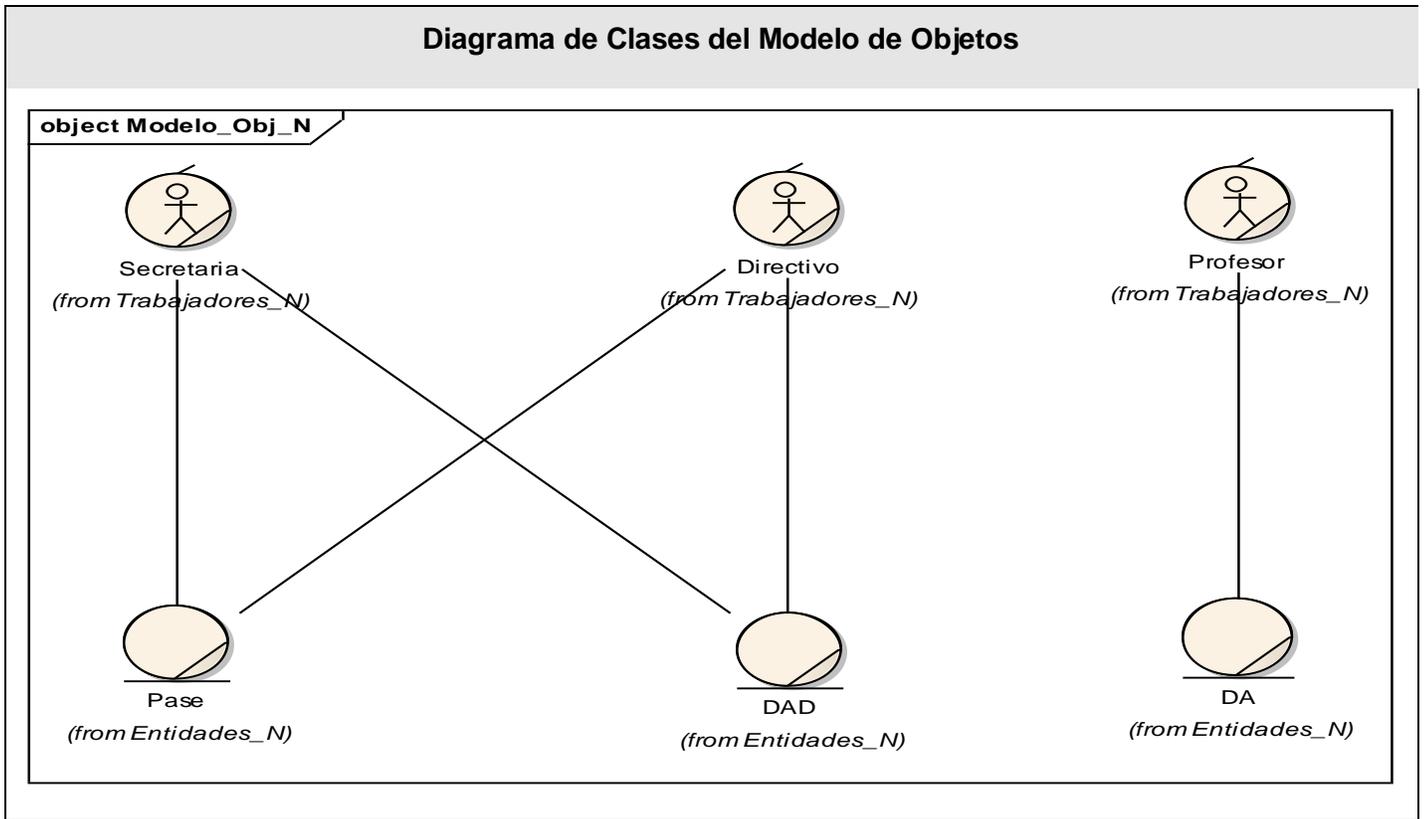


Figura 2.4 Diagrama de clases del modelo de objetos.

2.4 Especificación de los Requisitos de Software.

Los requisitos son las condiciones o capacidades que tienen que ser alcanzadas o poseídas por el sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Los requisitos son los que definen qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

2.4.1 Requerimientos Funcionales.

Los requisitos funcionales son condiciones o capacidades que debe alcanzar el software para satisfacer un contrato con los usuarios finales, estándar u algún otro documento que haya sido impuesto de manera formal. Los requisitos son los que definen lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones impuestas.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Para ello, se prevé que el sistema tenga las siguientes funcionalidades:

RF.1 Autenticar Usuario.

RF.1.1 Pedir nombre de usuario y contraseña para las opciones del sistema que lo requieran.

RF.1.2 Mandar directamente al usuario registrado a las opciones principales con que cuenta su nivel registrado.

RF.2 Solicitar Pase.

R.2.1 Permitir al estudiante realizar la solicitud de un pase no reglamentado explicando los motivos y por cuánto tiempo lo necesita.

RF.3 Notificar Llegada de pase del estudiante.

R.3.1 Permitir al estudiante informar su llegada de pase a la Universidad.

RF.4 Dar opinión de estudiante.

RF.4.1 Permitir al profesor dar su opinión sobre el estado del estudiante que solicita el pase.

RF.5 Ver Reportes.

RF.5.1 Reporte General.

RF.5.1.1 Dado un grupo, saber los estudiantes que se encuentran de pase.

RF.5.1.2 Dado un solapín, saber la cantidad de veces que ha salido de pase un estudiante.

RF.5.1.2 En un rango de fecha saber los estudiante que se encuentran de pase.

RF.5.2 Reporte Actual. Ver cantidad de estudiantes que se encuentran de pase en un determinado momento.

RF.5.3 Reporte por Causa. Ver los estudiantes que se encuentran de pase según la causa.

RF.6 Imprimir Pase.

RF.6.1 Permitir imprimir los pases aprobados según la elección, mediante una búsqueda o listándolos todos y mandándolos a imprimir.

RF.7 Otorgar Pase Urgente.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

RF.7.1 Permitir a los directivos de la Facultad 7 dar un pase urgente sin previa solicitud.

RF.7.2 Permitir elaborar el pase y mandarlo a imprimir al momento.

RF.8 Ver Solicitudes de Pase.

RF.8.1 Ver opiniones de los profesores en caso de que el estudiante tenga que ausentarse a clase o actividades productivas.

2.4.2 Requerimientos No Funcionales.

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto. Estas propiedades pueden verse como las características que hacen al producto atractivo, usable, rápido o confiable.

Se muestran a continuación, los siguientes requerimientos no funcionales que presentara la aplicación:

Apariencia o interfaz externa.

- Interfaz con pocas imágenes para no demorar las respuestas al usuario.
- Diseño sencillo y claro, con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus funcionalidades.
- Formal, serio y con una navegación sugerente, teniendo en cuenta el fin con el que se desarrolla la aplicación.

Usabilidad.

- El sistema puede ser usado por cualquier persona de la Facultad 7, que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.
- Instalar el sistema trae consigo una mayor rapidez de trabajo y por consiguiente un ahorro de materiales y personal.
- Cada usuario que se autentique sólo tendrá acceso a la información que le corresponde.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Rendimiento.

- Disponibilidad constante de trabajo en red contra el servidor.
- Rápido acceso de búsqueda a la información en tiempos relativamente cortos.
- El sistema deberá de ser lo más estable y confiable posible.
- El sistema debe soportar la conexión simultánea de más de 500 usuarios.

Soporte.

- Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.

Restricciones en el Diseño y la Implementación.

- La aplicación se implementará utilizando el lenguaje de programación PHP 5.2.5, Servidor Web Apache y como Gestor de Base de Datos: PostgreSQL.

Software.

- Para el funcionamiento del sistema en el servidor es necesario el S.O. Windows 98 o superior, Linux o Unix, en sus versiones de S.O. servidores.
- Para el funcionamiento del sistema en las terminales cliente es necesario el S.O. Windows 95 o superior, Linux o Unix.
- Los clientes tendrán acceso a la aplicación a través de cualquier navegador Web, recomendado Mozilla 1.5, Internet Explorer 5.0 o superior.

Hardware.

- Se necesitan como requerimientos mínimos una PC con procesador Pentium IV o superior.
- Impresora local o de red para imprimir los pases.

Portabilidad.

- El software podrá ser usado bajo los sistemas operativos Windows y Linux.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Seguridad.

- El sistema se encarga de controlar los diferentes niveles de acceso y roles de usuarios, garantiza que la información sea vista únicamente por quien tiene derecho a verla.
- Se hacen validaciones de la información tanto en el cliente como en el servidor, no obstante los usuarios acceden de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

Confidencialidad.

- Toda la información está protegida del acceso no autorizado.

Disponibilidad.

- Se garantiza a los usuarios del sistema el acceso a la información solicitada en todo momento (si tiene permiso para ello).

Confiabilidad.

- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.
- Requisitos para la documentación de usuarios en línea y ayuda del sistema.
- Mediante una opción de ayuda, disponer de instrucciones que sirvan de guía para el usuario.
- Mediante alguna metodología de desarrollo, disponer de una documentación del sistema.

2.5 Descripción del Sistema Propuesto.

El sistema propuesto automatiza el proceso de pases no reglamentados de los estudiantes de la Facultad 7 de la UCI, haciendo uso del gestor de base de datos PostgreSQL con el propósito de acelerar los tiempos de respuesta cuando los volúmenes de datos sean altos, así como la concurrencia a los mismos. Se destaca además por su importancia, el uso de herramientas de Software Libre, y se reafirma una vez más las ventajas en comparación con otros sistemas, debido a que esta práctica es gratuita y los requerimientos de hardware son relativamente bajos. Esta propuesta es una aplicación que se integrara al portal de la Facultad siendo una funcionalidad más de las que la misma ya presenta en la actualidad.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

La aplicación puede ser usada por el estudiante con la finalidad de solicitar un pase no reglamentado durante la semana y poder obtener una respuesta en el menor tiempo posible mediante la misma; la cual es emitida por los directivos de la Facultad, ya que éstos son los únicos autorizados a dar éste tipo de pase después de haber analizado dichas solicitudes, que contienen los motivos y el tiempo por el cual el estudiante solicita el pase y las opiniones de los profesores en caso de que dicho estudiante tenga afectaciones docentes. La base de datos contendrá la estructura apropiada para cumplir toda la gestión de estas solicitudes.

2.6 Modelo de Casos de Usos del Sistema.

Mediante las facilidades que brinda el UML, se capturan los requisitos funcionales del sistema, y se representan mediante un diagrama de casos de uso; para lo cual se define de acuerdo a lo planteado anteriormente, cuáles serán los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

2.6.1 Definición de los Actores.

Un actor del sistema no es más que otras personas u otros sistemas que eran trabajadores en el negocio y que ahora interactúan de alguna forma con el sistema. Se definen a continuación los siguientes actores:

Actores del Sistema	Descripción
Usuario	Representa un actor generalizado que puede comportarse como Estudiante, Directivo o Secretaria.
Estudiante	Representa a una persona que va a utilizar el sistema para hacer una solicitud de pase y si éste es aprobado también tiene que utilizarla para notificar la llegada a la Universidad de dicho pase.
Directivo	Representa a una persona autorizada a realizar todas las acciones sobre las solicitudes (aprobarlas o rechazarlas y firmarlas). Puede consultar además todos los reportes referentes a los pases que se han dado en la Facultad.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Profesor	Representa a una persona que va a utilizar el sistema para emitir su opinión sobre el estado del estudiante que solicita pase.
Secretaria	Representa a una persona que también puede consultar todos los reportes referentes a los pases que se han dado en la Facultad e imprime los pases otorgados.

Tabla 2.5 Justificación de los trabajadores del sistema.

2.7 Diagrama de Casos de Uso del Sistema.

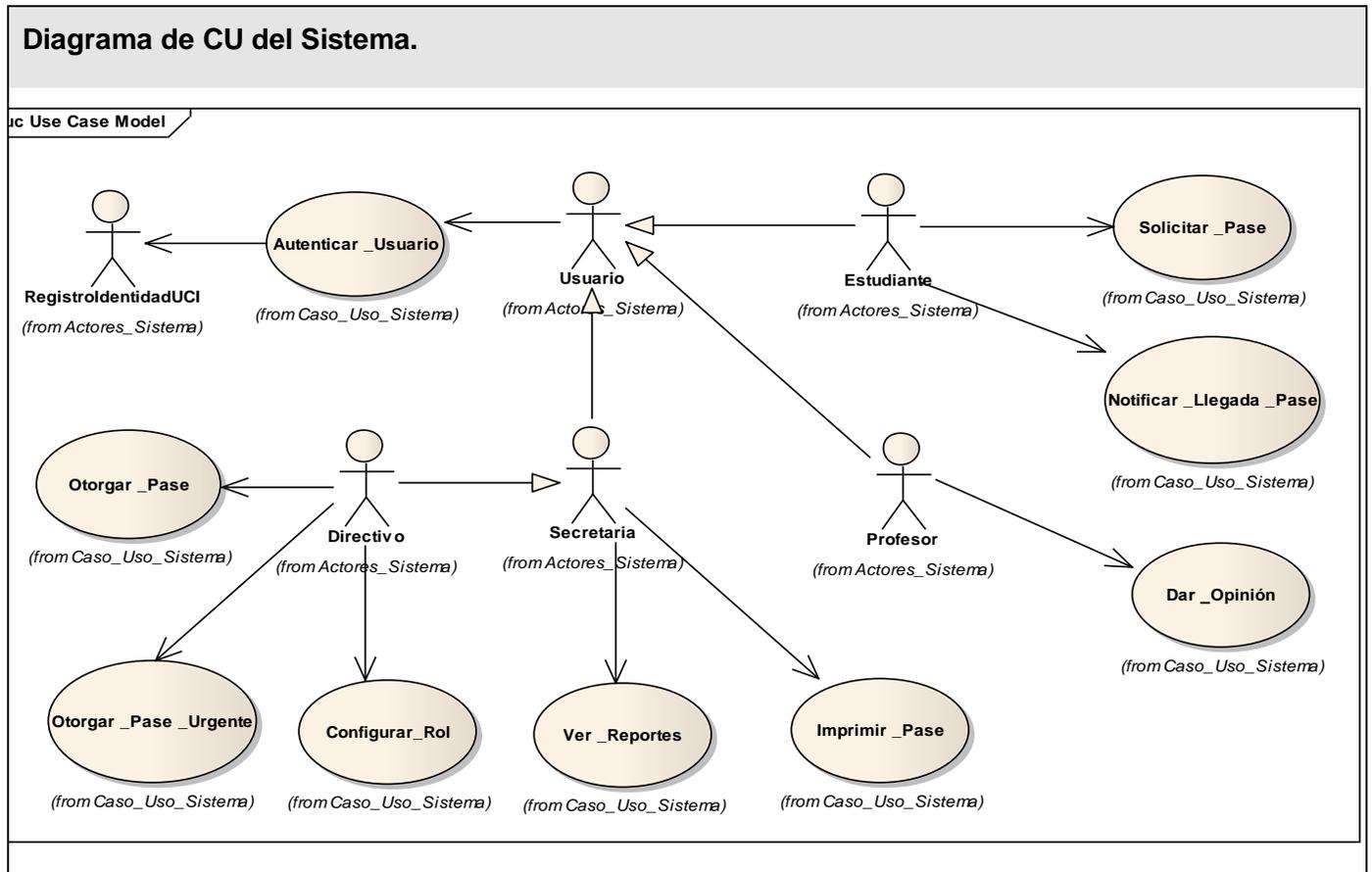


Figura 2.5 Diagrama de CU del Sistema.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.8 Especificación de los Casos de Uso del Sistema.

Mediante la especificación de los casos de uso del sistema se describe paso a paso la secuencia de eventos que los actores realizan para completar un proceso a través del sistema; constituyendo éste, el último paso en el análisis de los procesos, pasando luego, a la construcción de la solución propuesta.

2.8.1 Descripción del Caso de Uso del Sistema: Solicitar _Pase.

Caso de Uso:	Solicitar _Pase.	
Actores:	Estudiante.	
Resumen:	El CU se inicia cuando un estudiante accede al sistema y llena y envía la solicitud de pase, concluyendo el CU.	
Precondiciones:	Debe estar autenticado.	
Referencias	RF.2	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El estudiante llena la solicitud de pase y la envía.	2. El sistema valida los datos entrados por el estudiante.	
	3. El sistema archiva la solicitud para pendiente aprobación.	
Prototipo de Interfaz		
Flujos Alternos		

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Acción del Actor	Respuesta del Sistema
Línea 2. Datos incompletos o incorrectos.	2.1 El sistema emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos. Retorna al paso 3.
Prototipo de Interfaz	
Pos condiciones	

Tabla 2.6 CU: Solicitar _Pase.

2.8.2 Descripción del Caso de Uso del Sistema: Notificar _Llegada _Pase.

Caso de Uso:	Notificar _Llegada _Pase.
Actores:	Estudiante.
Resumen:	El CU se inicia cuando un estudiante accede al sistema e informa sobre su llegada de pase a la Universidad, terminando así el CU.
Precondiciones:	Debe estar autenticado.
Referencias	RF.3
Prioridad	Secundario.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El estudiante marca la opción de Notificar Llegada de Pase.	2. El sistema le da la posibilidad de seleccionar la fecha de llegada de pase.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

3. El estudiante selecciona la fecha de llegada y la manda a guardar.	4. El sistema archiva la fecha de notificación.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prototipo de Interfaz	
Pos condiciones	

Tabla 2.7 CU: Notificar _Llegada _Pase.

2.8.3 Descripción del Caso de Uso del Sistema: Autenticar _Usuario.

Caso de Uso:	Autenticar _Usuario.
Actores:	Usuario.
Resumen:	El CU se inicia cuando el usuario ingresa sus datos en el sistema (nombre de usuario y contraseña de dominio UCI). Se le informa si los mismos son correctos o no y se le da acceso a los lugares donde tiene permisos, terminando de esta forma el CU.
Precondiciones:	
Referencias	RF.1
Prioridad	Critico.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Usuario introduce su usuario y contraseña.	2. El sistema valida los datos entrados por el usuario.
	3. El sistema verifica que el usuario exista en su BD.
	4. El sistema permite el acceso del usuario registrado según los permisos que tenga.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Línea 2. Datos incompletos o incorrectos.	2.1 El sistema emite un mensaje de error. Retorna a paso 1.
Línea 3. El usuario no existe en la BD del sistema.	3.1 El sistema no permite el acceso del usuario a la aplicación.
Prototipo de Interfaz	
Pos condiciones	El usuario accede sólo donde tenga permiso.

Tabla 2.8 CU: Autenticar _Usuario.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.8.4 Descripción del Caso de Uso del Sistema: Dar _Opinión.

Caso de Uso:	Dar _Opinión.	
Actores:	Profesor.	
Resumen:	El CU se inicia cuando un profesor accede al sistema, consulta la solicitud de pase del estudiante y emite su opinión y la envía dándole fin al CU.	
Precondiciones:	El profesor debe recibir un aviso para que entre a la aplicación.	
Referencias	RF.4	
Prioridad	Secundario.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El profesor consulta la solicitud de pase del estudiante, emite su opinión y la envía.	2. El sistema archiva la opinión para ser consultada por los directivos.	
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
Prototipo de Interfaz		
Pos condiciones		

Tabla 2.9 CU: Dar _Opinión.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.8.5 Descripción del Caso de Uso del Sistema: Imprimir _Pase.

Caso de Uso:	Imprimir_ Pase.
Actores:	Autorizado(Directivo o Secretaria)
Resumen:	El CU se inicia cuando los autorizados acceden al sistema y seleccionan la opción Imprimir Pase. El sistema muestra los criterios de búsqueda y según la fecha seleccionada por el actor, se muestran los pases a imprimir en ese rango, pero antes se mandan a generar a PDF y seguidamente se mandan a imprimir, terminando así el CU.
Precondiciones:	Debe estar autenticado.
Referencias	RF.6
Prioridad	Critico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El autorizado selecciona la opción de Imprimir Pase.	2. El sistema muestra una interfaz que brinda la posibilidad de seleccionar la fecha de los pases que se quieren imprimir en ese rango.
3. El autorizado selecciona el rango de fecha.	4. El sistema muestra una interfaz que contiene los pases a imprimir.
5. El autorizado selecciona la opción Generar a PDF.	6. EL sistema genera a PDF.
7. El autorizado manda a imprimir el PDF.	8. El sistema imprime los pases del PDF.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Línea 7	8.1 El sistema muestra un mensaje que diga que no encuentra dispositivo de impresión.
<i>Prototipo de Interfaz</i>	
Pos condiciones	

Tabla 2.10 CU: Imprimir _Pase.

2.8.6 Descripción del Caso de Uso del Sistema: Otorgar _Pase _Urgente.

Caso de Uso:	Otorgar _Pase _Urgente.
Actores:	Directivo.
Resumen:	El CU se inicia cuando un directivo accede al sistema para aprobar o rechazar una solicitud de pase con carácter urgente. Cada vez que se aprueba una nueva solicitud es archivada en el sistema para poder generar posteriormente los reportes correspondientes.
Precondiciones:	Debe estar autenticado.
Referencias	RF.7
Prioridad	Critico.
Flujo Normal de Eventos	

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Acción del Actor	Respuesta del Sistema
1. El directivo accede a su sesión de trabajo y selecciona la opción Otorgar Pase Urgente.	2. El sistema muestra una interfaz que contiene un formulario que el directivo tiene que llenar con los datos que requiere el pase.
3. El directivo llena el formulario con los datos requeridos y lo envía.	4. El sistema valida los datos entrados por el directivo.
	5. El sistema archiva la solicitud con el estado de aprobada.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Línea 4. Datos incompletos o incorrectos.	4.1 El sistema emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos. Retorna al paso 3.
Prototipo de Interfaz	
Pos condiciones	

Tabla 2.11 CU: Otorgar _Pase _Urgente.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.8.7 Descripción del Caso de Uso del Sistema: Otorgar _Pase.

Caso de Uso:	Otorgar _Pase.
Actores:	Directivo.
Resumen:	El CU se inicia cuando el directivo accede a la aplicación y selecciona la opción Solicitudes Realizadas. El sistema le muestra todas las solicitudes existentes que contienen los datos de los estudiantes, las razones por las cuales solicita el pase y las opiniones de los profesores más la opción de que el directivo pueda emitir su criterio del estado del pase; finalizando así, el CU.
Precondiciones:	
Referencias	RF.8
Prioridad	Critico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El directivo selecciona la opción Solicitudes Realizadas.	2. El sistema muestra una interfaz que contiene el listado con las solicitudes realizadas.
3. El directivo selecciona una solicitud para consultar.	4. El sistema muestra una interfaz que contiene la solicitud seleccionada con los datos requeridos en la misma.
5. El directivo analiza la solicitud y consulta la opinión de los profesores en caso de que exista.	
6. El directivo aprueba la solicitud.	7. El sistema archiva la solicitud con el estado de

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

	aprobada.
	8. El sistema envía correo de notificación de estado de solicitud a estudiante y profesor.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Línea 6 El directivo rechaza la solicitud y emite criterio. Retornar al paso 8.	
Prototipo de Interfaz	
Pos condiciones	Es mostrado el listado con todas las solicitudes existentes.

Tabla 2.12 CU: Otorgar _Pase.

2.8.8 Descripción del Caso de Uso del Sistema: Ver _Reportes.

Caso de Uso:	Ver _Reportes.
Actores:	Autorizado(Directivo o Secretaria)
Resumen:	El CU se inicia cuando los autorizados acceden al sistema y seleccionan la opción Ver Reportes. El sistema muestra los criterios de búsqueda y según la fecha seleccionada por el actor, se muestra el tipo de reporte deseado y la opción de poder imprimirlos, terminando así el CU.
Precondiciones:	Debe estar autenticado.
Referencias	RF.5

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

Prioridad	Opcional.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El autorizado selecciona la opción Ver Reportes.	2. El sistema muestra una interfaz que brinda la posibilidad de seleccionar la fecha desde hasta cuando quiere obtener los reportes.
3. El autorizado selecciona el rango de fecha.	4. El sistema muestra una interfaz que contiene los datos estadísticos del reporte y una opción que le permite mandarlo a imprimir si lo desea.
5. El autorizado manda a imprimir el reporte si lo desea.	6. EL sistema imprime el reporte.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Línea 6	6.1 El sistema muestra un mensaje que diga que no encuentra dispositivo de impresión.
Prototipo de Interfaz	
Pos condiciones	

Tabla 2.13 CU: Ver _Reportes.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

2.8.9 Descripción del Caso de Uso del Sistema: Configurar _Rol.

Caso de Uso:	Configurar _Rol.
Actores:	Directivo.
Resumen:	El CU se inicia cuando un directivo accede al sistema y selecciona la opción Configurar Rol. El sistema muestra las 2 opciones posibles y finaliza el CU cuando el directivo asigna o elimina un rol.
Precondiciones:	Debe estar autenticado.
Referencias	
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El directivo selecciona la opción Configurar Rol.	2. El sistema muestra una interfaz que brinda la posibilidad de Asignar o Eliminar el rol que asumirá un determinado usuario en la aplicación.
3. El directivo selecciona una de las dos opciones: a) Asignar Rol, ir a la sección Asignar Rol. b) Eliminar Rol, ir a la sección Eliminar Rol.	
Sección Asignar Rol	
Acción del Actor	Respuesta del Sistema
.	4. El sistema muestra una interfaz que da la

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

	posibilidad de seleccionar el nuevo rol (Directivo o secretaria).
5. El directivo selecciona el nuevo rol y acepta.	6. El sistema verifica si el usuario existe.
	7. El sistema muestra un mensaje confirmando que se ha realizado correctamente la operación.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
Línea 6. El usuario no existe.	6.1 El sistema muestra un mensaje de error.
Sección Eliminar Rol	
Acción del Actor	Respuesta del Sistema
5. El directivo elimina el rol que tiene asignado	6. El sistema actualiza la información del usuario y acepta.
Pos condiciones	

Tabla 2.14 CU: Configurar _Rol.

Conclusiones

En el presente capítulo se inició el desarrollo de la solución propuesta como finalidad de éste trabajo, donde se obtuvo, mediante el análisis profundo de los procesos del negocio, todas las funciones que el sistema debe tener para su funcionamiento. Los mismos se representaron a través de un Diagrama de Casos de Uso, y se describieron todas las acciones de los actores del sistema con los casos de uso correspondientes para finalmente comenzar a construir el sistema siguiendo todos los requerimientos establecidos en éste capítulo.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

Capítulo III: Análisis y Diseño del Sistema.

En éste capítulo se traducen los requisitos a una especificación que describe como realizar la implementación del sistema. Los componentes de la aplicación se tratan como clases y se representan las relaciones entre ellas. Se tratan algunos principios de diseño, estándares de codificación a seguir y se presentan por último, los diagramas de despliegue e implementación, donde se definen los componentes que forman parte de la aplicación y los nodos que forman la topología de hardware, que soportará al sistema.

3.1 Modelo de Análisis.

El modelo de análisis es un modelo conceptual que realiza un trazado de cómo llevar a cabo la funcionalidad dentro de la aplicación e incluye la funcionalidad significativa para la arquitectura del software. Es descrito por el lenguaje del desarrollador y proporciona la estructura a la vista interna del sistema. En su construcción deben ser identificadas las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con toda esta información es construido el diagrama de clases del análisis. [17]

3.1.1 Diagrama de Clases del Análisis.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos de un dominio del problema, por lo que representa a las cosas del mundo real y no a la implementación automatizada de las aplicaciones. [18]

Se representan a continuación los diagramas de clases del análisis para cada uno de los casos de usos que comprende la aplicación.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1.1.1 Diagrama de Clases del CU: Solicitar Pase.

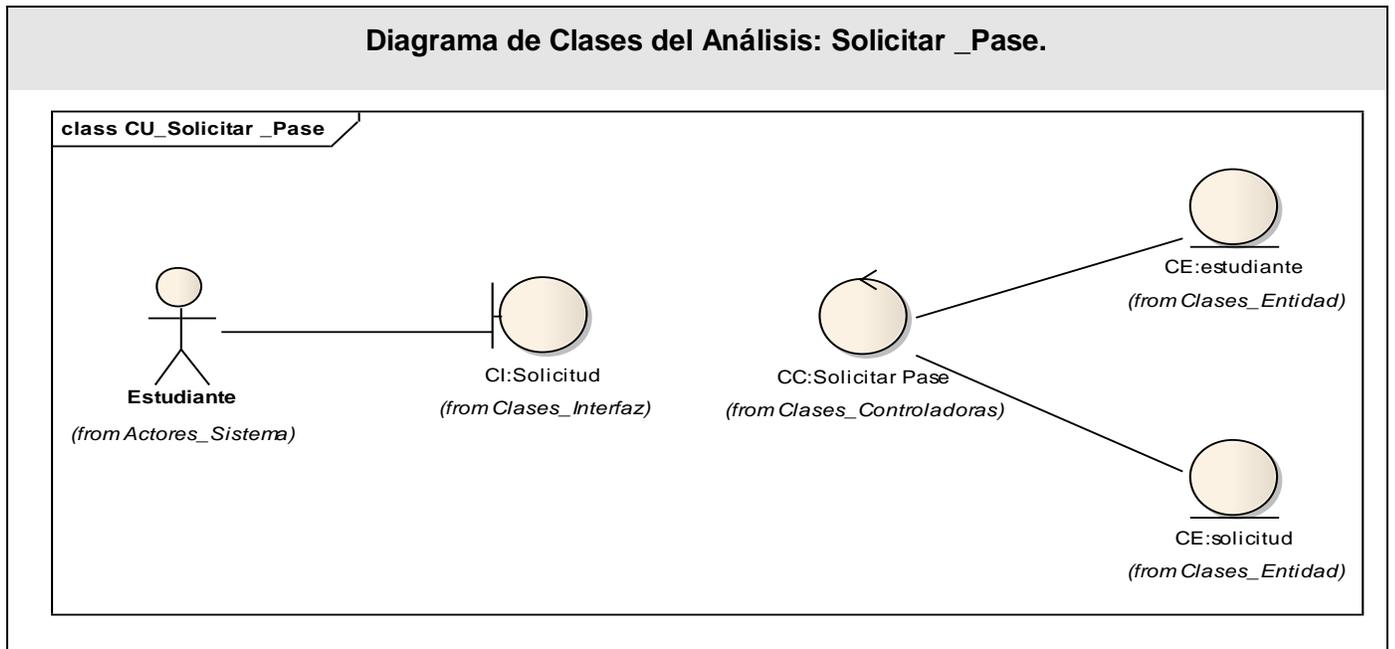


Figura 3.1 DC del análisis: Solicitar Pase.

3.1.1.3 Diagrama de Clases del CU: Dar Opinión.

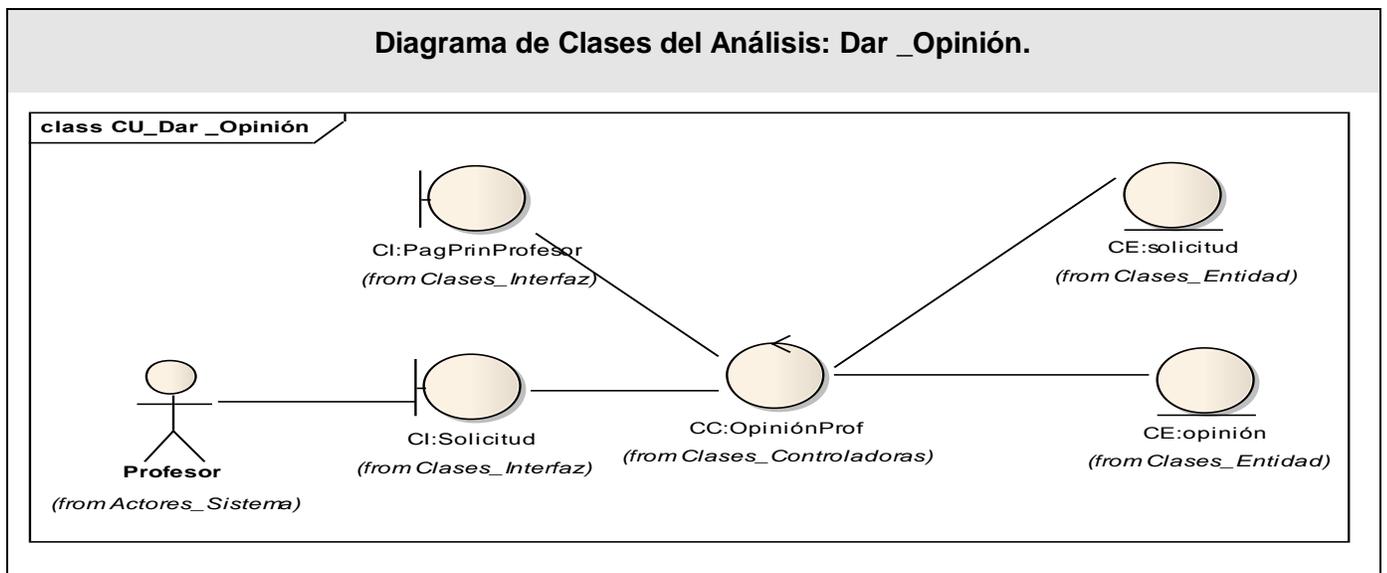


Figura 3.2 DC del análisis: Dar Opinión.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1.1.2 Diagrama de Clases del CU: Notificar Llegada Pase.

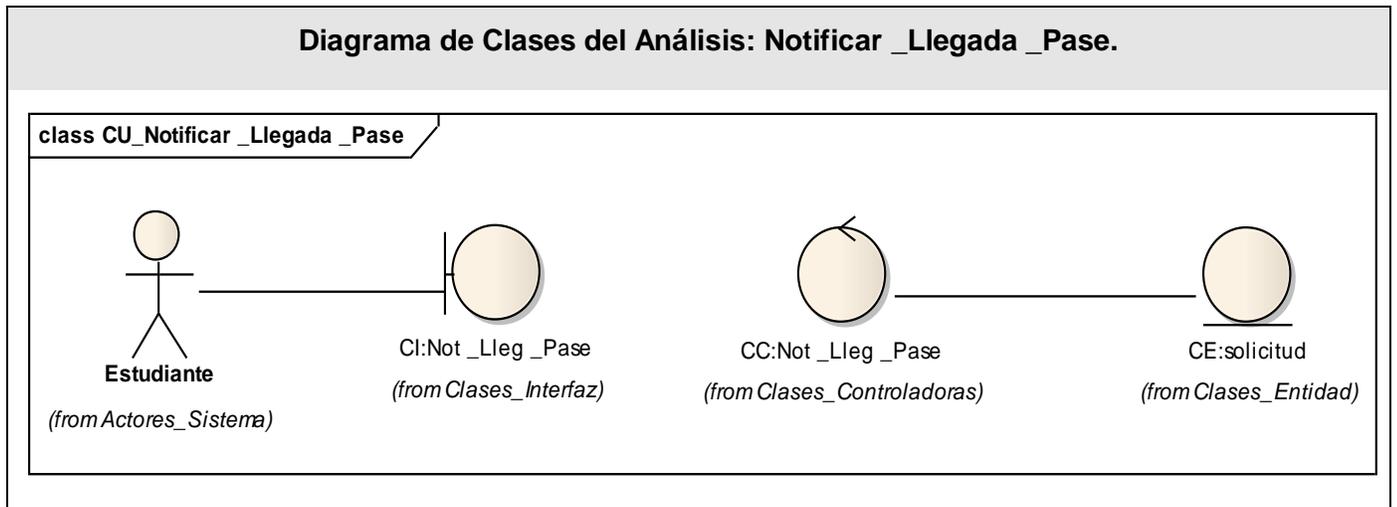


Figura 3.3 DC del análisis: Notificar Llegada Pase.

3.1.1.4 Diagrama de Clases del CU: Imprimir Pase.

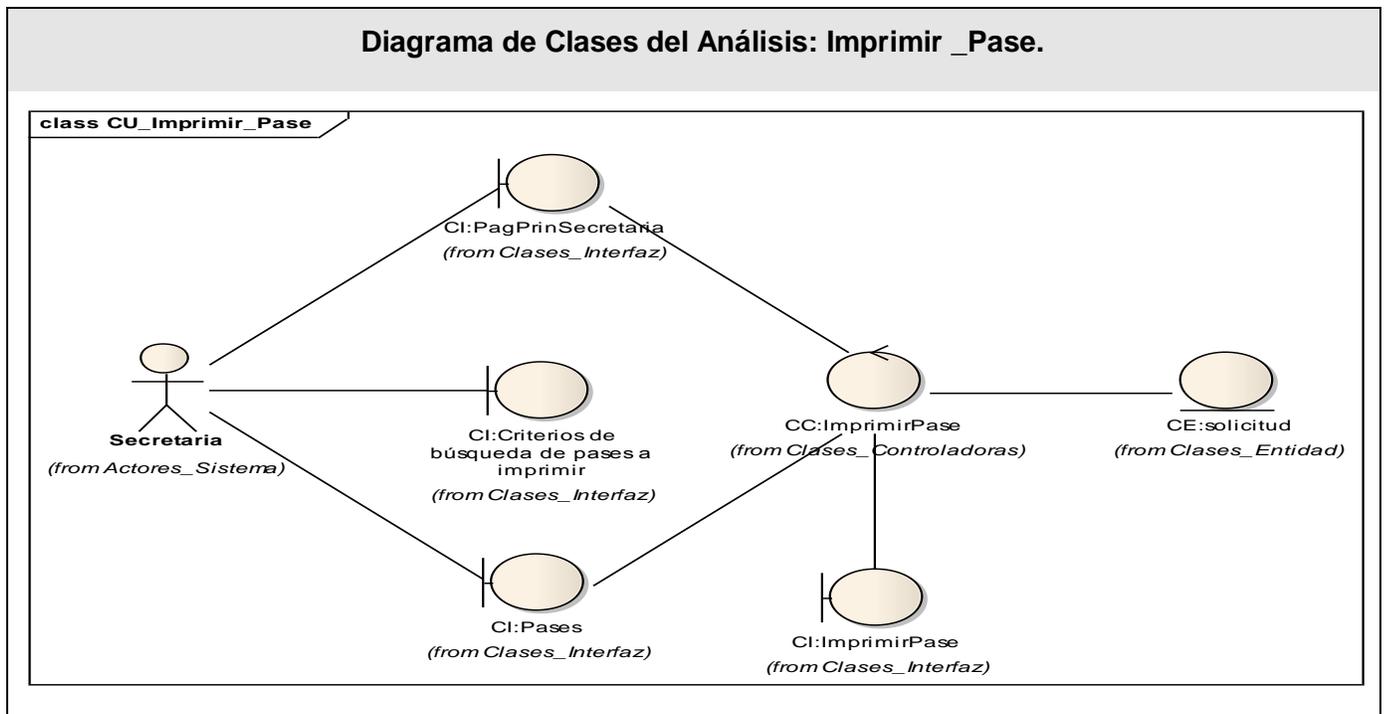


Figura 3.4 DC del análisis: Imprimir Pase.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1.1.5 Diagrama de Clases del CU: Otorgar Pase Urgente.

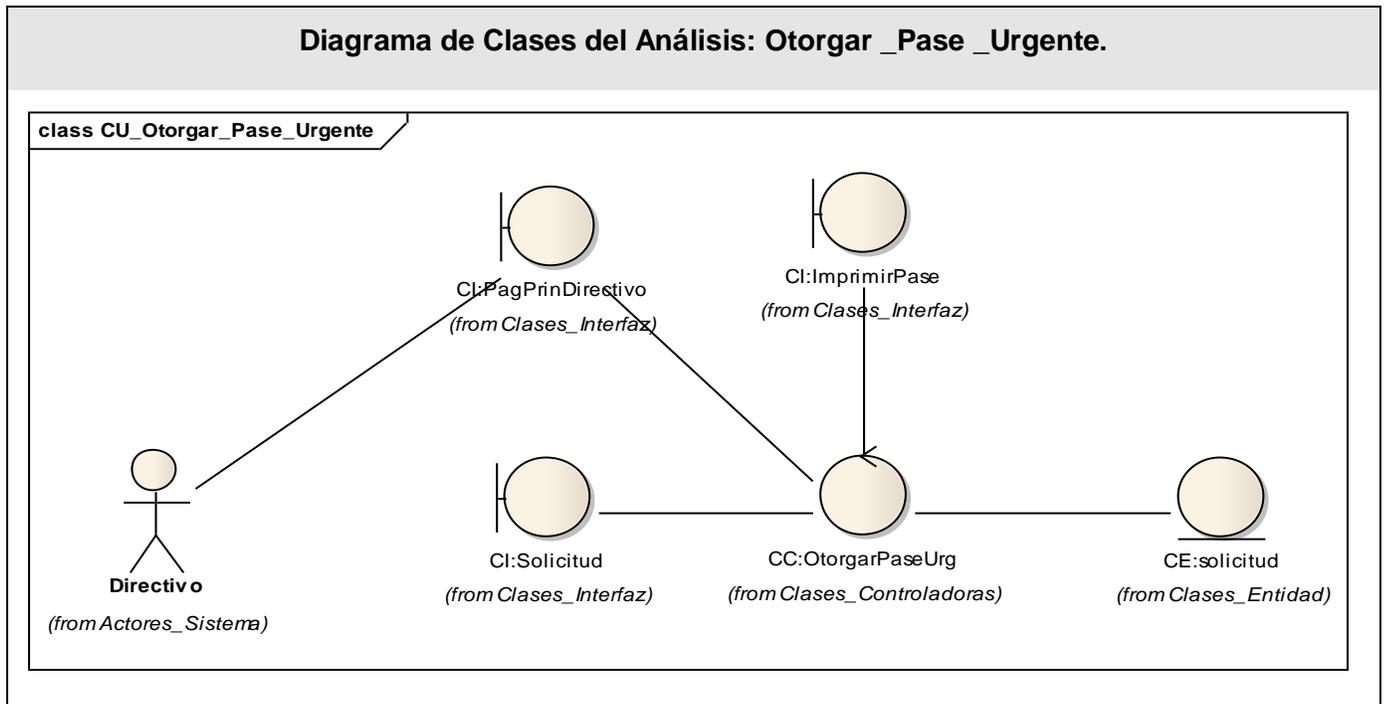


Figura 3.5 DC del análisis: Otorgar Pase Urgente.

3.1.1.8 Diagrama de Clases del CU: Autenticar Usuario.

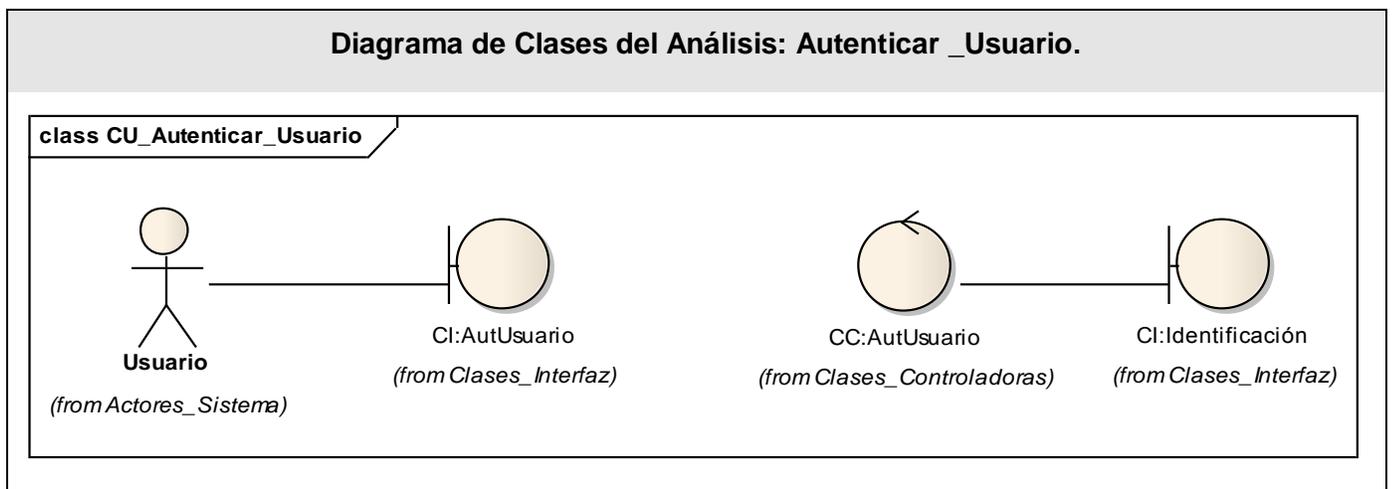


Figura 3.6 DC del análisis: Autenticar Usuario.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1.1.6 Diagrama de Clases del CU: Otorgar Pase.

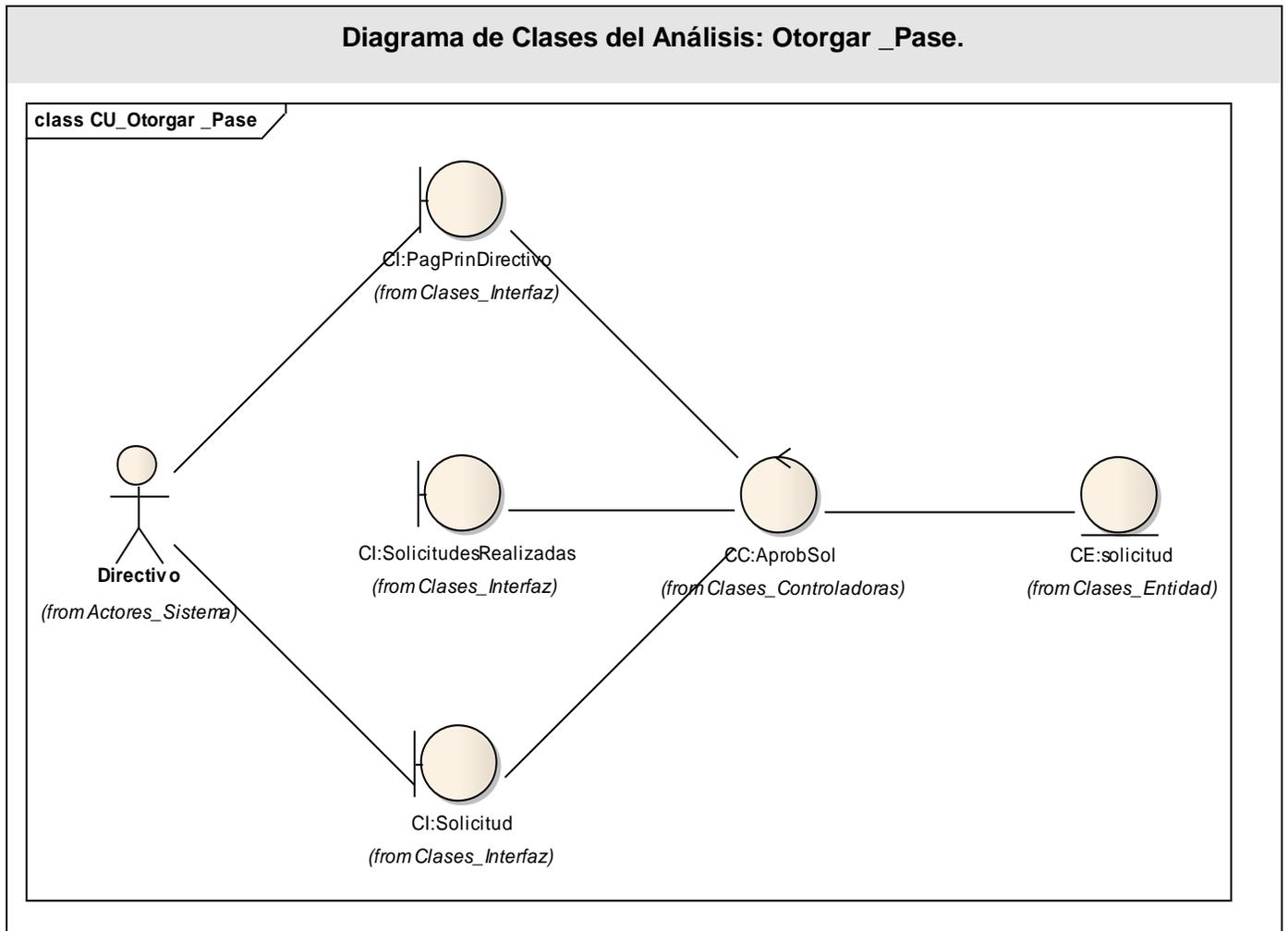


Figura 3.7 DC del análisis: Otorgar Pase.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1.1.7 Diagrama de Clases del CU: Ver Reportes.

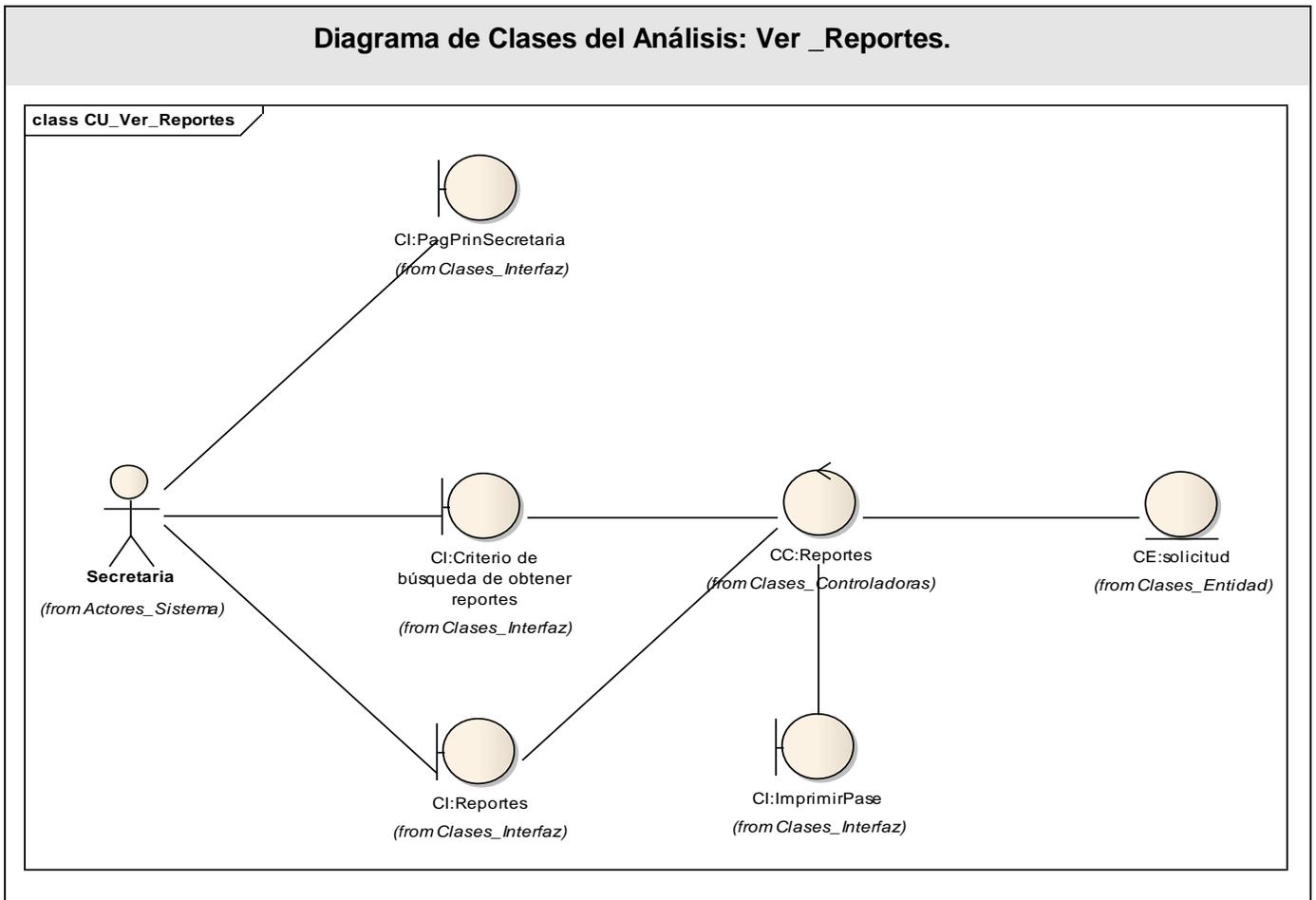


Figura 3.8 DC del análisis: Ver Reportes.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1.1.9 Diagrama de Clases del CU: Configurar Rol.

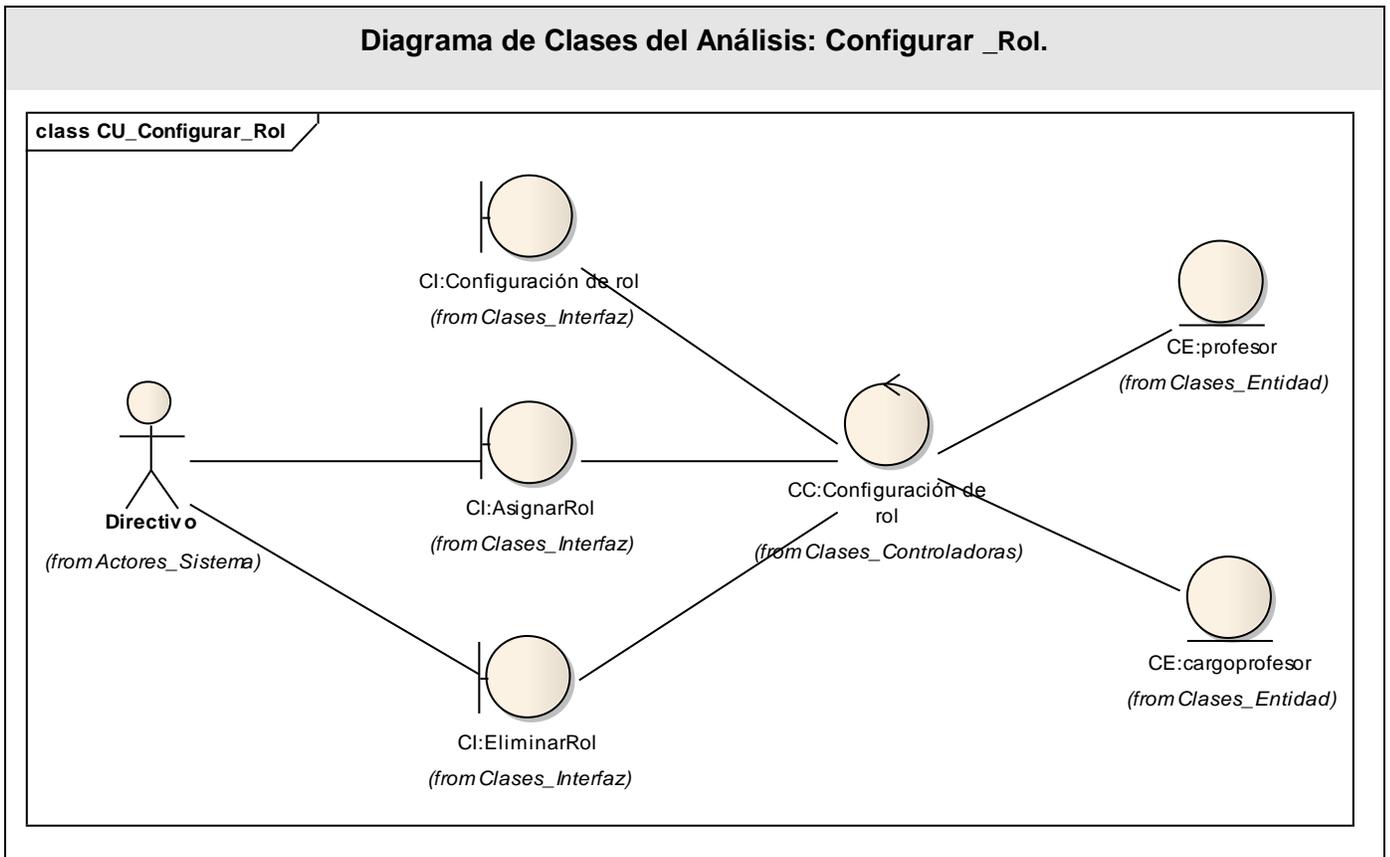


Figura 3.9 DC del análisis: Configurar Rol.

3.2 Modelo de Diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Lo que contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación.

Una entrada esencial en el diseño es el resultado del análisis que proporciona una comprensión detallada de los requisitos. Además, impone una estructura del sistema que se debe conservar lo más fielmente posible cuando se le dé forma al sistema.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

En el modelo del diseño, los casos de uso son realizados por las clases del diseño y sus objetos. Esto se representa por colaboraciones en el modelo de diseño y denota la realización de casos de uso del diseño.

3.2.1 Diagrama de Clases del Diseño Web.

El diagrama de clases del diseño web, no se realiza de igual forma a la tradicional de modular clases, éste difiere un tanto del resto de las aplicaciones que se construyen en que en ellas, en las aplicaciones web, es más importantes la modelación de la lógica y el estado del negocio, que los detalles de presentación. Es por ello que estos diagramas de clases presentan las páginas, los enlaces entre ellas y todo el código que irán creando, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente.

Estos diagramas de clases Web, son definidos a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para Web. A continuación se muestran los diagramas de clases para los distintos casos de uso.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.1 Diagrama de Clases del CU: Solicitar Pase.

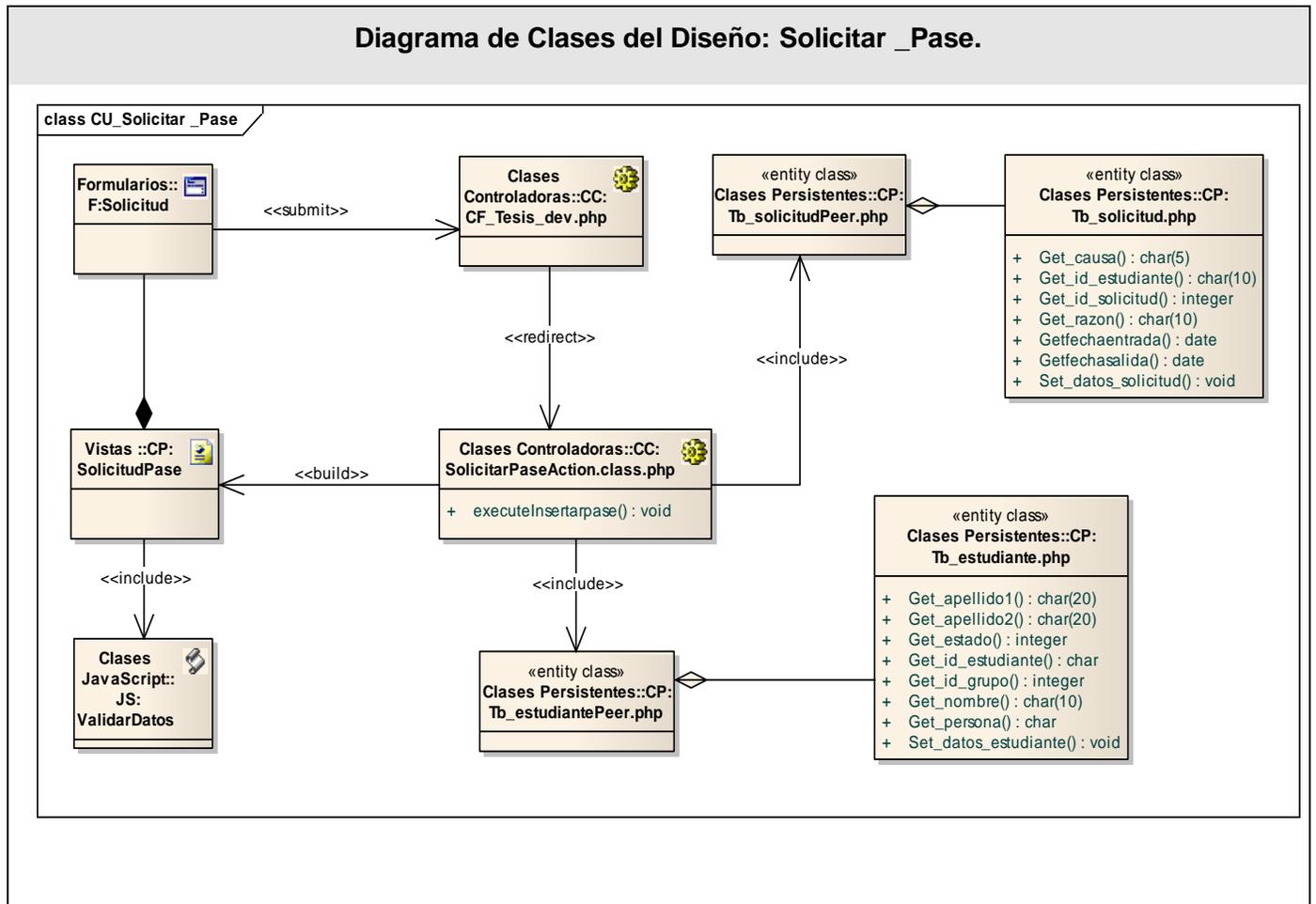


Figura 3.10 DC del diseño: Solicitar Pase.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.2 Diagrama de Clases del CU: Notificar Llegada Pase.

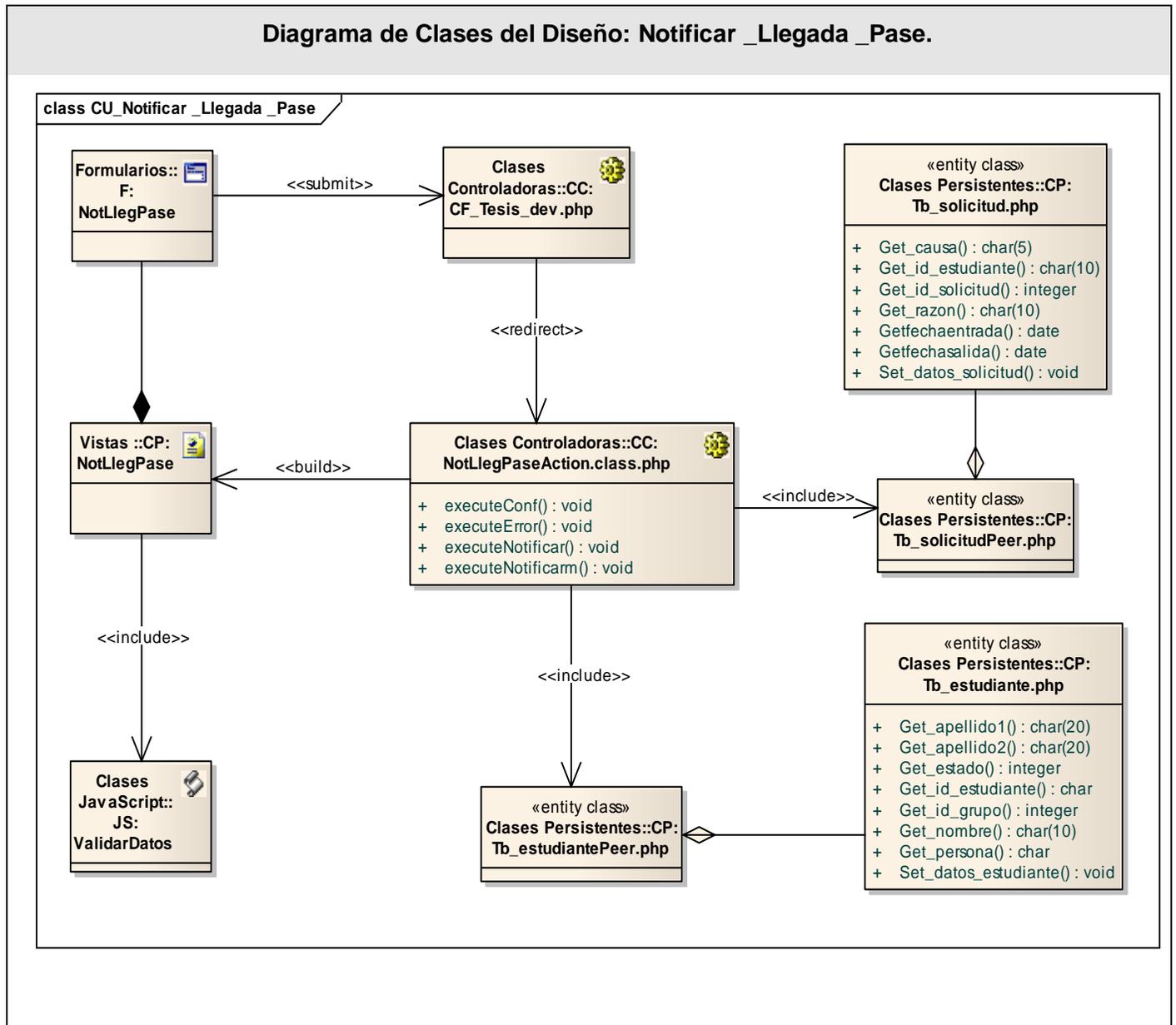


Figura 3.11 DC del diseño: Notificar Llegada Pase.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.4 Diagrama de Clases del CU: Imprimir Pase.

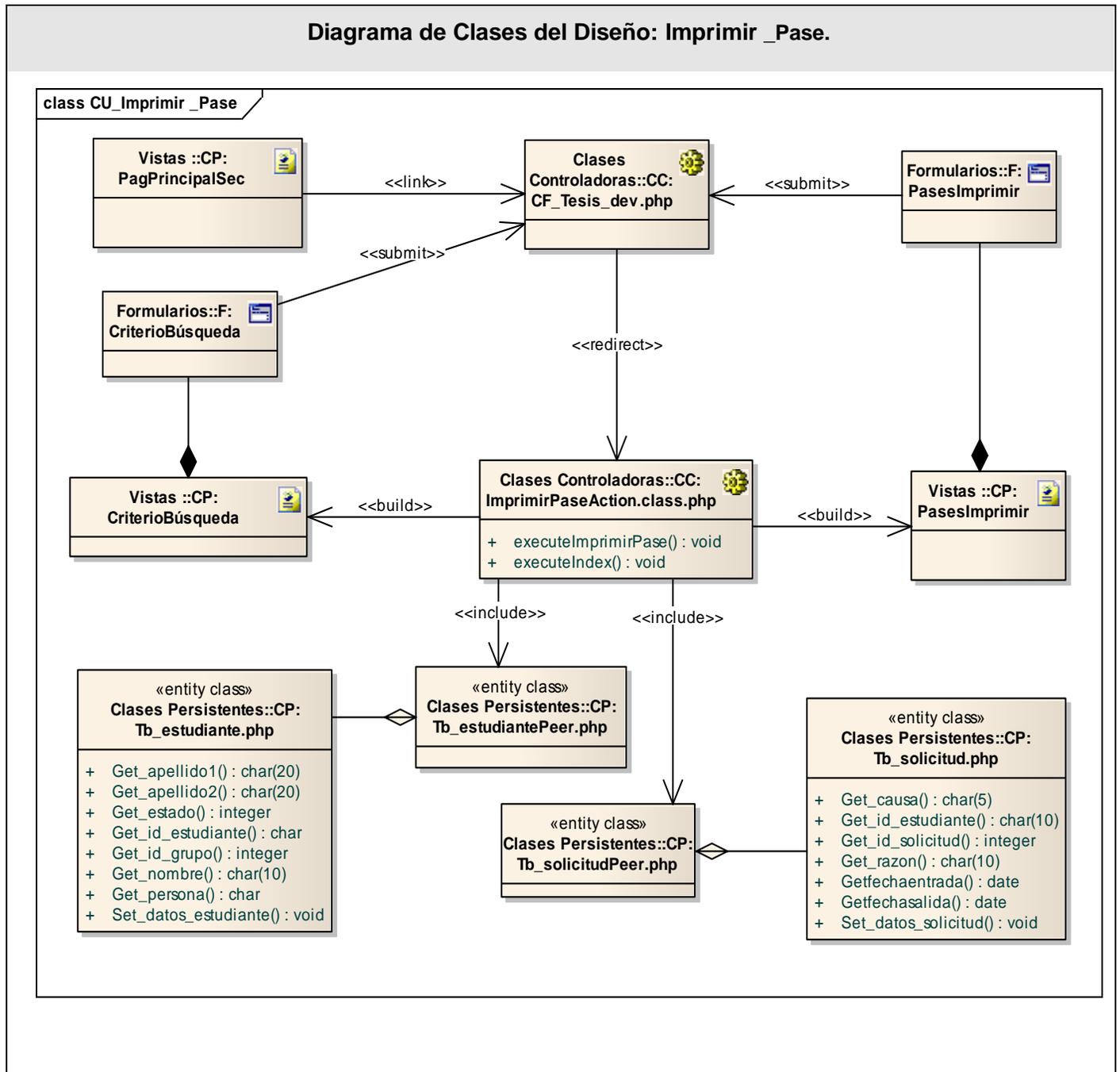


Figura 3.13 DC del diseño: Imprimir Pase.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.5 Diagrama de Clases del CU: Otorgar Pase Urgente.

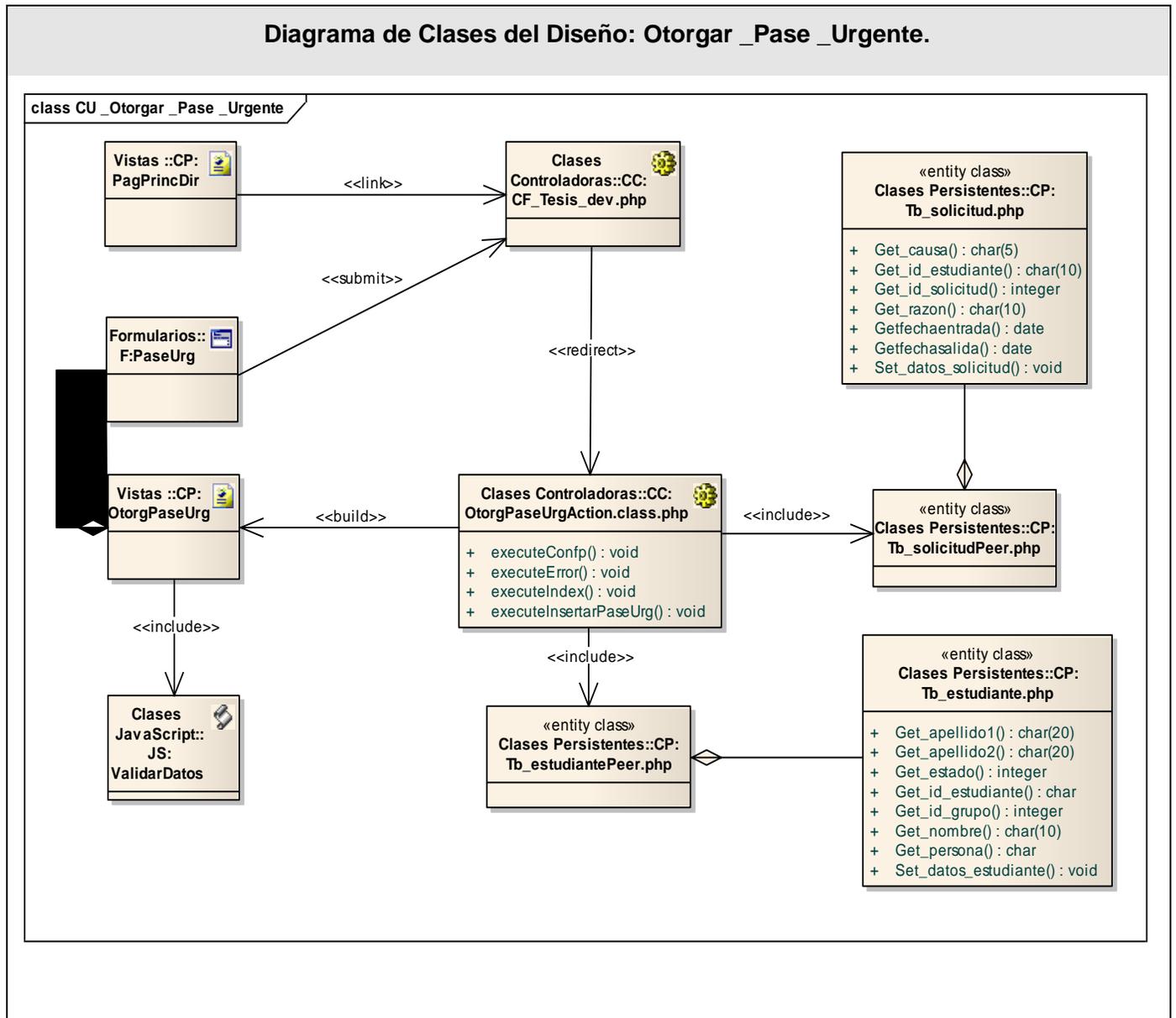


Figura 3.14 DC del diseño: Otorgar Pase Urgente.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.6 Diagrama de Clases del CU: Otorgar Pase.

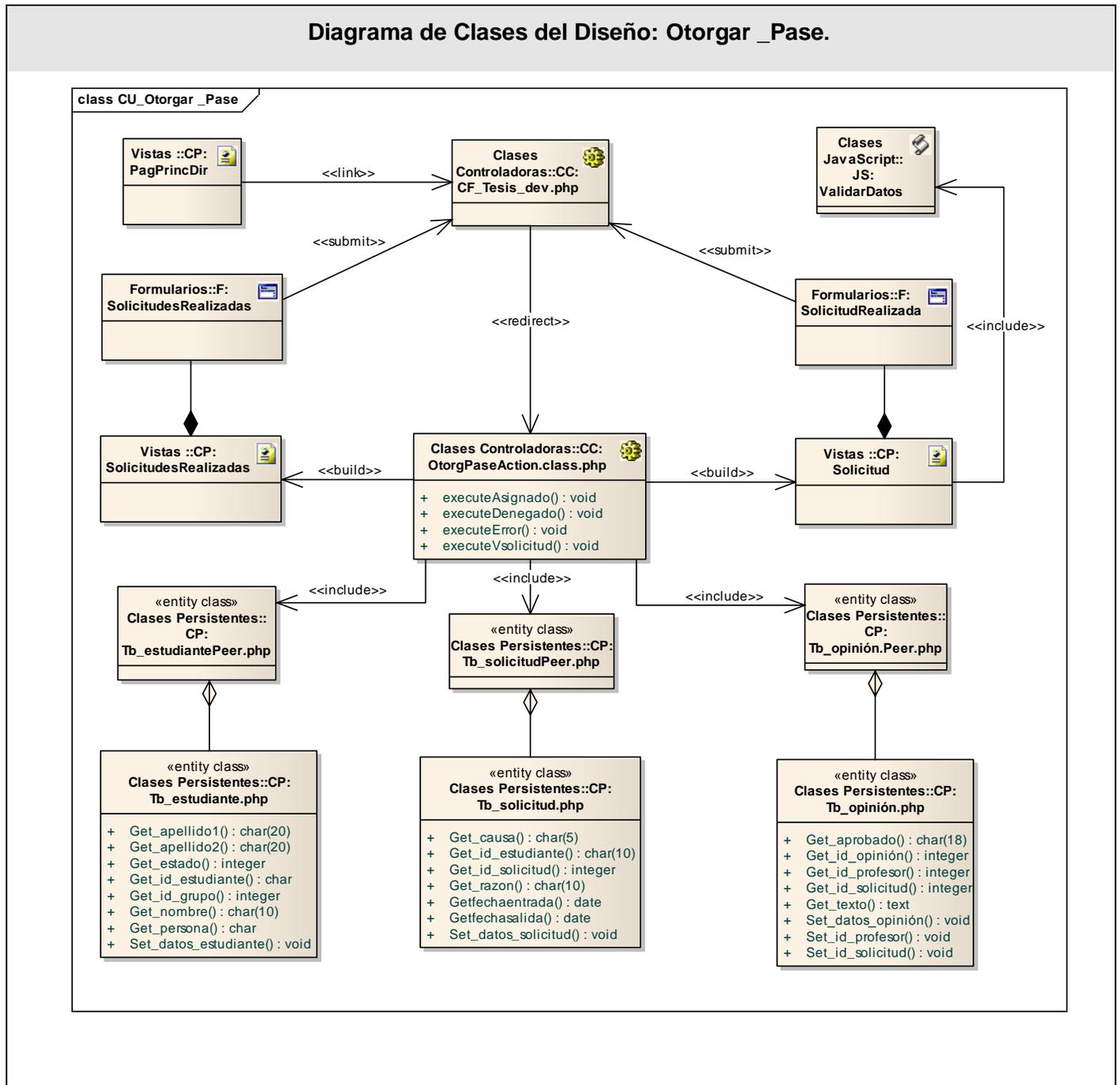


Figura 3.15 DC del diseño: Otorgar Pase.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.7 Diagrama de Clases del CU: Ver Reportes.

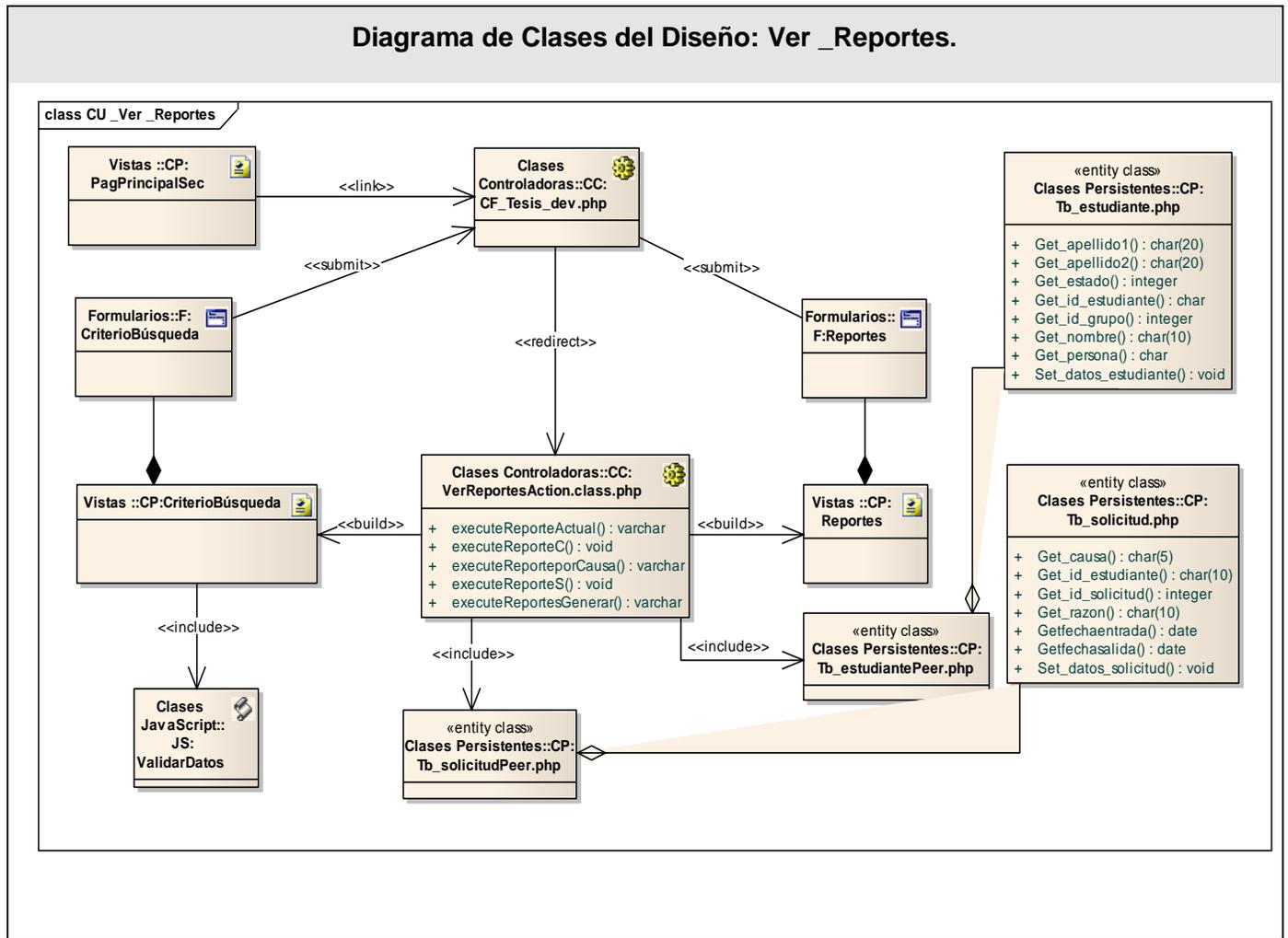


Figura 3.16 DC del diseño: Ver Reportes.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.8 Diagrama de Clases del CU: Autenticar Usuario.

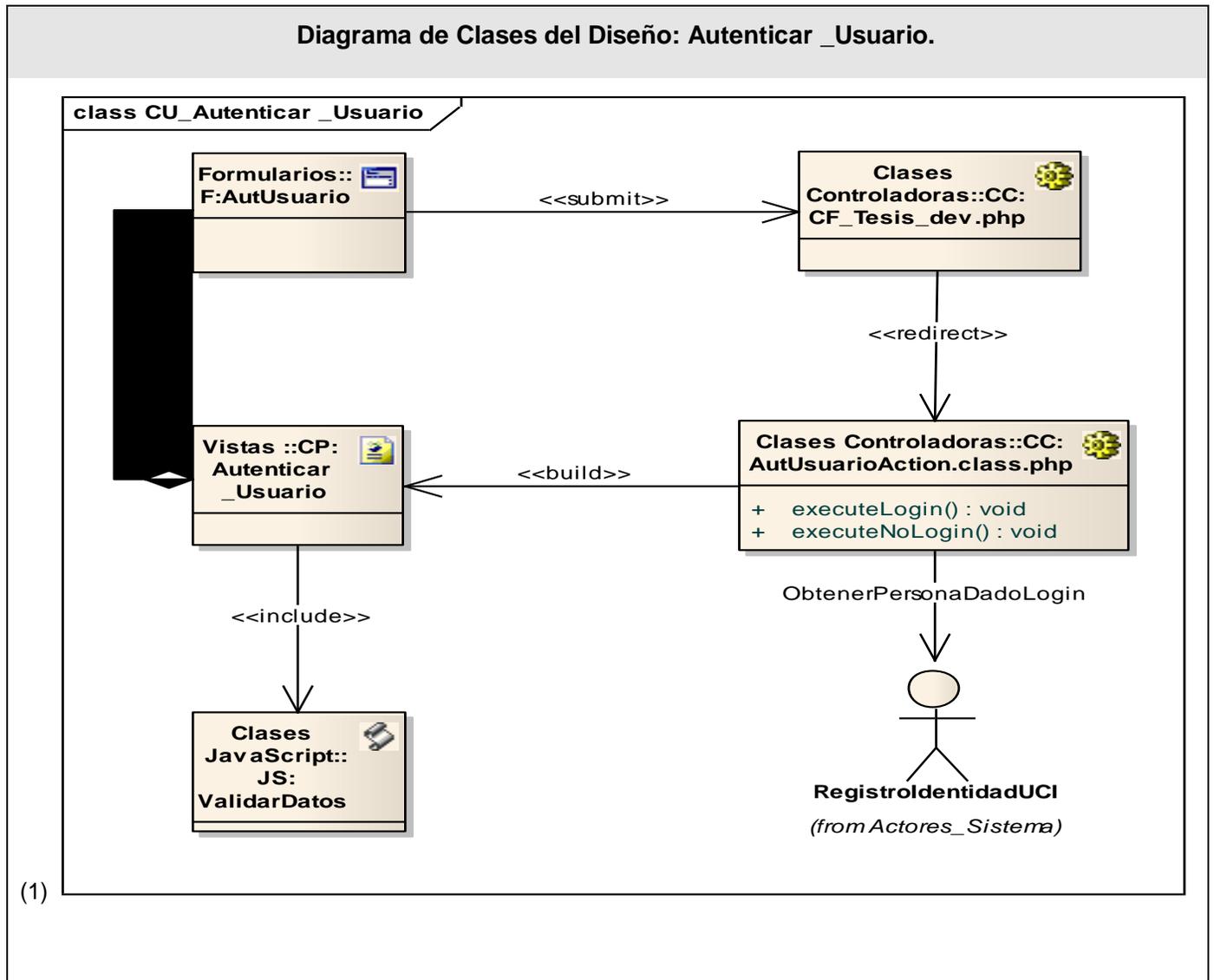


Figura 3.17 DC del diseño: Autenticar Usuario.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2.9 Diagrama de Clases del CU: Configurar Rol.

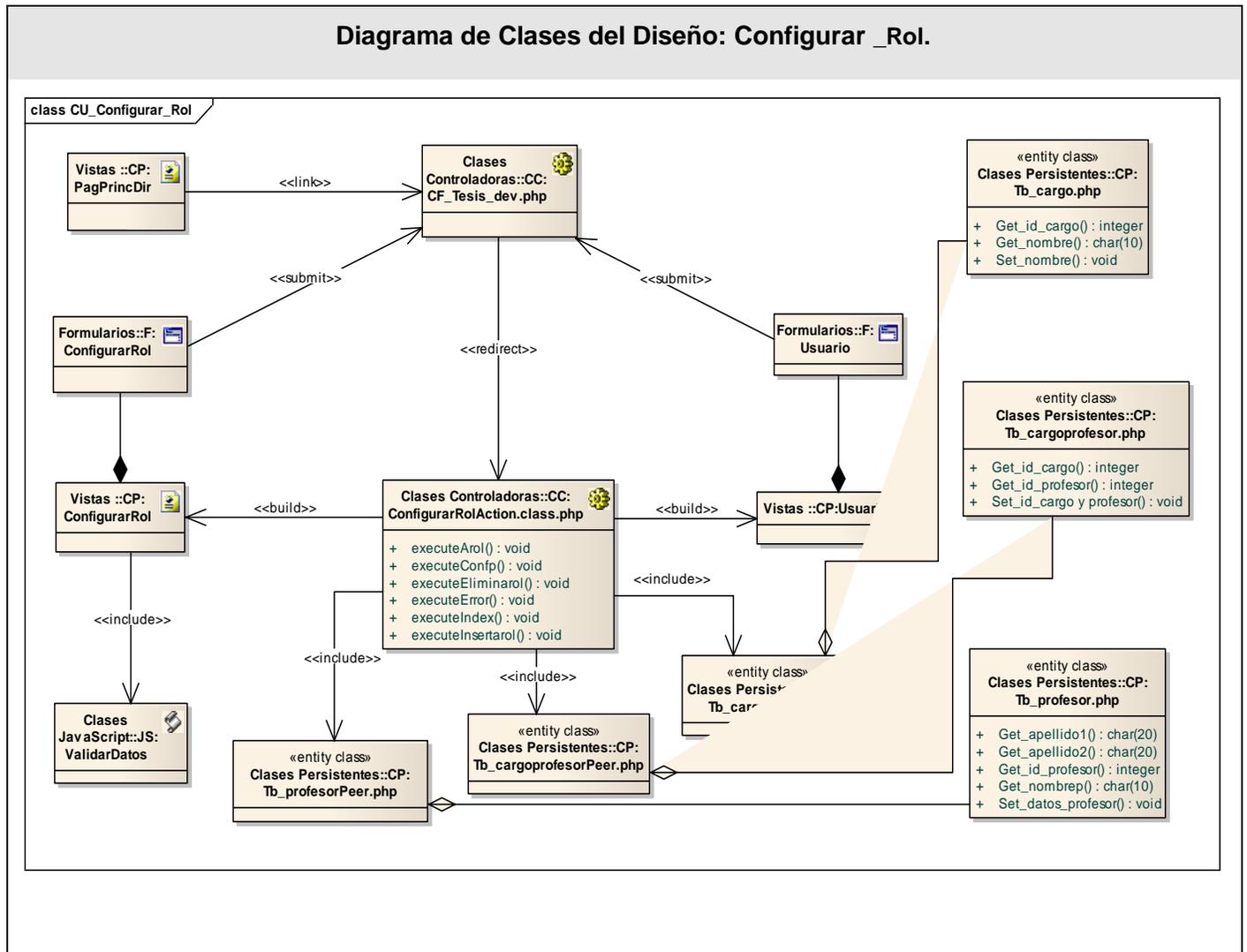


Figura 3.18 DC del diseño: Configurar Rol.

3.2.2 Diagramas de Secuencia del Diseño.

Los diagramas de secuencia son diagramas de interacción que hacen énfasis en la ordenación temporal de los mensajes. Para ver estos diagramas, ver [\[Anexo 1,2, 3, 4, 5, 6, 7, 8, 9\]](#).

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3 Diseño de la Base de Datos.

Para diseñar la base de datos del sistema, se utiliza el diagrama de clases persistentes y el modelo de datos, que están basados en el modelado de las clases del epígrafe anterior. Algunas de las clases representan los datos que se obtienen y almacenan durante los procesos de la aplicación, estas son las que pueden modelarse a través de un diagrama de clases persistentes, estas clases son las que son capaces de guardar su estado en un medio permanente cuya necesidad está dada por el almacenamiento físico permanente de la información de la clase, para la copia de seguridad en caso de caída del sistema, o para el intercambio de información ; lo que permitirá ver la relación entre los datos, y completará el modelado de la lógica de negocio de la aplicación. Por su parte el modelo de los datos describe la representación lógica y física de dichos datos persistentes en el sistema.

3.3.1 Diagrama de Clases Persistentes.

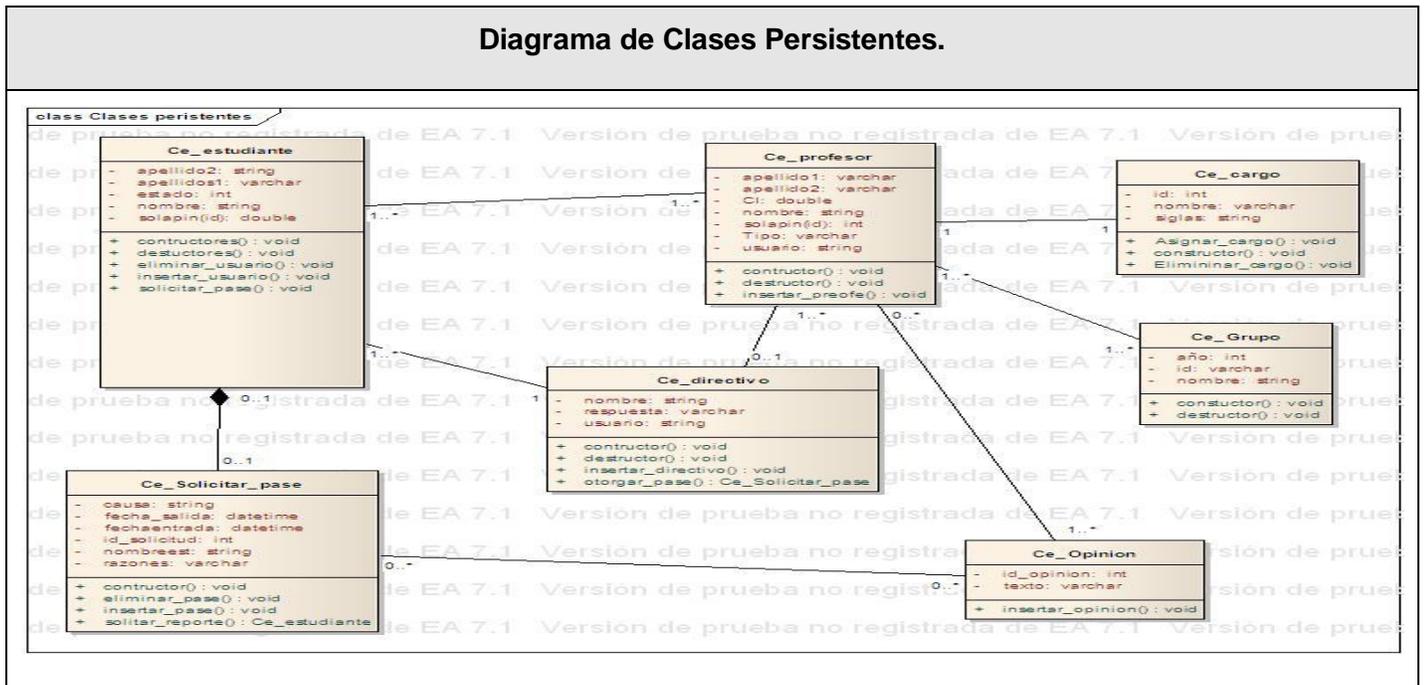


Figura 3.19 DC Persistentes.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.3.2 Modelo de Datos.

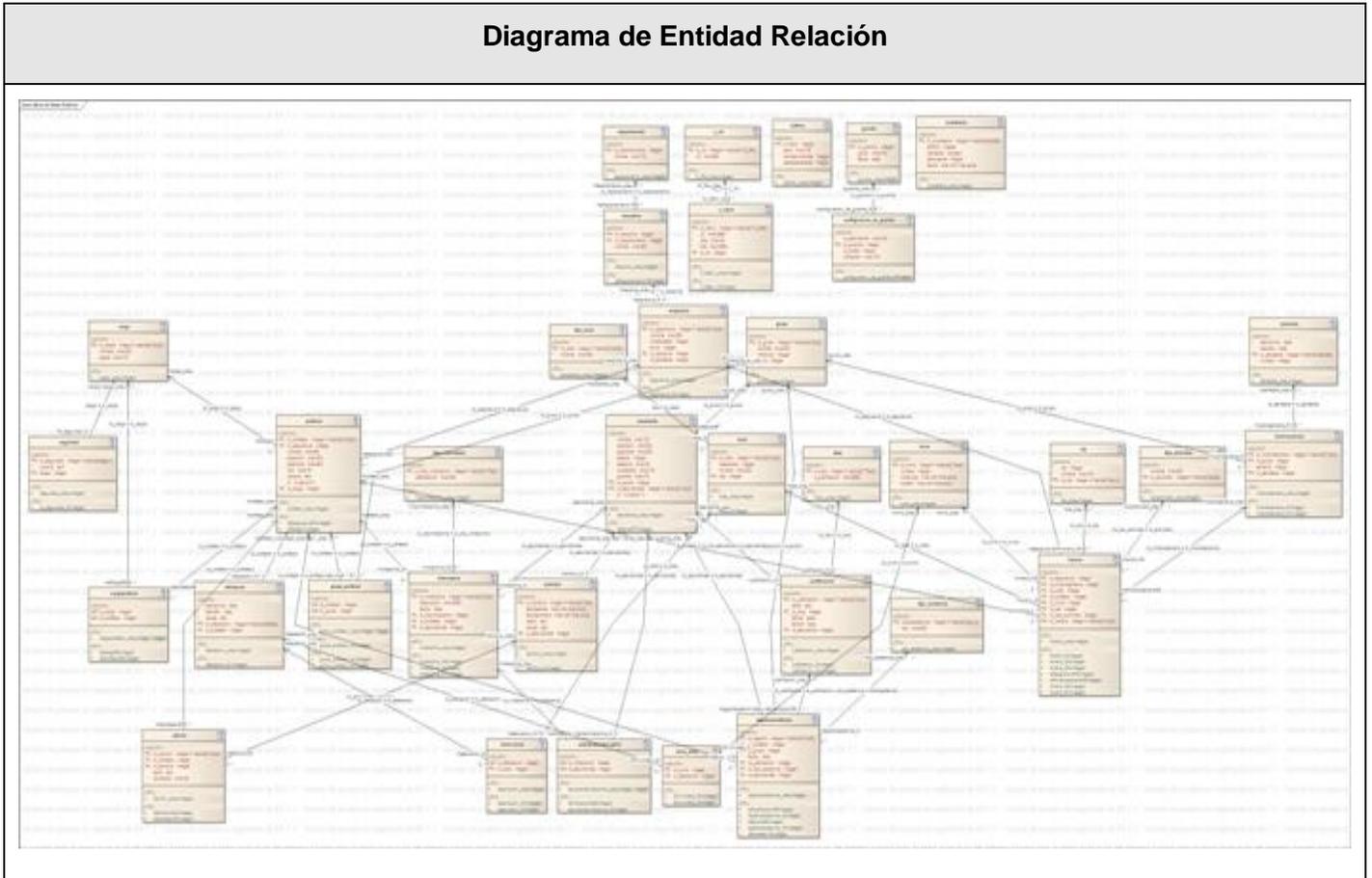


Figura 3.20 Diagrama de Entidad Relación.

3.4 Principios de Diseño Empleados.

En el éxito o el fracaso de una aplicación web, ya sea esta pequeña, mediana o grande, tienen gran influencia el correcto diseño de la interfaz, la elaboración de una completa y detallada ayuda y el tratamiento de los errores. Por lo que a continuación son descritos los principios de diseño utilizados para el desarrollo de la presente aplicación.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.4.1 Estándares en la Interfaz de la Aplicación.

El correcto diseño de una interfaz de usuario es una tarea que ha adquirido una gran relevancia en el desarrollo de sistemas, pudiera definirse como interfaz de usuario a: “el conjunto de trabajos y pasos que seguirá el usuario durante todo el tiempo que se encuentre en relación con el software, detallando lo que verá y escuchará en cada momento, las acciones que realizará, así como las respuestas que el software brindará”. [19]

La calidad con que sea desarrollada la interfaz de usuario es uno de los principales motivos para lograr el éxito o el fracaso de una aplicación, es por ello que en gran medida uno de los principales aspectos de la usabilidad es la consistencia de su interfaz de usuario y por tanto la misma debe ser lo más amigable posible y debe proporcionarle funciones fáciles de aprender al usuario para que se sienta mejor identificada con la misma, siendo esta, el medio por el cual se interactúa con el sistema.

Para el desarrollo de la interfaz de usuario de la siguiente aplicación se tuvieron en cuenta los siguientes aspectos:

- Reducir la carga a la memoria.
- Previsiones de errores y manejo de errores de entrada de datos por parte del usuario.
- Se utilizaron colores gris, blanco y negro, estándares y acordes a las normas de diseño de la Facultad.
- Obtención en todo momento de información de retroalimentación.
- Interfaz sencilla para que el usuario en ningún momento se sienta desorientado.
- La letra usada fue tipo Arial de tamaño 12, porque permite una separación entre las letras óptima para presentar información.
- Lograr que los usuarios realicen las acciones de forma fácil.
- Se utilizó una hoja de estilo para guardar la configuración del diseño para todas las páginas, para los botones y las líneas se utilizaron estos estilos, eliminando así el número de imágenes que hagan más lenta la presentación de la página.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

3.4.2 Tratamiento de Errores.

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que en éste se registra y muestra. Los errores se tratan en una página especial que incluye el fichero de configuración general, y está preparada para recoger el número del error y presentar la pantalla con el error que le corresponde a ese código. En algunos casos incluye la forma de solucionar el error, como es el caso de la sesión, y la autenticación de los usuarios.

Algunos errores serán generados por funciones JavaScript para evitar la ejecución de la página en vano y otros en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente y continuar. Se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios. Cuando se introduce información en un formulario y faltan datos, sale un cuadro de alerta indicando el campo o dato que falta. Similar ocurre cuando se introduce información errónea en un campo numérico o e-mail.

3.4.3 Formatos de Reportes.

Generar reportes que permitan un control de las solicitudes es una de las principales funcionalidades del sistema propuesto. Estos han sido concebidos como un módulo dentro de la aplicación y se obtienen en dependencia de las necesidades del usuario.

Cada reporte tiene un encabezado que le identifica y describe brevemente, luego se muestra la información obtenida de manera legible y organizada.

3.4.4 Seguridad.

El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios en la aplicación, identificando al usuario antes de que éste pueda realizar cualquier acción sobre el mismo; garantizando que la información sea vista y manipulada únicamente por quien tiene derecho a hacerlo.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

Conclusiones

En el capítulo fueron tratadas las concepciones básicas para el análisis y diseño del sistema. Se desarrollaron los diagramas de clases de la aplicación y el diseño de la base de datos de la misma. Además, se describieron los principios de diseño seguidos, específicamente, los estándares de la interfaz, la concepción del tratamiento de errores, formatos de reportes y seguridad.

CAPÍTULO IV

IMPLEMENTACIÓN

Capítulo IV: Implementación

En éste capítulo se describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Para ello se confeccionan los diagramas de despliegue y componentes.

4.1 Diagrama de Despliegue.

El Diagrama de Despliegue es un Modelo de Objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar y cada arco la conectividad que existe entre dichos nodos. [20]

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En éste tipo de diagrama intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un modelo físico en tiempo de ejecución, es decir un computador que se compone habitualmente de: por lo menos, memoria y capacidad de procesamiento, a su vez puede estar conformada por otros componentes. [21]

El Diagrama de Despliegue muestra la topología del hardware sobre la que es ejecutado el sistema. A continuación es mostrado el Diagrama de Despliegue para la propuesta de aplicación a desarrollar:

CAPÍTULO IV

IMPLEMENTACIÓN

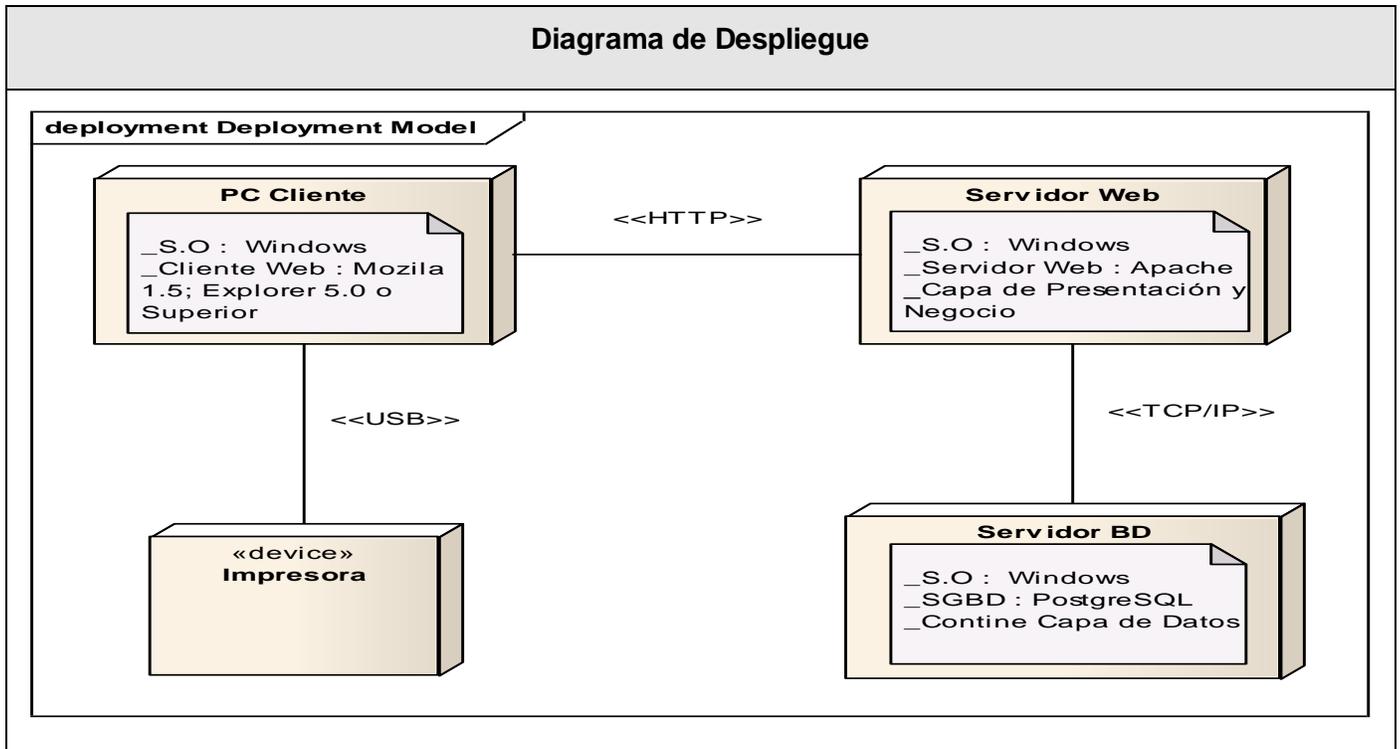


Figura 4.1 Diagrama de despliegue.

4.2 Diagrama de Componentes.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Normalmente contienen componentes, interfaces y relaciones entre ellos y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Dado que los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases, o un diagrama de casos de uso.

CAPÍTULO IV

IMPLEMENTACIÓN

Un diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también que un componente software contiene una interfaz, es decir, la soporta.

Normalmente los diagramas de componentes se utilizan para modelar código fuente, versiones ejecutables, bases de datos físicas, entre otros.

En el diagrama que se muestra a continuación, se representa la estructura y organización de la presente Aplicación Web en términos de componentes, mediante la arquitectura Modelo Vista Controlador que propone el Framework Symfony.

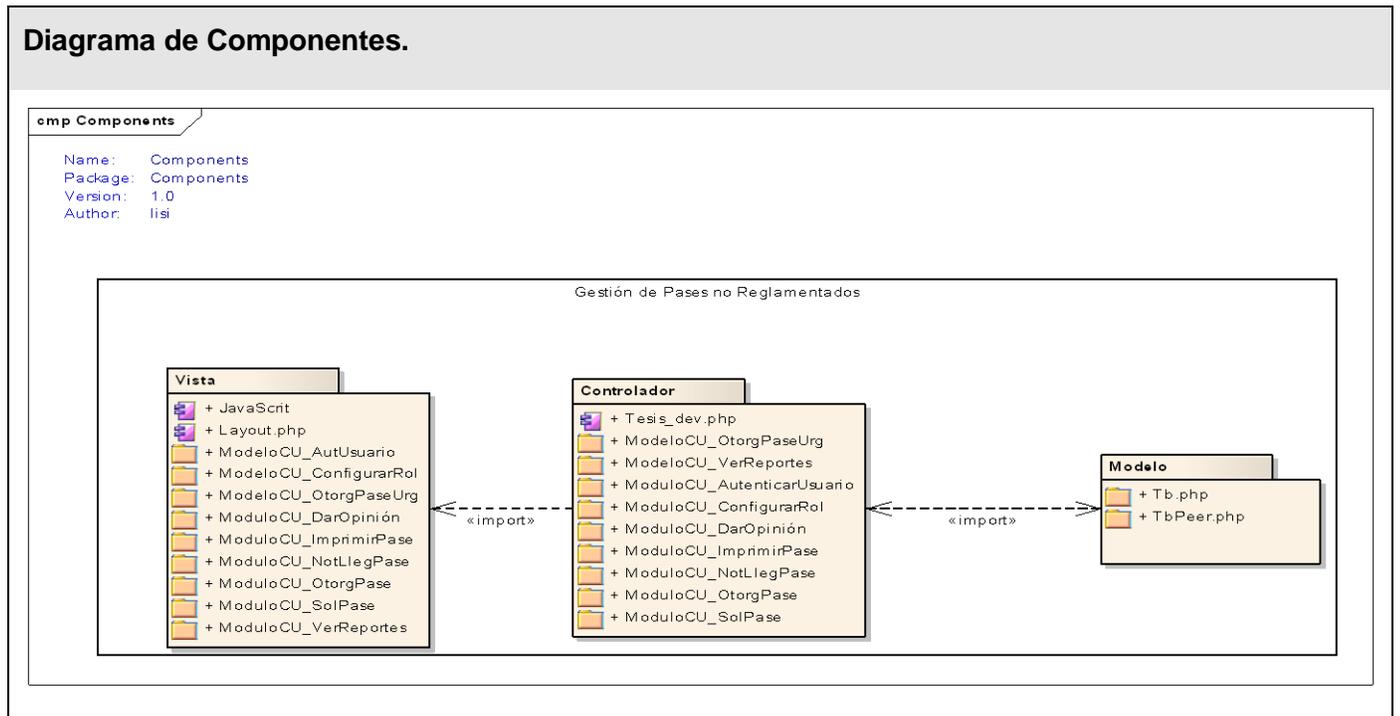


Figura 4.2 Diagrama de componentes.

CAPÍTULO IV

IMPLEMENTACIÓN

La Capa Vista permite mostrar al usuario una página web y le posibilita interactuar con la misma, para ello cuenta con la página cliente y la biblioteca JavaScript que la complementa proporcionándole gran dinamismo a la misma.

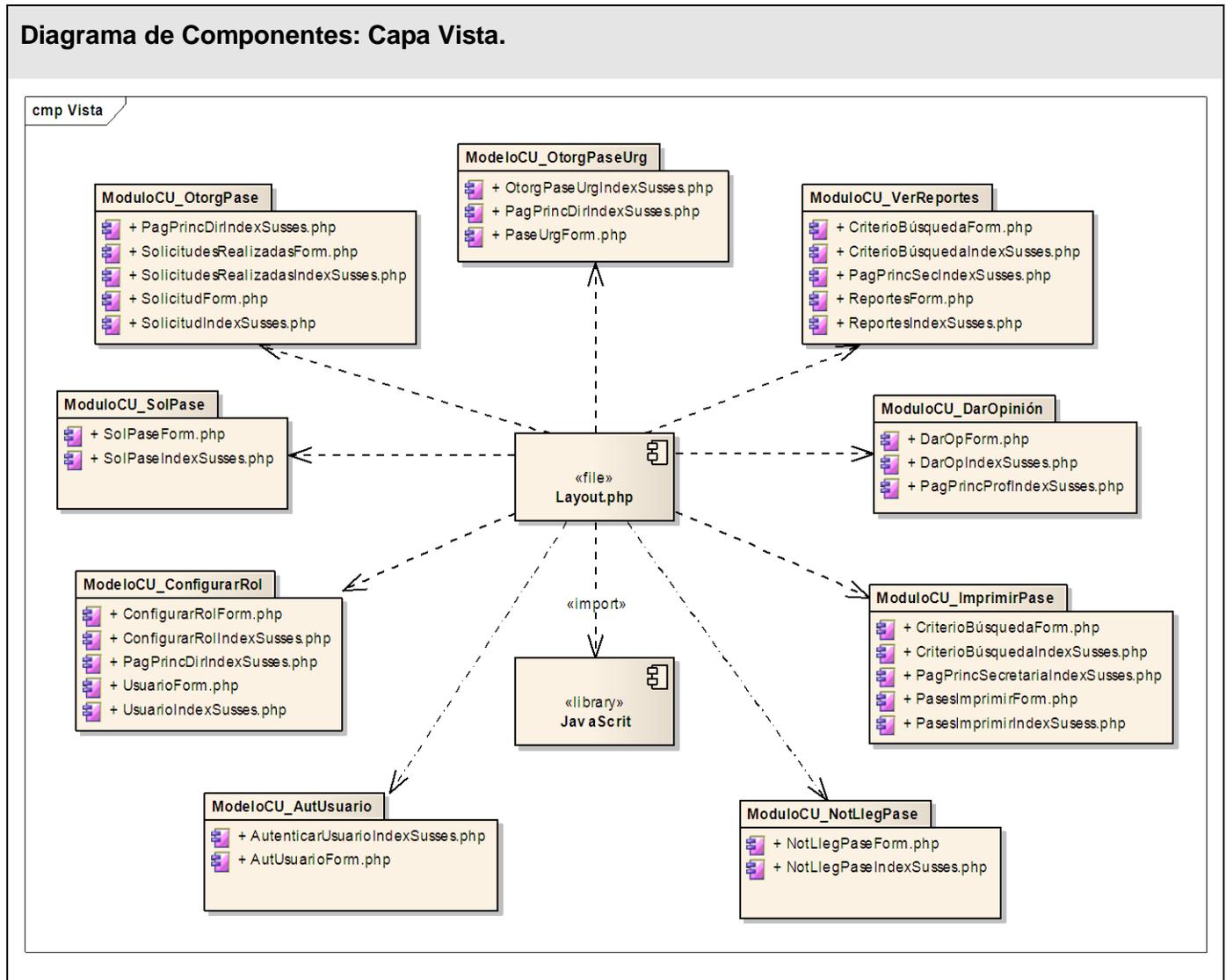


Figura 4.3 Diagrama de componentes: Capa Vista.

CAPÍTULO IV

IMPLEMENTACIÓN

La Capa Modelo proporciona el acceso a datos, está constituida por las clases Peer quienes contienen métodos estáticos para obtener los datos de las clases, y las clases php que son las clases que manipulan los métodos de inserción y devolución de la BD.

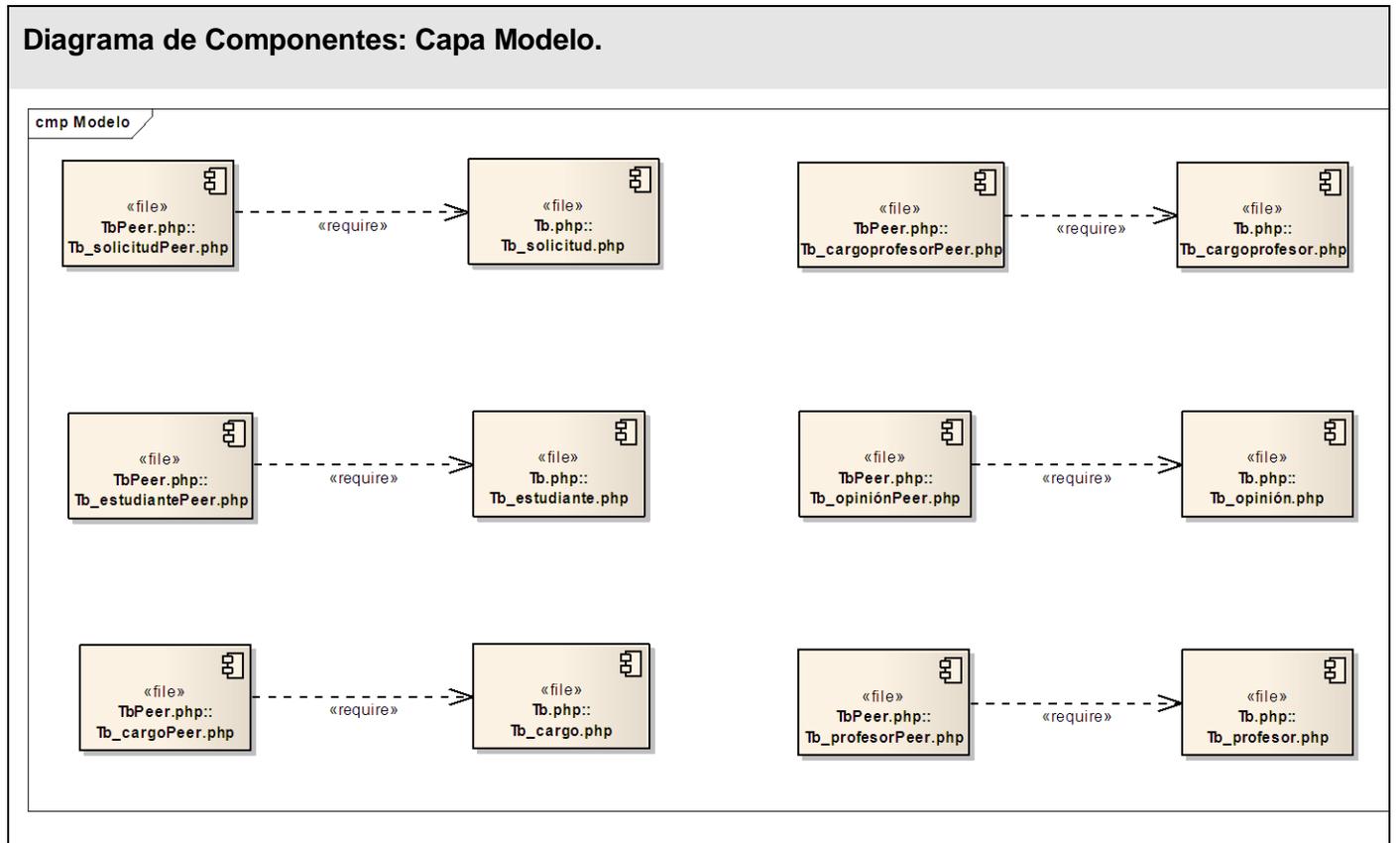


Figura 4.4 Diagrama de componentes: Capa Modelo.

En la Capa del Controlador se encuentra el Controlador Frontal quien actúa como único punto de entrada a la aplicación, se encarga de gestionar las peticiones realizadas en la aplicación y determina la acción a ejecutar.

CAPÍTULO IV

IMPLEMENTACIÓN

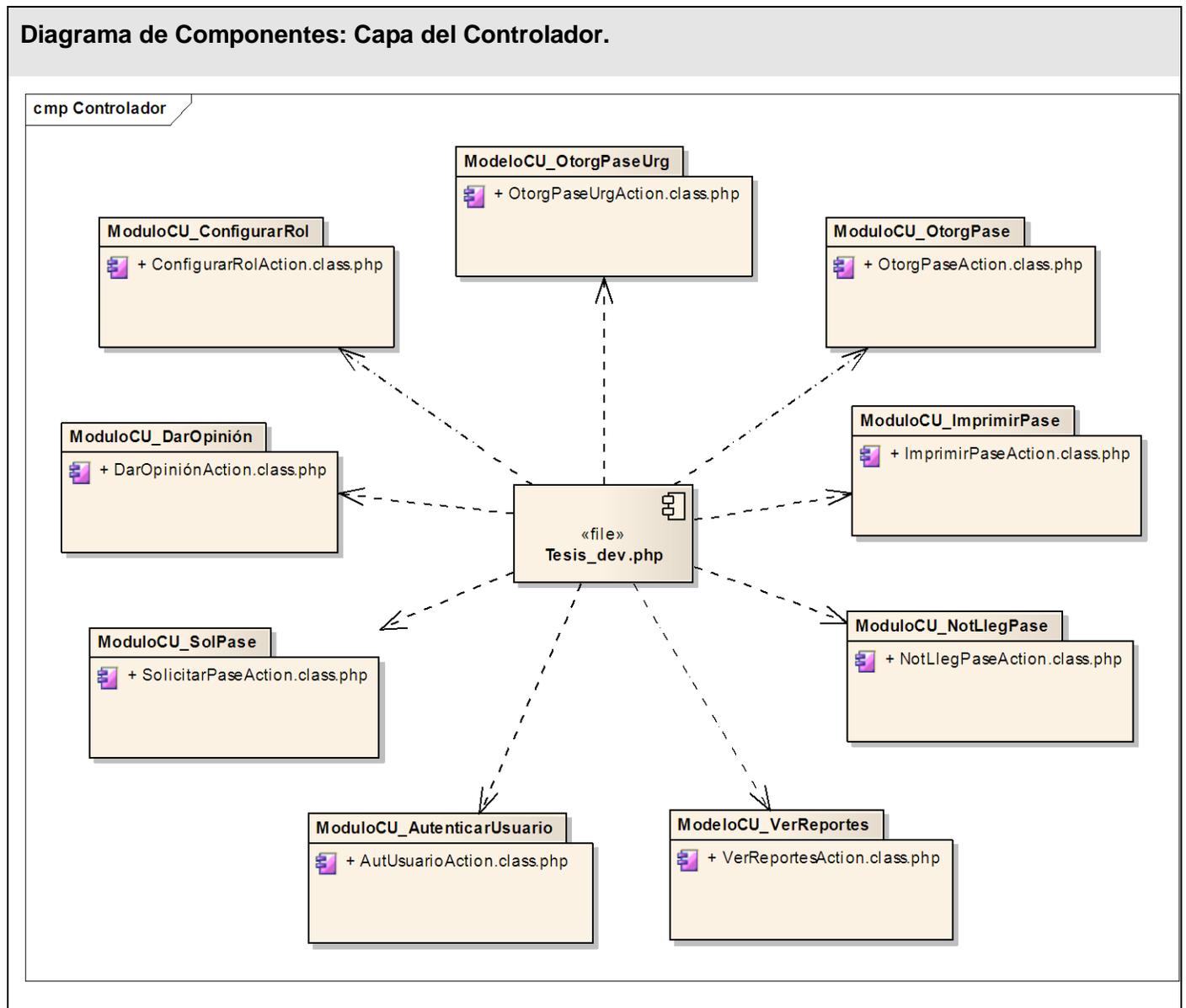


Figura 4.5 Diagrama de componentes: Capa del Controlador.

CAPÍTULO IV

IMPLEMENTACIÓN

Conclusiones

En éste capítulo es mostrada la topología de hardware sobre la que estará corriendo el sistema a través del modelo de despliegue y fue definida la organización de las clases y objetos en componentes. A partir de éste momento el sistema cuenta con funcionalidad operativa.

Conclusiones



CONCLUSIONES

Llegado éste punto de culminación del presente trabajo, y después de arduas e intensas jornadas de labor llenas de dedicación y entrega, se llega a las siguientes conclusiones:

- Se documentó y desarrolló una aplicación web, que gestiona la información relacionada con el proceso de solicitud de pases no reglamentados, por parte de los estudiantes de la Facultad 7, en la Universidad de las Ciencias Informáticas (UCI).
- Se centralizó la información que se genera a partir del proceso de solicitud de pases no reglamentados en la Facultad 7 de la UCI.
- Se implementó la aplicación en el Portal de la Facultad 7 lográndose de esta manera un ambiente de trabajo integrado.

Recomendaciones



RECOMENDACIONES

Para un posterior desarrollo y perfeccionamiento del sistema implementado se recomienda:

- Desplegar el sistema desarrollado en todos los Puntos de Control de acceso a la Universidad.
- Llevar la aplicación a un componente de Joomla para una mayor operatividad a la hora de integrarla al Portal de la Facultad.
- Continuar el estudio del sistema en general con el objetivo de añadir nuevas funcionalidades.
- Integrar la aplicación a otros sistemas existente en la Universidad como el de la planificación de alimentos.

R eferencias

B ibliográficas



REFERENCIAS BIBLIOGRÁFICAS

[1] Aplicaciones Web. [Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web].

[2] Corsi, M. "Beneficios De Las Aplicaciones Basadas En Web". [Disponible en: Http://Www.Masternewmedia.Org/Es/Aplicaciones_Web/Temas_De_Aplicaciones_Web/Beneficios_De_Las_Aplicaciones_Basadas_En%20_Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.Htm].

[3] Corsi, M. "Beneficios De Las Aplicaciones Basadas En Web". [Disponible en: Http://Www.Masternewmedia.Org/Es/Aplicaciones_Web/Temas_De_Aplicaciones_Web/Beneficios_De_Las_Aplicaciones_Basadas_En%20_Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.Htm].

[4] Aplicaciones Web. [Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web].

[5]. Peñalver, J. "Ventaja e inconvenientes de las aplicaciones Web". [Disponible en: <http://www.avidos.net/blogold/aplicaciones-web/>].

[6] Soto, N. y Saborit, Y. Hubble. Propuesta para un Sistema de Catalogación y Recuperación de Recursos de Información. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico "José Antonio Echeverría", Ciudad de la Habana, junio 2004.

[7] Servicio web. [Disponible en: http://wiki.prod.uci.cu/index.php/Servicio_web].

[8] Alemán, I y Rodríguez, JC. Sistema Automatizado para la Gestión de las Solicitudes de Servicios Telemáticos, Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, Facultad 2 , Ciudad de la Habana, Junio del 2007.

REFERENCIAS BIBLIOGRÁFICAS

- [9] Alonso L. Modulo Admisión del Sistema Automatizado para la Gestión de Información de la Misión Milagro. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2006
- [10] Arquitectura cliente/servidor [Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>].
- [11] Cliente-servidor [Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>].
- [12] Álvarez, M. “Evaluando Zend Studio”. [Disponible en: <http://www.maestrosdelweb.com/editorial/zendstudio/>].
- [13] Álvarez, M. “Evaluando Zend Studio”. [Disponible en: <http://www.maestrosdelweb.com/editorial/zendstudio/>].
- [14] Álvarez, M. “Evaluando Zend Studio”. [Disponible en: <http://www.maestrosdelweb.com/editorial/zendstudio/>].
- [15] Modelo del Negocio. Conferencia 2 de Ingeniería de Software I, Curso: 2007-2008, UCI.
- [16] Modelo del Negocio. Conferencia 2 de Ingeniería de Software I, Curso: 2007-2008, UCI.
- [17] Jacobson, I y Booch, G y Rumbaugh, J. “El Proceso Unificado de Desarrollo del Software”. España. Addison Wesley. 2000.458.
- [18] Jacobson, I y Booch, G y Rumbaugh, J. “El Proceso Unificado de Desarrollo del Software”. España. Addison Wesley. 2000.458.

REFERENCIAS BIBLIOGRÁFICAS

[19] Eliurkis Díaz Terrero. "Módulo flujo médico del Sistema Automatizado para la Gestión de la Información de la Misión Milagro". Instituto Superior Politécnico "José Antonio Echeverría". Ciudad de la Habana, 2006 7.

[20] Jacobson, I y Booch, G y Rumbaugh, J. "El Proceso Unificado de Desarrollo del Software". España. Addison Wesley. 2000.458.

[21] Jacobson, I y Booch, G y Rumbaugh, J. "El Proceso Unificado de Desarrollo del Software". España. Addison Wesley. 2000.458.

BIBLIOGRAFÍA

- Alemán, I y Rodríguez, JC. Sistema Automatizado para la Gestión de las Solicitudes de Servicios Telemáticos. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, Facultad 2, Ciudad de la Habana, Junio del 2007.
- Alonso L. Modulo Admisión del Sistema Automatizado para la Gestión de Información de la Misión Milagro. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2006.
- Álvarez, M. “Evaluando Zend Studio”. [Disponible en: <http://www.maestrosdelweb.com/editorial/zendstudio/>]. Modelo del Negocio. Conferencia 2 de Ingeniería de Software I, Curso: 2007-2008, UCI.
- Aplicaciones Web. [Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web].
- Arquitectura cliente/servidor [Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>].
- Article_MySQL-PostGreSQL. [Disponible en: http://www.netpecos.org/docs/mysql_postgres/index.html].
- Article_MySQL-PostGreSQL; Artículo comparativo. [Disponible en: <http://www.phpbuilder.com/columns/tim20000705.php3?page=1>].
- Benchmark. [Disponible en: <http://es.wikipedia.org/wiki/Benchmark>].
- Cliente-servidor [Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>].

BIBLIOGRAFÍA

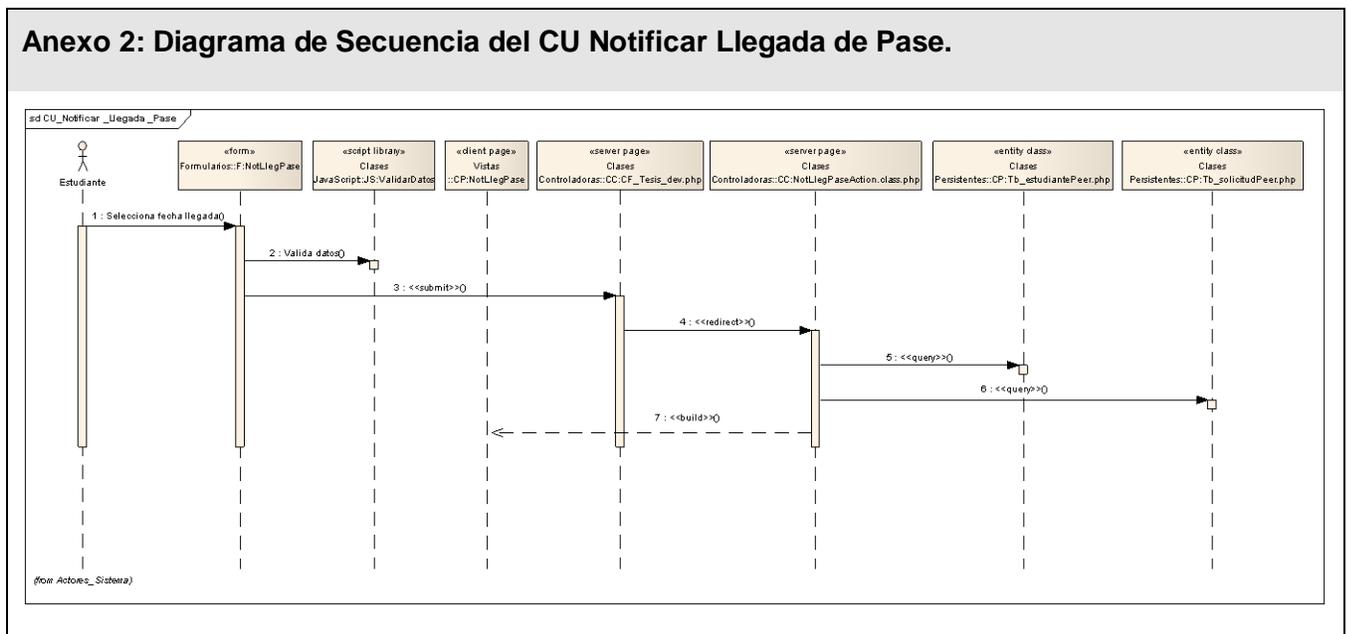
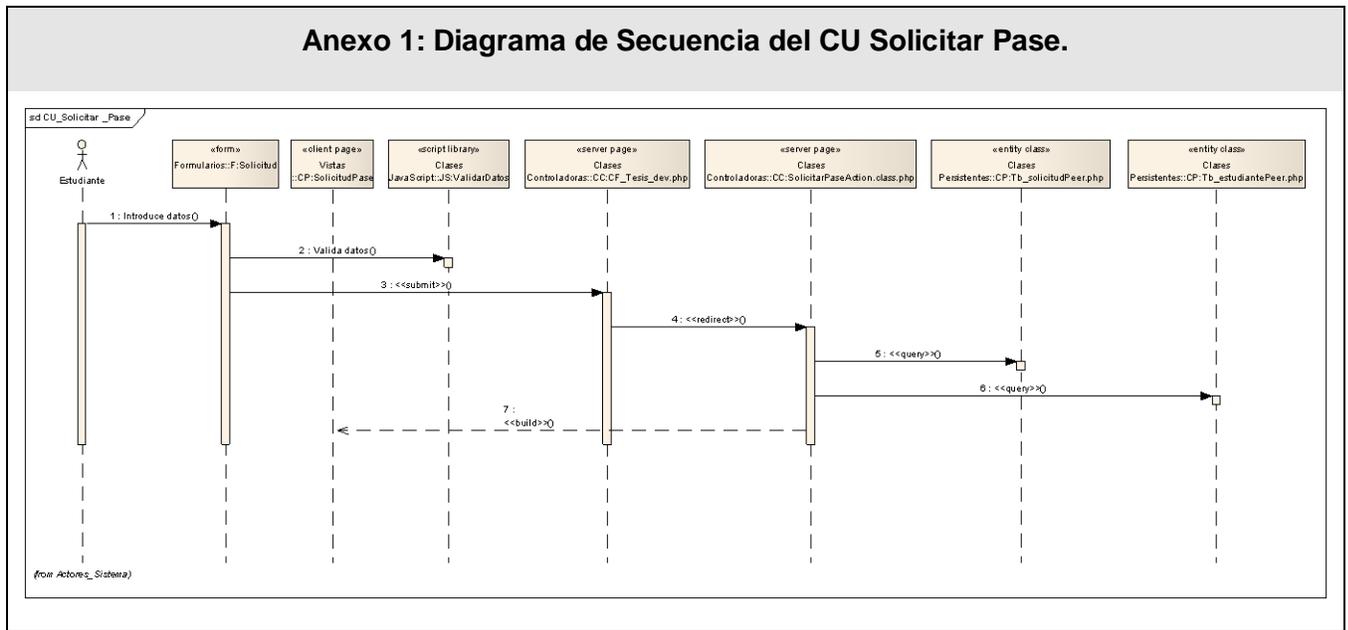
- Corsi, M. “Beneficios De Las Aplicaciones Basadas En Web”. [Disponible en: [Http://Www.Masternewmedia.Org/Es/Aplicaciones_Web/Temas_De_Aplicaciones_Web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.Htm](http://Www.Masternewmedia.Org/Es/Aplicaciones_Web/Temas_De_Aplicaciones_Web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.Htm)].
- Eliurkis Díaz Terrero. “Módulo flujo médico del Sistema Automatizado para la Gestión de la Información de la Misión Milagro”. Instituto Superior Politécnico “José Antonio Echeverría”. Ciudad de la Habana, 2006 7.
- Frameworks-de-php. [Disponible en: <http://pwneds.blogspot.com/2007/11/comparacion-de-10-frameworks-de-php.html>].
- Jacobson, I y Booch, G y Rumbaugh, J. “El Proceso Unificado de Desarrollo del Software”. España. Addison Wesley. 2000.458.
- Lee Babin, Nathan A. Good, Frank M. Kromann, Jon Stephens.PHP 5 Recipes: A Problem-Solution Approach; 2005.
- Manual de MySQL. [Disponible en: <http://www.mysql.com/documentation/index.html>].
- Manual de PostGreSQL. [Disponible en: <http://www.mysql.com/documentation/index.html>.]
- Marrero, P y Rodríguez, Y. Módulo pases programados para el Sistema de Reservación de Transporte. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, Ciudad de la Habana, Junio del 2007.

BIBLIOGRAFÍA

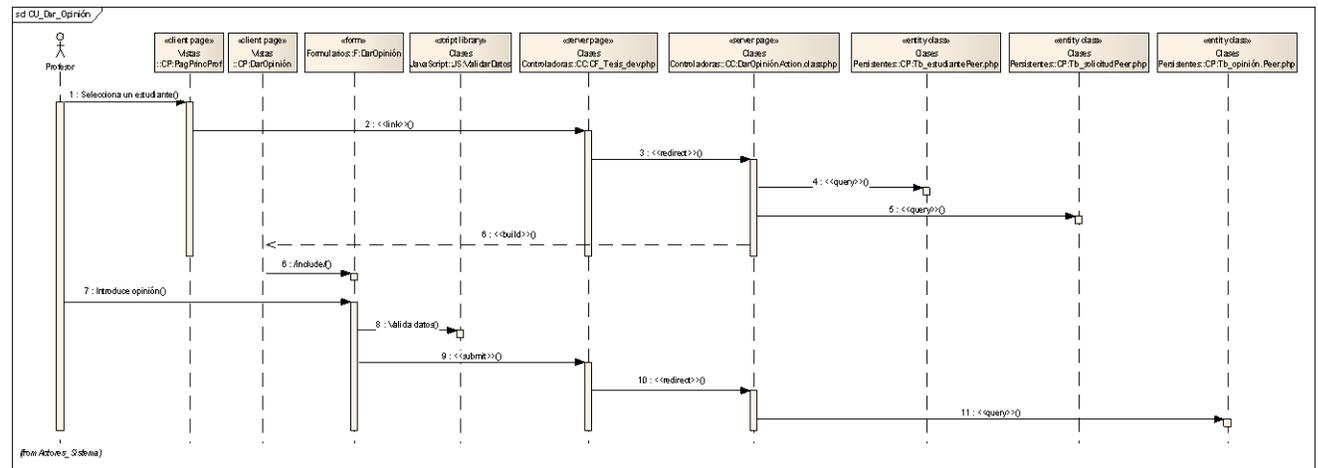
- Patrones de Arquitectura. [Disponible en: <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>].
- Peñalver, J. “Ventaja e inconvenientes de las aplicaciones Web”. [Disponible en: <http://www.avidos.net/blogold/aplicaciones-web/>].
- Proceso Unificado de Desarrollo de Software (RUP). [Disponible en: <http://www.scribd.com/doc/297224/RUP>].
- Servicio web. [Disponible en: http://wiki.prod.uci.cu/index.php/Servicio_web].
- WYSIWYG. [Disponible en: <http://www.webopedia.com/TERM/W/WYSIWYG.html> y en <http://es.wikipedia.org/wiki/WYSIWYG>].

*A*nexos

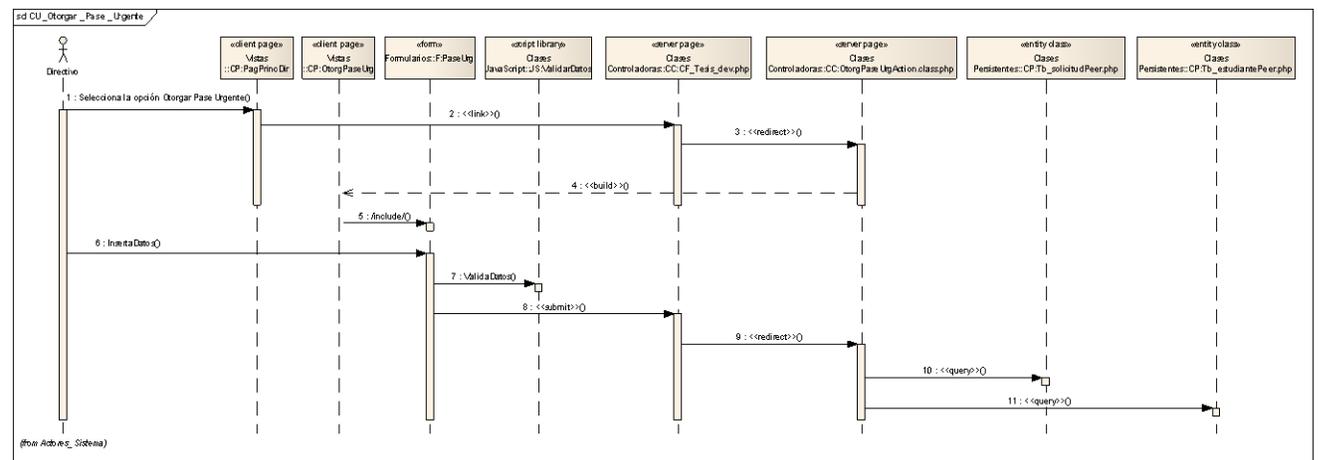




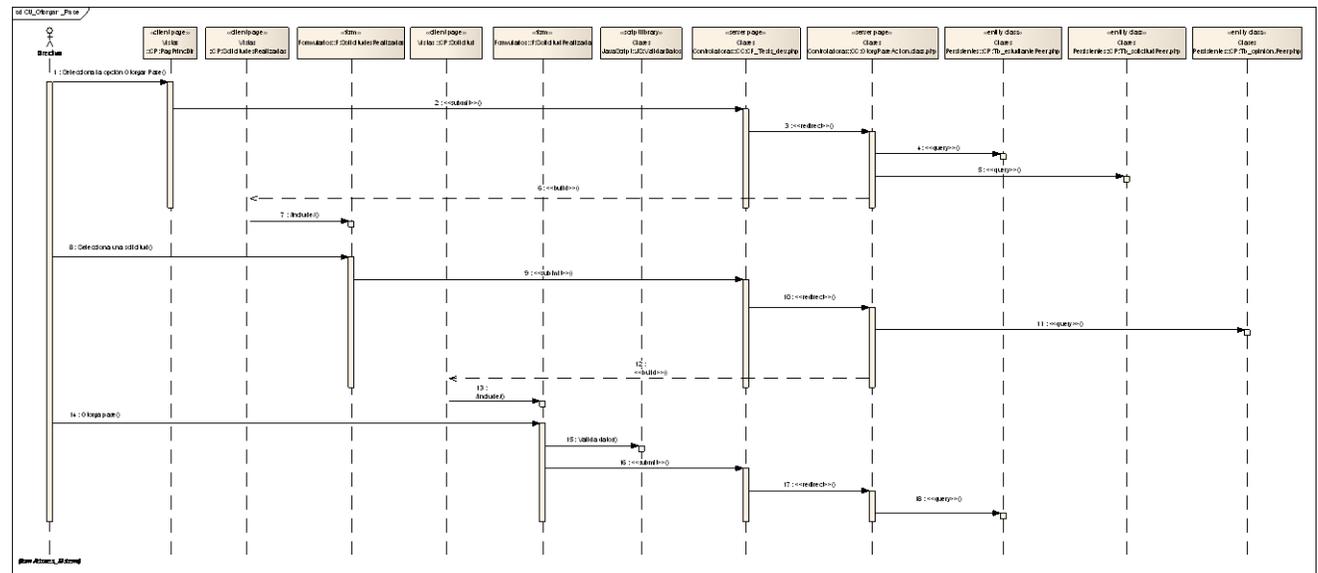
Anexo 3: Diagrama de Secuencia del CU Dar Opinión.



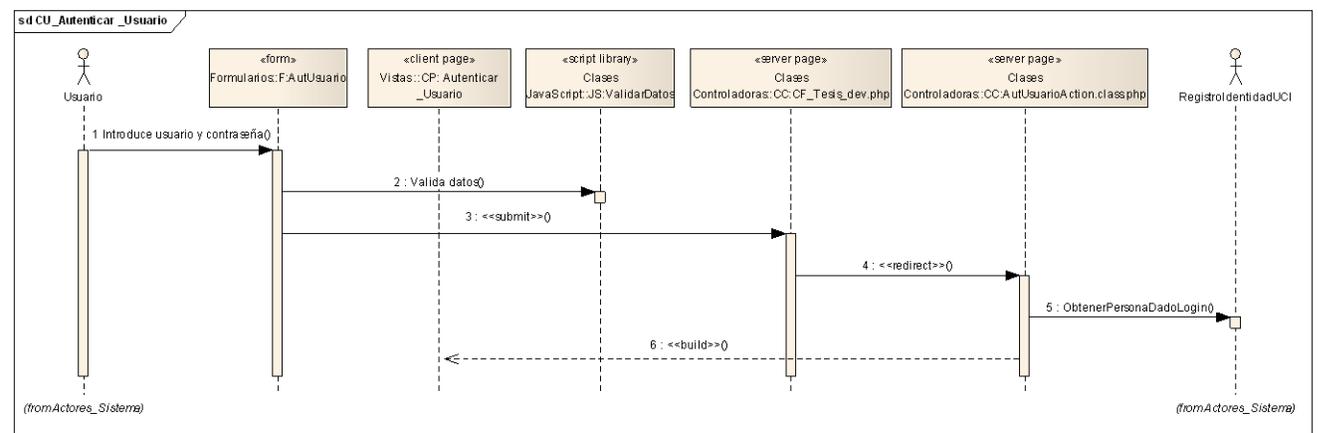
Anexo 4: Diagrama de Secuencia del CU Otorgar Pase Urgente.



Anexo 5: Diagrama de Secuencia del CU Otorgar Pase.



Anexo 6: Diagrama de Secuencia del CU Autenticar Usuario.



Glosario



Glosario de Términos

- **APACHE:** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.
- **Arquitectura Cliente/Servidor:** es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.
- **Benchmarks:** Es una técnica utilizada para medir el rendimiento de un sistema o componente de un sistema, es decir, comparar el rendimiento de un software contra otro o de parte del mismo, por ejemplo, comparar distintas consultas a una base de datos para saber cuál es la más rápida o directamente partes de código.
- **CSS:** Cascading Style Sheets. Lenguaje de hojas de estilo en cascada utilizadas para definir la presentación de un documento HTML o XML.
- **CU:** Casos de Uso.
- **CUN:** Caso de Uso del Negocio.
- **DC:** Diagrama de clase.
- **Directivo:** es la persona encargada de otorgar al estudiante un pase no reglamentado y que ocupa el cargo de Decano o Vicedecano de la Facultad.
- **Framework:** Estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.
- **HTML:** HyperText Markup Language. Lenguaje de marcas usado para escribir documentos para servidores World Wide Web. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

GLOSARIO DE TÉRMINOS

- **HTTP:** HyperText Transfer Protocol. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- **Pases no Reglamentados:** son los pases que solicitan los estudiantes durante el transcurso de la semana.
- **PHP:** Hypertext Preprocessor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.
- **PostgreSQL:** es un Sistema de Gestión de bases de datos Objeto-Relacionales (ORDBMS) libre.
- **Profesor:** es la persona encargada de dar una opinión del estudiante que solicita un pase no reglamentado sobre cómo se encuentra en su asignatura.
- **Secretaria:** es la persona encargada de llevar el control en la Facultad de los estudiantes que salen de pase.
- **SOAP:** Simple Object Access Protocol. Protocolo utilizado en los servicios Web. Define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
- **UCI:** Universidad de las Ciencias Informáticas.
- **UML:** Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.
- **Web Service:** (Servicio Web). Conjunto de documentos web en formatos tales como .aspx(ASP), .php(PHP), .pl(PERL), etc. accesible desde http como desde cualquier aplicación, su código es generalmente más extensivo que el de una simple web en html.
- **WYSIWYG:** es el acrónimo de **What You See Is What You Get** (en español, "lo que ves es lo que obtienes"). Es una aplicación que le permite al usuario ver en la pantalla exactamente lo que aparecerá cuando se imprime el documento. Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.