



Universidad de las Ciencias Informáticas

Módulo pases programados para el Sistema de Reservación de Transporte

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

AUTORES

Pável Marrero Pulgar
Yalian Rodríguez Leyva

TUTORES

Ing. Yadilka Suárez-Inclán Rivero
Ing. José Fidalgo Hidalgo

CONSULTANTE

Ing. Eliurkis Díaz Terrero

Ciudad de la Habana

Junio del 2007

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los __ días del mes de junio del 2007.

Firma del Autor
Pável Marrero Pulgar

Firma del Autor
Yalian Rodríguez Leyva

Firma del Tutor
Ing. Yadilka Suárez-Inclán Rivero

Firma del Tutor
Ing. José Fidalgo Hidalgo

Opinión del Usuario del trabajo de Diploma

El Trabajo de Diploma, titulado: "Módulo pases programados para el Sistema de Reservación de Transporte", fue realizado en la Universidad de las Ciencias Informáticas (UCI). Este centro considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta Universidad los beneficios siguientes:

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a _____.

Y para que así conste, se firma la presente a los _____ días del mes de _____ del año _____

Representante de la entidad

Cargo

Firma

Cuño

Opinión del tutor del trabajo de Diploma

Título: Módulo pases programados para el Sistema de Reservación de Transporte.

Autores: Pável Marrero Pulgar y Yalian Rodríguez Leyva.

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____.

Firma

Fecha

“Las proposiciones matemáticas, en cuanto tienen que ver con la realidad, no son ciertas;
y en cuanto que son ciertas, no tienen nada que ver con la realidad”

Albert Einstein

Dedicatoria

...A Misle.

...A Keila.

...A Tomás.

A mi madre, que era su sueño y el mío.

A mi hermanita que la quiero mucho.

A toda mi familia por apoyarme y darme fuerzas.

Pavel Marrero Pulgar

Yalian Rodríguez Leyva

Agradecimientos

Agradecemos a la Revolución Cubana y en especial a nuestro Comandante en Jefe por habernos dado la posibilidad de formarnos como lo que somos, y de ver nuestros sueños hechos realidad.

Pavel Marrero Pulgar:

A mis padres Tomás y Keila por comprenderme y estar conmigo en todo momento.

A mi hermana Misleidis por estar siempre a mi lado.

A Edith Tort pues a pesar de estar tan lejos ha sabido dar su modesto aporte.

A Dina por estar siempre a mi lado cuando la necesitaba.

A Eliurkis por brindar siempre su ayuda desinteresada.

A Deivis por ayudar a encontrar soluciones a problemas presentados.

A Yalian por ser un excelente compañero de tesis.

A todos los que tan siquiera una sola vez me preguntaron: ¿cómo va la tesis?

Yalian Rodríguez Leyva:

A mi madre, por ser mi inspiración y por apoyarme en todo cuanto pudo.

A Yaniet, mi hermanita del alma, por tener tanta confianza en mi.

A Iris, por todo el apoyo y confianza.

A Rodisnel, por todos los buenos consejos en todo momento.

A toda mi familia por la fuerza brindada.

A nuestros tutores, por el apoyo incondicional.

A Pável por ser también un excelente compañero de tesis.

A todos los que de una u otra forma se han hecho parte de este éxito, llegue hasta ustedes nuestro más sincero agradecimiento.

Resumen

Con el desarrollo de la informática y el aumento de su impacto social, son cada vez más las instituciones u organizaciones que optan por incorporar aplicaciones que gestionen su información, logrando así una mayor dinámica en sus procesos de negocio.

La Universidad de las Ciencias Informáticas es una institución completamente comprometida con la Revolución Cubana, que se propone, mediante la formación integral y continua de profesionales, la actividad científico-técnica y la extensión universitaria, contribuir de forma significativa al desarrollo sostenible de la sociedad cubana; mantener un liderazgo nacional en el campo de la tecnología, y con soluciones creativas y autóctonas se nutre la UCI para brindar disímiles servicios a favor de construir una ciudad completamente informatizada.

Este trabajo ha sido realizado en la Universidad de las Ciencias Informáticas, donde se brindan diferentes servicios a la comunidad universitaria, como el Sistema de Reservación de Transporte, el cual aun no satisface completamente las exigencias planteadas para su completo funcionamiento.

Con este trabajo se pretende brindar una solución automatizada, flexible y única a todo el proceso de programación del pase para el transporte del fin de semana, por parte de los estudiantes de la universidad, para así proporcionar una mayor funcionalidad al sistema.

Es por ello, que para solucionar dichos problemas se decidió desarrollar una aplicación Web, basada en tecnología PHP4 y con gestor de base de datos PostgreSQL la cual contribuye al eficiente funcionamiento.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1 Introducción	5
1.2 Sistemas de Reservasiones	5
1.2.1 Czech Transport.com	5
1.2.2 easyJet.com	6
1.2.3 Avis	7
1.2.4 3 Milenio	8
1.2.5 Sistema de Reservación de Transporte (versión inicial)	9
1.3 Tendencias y tecnologías actuales	10
1.3.1 Aplicaciones Web	10
1.3.1.1 Modelo Cliente/Servidor	12
1.3.2 PHP (Hyper Text Preprocessor)	13
1.3.3 Servidor Web Apache	17
1.3.4 AJAX	19
1.3.4.1 Ventajas de AJAX	20
1.3.4.2 Desventajas de AJAX	20
1.3.5 Patrones de Diseño	21
1.3.6 Modelo Vista Controlador (MVC)	22
1.3.7 Sistema Gestor de bases de datos	23
1.3.7.1 PostgreSQL	25
1.3.7.2 MVCC	26
1.3.7.2.1 Cliente/Servidor	27
1.3.8 Proceso de Desarrollo	27
1.3.8.1 Lenguaje Unificado de Modelado (UML)	28
1.3.9 Herramientas utilizadas en el desarrollo	29
1.3.9.1 Macromedia Dreamweaver 8	30
1.3.9.2 Zend Studio	30
1.3.9.3 Rational Rose	30
1.3.9.4 PgAdmin	31
1.4 Conclusiones	31
Capítulo 2: Características del Sistema	33
2.1 Introducción	33
2.2 Objeto de estudio	33
2.2.1 Situación problemática	33
2.2.1.1 Flujo actual de los procesos	34
2.2.2 Objeto de automatización	35
2.2.3 Información que se maneja	35
2.3 Modelo del Negocio	35
2.3.1 Actores del Negocio	36
2.3.2 Trabajadores del Negocio	37

2.3.3	Diagrama de casos de uso del negocio.	37
2.3.4	Descripción de los casos de uso del negocio.....	38
2.3.4.1	Especificación textual del caso de uso del negocio: Reservar Pase.	38
2.3.4.2	Diagrama de actividad del caso de uso del negocio: Reservar Pase.	39
2.3.4.3	Diagrama de clases del modelo de objetos del caso de uso del negocio: Reservar Pase. .	40
2.3.4.4	Especificación textual del caso de uso del negocio: Solicitar Transporte.	40
2.3.4.5	Diagrama de actividad del caso de uso del negocio: Solicitar Transporte.....	41
2.3.4.6	Diagrama de clases del modelo de objetos del caso de uso del negocio: Solicitar Tpte.	41
2.3.4.7	Especificación textual del caso de uso del negocio: Asignar Transporte.....	42
2.3.4.8	Diagrama de actividad del caso de uso del negocio: Asignar Transporte.	43
2.3.4.9	Diagrama de clases del modelo de objetos del caso de uso del negocio: Asignar Tpte.	43
2.3	Especificación de los requisitos de software.	44
2.3.4	Requerimientos funcionales.	44
2.3.5	Requerimientos no funcionales.	46
2.4	Descripción del sistema propuesto.....	48
2.5	Modelo de casos de usos de sistema.	48
2.5.4	Definición de los actores.	48
2.5.5	Listado de los casos de uso.	49
2.5.6	Diagrama de casos de uso.....	52
2.5.7	Casos de uso expandidos.	53
2.6	Conclusiones.	65
Capítulo 3: Análisis y Diseño del Sistema.....		66
3.1	Introducción.	66
3.2	Modelo de análisis.....	66
3.2.1	Diagrama de clases del análisis.....	66
3.2.1.1	Diagrama de clases del caso de uso “Gestionar Pase”.	67
3.2.1.2	Diagrama de clases del caso de uso “Autenticar”.....	67
3.2.1.3	Diagrama de clases del caso de uso “Imprimir boletines”.....	68
3.3	Modelo de diseño.	68
3.3.1	Clases Base.	68
3.3.1.1	Diagrama de las clases base.	69
3.3.1.2	Diagrama de secuencia.	70
3.3.2	Diagrama de clases del Diseño Web.	70
3.3.2.1	Caso de uso “Gestionar Pase”.	71
3.3.2.2	Caso de uso “Autenticar”.	72
3.3.2.3	Caso de uso “Imprimir boletines”.	73
3.4	Diseño de la base de datos.....	73
3.4.1	Diagrama de clases persistentes.	74
3.4.2	Modelo de datos.	75
3.5	Principios de diseño aplicados.	75
3.5.1	Interfaz de usuario.....	76
3.5.2	Formato de salida de los reporte.....	77
3.5.3	Tratamiento de errores.....	77
3.5.4	Seguridad.	78
3.6	Conclusiones.	78

Capítulo 4: Implementación del Sistema	79
4.1 Introducción.	79
4.2 Diagrama de despliegue.	79
4.2.1 Diagrama de despliegue.	80
4.3 Diagramas de componentes.	80
4.3.1 Diagrama de componentes “Acceso a Datos”.	81
4.3.2 Diagrama de componentes “Componentes Web o código ejecutable”.	82
4.3.3 Diagrama de componentes “Código Fuente”.	83
4.4 Conclusiones.	83
Capítulo 5: Estudio de Factibilidad	84
5.1 Introducción.	84
5.2 Planificación Basada en Puntos de Casos de Uso.	84
5.2.1 Cálculo de Puntos de Casos de Uso sin ajustar.	84
5.2.2 Cálculo de los puntos de casos de uso ajustados.	85
5.2.3 Estimación de esfuerzo a través de los puntos de casos de uso.	87
5.2.4 Calcular el esfuerzo de todo el proyecto (ET).	88
5.2.5 Costo total del proyecto.	89
5.3 Beneficios tangibles e intangibles.	89
5.4 Análisis de costos y beneficios.	90
5.4 Conclusiones.	90
Conclusiones	91
Recomendaciones	92
Referencias bibliográficas	93
Bibliografía	97
Anexos	98
Glosario de Términos	133

Introducción

A raíz de la idea de nuestro comandante en jefe Fidel Castro de crear una universidad de nuevo tipo, nace la Universidad de las Ciencias Informáticas (UCI). Proyecto que tiene como misión: la formación de profesionales, comprometidos con su Patria y altamente calificados en la rama de la informática.

Desde el primer año de vida de la universidad se hizo necesario desarrollar un sistema que resolviera la reservación de transporte para la gestión del pase estudiantil, ya que resultaba demasiado engorrosa hacer su planificación manualmente, dependiendo de un trabajo exhaustivo y no exento de errores. Dicha aplicación resolvió en gran medida las metas propuestas en ese momento.

Trascurridos ya cuatro cursos, desde el surgimiento de la UCI e incrementado considerablemente su población. Esta, la mayor residencia estudiantil del país, es prácticamente una ciudad, contando en su recinto cerca de diez mil estudiantes. Dadas estas circunstancias, la primera versión del sistema no satisfacía las necesidades actuales y se hace necesaria la incorporación de nuevas funcionalidades, lo que conllevó a la realización de una nueva versión del mismo.

Actualmente se trabaja en la nueva versión del Sistema de Reservación de Transporte, corrigiendo las deficiencias del anterior e incorporándosele nuevas funcionalidades. En este momento el sistema brinda muchas ventajas tanto a estudiantes como a directivos, por ejemplo: carga automáticamente los datos de los familiares, si el estudiante tenía reservaciones previas; también implementa el servicio de impresión de boletines, aspecto bastante engorroso en la anterior versión, puesto que dicho servicio se manejaba con una aplicación independiente al sistema y ocurrían errores frecuentes.

Sin embargo hoy en día el estudiante no puede hacer una programación de su pase, como: reservar para varios fines de semana, o para todos, o para alguno en particular en el período del curso académico; tampoco tiene la posibilidad de configurar sus salidas/entradas de pase (reservación) de forma periódica. Es decir no puede reservar su pase si no está en transcurso la semana correspondiente a este. Por otra parte el sistema no posee documentación alguna, y por ser bastante grande, se hace necesario realizar esta tarea para su posterior mantenimiento.

La no existencia de una solución viable a las necesidades antes expuestas, conlleva a la realización de este trabajo; surgiendo el **problema** de ¿Cómo incorporar nuevas funcionalidades al Sistema de Reservación de Transporte para permitirle a los estudiantes realizar una programación de su pase?

Por lo que el **objeto de estudio** del presente trabajo es los procesos para la reservación de transporte para el pase de fin de semana de los estudiantes de la UCI.

De donde se deriva que el **campo de acción** comprende la automatización del proceso de pases programados por parte de los estudiantes de la UCI.

Los aportes prácticos que se obtienen mediante el presente trabajo son:

1. Facilitación de una nueva herramienta del Sistema de Reservación de Transporte para que los estudiantes puedan programar la planificación de su pase, durante cualquier período del curso académico, acorde a sus necesidades.
2. Proporcionar una documentación para el Sistema de Reservación de Transporte.

Se ha propuesto como **objetivo general** la implementación de un módulo para el Sistema de Reservación de Transporte que permita la programación del pase por parte de los estudiantes de la UCI mediante el uso de herramientas de Software Libre que propicien la integración de dicho módulo al sistema.

Los **objetivos específicos** del presente trabajo son:

1. Definir y fundamentar los principales elementos teóricos - metodológicos para el análisis del problema a resolver.
2. Desarrollar la documentación del Sistema de Reservación de Transporte. (Desarrollar ingeniería inversa al Sistema de Reservación de Transporte)
3. Desarrollar el análisis y diseño de los procesos relacionados con el módulo de pases programados.
4. Implementar el módulo de pases programados con las características definidas en los procesos de análisis y diseño para este módulo.

Se plantean entonces un grupo de **tareas** para satisfacer los objetivos y darle solución a la situación problemática planteada:

1. Análisis de las tecnologías que se utilizan para llevar a cabo sistemas como el que se pretende desarrollar, a nivel internacional.
2. Análisis del Sistema de Reservación de Transporte.
3. Estudio de las principales metodologías para el desarrollo de la documentación de un software.
4. Selección de la metodología de desarrollo de sistemas informáticos, que facilite la creación y garantice la calidad del módulo.
5. Estandarización en la implementación de la herramienta, para su total compatibilidad con el sistema.

Los métodos científicos utilizados en el transcurso de la investigación permitieron apreciar el problema con objetividad, integralmente, en todas sus dimensiones. La entrevista al cliente es esencial pues define qué es lo que el sistema debe hacer, identificando las funcionalidades requeridas por el mismo y las restricciones que se imponen. El método de la modelación resultó imprescindible por el gran peso que tienen los modelos en el trabajo, donde la realidad es representada a través de un modelo que guía todo el proceso hasta la obtención del producto esperado, el cual será capaz de resolver los problemas planteados en la gestión de pases programados para el Sistema de Reservación de Transporte.

Finalmente se obtuvo una herramienta a la altura de las exigencias actuales de producción de software, acorde con los estándares utilizados en el sistema. Proporcionando de tal manera nuevas funcionalidades a favor de la comunidad universitaria.

El contenido de este trabajo se encuentra estructurado en cinco capítulos de la siguiente manera:

En el Capítulo 1, “Fundamentación Teórica”, se realiza una descripción detallada del objeto de estudio, se valoran sistemas existentes vinculados al campo de acción, se enuncian las tendencias y tecnologías actuales seleccionadas a emplear en el desarrollo de la propuesta y por qué su utilización.

El capítulo 2, “Características del Sistema”, 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.

El capítulo 3, “Análisis y diseño 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.

En el capítulo 4, “Implementación 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.

En el capítulo 5, “Estudio de Factibilidad”, se estudia la factibilidad económica para este proyecto. Se analizan los diferentes criterios que influyen en el cálculo del esfuerzo, tiempo de desarrollo y costo del proyecto, determinando si es factible o no el desarrollo del software propuesto.

Capítulo 1 Fundamentación Teórica

1.1 Introducción.

Este capítulo contiene una perspectiva de los principales aspectos relacionados con los servicios de reservación de transporte y problemas específicos que fundamentan la propuesta de solución. Además de brindar un enfoque general de sistemas automatizados existentes, vinculados al campo de acción y el análisis comparativo de las soluciones existentes con la propuesta dada en este 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.

1.2 Sistemas de Reservas.

Las múltiples ventajas competitivas que para las empresas del sector del transporte, aporta un sistema automatizado de reservas *on-line*, se traducen entre otras, en un considerable aumento de la calidad de los servicios prestados a sus clientes. Algunos de los principales servicios que actualmente se ofrecen a través de estos sistemas consisten en poner a disposición del usuario información sobre todos los posibles servicios de transportación disponibles; reservas; listas de espera de plazas en vuelos, barcos, autos; tarifas ya sean confidenciales o normales; confeccionar agendas, cambios y anulaciones; emisión de billetes y tarjetas de embarque; billete electrónico (el cliente no lleva ningún documento, aparece su reserva con un número de código y obtiene su tarjeta de embarque en el aeropuerto, lo que conlleva a un ahorro operativo y de papel); chip-card (tarjeta electrónica con soporte magnético del estilo Visa).

1.2.1 Czech Transport.com.

Sistema de Reservación de Transporte en la República Checa. Facilita desde el aeropuerto Ruzyne transportación segura, barata, rápida y confiable, operando los siete días de la semana las veinticuatro horas del día. Los choferes conducen coches cómodos y hablan idioma inglés.

Capítulo 1 Fundamentación Teórica

El Sistema de Reservación de Transporte en la República Checa permite reservar el traslado desde el aeropuerto Ruzyne hacia cualquier hotel de Praga. La base de datos de este sistema contiene más de 800 hoteles de Praga. Permite calcular **on-line** el precio de la transportación. Permite además la reservación de transporte para el traslado desde el aeropuerto Ruzyne a cualquier ciudad de la República Checa.

Ventajas de Czech Transporte.com:

- Fácil de reservar a través del sistema de reservación en línea.
- Posee los mejores precios garantizados.
- No se requiere tarjeta de crédito.
- No hay necesidad de buscar taxis en el aeropuerto con tarifas inseguras.
- Permite conocer el precio por adelantado.
- Facilidad de pago. El cliente puede pagar en Coronas Checas, Dólares Americanos o Euros.
- Contacto **on-line** para brindar información adicional.

El sistema muestra a los clientes una lista de los puntos hacia los cuales el cliente desea realizar reservación, le brinda además la posibilidad de que el cliente especifique la cantidad de pasajeros que viajarán junto a él (como máximo 7). El usuario puede calcular el costo de la transportación y ver el lugar exacto en que se encuentra el punto al que desea trasladarse.

1.2.2 easyJet.com.

Sistema de Reservación de Transporte **on-line** con ubicación en Reino Unido. El sistema de reservación de transporte easyJet tiene como objetivo principal proporcionar a sus clientes servicios aéreos seguros y a un buen precio, desarrollar y ofrecer un producto y unas tarifas coherentes y de confianza que sean atractivas tanto para el mercado turístico, como para el de los viajes de negocio, en una serie de rutas europeas. EasyJet realiza 289 rutas entre 74 importantes aeropuertos del Reino Unido, Francia, España, Suiza, los Países Bajos, Dinamarca, Italia, la República Checa, Grecia, Alemania y Portugal. Easyjet opera desde 1995 y en continua expansión hasta ser hoy día una de las mejores compañías de reservación de transporte **on-line**.

Capítulo 1 Fundamentación Teórica

Ventajas de easyJet.com:

- Tarifas para reservaciones de vuelos muy reducidas.
- Gran cantidad de conexiones.
- Flexibilidad (ofertas de trayectos de ida, cambios de vuelo...)

Ventajas de reservas **on-line** directa del cliente (más barato) y posibilidad de cambiar en línea las reservas.

1.2.3 Avis.

El sistema de reservación de transporte nace en 1996 cuando Avis (nacida en 1946 en Detroit, EE.UU) se convierte en la primera compañía de alquiler de vehículos que disponía de una página Web en Internet: www.avis.com la cual permite el acceso al usuario de internet a información relevante acerca del servicio de alquiler, en cualquier punto del mundo, así como tramitar y confirmar personalmente su reserva. El sistema Avis es el sistema informatizado de gestión de reservas y alquileres tecnológicamente más avanzado dentro del sector de alquiler de vehículos. Las más de 16.000 terminales con que cuenta el sistema permiten:

- Confirmar más de 100.000 reservas diariamente desde o para cualquier oficina de Avis a nivel mundial.
- Realizar el procesamiento de más de 2,5 millones de transacciones diarias.
- Facilitar a los usuarios información instantánea sobre cualquier aspecto relacionado con el alquiler y/o servicios brindados.

El sistema automatizado de reservación de transporte Avis se encuentra conectado directamente a los sistemas de reservas informáticos mas importantes del mundo como son: Amadeus, Galileo, Sabre, Apollo, Sita, System One y Worldspan, lo que permite el acceso al sistema en tiempo real a más de 40.000 terminales de agencias de viaje, y departamentos de empresas a nivel mundial.

Capítulo 1 Fundamentación Teórica

Ventajas de Avis Rent a Car:

- El sistema Avis en conjunto con la empresa ofrece una amplia gama de coches para elegir el que mejor se acoge al viaje de usuario.
- Un servicio de calidad que garantiza un viaje en toda tranquilidad.
- Una oficina en cada aeropuerto, para la facilitación de los traslados del usuario.

1.2.4 3 Milenio.

3 Milenio es una nueva iniciativa pensada para todos aquellos que están interesados en vender uno de los destinos turísticos más solicitados y gustados del mundo: el Caribe. Por diversas razones, los principales sistemas de reservas del mundo venden muy pocos servicios relacionados con Cuba. Mediante 3 Milenio los usuarios pueden gestionar las reservaciones a más de 150 hoteles de todas las categorías y en todos los rincones de la isla, así como vuelos domésticos, autos, paquetes de vacaciones, eventos, excursiones, autocaravanas y otros que conforman el producto turístico cubano.

Ventajas de 3 Milenio:

- Mayor eficiencia y rapidez en la gestión de reservaciones.
- Ofrece un control en tiempo real de todas las disponibilidades, precios y ofertas especiales.
- Información completamente actualizada sobre los servicios turísticos en Cuba.
- Cálculo del precio exacto de todos los productos.
- Búsqueda profesional haciendo uso de múltiples criterios.
- Posibilidad de vender ofertas de última hora.
- Fácil de usar.
- Disponible las 24 horas, los 365 días del año.
- Se puede utilizar desde cualquier lugar a través de Internet.

Para informar detalladamente a sus clientes de los servicios que puede comprar, 3 Milenio cuenta con la más actualizada fuente de información turística de Cuba: el directorio turístico de Cuba. Actualizado semanalmente, contiene fotos, y la descripción de todos los hoteles de Cuba, así como de las más

Capítulo 1 Fundamentación Teórica

importantes atracciones turísticas de la Isla Grande. Un avanzado sistema de mapas incorporado en 3 Milenio le permite ubicar cualquier servicio o atracción turística.

1.2.5 Sistema de Reservación de Transporte (versión inicial).

El Sistema de Reservación de Transporte (versión inicial) desarrollado en el área de informatización de la UCI, con la finalidad de permitir la reservación de transporte para el pase de fin de semana, por parte de los universitarios, desde cualquier ordenador conectado a la red interna. Este permitía al usuario realizar modificaciones sobre una reservación realizada con anterioridad, es decir el usuario tenía la posibilidad de volver a formular su reservación y por ende volver a especificar los datos de los lugares para donde reservaría el transporte. El sistema brindaba la posibilidad de cancelar una reservación previamente realizada, en caso de que el usuario se hubiese arrepentido de dicha reservación.

El hecho de que esta aplicación resolviera uno de los grandes problemas existentes en la universidad no establecía un punto final a las necesidades. No cumplía con todos los aspectos necesarios que se requerían, con el transcurso del tiempo comenzaron a aparecer deficiencias en el software. Algunas de las insuficiencias encontradas en la primera versión fueron:

- El sistema no era capaz de brindarle la posibilidad al administrador de modificar las rutas ya existentes. Si era necesario realizar alguna modificación de las rutas ya existentes o agregar algunas tantas más había que realizarlo de forma manual, entrar directamente al código de la aplicación y cambiarlas/agregarlas. Las rutas eran totalmente estáticas. La aplicación no soportaba varios puntos de recogida dentro de una misma ruta.
- Los boletines para abordar los ómnibus de la transportación eran generados por una aplicación externa de escritorio que tenían los directivos instalado en sus ordenadores. Esto traía consigo la necesidad de tener una persona con conocimientos para dar mantenimiento al software de impresión en caso de que el mismo presentara problemas.
- La tarea de dar las baja de la alimentación (a los estudiantes reservados) para ese fin de semana, era de forma manual, siendo esta labor compleja y rigurosa por parte de los implicados.

Capítulo 1 Fundamentación Teórica

- Sistema desarrollado en Perl y gestor de base de datos Microsoft SQL Server. Haciéndose complejo el mantenimiento de la aplicación por la complejidad y falta de desarrolladores en el lenguaje empleado.

1.3 Tendencias y tecnologías actuales.

Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio de las tendencias y tecnologías actuales posibles a emplear, descritas a continuación.

1.3.1 Aplicaciones Web.

Las aplicaciones Web son una especialización y concreción de las aplicaciones Cliente/Servidor, o sea, su arquitectura general es la de un sistema Cliente/Servidor, donde tanto el cliente (navegador) como el servidor (servidor Web), y el protocolo mediante el cual se comunican (el HTTP: Hyper Text Transfer Protocol) son un estándar y no deben ser creados por el desarrollador de aplicaciones. [1]

La parte del cliente de las aplicaciones Web 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 (JavaScript, VBScript, PerlScript) o mediante pequeños programas (applets) en Java. 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. [1]

La creciente popularidad de las aplicaciones Web se debe a sus múltiples ventajas, entre las cuales podemos destacar: [2]

Multiplataforma: con un solo programa, un único ejecutable, nuestras aplicaciones 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 solo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.

Capítulo 1 Fundamentación Teórica

Suave curva de aprendizaje: los usuarios, como emplean la aplicación a través de un navegador, hacen empleo del sistema tal como si estuvieran navegando por Internet, debido a esto el acceso es mucho más intuitivo.

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, desde una agenda electrónica, desde un celular, desde la oficina, desde la casa o cualquier otra parte del mundo.

El desarrollo de aplicaciones Web está siendo empleado en muchas organizaciones, ésta situación continuará en ascenso indefinidamente. Es por ello que día a día se requieren más programadores capacitados para desarrollos basados en la World Wide Web (WWW).

No obstante a la serie de ventajas que presenta tiene además algunas desventajas, las cuales se refieren 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 obtenida la reacción del sistema.

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.

Capítulo 1 Fundamentación Teórica

1.3.1.1 Modelo Cliente/Servidor.

Es conocido que 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. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes: [3]

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, solo 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 de la arquitectura Cliente/Servidor: [4]

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte entre los clientes.
- Se reduce el tráfico de la red sustancialmente. El cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando libre la red.

Las arquitecturas de dos capas contienen tres componentes distribuidos en dos capas: cliente (solicitante de servicios) y servidor (proveedor de servicios). [Anexo 1]. Los tres componentes se reflejan a continuación: [4]

- Interfaz de usuario al sistema. Tales como una sesión, entradas de texto, desplegados de menús, etc.
- Administración de procesamiento. Tales como la ejecución de procesos, el monitoreado de los mismos y servicios de procesamiento de recursos.
- Administración de bases de datos. Tales como los servicios de acceso a datos y archivos.

Capítulo 1 Fundamentación Teórica

La arquitectura de software de tres capas surgió en la década de los noventa para solucionar las limitaciones de la arquitectura de dos capas. La tercera capa (capa de servicios) se localiza entre la interfaz de usuario (cliente) y el administrador de datos (servidor). Esta capa intermedia provee de servicios para la administración de procesos, los cuales son compartidos por múltiples aplicaciones.

[\[Anexo 2\]](#)

1.3.2 PHP (Hyper Text Preprocessor).

El PHP (acrónimo de "PHP: Hyper Text Preprocessor"), es un lenguaje interpretado de alto nivel embebido en páginas HTML (acrónimo de "HTML: Hyper Text Markup Language") y ejecutado en el servidor.

El PHP inició como una modificación a Perl escrita por el programador Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quien visitaba su currículum vitae en su Web. En los siguientes tres años, se fue convirtiendo en lo que se conoce como PHP/FI 2.0. Esta forma de programar llegó a muchos usuarios, pero el lenguaje no tomó el peso actual hasta que dos programadores de Israel Zeev Surasky y Andi Gutmans le incluyeron nuevas características en 1997, que dio por resultado el PHP 3.0. En el invierno de 1998, poco después del lanzamiento oficial de PHP 3.0, Andi Gutmans y Zeev Surasky comenzaron a trabajar en la reescritura del núcleo de PHP. Los objetivos del diseño fueron mejorar la ejecución de aplicaciones complejas, y mejorar la modularidad del código base de PHP. Estas aplicaciones se hicieron posibles por las nuevas características de PHP 3.0 y el apoyo de una gran variedad de bases de datos y APIs de terceros, pero PHP 3.0 no fue diseñado para el mantenimiento tan complejo de aplicaciones eficientemente. [\[5\]](#)

El nuevo motor, apodado Zend Engine 1.0 o 'Motor Zend' (comprimido de sus apellidos, Zeev y Andi), alcanzó estos objetivos de diseño satisfactoriamente, y se introdujo por primera vez a mediados de 1999. PHP 4.0, basado en este motor, y acoplado con un gran rango de nuevas características adicionales, fue oficialmente liberado en Mayo de 2000, casi dos años después que su predecesor, PHP 3.0. Además de la mejora de ejecución de esta versión, PHP 4.0 incluía otras características clave como el soporte para la mayoría de los servidores Web, sesiones HTTP, buffers de salida, formas más seguras de controlar las entradas de usuario y muchas nuevas construcciones de lenguaje. A pesar de que la potencia de PHP está más que demostrada en la versión 4, arrastra una serie de limitaciones en el soporte de clases y

Capítulo 1 Fundamentación Teórica

objetos desde que fue implementado en PHP 3. Es por esto que sus desarrolladores deciden trabajar en ello y en verano de 2004 publican PHP 5 con el nuevo Zend Engine 2.0. Esta vez sí, PHP es un lenguaje con un auténtico motor de objetos, si bien basa su funcionamiento en el uso de funciones. La versión más reciente de PHP es la 5.2.0 que incluye todas las ventajas que provee el nuevo Zend Engine II, la cual por su gran cantidad de correcciones, mejoras de seguridad y nuevas características, es un cambio significativo en la rama 5.x. Tanto que PHP 5.2.0 quien también incluye el novedoso PDO (PHP Data Objects) finalmente hace obsoletas a todas las anteriores versiones 5.1.x. Según estudios realizados, poco más de un millón de servidores tienen esta capacidad implementada y los números continúan en ascenso. [5]

Una de las características más poderosas del PHP es el soporte nativo incluido para una gran cantidad de bases de datos. Entre las bases de datos soportadas se encuentran: MySQL, Oracle, PostgreSQL, MSSQL Server, DBM, DBase, Frontbase, filePro, SQLite entre otras. PHP ofrece además una potente integración con muchas bibliotecas externas, las cuales permiten que el desarrollador realice prácticamente cualquier cosa desde generar documentos en formato PDF (Portable Document Format) hasta analizar código XML (Extensive Markup Language), permite además mediante su integración con la librería GTK+ crear programas GUI (Graphics User Interface, al igual que los programas de Visual Basic o Delphi). Dichos programas creados con PHP-GTK son altamente portables entre plataformas. [5]

La principal ventaja de PHP viene dada por ser un lenguaje multiplataforma, funciona tanto para Unix (con el popular Apache) como para Windows (con la versión de Apache para el mismo o el servidor WWW incorporado Internet Information Server) permitiendo que el código que haya sido creado para cualquiera de las plataformas sea completamente funcional en la otra y no haya por que realizarse modificación alguna.

Es software completamente libre, lo cual implica menos costes y servidores WWW mucho mas baratos que alternativas propietarias. Es sumamente rápido y su integración con Sistemas Gestores de Bases de Datos libres (MySQL, PostgreSQL) y servidores igualmente libres como Apache lo hacen erguirse como una de las alternativas más económicas y eficientes del actual mercado. [5]

Capítulo 1 Fundamentación Teórica

La sintaxis de PHP está en gran parte inspirada en el potente lenguaje C, la cual ha sido ligeramente modificada para poderlo adaptar al entorno en que trabaja, de manera que a los desarrolladores que están familiarizados con C les sea mas fácil entenderlo. [5]

PHP posee una de las más grandes comunidades en Internet, de esta manera no se hace nada difícil encontrar a lo largo de toda la red cualquier tipo de ayuda, documentación, noticias, artículos, código fuente y muchos más recursos. [5]

Debido a su extensa distribución PHP está siendo sostenido por una inmensa comunidad de programadores. Siendo software de fuente totalmente abierta, PHP se fortalece cada vez más de la ayuda de un gran grupo de desarrolladores, logrando así que los problemas de funcionamiento sean encontrados y reparados en el menor tiempo posible. El código fuente se mantiene al día de forma continua con mejoras y extensiones del mismo para lograr una ampliación de las capacidades del lenguaje. [5]

La funcionalidad de PHP se describe a través de los siguientes pasos. [6]

- Escritura en las páginas HTML pero con el código de PHP embebido.
- Guardar la página creada en el servidor WWW.
- Un usuario mediante un navegador solicita una página al servidor WWW.
- El servidor interpreta en código PHP embebido en la página solicitada.
- El servidor envía de vuelta al usuario que realizó la petición el resultado del conjunto de código HTML y el resultado de la interpretación del código PHP incrustado que es también HTML.

Bajo ningún concepto el servidor envía al navegador del usuario que realiza la petición código PHP, todas las operaciones son realizadas de forma transparente al usuario, el código PHP es ejecutado en el servidor por el motor PHP y solo el resultado de dicha ejecución es enviado en forma de HTML al navegador del usuario. El resultado es lo que se conoce como una página puramente HTML, de esta forma el usuario aparentemente estará visitando una página HTML que dichas páginas son interpretadas por cualquier navegador. [6]

Por ser PHP un lenguaje ejecutado en el servidor no se hace necesario que el navegador lo soporte, es totalmente independiente del navegador empleado por cualquier usuario, no siendo así para el

Capítulo 1 Fundamentación Teórica

funcionamiento de las páginas alojadas en el servidor, este si debe soportar PHP para un correcto funcionamiento de dichas páginas. Además de que podemos encontrarlo de forma libre en el mercado y podemos acceder a él mediante Internet. [6]

Después de transcurridos seis largos años, y después de una profunda revisión realizada por la comunidad de desarrolladores a todas las cosas buenas que ha dejado PHP, se han realizado cambios estructurales en el lenguaje ofreciendo innovaciones en PHP 5 y solucionando muchos problemas encontrados en su antecesor PHP 4.

Favorablemente, lo actual incluido a PHP 5 mejora muchas e importantes áreas en el lenguaje y también su ejecución, por ejemplo: [7]

- Soporta una compleja Programación Orientada a objetos (OPP).
- XML.
- Integración nativa con Zend Engine.

Los desarrolladores de PHP 5 han realizado un cambio substancial en el tratamiento de las variables objeto: en PHP 5 todas las variables que nombran objetos son más bien referencias. No es necesario emplear el operador ‘&’ ni en asignaciones, ni en el paso de parámetros que son objetos, ahorrándose con ello gran cantidad de errores. [6]

La principal novedad recogida en las clases PHP 5 es la inclusión de modificadores de control de acceso con los cuales implementar la ocultación de información, principio básico en la programación orientada a objetos de la cual carecía PHP 4.

PHP 5 introduce tres palabras claves (*private*, *protected* y *public*) las cuales surgen para sustituir a la palabra clave *var* en la definición de variable miembro —atributos— de las clases, y que anteceden a la definición de funciones miembro —métodos—.

Otros lenguajes como Perl (Practical Extraction and Report Language), ASP (Active Server Pages) y JSP (Java Server Pages) tienen características muy similares a PHP aunque además poseen características especiales de cada uno de ellos mediante las cuales son distinguidos, entre ellas se pueden destacar: [7]

Capítulo 1 Fundamentación Teórica

Características multiplataforma: con excepción de ASP (quien es solamente soportado por la plataforma Windows de Microsoft) los demás lenguajes están actualmente soportados en múltiples y diversas plataformas.

Velocidad de ejecución: la mayor velocidad de ejecución es alcanzada por PHP, seguido por PERL, quien a su vez es seguido por JSP.

Disponibilidad de recursos: en la actualidad los lenguajes más utilizados en toda la Internet son el PHP y el JSP, siendo los más empleados en la publicación de artículos y código fuente de ejemplo. PHP posee una de las comunidades más grandes en Internet, de igual envergadura que la comunidad de Java.

Familiaridad con el lenguaje: en la universidad los lenguajes más utilizados por los desarrolladores de aplicaciones WWW son el ASP y el PHP.

De acuerdo a estas comparaciones realizadas, el PHP resulta ser un lenguaje favorito, siendo este el lenguaje empleado en el actual sistema (Sistema de Reservación de Transporte [ver 2.0]), por lo que es preciso utilizarlo en la implementación de la propuesta de solución del presente trabajo.

1.3.3 Servidor Web Apache.

Un servidor Web no es más que un programa especial que permite al usuario acceder a páginas Web residentes en un computador. Actualmente Apache es el servidor Web más utilizado en todo el mundo, encontrándose por encima de todos sus competidores tanto gratuitos como comerciales. Es un software de código completamente abierto el cual funciona sobre cualquier plataforma ya sea Windows, Linux, Unix, entre otras. Desde su surgimiento ha estado en constante evolución hasta convertirse en uno de los mejores servidores Web en términos de eficiencia, velocidad de procesamiento y funcionalidad, Apache surge en abril de 1996 y ya en julio del 2002 era utilizado por el 57% de los sitios Web de Internet. [8]

Apache posee la capacidad de prestar servicio tanto para el alojamiento de páginas con contenido estático, función para la cual sería útil un antiguo computador Intel 486, como de contenido dinámico mediante el uso de otras herramientas que facilitan la actualización de los contenidos de las páginas Web

Capítulo 1 Fundamentación Teórica

como lo son: bases de datos, ficheros o cualquier otra fuente de información, es altamente potente y a la vez completamente configurable. [8]

Los servidores Web no hacen otra tarea que suministrar páginas Web a los navegadores que lo soliciten. Los servidores Web mantienen soporte para Protocolo de Transferencia de Hyper Texto HTTP, el estándar de Internet para las comunicaciones Web. Mediante el empleo de HTTP un servidor Web envía páginas Web en HTML y CGI (Common Gateway Interface) así como otro tipo de información al navegador del cliente cuando este lo requiera. Cuando un usuario acciona el clic sobre un enlace Web es inmediatamente enviada una solicitud al servidor con el objetivo de localizar los datos nombrados por ese enlace. El servidor Web recibe esta solicitud y emite los datos que le han sido solicitados o devuelve un mensaje de error. [8]

El servidor Apache es un gran software que está completamente estructurado en módulos, está dividido en muchos fragmentos de código quienes hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada pues la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos de Apache son clasificables en tres categorías: [8]

- Módulos Base: módulo con las funcionalidades básicas de Apache.
- Módulos Multiproceso: son los que se responsabilizan de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- Módulos Adicionales: cualquier otro módulo que le añada algún tipo de funcionalidad al servidor Web.

Las funcionalidades elementales de Apache se encuentran en el Módulo Base, siendo necesario un Módulo Multiproceso para manejar las peticiones. Se han diseñado varios Módulos Multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando así la rapidez, la funcionalidad y el rendimiento del código. [8]

Capítulo 1 Fundamentación Teórica

El resto de las funcionalidades del servidor se consiguen mediante módulos adicionales que se pueden cargar. Si se desean añadir un conjunto de utilidades al servidor simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software. [8]

1.3.4 AJAX.

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asincrónicos), es una actual técnica de desarrollo Web mediante la cual se crean aplicaciones en gran medida interactivas. Dichas aplicaciones son ejecutadas en el cliente, lo que es lo mismo en el navegador del usuario, y mantiene comunicación asincrónica con el servidor en un segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa un aumento considerable de la velocidad en la misma. [9]

AJAX no constituye una tecnología en si, sino que es más bien un término mediante el cual se engloba a un grupo de tecnologías que trabajan en conjunto y las cuales son mostradas a continuación: [9]

- XHTML (o HTML) y hojas de estilo en cascada (CSS) para el diseño que acompaña a la información reflejada en las páginas Web.
- DOM (Document Object Model) accedido con un lenguaje de script por parte del usuario, específicamente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada al cliente.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se hace empleo de un objeto llamado IFrame en lugar de XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar correctamente, incluyendo HTML pre-formateado, texto plano, JSON y hasta EBML.

Existen un gran número de diferencias significativas entre las aplicaciones Web tradicionales y las aplicaciones desarrolladas mediante AJAX así como sus patrones de interacción sincrónica para una aplicación Web tradicional y asincrónica para una aplicación AJAX. [Anexos 3 y 4]

1.3.4.1 Ventajas de AJAX.

Interactividad

Las aplicaciones AJAX son ejecutadas en el cliente, es decir en los ordenadores de los usuarios, manipulando la página actual dentro de sus navegadores usando métodos de Document Object Model. Puede ser empleado para multitud de tareas como lo son actualizar o eliminar registros, expandir formularios Web, devolver peticiones simples de búsqueda o editar árboles de categorías; todo sin tener que recargar la página HTML cada vez que el usuario realiza un cambio. Generalmente solo requiere enviar pequeñas peticiones al servidor, y se devuelven respuestas relativamente cortas. Esto permite el desarrollo de aplicaciones Web con interfaz de usuario más interactivas gracias al uso de las técnicas DHTML. [10]

Portabilidad

Las aplicaciones construidas con AJAX emplean características sumamente documentadas presentes en todos los navegadores importantes en la mayoría de las plataformas existentes. Esta situación bien pudiera cambiar en el futuro, en este momento los usos de AJAX son efectivos entre plataformas. Mientras que la plataforma de AJAX está más restringida que la plataforma de Java, las aplicaciones actuales de AJAX llenan con eficacia la parte de los Java Applets: ampliar el navegador con mini-aplicaciones ligeras. [10]

1.3.4.2 Desventajas de AJAX.

Críticas de uso

Una de las mayores críticas de las que ha sido objeto el uso de AJAX en aplicaciones Web es que puede fácilmente acabar con el comportamiento normal del botón atrás del navegador. Otro problema relacionado a todo esto del dinamismo es que las actualizaciones dinámicas hacen muy difícil al usuario agregar a los marcadores/favoritos un momento en particular de la aplicación. [11]

Capítulo 1 Fundamentación Teórica

JavaScript

Aunque AJAX no necesita ningún tipo de plug-in agregado para el navegador, requiere que los usuarios tengan el JavaScript activado. Esto se aplica a todos los navegadores por igual que soporten esta tecnología excepto para el Microsoft Internet Explorer 6 y sus antecesores los cuales necesitan también tener el ActiveX activado, ya que el objeto XMLHttpRequest está implementado junto con el ActiveX en este navegador.

Como ocurre con las aplicaciones DHTML, las de AJAX deben ser probadas rigurosamente para adaptarse a los diferentes navegadores y plataformas. [11]

1.3.5 Patrones de Diseño.

Un patrón es un modelo el cual se sigue para realizar algo. Los patrones surgen de la experiencia de los seres humanos de tratar y a la vez lograr ciertos objetivos. Los patrones capturan la experiencia existente y probada para promover nuevas prácticas. [12]

Los Patrones de Diseño (Design Patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo del software y otros ámbitos referentes al diseño de interacción o interfaz. [12]

Estos se encuentran divididos en tres grandes categorías: [12]

- Patrones Creacionales: solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación.
- Patrones Estructurales: solucionan problemas de composición (agregación) de clases y objetos.
- Patrones de Comportamiento: soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

Un Patrón de Diseño es: [12]

- Una solución estándar para un problema común de programación.
- Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.

Capítulo 1 Fundamentación Teórica

- Un proyecto o estructura de implementación que logra una finalidad determinada.
- Un lenguaje de programación de alto nivel.
- Una manera más práctica de escribir ciertos aspectos de la organización de un programa.
- Conexiones entre componentes de programas.
- La forma de un diagrama de objeto o de un modelo de objeto.

Los patrones de diseño han contribuido grandemente a la flexibilidad. Y en adición, han demostrado ser una forma muy útil de reutilizar diseño, ya que ellos no solo nombran, abstraen e identifican aspectos claves de estructuras comunes de diseño, sino que generalmente son descritos en una forma específica documental, haciendo su comprensión y aplicación fácil para el conjunto de desarrolladores.

Se pueden destacar que los beneficios que produce un patrón pueden ser medidos en varios aspectos:

Contribuyen a reutilizar diseños, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones. La importancia de la reutilización del diseño no es despreciable, ya que esta provee numerosas ventajas: reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad, eficiencia y consistencia de los diferentes diseños, y proporciona un considerable ahorro en la inversión.

Mejoran (aumentan, elevan) la flexibilidad, modularidad y extensibilidad, factores internos e íntimamente relacionados con la calidad percibida por el usuario.

Los patrones incrementan las diferentes gamas de diseño, ayudando a diseñar desde un mayor nivel de abstracción.

1.3.6 Modelo Vista Controlador (MVC).

El Modelo Vista Controlador (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 [\[Anexo 5 y 6\]](#). Este patrón se 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. [\[13\]](#)

Capítulo 1 Fundamentación Teórica

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. [13]

El **Modelo** es completamente acceso a datos, y las funciones que llevan a lo que comúnmente llamamos “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. Teniendo 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 (un clic, un cambio de campo de texto, la opresión de una tecla, un tiempo prefijado, etc.). 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.

1.3.7 Sistema Gestor de bases de datos.

Un Sistema de Gestión de Bases de Datos (SGBD) podría definirse como un paquete generalizado de software que es ejecutado en un ordenador anfitrión, quien centraliza los accesos a los datos y actuando de interfaz mediando entre los datos físicos y el usuario que realiza la petición. Las principales funciones

Capítulo 1 Fundamentación Teórica

que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de acceso, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias, inconsistencias, tratar la concurrencia y mantener la integridad de los datos almacenados. [14]

Un SGBD posee los siguientes objetivos específicos: [14]

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia de los datos almacenados.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de la información.
- Control centralizado.

La información es representada a través de tuplas, las cuales describen el fenómeno, proceso o ente de la realidad objetiva que está siendo analizada y es representada a través de tablas. [14]

Entre los SGBD más comúnmente empleados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Internase, DB2, entre otros. Todos los aquí expuestos presentan un enfoque relacional con una excelente base matemática centrada en el Álgebra Relacional. [14]

Todos los SGBD expuestos anteriormente facilitan el trabajo con la base de datos y poseen características comunes que los diferencian, por ejemplo: [14]

Oracle: conocido como uno de los mejores gestores del mundo requiere de una licencia para su posterior empleo, es necesario pagar un alto precio para poder hacer uso de este SGBD.

Microsoft SQL Server: no es multiplataforma, solo puede ser empleado en conjunto al sistema operativo Windows que es propiedad de la gran compañía Microsoft Corp.

Capítulo 1 Fundamentación Teórica

MySQL: es un SGBD de fuente abierta, especial para aplicaciones de pequeña y de mediana envergadura.

PostgreSQL: es también un SGBD de fuente abierta con mayores prestaciones que MySQL, incluye además gran potencia para aplicaciones de mayor tamaño. Soporta un subconjunto de SQL92 mayor que el soportado por MySQL.

Como SGBD es seleccionado PostgreSQL por las múltiples ventajas que este ofrece con respecto a los demás gestores y por compatibilidad con el sistema actual.

1.3.7.1 PostgreSQL.

PostgreSQL es un SGBD Objeto-Relacionales (ORDBMS) libre, liberado bajo la licencia de BSD (Berkeley Software Distribution). Es una alternativa mucho más eficaz a otros sistemas de bases de datos de código abierto (MySQL, Firebird y MaxDB), así como a sistemas propietarios como Oracle y Microsoft SQL Server. PostgreSQL ha sido desarrollado de varias formas desde el año 1997. En 1994, Andrew Yu y Jolly Chen añadieron un intérprete de lenguaje SQL a Postgres. PostgreSQL fue lanzada a la WWW para encontrar su propio hueco en el mundo como un hijo de dominio público y código abierto del código original Postgres de Berkeley. [15]

Para el año 1996, debido a un nuevo esfuerzo de código abierto y a la gran funcionalidad del software, Postgres fue renombrado a PostgreSQL. El proyecto PostgreSQL sigue siendo actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores del código abierto. [15]

PostgreSQL es ampliamente considerado como el sistema de bases de datos de fuente abierta más avanzado del mundo. Posee un gran número de características que históricamente solo pueden ser vistas en productos comerciales de alta confiabilidad. [15]

A continuación se refleja una lista de algunas de las principales características, a partir de PostgreSQL 7.1.x. [16]

Capítulo 1 Fundamentación Teórica

- DBMS Objeto-Relacional: PostgreSQL aproxima grandemente los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte para multi-usuario, transacciones, consultas optimizadas, herencia, arreglo, etc.
- Altamente Extensible: PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.
- Soporte SQL comprensivo: PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (join) SQL92.
- Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- Lenguajes Procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este es comparable al lenguaje procedural de Oracle, PL/SQL.
- Otra de las tantas ventajas de PostgreSQL es su habilidad para usar Perl, Python, o TCL (Tool Command Language) como lenguaje procedural embebido.

1.3.7.2 MVCC.

MVCC, o lo que es lo mismo Control de Recurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL emplea para evitar bloqueos innecesarios. Los bloqueos son provocados por usuarios que están escribiendo en la base de datos. Realizando un resumen, el lector se encuentra bloqueado debido a los escritores que se encuentran actualizando registros de forma simultánea a la lectura. [16]

Mediante el uso de MVCC, PostgreSQL evita este problema completamente. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los distintos usuarios de la base de datos. PostgreSQL es capaz de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Capítulo 1 Fundamentación Teórica

1.3.7.2.1 Cliente/Servidor.

PostgreSQL usa una arquitectura proceso-por-usuario Cliente/Servidor. Esta es similar al método del Apache 1.3.x para realizar manejo a los procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

A continuación se ofrece una lista con algunas de las principales mejoras que incluye PostgreSQL: [\[16\]](#)

- Los bloqueos de tabla han sido sustituidos por el control de la concurrencia multi-versión, el cual permite a los accesos de solo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite además realizar copias de seguridad en caliente desde pg_dump la base de datos permanece disponible para consultas.
- Se han realizado implementaciones de características importantes del motor de datos, incluyendo sub-consultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- Se han agregado características adicionales que cumplen el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadenas literales, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue puesta en el mercado.

1.3.8 Proceso de Desarrollo.

Cada día la producción de software busca adecuarse más a las necesidades del usuario, eso trae consigo que aumenten el tamaño y la complejidad.

Para lograr la productividad del software es necesario un proceso que sea capaz de integrar las múltiples facetas del desarrollo. [\[Anexo7\]](#)

Capítulo 1 Fundamentación Teórica

Se hace necesario definir la metodología de ingeniería del software que guiará el proceso de automatización, se ha escogido el Proceso Unificado de Desarrollo de Software (RUP).

El proceso Unificado de Rational, (Rational Unified Process), fue publicado en el año 1998 como resultado de varios años de trabajo y experiencia. [17]

RUP es un proceso de desarrollo de software, más bien, un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. [17]

RUP es un proceso basado en componentes, que emplea el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, todos los verdaderos aspectos definitorios del Proceso Unificado se resumen en que está dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, se parte de casos de uso, es centrado en la arquitectura y es iterativo e incremental. [17]

Está acompañado de una herramienta muy buena que soporta cada uno de los procesos que nos son necesarios: Rational Rose Enterprise Edition 2003.

Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a emplear en el modelo de desarrollo de software.

1.3.8.1 Lenguaje Unificado de Modelado (UML).

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. [17]

Los creadores pretendieron mediante la creación de este lenguaje, unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar.

Capítulo 1 Fundamentación Teórica

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en forma convencional y fácil de comprender para comunicar todas las personas que estén vinculadas en el desarrollo del software, esto es llevado a cabo mediante un conjunto de símbolos y diagramas. [17]

El UML 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. [17]

Seguidamente se exponen de forma general las principales características: [17]

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y Cliente/Servidor.

Existen varias herramientas CASE (Computer-Aided Systems Engineering), que dan asistencia a analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software, pero es Rational Rose líder en el modelado del desarrollo de los proyectos y es esta precisamente la que se emplea en la modelación de este proyecto. La herramienta fue desarrollada por los creadores de UML, empleando la notación estándar en la arquitectura de software. Esta herramienta integra todos los elementos que proponen la metodología RUP para cubrir el ciclo de vida de un proyecto. [17]

1.3.9 Herramientas utilizadas en el desarrollo.

Para realizar la propuesta de este trabajo se hace indispensable tener en cuenta la utilización de algunas herramientas necesarias en el diseño de interfaz, y el modelado del proyecto. Seguidamente haremos referencias a las mismas.

Capítulo 1 Fundamentación Teórica

1.3.9.1 Macromedia Dreamweaver 8.

Macromedia Dreamweaver 8 es uno de los editores de desarrollo Web más utilizados en el mundo hoy por hoy a escala profesional para la creación de sitios Web. Su amplia cantidad de herramientas permite crear desde la más simple página Web hasta el sitio Web más completo y complejo para una pequeña o gran empresa y hacer empleo de todos los recursos de la Web. [18]

Este editor HTML profesional para el diseño, programación y desarrollo de páginas, sitios y aplicaciones Web; permite la edición visual, más explícitamente, permite crear páginas rápidamente sin necesidad de escribir una sola línea de código, así como también la codificación manual. Ayuda además a construir aplicaciones Web dinámicas vinculadas a bases de datos. [18]

Es un software de alta calidad y completamente personalizable. Pueden ser creados objetos y comandos propios del usuario, realizar modificaciones a los accesos directos de teclado, e incluso escribir código JavaScript para extender las capacidades del Dreamweaver con diferentes comportamientos. Soporta varias tecnologías del servidor para la construcción de aplicaciones Web, tales como: Macromedia ColdFusion, Microsoft ASP, Microsoft ASP.NET, Sun JavaServer Pages, JSP y PHP. [18]

1.3.9.2 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. [19]

1.3.9.3 Rational Rose.

Existen herramientas CASE de trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos, en la actualidad la mejor y más empleada en todo el mundo es Rational Rose y es la que se emplea en la modelación de este proyecto. [20]

Capítulo 1 Fundamentación Teórica

Rational Rose cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, contracción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Rose es 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 las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores poder visualizar el sistema completo empleando un lenguaje común, además los diseñadores pueden moldear sus componentes e interfaz en forma individual y luego unirlos con otros componentes del proyecto. [20]

Se decidió que se emplearía Rational Rose Enterprise Edition 2003, para sustentar la documentación, como modelador visual de la notación UML para la confección de los diagramas que se ilustran en nuestro documento. Esta herramienta es sumamente completa y ofrece amplias potencialidades.

1.3.9.4 PgAdmin.

Es desarrollado por una comunidad de los expertos de PostgreSQL de varias partes del mundo y está disponible en poco más de doce lenguas. Se trata de una herramienta para la administración de bases de datos PostgreSQL. Su empleo se puede extender hacia las plataformas de Linux, FreeBSD, Solaris, Mac OSX y Windows. [21]

1.4 Conclusiones.

En este, el primer capítulo es expuesto las condiciones y problemas que rodean el objeto de estudio. Se evidencia la necesidad de desarrollar un nuevo módulo que permita controlar de forma precisa y organizada la transportación de los estudiantes los fines de semana. Para el desarrollo del nuevo módulo se hace empleo de la tecnología para la programación de páginas dinámicas el lenguaje PHP4 y con soporte de bases de datos en PostgreSQL manteniendo así compatibilidad con el Sistema de Reservación de Transporte que funciona actualmente. El proceso de desarrollo es RUP, el cual está basado en la

Capítulo 1 Fundamentación Teórica

orientación a objetos y el modelamiento visual usando UML, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios para nuestro módulo.

Capítulo 2 Características del Sistema

2.1 Introducción.

En el presente capítulo se identifican los principales problemas que fundamentan la propuesta de solución, abordando aspectos como el objeto de estudio y el flujo actual del proceso. Se realiza un Modelo del Negocio, para capturar correctamente los requisitos y poder construir un sistema correcto. Además se enumeran los requisitos funcionales y no funcionales que debe tener el sistema propuesto, lo que permite tener una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

2.2 Objeto de estudio.

La Universidad de las Ciencias Informáticas (UCI), basada en su estrategia de desarrollo, está enfrascada en el proyecto UCI-Ciudad Digital cuyo objetivo, es crear una micro-sociedad completamente informatizada, que pudiera constituir el modelo de sociedad cubana del futuro. Como parte de esta misión el área de Informatización realiza diferentes proyectos para la automatización de los servicios que se brindan en el centro.

Justamente, el objeto de estudio de este trabajo son los diferentes procesos que se desarrollan, los servicios y recursos que brinda la UCI para garantizar la gestión de la reservación de transporte para el pase de fin de semana por parte de la comunidad universitaria.

2.2.1 Situación problemática.

El Sistema de Reservación de Transporte, para la salida de pase de la comunidad universitaria, en la actualidad resuelve las necesidades básicas, aunque no cumple con el alcance trazado por la dirección de Informatización. Esta última requiere que el sistema brinde la posibilidad de hacer una programación de las reservaciones de transporte para el pase de los estudiantes.

Capítulo 2 Características del Sistema

Hoy en día el estudiante no puede hacer una programación de su pase, como: reservar para varios fines de semana, o para todos, o para alguno en particular en el período del curso académico; tampoco tiene la posibilidad de configurar sus salidas/entradas de pase (reservación) de forma periódica. Es decir no puede reservar su pase si no está en transcurso la semana correspondiente a este. Por otra parte el sistema no posee documentación alguna, y por ser este bastante grande, se hace necesario realizar esta tarea para su posterior mantenimiento.

2.2.1.1 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. [22]

Actualmente la Universidad de Ciencias Informáticas, brinda a la comunidad universitaria el servicio de Reservación de Transporte, solución única para la gestión de todo un flujo de información que conlleva la realización del mismo.

Los estudiantes al hacer uso del proceso de Reservación de Transporte, el sistema realizará toda la gestión de la información, la cual está compuesta por una serie de procesos los cuales están relacionados, existiendo entre ellos una serie de dependencias, que tienen como objetivo final brindar un servicio de calidad.

Las reservaciones de transporte para la salida del fin de semana se pueden realizar a partir del lunes a las 01:00 AM hasta el viernes a las 8:30 AM. A partir del viernes a las 8:30 AM y hasta el sábado a las 9:00am sólo será posible cancelar una reservación previamente realizada.

Una vez reservada una solicitud de transportación y pasado el horario de posible cancelación, es obligatorio utilizar el transporte reservado, en caso contrario se retirará el derecho a reservar transporte por un período de tiempo especificado por el decano, o cualquier otra medida que se estime conveniente. La autenticación para acceder al sistema se controla con el usuario y contraseña del dominio UCI. Tras la

Capítulo 2 Características del Sistema

solicitud de reservación de transporte que hace el estudiante para su correspondiente pase, se registran sus datos personales, dirección específica hacia donde se dirige, y persona o familiar con la cual se va a reunir, todo esto se gestiona en una base de datos centralizada la cual permite los diferentes procesos asociados a dicha reservación. Cuando se produce una reservación con los requisitos antes trazados, el sistema lanza un aviso de confirmación al estudiante en cuestión, lo que verifica este si su pase fue reservado correctamente o se percata si alguien ha reservado con su mismo usuario de dominio. En la fecha próxima a la salida y sin haber cancelado la reservación, se le hace llegar un boletín de reservación, el cual es necesario para abordar al ómnibus.

Los decanos son los encargados de imprimir y distribuir los boletines de reservación, ya sea de ida, regreso, o ambos, a los estudiantes de sus facultades, tarea que pueden delegar en sus correspondientes secretarías. La dirección de transporte recoge los datos necesarios para determinar la cantidad de ómnibus a emplear en la tarea.

2.2.2 Objeto de automatización.

Con el módulo se pretende automatizar el siguiente proceso.

- Programación de las reservaciones de transporte.

2.2.3 Información que se maneja.

La reservación de transporte contiene implícita datos personales y datos propios de la reservación siendo esta información sensible. Se debe lograr, y es uno de los objetivos que persigue la implementación del módulo, 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

Capítulo 2 Características del Sistema

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. [23]

En los inicios de la UCI, la dirección de la misma decidió facilitar pase a los estudiantes cada fin de semana hacia Ciudad de la Habana. Para entonces el pase se hacía de manera manual. El estudiante que decidía salir de pase se presentaba a la secretaría de su facultad y lo solicitaba. La secretaria preguntaba hacia que ruta deseaba salir, le pedía el solapín y luego lo anotaba en la lista por rutas. Al finalizar cada semana las secretarias correspondientes a cada una de las facultades, entregaban al jefe de transporte la relación de los estudiantes que saldrían de pase. Este por su parte confeccionaba el listado total y la cantidad por ruta. Esta cantidad total era enviada hacia el Consejo de Estado, que es el organismo encargado de asignar los ómnibus para esta actividad. El responsable para esta tarea de dicho organismo, asigna los ómnibus necesarios a la escuela y le informa al jefe de transporte la cantidad de ómnibus asignados por cada ruta. El jefe de transporte confecciona un listado de alumnos por ómnibus y lo envía a los profesores encargados de cada ómnibus en la salida de pase. Estas actividades se hacen semanalmente.

2.3.1 Actores del Negocio.

Un actor del Negocio es cualquier individuo, grupo, organización, ordenador o sistema de información externo que interactúa con el Negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el Negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del Negocio representa un tipo particular de usuario del Negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor. [24]

Actor del Negocio	Justificación
Estudiante	El estudiante es el que inicia las acciones de reservar el transporte para el pase, las cuales dan lugar al comienzo de los procesos y es el principal beneficiado al concluir dicho proceso.

Tabla 2.1 Justificación de los actores del negocio.

Capítulo 2 Características del Sistema

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. [25]

Trabajadores del negocio	Justificación
Secretaria docente	Es la encargada de llevar el control en su facultad de los estudiantes que salen de pase, así como reportar la cantidad de estudiantes por ruta que saldrán de pase para contribuir a la reservación del transporte. No se beneficia con ningún resultado obtenido al terminar los procesos.
Jefe de transporte	Responsable de realizar según la información que le suministra cada secretaria, la reservación de transporte según la necesidad de ese fin de semana. No se beneficia con ningún resultado obtenido al terminar los procesos.
Responsable del Consejo de Estado	Es el encargado de asignar la cantidad de ómnibus necesarios para la salida y entrada de estudiantes cada fin de semana de la universidad. No se beneficia con ningún resultado obtenido al terminar los procesos.
Profesor encargado de cada ómnibus	Es el responsable de comprobar que cada estudiante salga de pase en el ómnibus que le corresponde. No se beneficia con ningún resultado obtenido al finalizar los procesos.

Tabla 2.2 Justificación de los trabajadores del negocio.

2.3.3 Diagrama de casos de uso del negocio.

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.), permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo. [17]

Capítulo 2 Características del Sistema

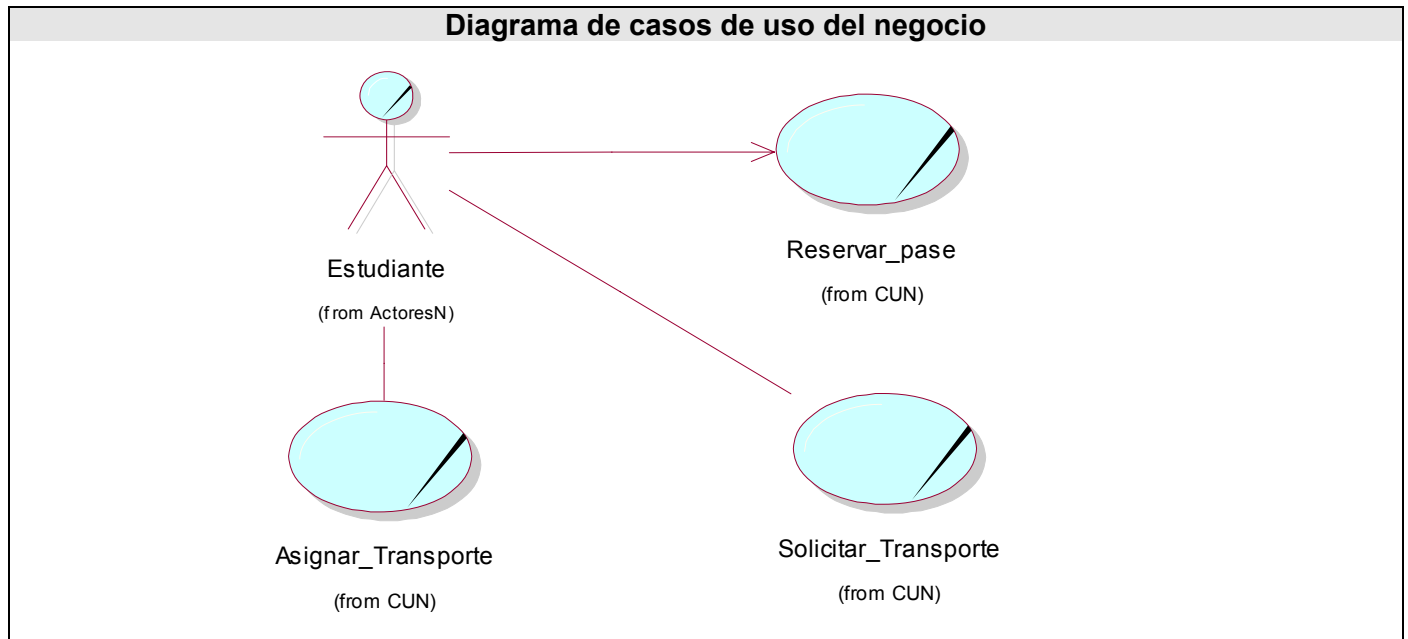


Figura 2.1 Diagrama de CUN

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. 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: Reservar Pase.

Caso de Uso:	Reservar Pase
Actores:	Estudiante
Trabajadores:	Secretaria docente
Resumen:	El caso de uso se inicia cuando un estudiante llega a la secretaría docente donde es atendido por la secretaria y le informa que desea reservar pase. La secretaria le pide los datos necesarios al estudiante y lo agrega al listado de estudiantes según la ruta que este mismo escogió, finalizando así el caso de uso.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

Capítulo 2 Características del Sistema

1. El estudiante se presenta en la secretaría docente para reservar el pase.	1.1 La Secretaria docente le pregunta en cual ruta de las existentes desea salir de pase.
2. Le informa la ruta deseada.	2.1 La Secretaria docente busca en el listado de estudiantes por rutas, la sesión de la ruta que eligió. 2.2 La Secretaria docente le pide el solapín y le pregunta otros datos importantes para la reservación.
3. Le entrega los datos pedidos.	3.1 La Secretaria docente lo anota en la lista según la ruta elegida. 3.2 La Secretaria docente le informa que ya se ha realizado la reservación.
Casos de Uso asociados:	
Pos condiciones	Se obtienen los listados por cada facultad, de los alumnos que saldrán de pase, ordenados por rutas.

Tabla 2.3 Descripción textual del caso de uso del negocio: Reservar Pase.

2.3.4.2 Diagrama de actividad del caso de uso del negocio: Reservar Pase.

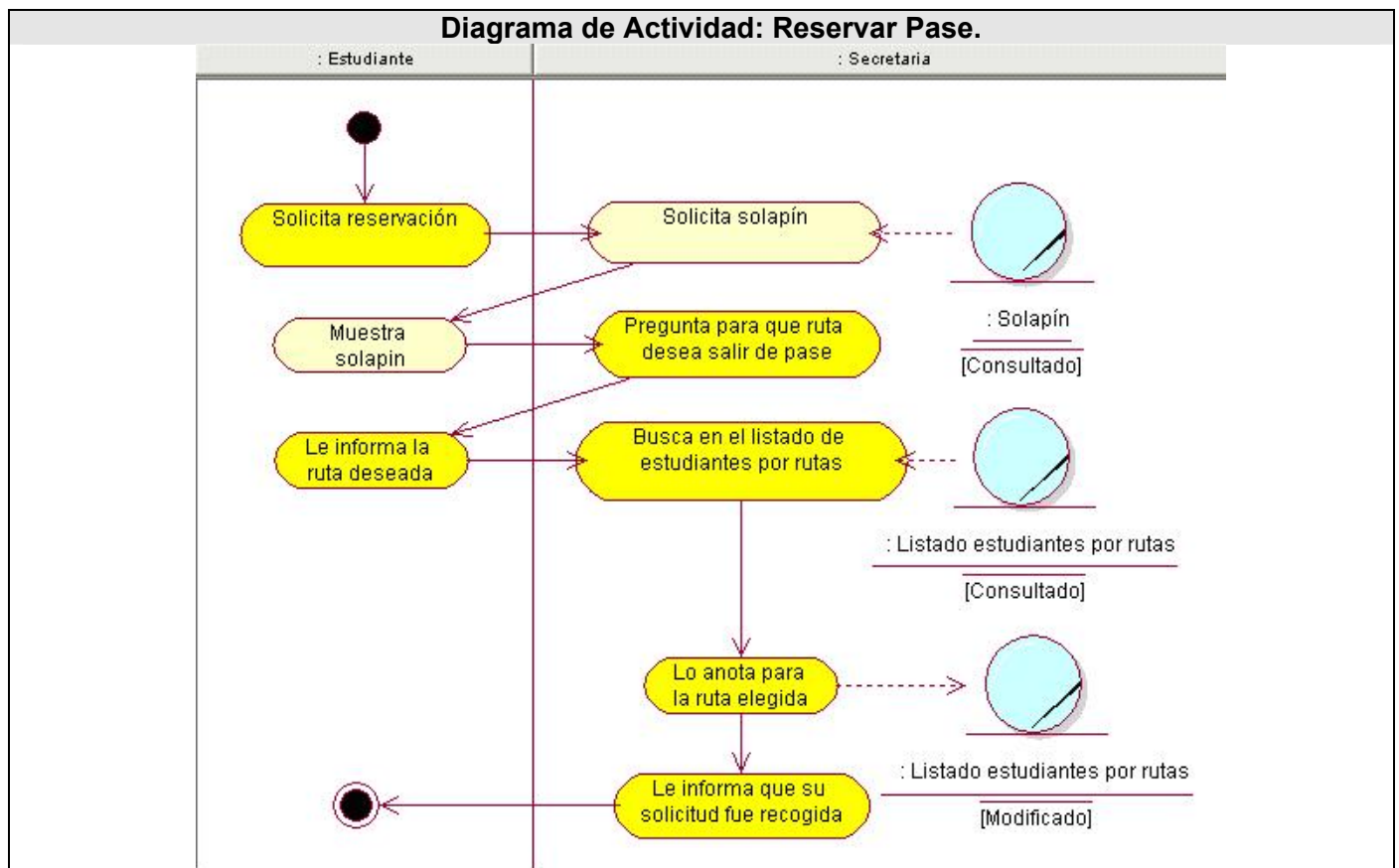


Figura 2.2 Diagrama de actividad del caso de uso del negocio: Reservar Pase.

Capítulo 2 Características del Sistema

2.3.4.3 Diagrama de clases del modelo de objetos del caso de uso del negocio: Reservar Pase.

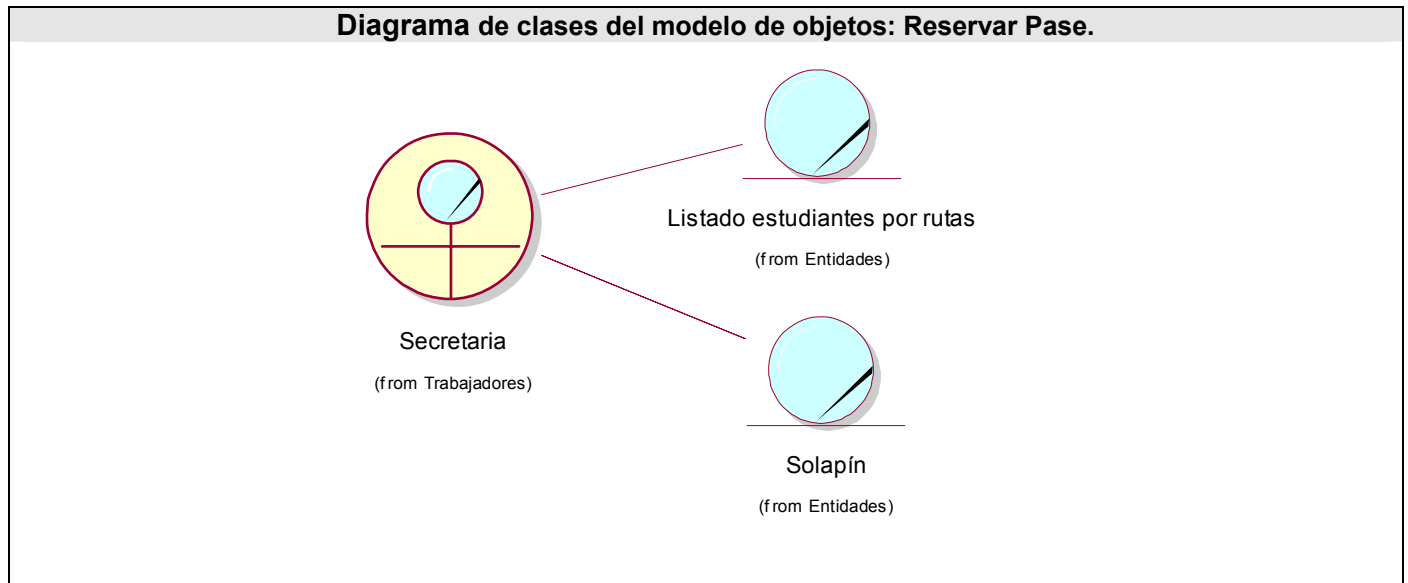


Figura 2.3 Diagrama de clases del modelo de objetos: Reservar Pase.

2.3.4.4 Especificación textual del caso de uso del negocio: Solicitar Transporte.

Caso de Uso:	Solicitar Transporte
Actores:	Estudiante
Trabajadores:	Secretaria docente, Jefe de Transporte, Responsable del Consejo de Estado.
Resumen:	El caso de uso se inicia cuando la Secretaria docente le envía al jefe de transporte el listado de estudiantes que saldrán de pase. Este confecciona el listado general por rutas y saca la cantidad de estudiantes por rutas enviándoselo al Responsable del Consejo de Estado, finalizando así el Caso de Uso.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
	1.1 La secretaria le envía al jefe de transporte el listado de los estudiantes por ruta. 1.2 El Jefe de transporte confecciona el listado general por rutas, según lo que le enviaron las secretarias de cada facultad. 1.3 El jefe de transporte envía la cantidad de estudiantes según el listado que saldrán de pase por rutas al Responsable del Consejo de Estado.
Casos de Uso asociados:	
Pos condiciones	Se obtiene el listado general por cada ruta de los estudiantes que van a salir de pase.

Tabla 2.4 Descripción textual del caso de uso del negocio: Solicitar Transporte.

Capítulo 2 Características del Sistema

2.3.4.5 Diagrama de actividad del caso de uso del negocio: Solicitar Transporte.

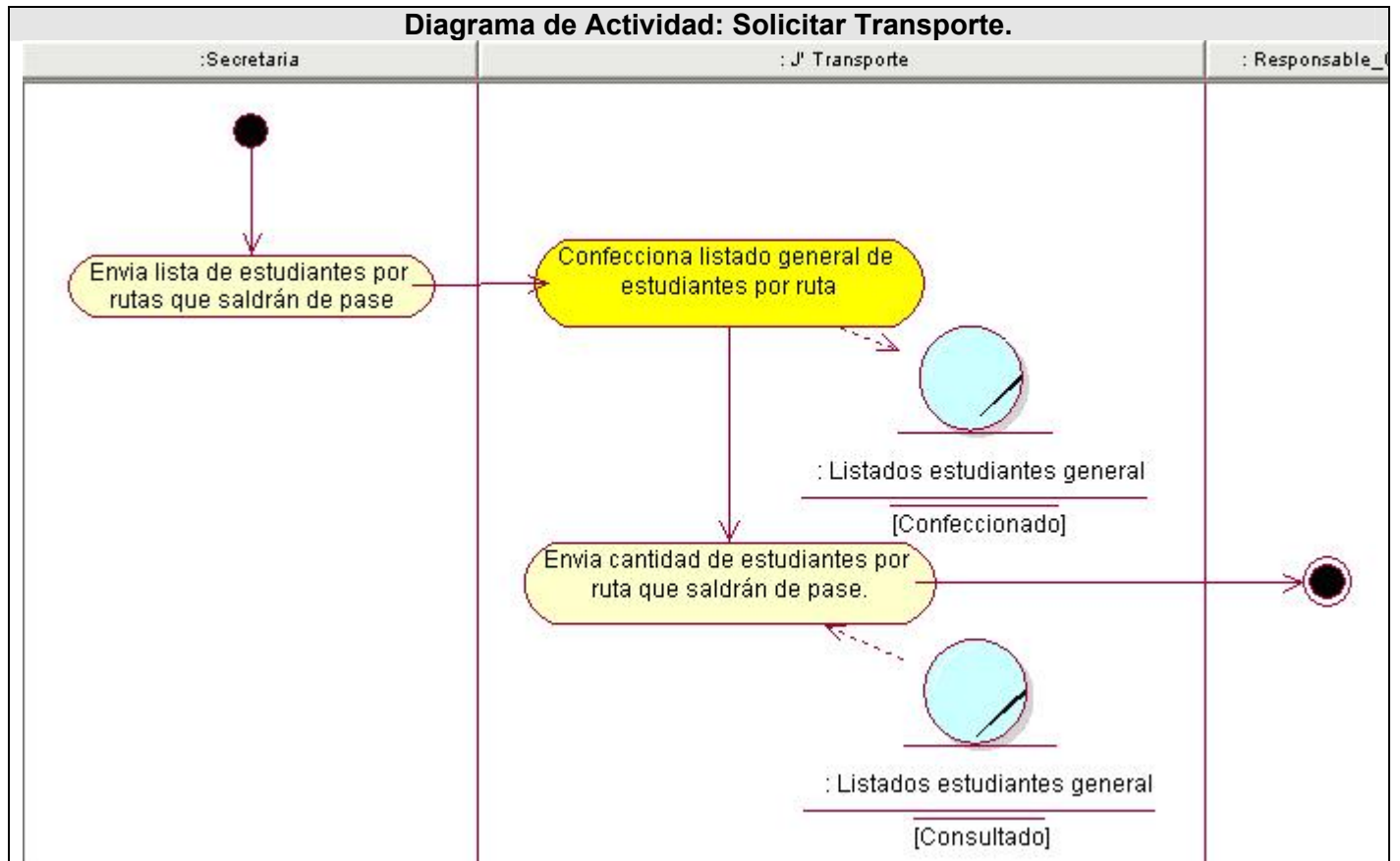


Figura 2.4 Diagrama de actividad del caso de uso del negocio: Solicitar Transporte.

2.3.4.6 Diagrama de clases del modelo de objetos del caso de uso del negocio: Solicitar Transporte.

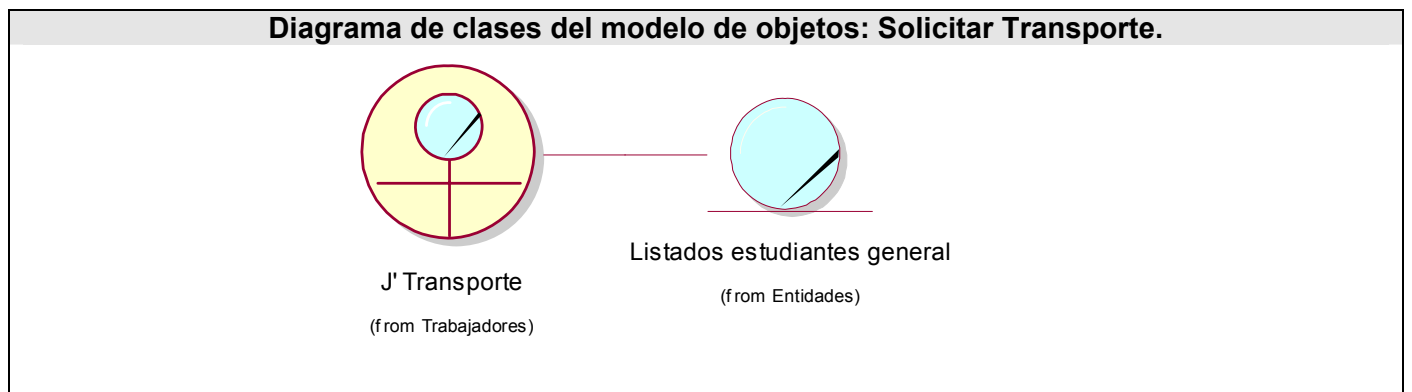


Figura 2.5 Diagrama de clases del modelo de objetos: Solicitar Transporte.

Capítulo 2 Características del Sistema

2.3.4.7 Especificación textual del caso de uso del negocio: Asignar Transporte.

Caso de Uso:	Asignar Transporte	
Actores:	Estudiante	
Trabajadores:	Jefe de Transporte, Responsable del Consejo de Estado, Profesor Responsable de ómnibus.	
Resumen:	El caso de uso se inicia cuando el Responsable del Consejo de Estado (CE) recibe la cantidad de estudiantes que saldrán de pase por rutas. Luego calcula la cantidad de ómnibus necesarios por rutas para el pase y le informa al Jefe de Transporte la cantidad de ómnibus asignados para el pase por rutas. El Jefe de Transporte confecciona el listado de estudiantes por ómnibus y se lo envía a los profesores responsables de los ómnibus, finalizando así el Caso de Uso.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
	1.1 El Responsable del Consejo de Estado recibe la cantidad de estudiantes que saldrán por rutas. 1.2 Calcula la cantidad de ómnibus necesarios por rutas para el pase. 1.3 Informa al Jefe de Transporte la cantidad de ómnibus asignados para el pase. 1.4 El Jefe de Transporte confecciona el listado de estudiantes por ómnibus. 1.5 El Jefe de Transporte envía el listado de estudiantes por ómnibus a los profesores responsables de estos. 1.6 Los profesores verifican que cada estudiante arribe al ómnibus correspondiente.	
Casos de Uso asociados:		
Pos condiciones	Se obtienen el listado por cada ómnibus de los estudiantes que van a salir de pase.	

Tabla 2.5 Descripción textual del caso de uso del negocio: Asignar Transporte.

Capítulo 2 Características del Sistema

2.3.4.8 Diagrama de actividad del caso de uso del negocio: Asignar Transporte.

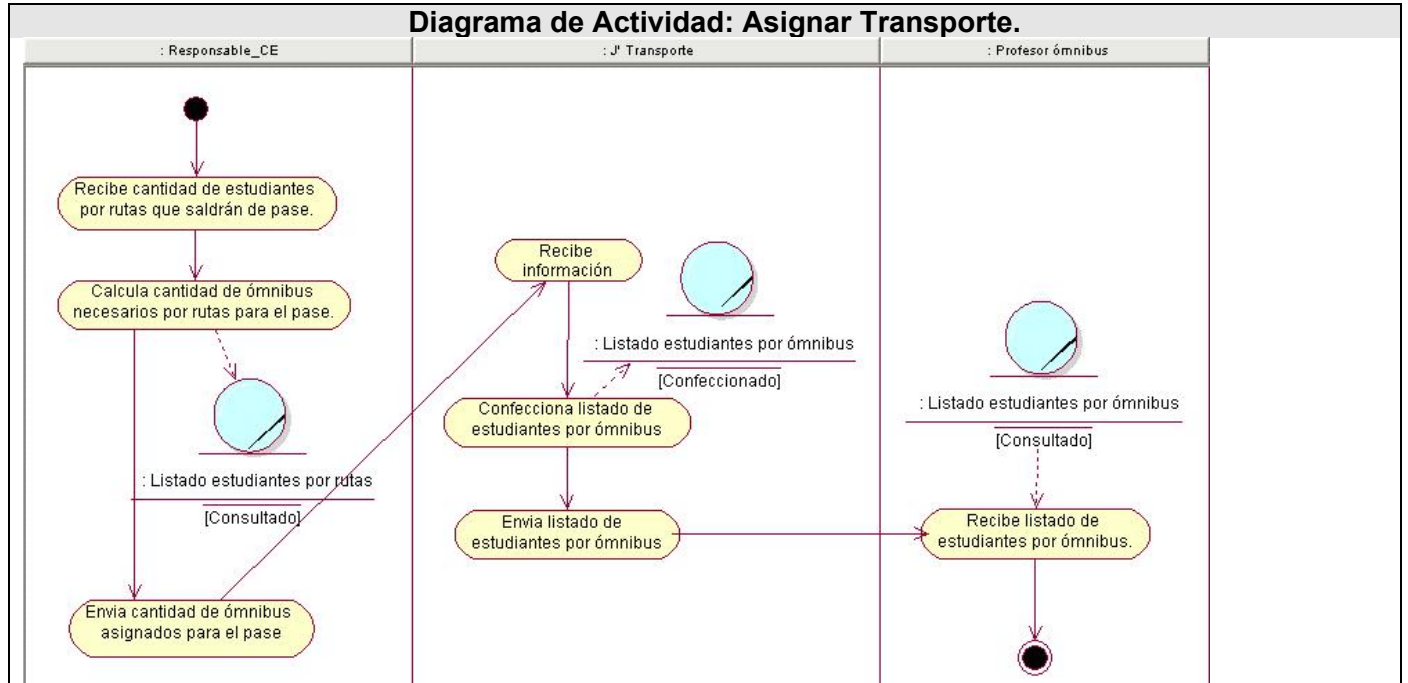


Figura 2.6 Diagrama de actividad del caso de uso del negocio: Asignar Transporte.

2.3.4.9 Diagrama de clases del modelo de objetos del caso de uso del negocio: Asignar Transporte.

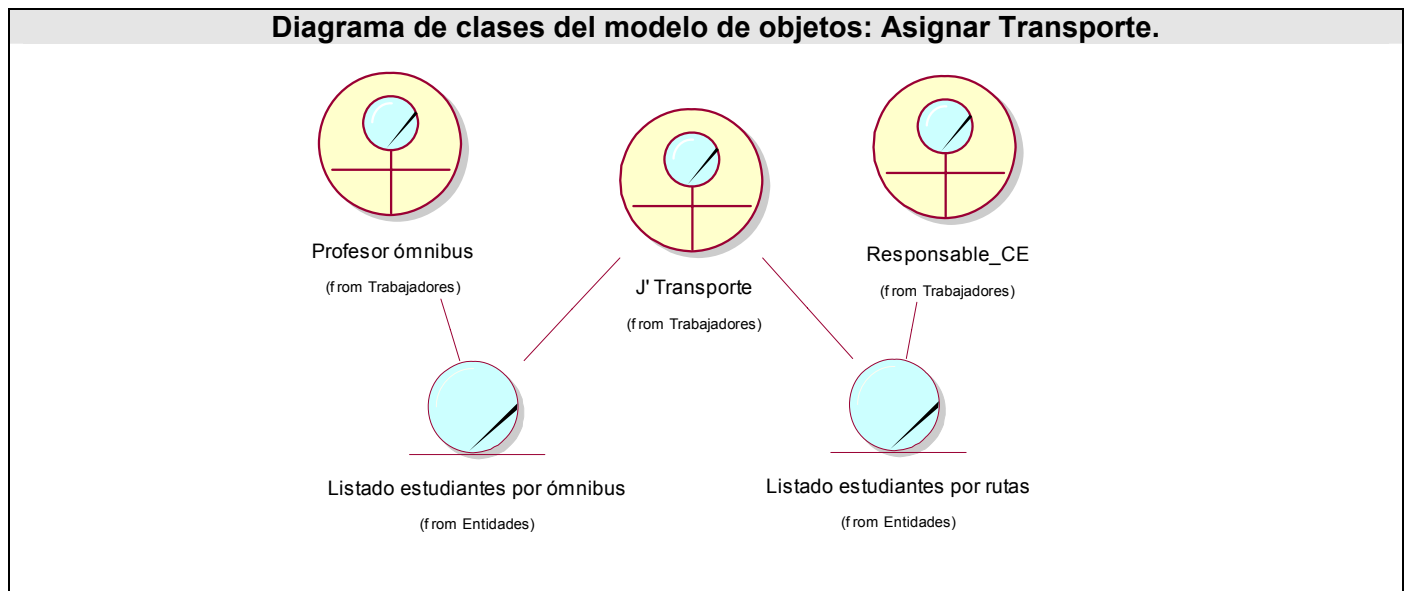


Figura 2.7 Diagrama de clases del modelo de objetos: Asignar Transporte.

Capítulo 2 Características del Sistema

2.3 Especificación de los requisitos de software.

Los requisitos 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. [26]

2.3.4 Requerimientos funcionales.

Los requerimientos funcionales son aquellos requisitos que, desde un punto de vista de las necesidades del usuario, debe cumplir el sistema y los cuales están fuertemente ligados a las opciones brindadas por el software. [27]

Para el cumplimiento de los objetivos propuestos se prevé que el sistema posea las siguientes funcionalidades:

R.1 Autenticar.

R.2 Gestionar Pase.

R.2.1 Permitir al estudiante realizar reservación y/o series de reservaciones de transporte para ida y/o regreso para cuando desee, en el período del curso académico.

R.2.2 Permitir al estudiante cancelar reservación y/o series de reservaciones de transporte siempre y cuando esté realizada, y sea tiempo de cancelación.

R.2.3 Permitir al estudiante modificar reservación y/o series de reservaciones de transporte siempre y cuando esté realizada, y sea tiempo de modificación.

R.2.4 Ver Listado de Reservaciones.

R.3 Imprimir boletines.

R.4 Realizar búsqueda avanzada de estudiantes que reservaron transporte.

R.5 Dar baja de la planificación alimenticia para el fin de semana de los estudiantes reservados.

R.6 Gestionar informes semanales.

R.6.1 Crear informes semanales de estudiantes que salen de pase según la facultad.

R.6.2 Crear informes semanales de estudiantes que salen de pase según la ruta.

Capítulo 2 Características del Sistema

R.6.3 Crear informes semanales de estudiantes que salen de pase según el itinerario.

R.6.4 Crear informes semanales de estudiantes que salen de pase según el año.

R.7 Gestionar usuarios del sistema.

R.7.1 Agregar usuarios del sistema.

R.7.2 Eliminar usuarios del sistema.

R.7.3 Ver listado de usuarios del sistema.

R.8 Gestionar Rutas de ómnibus.

R.8.1 Ver Listado de Rutas de ómnibus.

R.8.2 Agregar Rutas de ómnibus.

R.8.3 Editar Rutas de ómnibus.

R.8.4 Activar/Desactivar Rutas de ómnibus.

R.9 Gestionar Municipios.

R.9.1 Insertar municipios a los cuales se dirigen los ómnibus.

R.9.2 Modificar Municipio a los cuales se dirigen los ómnibus.

R.9.3 Ver Listado de Municipios.

R.10 Gestionar Horarios.

R.10.1 Insertar horarios de entrada y salida del pase.

R.10.2 Modificar Horarios de entrada y/o de salida del pase.

R.10.3 Insertar horario de fin de curso.

R.10.4 Modificar horario de fin de curso.

R.11 Gestionar Puntos de Salida de la transportación.

R.11.1 Insertar Punto de salida de los ómnibus.

R.11.2 Editar Punto de salida de los ómnibus.

R.11.3 Ver Puntos de salida de los ómnibus.

R.12 Administrar mapa de los puntos de salida de la Universidad.

R.13 Permitir el envío de inquietudes (quejas y sugerencias).

R.14 Ver Reportes.

R.14.1 Ver reportes por rutas.

R.14.2 Ver reportes por años.

R.14.3 Ver reportes por facultades.

Capítulo 2 Características del Sistema

2.3.5 Requerimientos no funcionales.

Los requerimientos no funcionales son características que describen alguna forma o restricción para la correcta realización de algún requerimiento (funcionalidad) o conjunto de ellas o incluso todas las funcionalidades. Se consideran los atributos que deberá poseer el software, propiedades que debe tener el producto final. [27]

A continuación son descritos los requerimientos no funcionales:

Apariencia o Interfaz externa: la interfaz contiene la menor cantidad de imágenes posibles y las que contiene lo hace en pequeñas dimensiones contribuyendo así al aceleramiento de las respuestas al usuario. El diseño de la interfaz es sencillo y fácil de utilizar.

Usabilidad: el sistema ha sido desarrollado de forma tal que permite a los usuarios ejecutar correctamente las acciones para las cuales posee los permisos necesarios. Para cada proceso el sistema muestra a través de mensajes las acciones ejecutadas por los usuarios.

Rendimiento: la disponibilidad del trabajo en red ejecutándose contra el servidor es constante. Es garantizado que la respuesta a solicitudes de los usuarios del sistema sea en períodos de tiempos relativamente cortos evitando así las demoras excesivas.

Soporte: el sistema es de fácil instalación y configuración.

Software: para el correcto funcionamiento del sistema en la parte del servidor es necesario el empleo de Windows 98 o superior, Linux o Unix en sus versiones de servidor. Para un correcto funcionamiento del sistema en las terminales del cliente es necesario el sistema operativo Windows 98 o superior, Linux, Unix, entre otros, que posean navegadores con tecnología JavaScript con excepción de Microsoft Internet Explorer 6 y sus antecesores los cuales necesitan también tener el ActiveX activado.

Hardware: es necesario como requerimiento mínimo una PC con procesador Pentium III y 128 MB de RAM.

Capítulo 2 Características del Sistema

Portabilidad: el sistema es usado bajo los sistemas operativos Windows, Linux, Unix. El producto corre sobre plataforma Web, desarrollada en PHP4 y gestor de bases de datos en PostgreSQL.

Seguridad: el sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios en el sitio, de realizar identificaciones de los usuarios antes de que estos puedan realizar cualquier acción sobre el sistema. Garantiza que la información sea vista y manipulada única y exclusivamente por aquellos usuarios que tienen derecho a hacerlo.

Se emplean mecanismos de encriptación (Base64) de los datos que por cuestiones de seguridad no deben viajar al servidor en texto plano como es el caso de las contraseñas de los sistemas a los cuales se les solicita información confidencial.

Son realizadas validaciones de la información tanto en el cliente como en el servidor, esto no impide que los usuarios tengan un rápido acceso al sistema sin que los requerimientos de seguridad se conviertan en una demora para ellos.

Confidencialidad: toda la información está protegida de los accesos no autorizados, los administradores del sistema son los únicos que podrán alterar la información, los usuarios de impresión y usuarios de consulta solo tendrán acceso a ver los listados de información.

Disponibilidad: esta aplicación garantiza a los usuarios finales acceso a la información solicitada en todo momento (solo si tiene permiso para ejecutar la acción).

Restricciones en el diseño y la implementación: es una aplicación desarrollada con el empleo de la tecnología para la creación de páginas Web dinámicas PHP4 y bases de datos PostgreSQL.

Legales: la plataforma escogida para el desarrollo del software esta basada en la licencia GNU/GPL.

Restricciones: en el desarrollo de la aplicación se utiliza UML como lenguaje de modelado para lograr una mejor documentación del sistema y como herramienta de apoyo Rational Rose 2003. Se emplea como lenguaje de programación PHP4 y como gestor de bases de datos PostgreSQL.

Capítulo 2 Características del Sistema

2.4 Descripción del sistema propuesto.

El sistema propuesto automatiza el proceso de pases programados, el mismo hace uso del gestor de bases de datos PostgreSQL con el objetivo de disminuir los tiempos de respuesta cuando son altos los volúmenes de datos, así como la concurrencia a los mismos. Es importante destacar el uso de herramientas de Software Libre, pues confirma 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 un módulo el cual se va a integrar con todos los procesos y actividades que se desarrollan durante la reservación de transporte por parte de los estudiantes de la UCI.

Teniendo en cuenta todos los requerimientos planteados, el módulo propuesto debe ser capaz de integrarse al Sistema de Reservación de Transporte.

El módulo de pases programados puede ser usado por el estudiante con la finalidad de reservar transporte para el pase de fin de semana en la fecha que decida o efectuar una planificación del mismo. Una vez realizada alguna reservación, el estudiante tiene el derecho de modificar o eliminarla, siempre y cuando se encuentre en tiempo para realizar dicha acción. La base de datos contendrá la estructura apropiada para cumplir toda la gestión de la reservación de transporte.

2.5 Modelo de casos de usos de sistema.

Utilizando las facilidades que brinda el UML, se capturan los requisitos funcionales del sistema y se representan mediante un diagrama de casos de uso. Para ello se define de acuerdo a lo planteado en los epígrafes anteriores, cuales serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

2.5.4 Definición de los actores.

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema como tal y pueden intercambiar con él o ser recipientes

Capítulo 2 Características del Sistema

pasivos de información. En este caso los actores que interactúan con el sistema son definidos a continuación: [24]

Actores	Justificación
Administrador	Se encarga de realizar los cambios administrativos necesarios en el sistema. Ejemplo: agregar, modificar y eliminar un usuario.
Invitado	Puede ver las rutas hacia las cuales está destinada la transportación y el mapa de las salidas de la transportación en la UCI, además puede enviar inquietudes.
Directivo	Se encarga de ver los reportes para facilitar la gestión de ómnibus, puede realizar búsquedas avanzadas de estudiantes y realizar la impresión de boletines.
Estudiante	Se encarga de realizar reservación y/o series de reservaciones de transporte para el pase de fin de semana, en el período del curso académico, cancelar o modificar reservaciones y/o series de reservaciones de transporte previamente realizadas.
Reloj	Se encarga de que la reservación sea realizada a la hora debida, y cancela la planificación alimenticia para el fin de semana del estudiante reservado también a una hora determinada.

Tabla 2.6 Justificación de los actores del sistema.

En algunos sistemas en particular, como el presente, existen actividades que deben ser ejecutadas periódicamente cuando llega cierto instante de tiempo. Para realizar la modelación de esto se define un actor ficticio (Reloj) el cual se encarga de “disparar” la ejecución del caso de uso.

2.5.5 Listado de los casos de uso.

CU-1	Gestionar Pase.
Actor	Estudiante.
Descripción	El Estudiante reserva transporte para el pase del fin de semana en la fecha que decida, modifica las reservaciones existentes y cancela las reservaciones

Capítulo 2 Características del Sistema

	existentes siempre y cuando se encuentre en tiempo para realizar dichas acciones.
Referencia	R2.

CU-2	Autenticar.
Actor	Usuario.
Descripción	El Usuario solicita autenticación para entrar al sistema con su nombre de usuario y contraseña, el sistema evalúa los datos de acceso, si son correctos le concede la entrada, si son incorrectos le deniega el acceso.
Referencia	R1.

CU-3	Enviar Inquietud.
Actor	Usuario.
Descripción	El Usuario envía sus inquietudes (críticas y/o sugerencias), y el sistema se encarga de tramitar su inquietud hacia los administradores del sistema.
Referencia	R13.

CU-4	Ver Rutas.
Actor	Usuario.
Descripción	El Usuario solicita al sistema que le muestre las rutas disponibles y el sistema muestra todas las rutas disponibles hacia las cuales puede reservar transporte.
Referencia	R8.

CU-5	Ver Puntos de Salida.
Actor	Usuario.
Descripción	El Usuario solicita al sistema que le muestre los puntos de salida desde donde sale cada uno de los ómnibus, el sistema le muestra al Usuario todos los puntos de salida de cada uno de los municipios.
Referencia	R11.

CU-6	Buscar Estudiante.
Actor	Directivo.
Descripción	El Directivo solicita al sistema buscar un estudiante según parámetros establecidos y el sistema le muestra al Directivo el resultado de la búsqueda, le muestra al estudiante en caso de encontrarlo o un mensaje de error en caso de que el estudiante solicitado no se encuentre.
Referencia	R4.

CU-7	Ver Reportes.
Actor	Directivo.
Descripción	El Directivo le pide al sistema que le muestre un reporte según criterios de búsqueda, el sistema en caso de encontrar resultados con los criterios establecidos los devuelve, si no, devuelve un mensaje de error.

Capítulo 2 Características del Sistema

Referencia	R14.
CU-8	Imprimir boletines.
Actor	Directivo.
Descripción	El Directivo le pide al sistema que imprima los boletines según criterios de búsqueda proporcionados, el sistema en caso de encontrar resultados con los criterios establecidos imprime el/los boletines, si no, devuelve un mensaje de error.
Referencia	R3.
CU-9	Modificar Horarios.
Actor	Administrador.
Descripción	El Administrador puede modificar los horarios por los cuales está regida la aplicación, el sistema modifica los horarios según los nuevos horarios establecidos por el administrador.
Referencia	R10.
CU-10	Gestionar Usuarios.
Actor	Administrador.
Descripción	El Administrador puede agregar, modificar o eliminar un usuario.
Referencia	R7.
CU-11	Gestionar Puntos de salida.
Actor	Administrador.
Descripción	El Administrador puede agregar, modificar o eliminar un punto de salida desde donde saldrán los ómnibus para la transportación.
Referencia	R11.
CU-12	Gestionar Municipio.
Actor	Administrador.
Descripción	El Administrador puede agregar, modificar o eliminar un municipio hacia los cuales están dirigidos los ómnibus de la transportación.
Referencia	R9.
CU-13	Gestionar Rutas.
Actor	Administrador.
Descripción	El Administrador puede agregar, modificar o eliminar una ruta por la cual se dirigen los ómnibus de la transportación.
Referencia	R8.
CU-14	Reservar comida.
Actor	Relej.
Descripción	El Relej dispara una acción, actualizar en las base de datos de los comedores la

Capítulo 2 Características del Sistema

	cantidad de estudiantes que se irán de pase, así como un listado con cada uno de los estudiantes para impedir el acceso a los comedores.
Referencia	R5.

2.5.6 Diagrama de casos de uso.

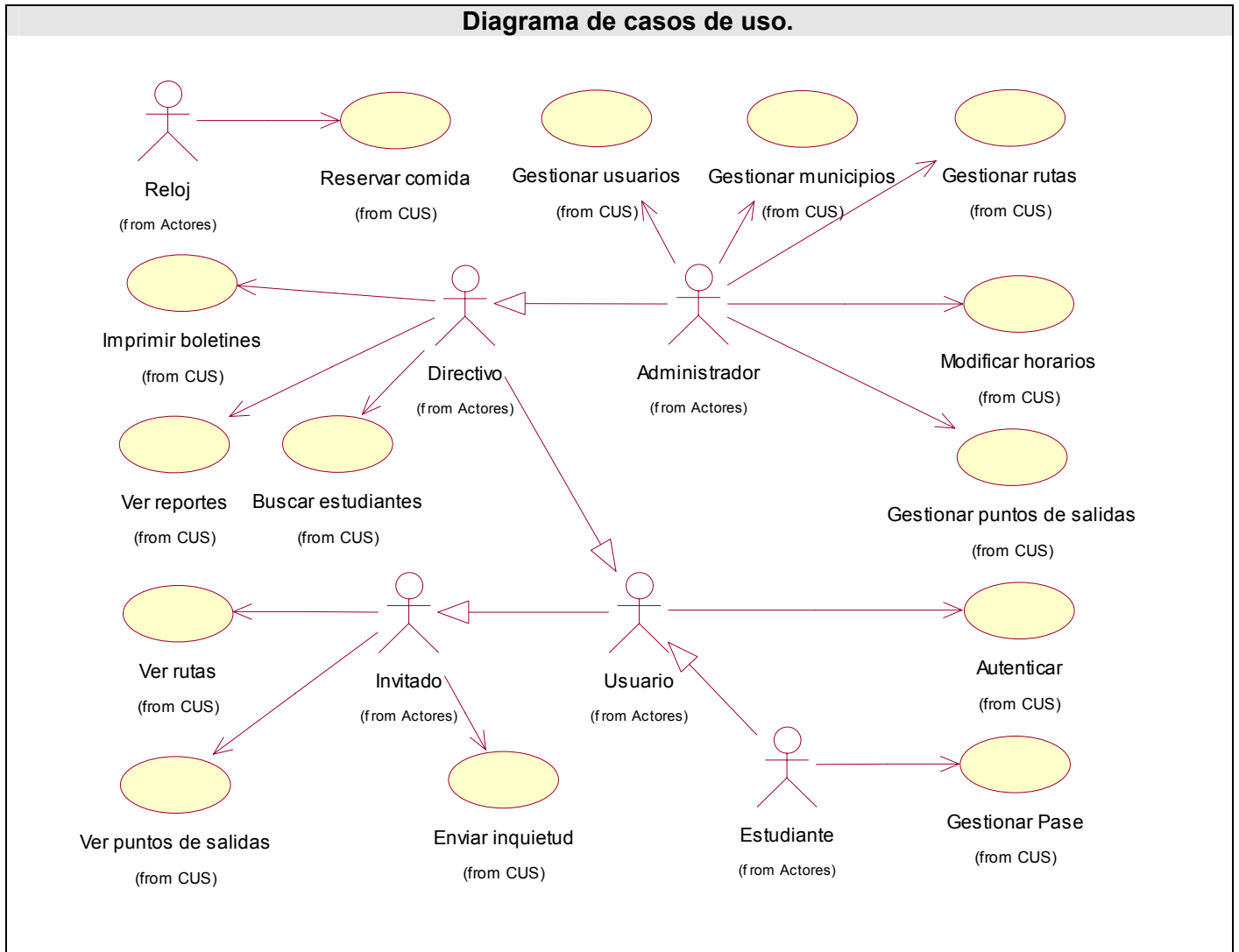


Figura 2.8 Diagrama de casos de uso

Capítulo 2 Características del Sistema

2.5.7 Casos de uso expandidos.

A través de la expansión de los casos de uso, es descrita la secuencia de eventos que los actores utilizan para poder completar un proceso a través del sistema. Este sería el último paso en el análisis de los procesos, para pasar a la construcción de la solución propuesta.

Caso de uso	
CU-1	Gestionar Pase.
Propósito	Garantizar que los estudiantes reserven, modifiquen, cancelen y planifiquen el pase en el sistema.
Actores	Estudiante.(inicia)
Precondiciones	Debe estar autenticado. Para realizar cancelación o modificación debe haber reservado.
Poscondiciones	El Estudiante debe haber reservado, modificado o cancelado una reservación o una serie de reservaciones.
Resumen: el caso de uso se inicia cuando el Estudiante accede al sistema y solicita Gestionar Reservación. Inmediatamente se visualiza una pantalla donde el sistema brinda la posibilidad de efectuar operaciones como: reservar, modificar y cancelar reservación de transporte y/o serie de reservaciones de transporte.	
Referencias	R2, R8, R9
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Estudiante decide Gestionar Pase y selecciona la opción.	1.1. Si el Estudiante no ha realizado reservaciones con anterioridad, el sistema muestra una interfaz con el nombre, usuario, número de solapín, IP del ordenador y un formulario a llenar con los datos del familiar.
2. El Estudiante selecciona la operación reservar ida y/o reservar regreso e introduce los datos.	2.1. El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 2.2 La reservación se registra en el sistema. 2.3 El sistema muestra las reservaciones realizadas. 2.4 El sistema envía un correo de notificación asociado a la semana de reservación.
Flujos Alternos	
	2.1 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
	1.1 Si el estudiante ha realizado reservaciones semanales con anterioridad, el sistema muestra una interfaz con todas estas reservaciones y sus datos, además de: el nombre del Estudiante, usuario, número de solapín, IP del ordenador.

Capítulo 2 Características del Sistema

2. El Estudiante selecciona una de las reservaciones realizadas.	2.1 El sistema muestra los detalles de dicha reservación semanal. 2.2 El sistema verifica que esta reservación semanal esté en tiempo de ser cancelada o modificada.
3. El Estudiante selecciona una de las operaciones a realizar.	3.1 El sistema ejecuta una de las siguientes acciones: Si se decide modificar, ir a la sección "Modificar". Si se decide cancelar, ir a la sección "Cancelar".
Sección "Modificar"	
Acción del Actor	Respuesta del sistema
	3.1 El sistema verifica que la reservación semanal a modificar no es parte de una serie hecha por el usuario.
4. El Estudiante introduce los datos de la reservación.	4.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 4.2 El sistema actualiza la información. 4.3 El sistema envía un correo de notificación asociado a la semana de modificación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.3 El Estudiante selecciona modificar la serie.	3.1 El sistema verifica que la reservación semanal a modificar es parte de una serie. 3.2 El sistema pregunta si desea modificar toda la serie o solamente la semana seleccionada. 3.4 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.5 El sistema actualiza la información. 3.6 El sistema envía un correo de notificación.
Curso Alterno de los eventos.	
3.3 El Estudiante selecciona modificar solamente la semana.	3.4 Retornar al paso 4.
Sección "Cancelar"	
Acción del Actor	Respuesta del sistema
	3.1 El sistema verifica que la reservación semanal a cancelar no es parte de una serie hecha por el usuario.
4. El Estudiante confirma la cancelación.	4.1 El sistema cancela la reservación seleccionada. 4.3 El sistema envía un correo de notificación asociado a la semana de cancelación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	3.1 El sistema verifica que la reservación semanal a cancelar es parte de una serie.

Capítulo 2 Características del Sistema

3.3 El Estudiante selecciona cancelar la serie.	3.2 El sistema pregunta si desea cancelar toda la serie o solamente la semana seleccionada. 3.4 El sistema cancela la serie. 3.6 El sistema envía un correo de notificación.
Curso Alterno de los eventos.	
3.3 El Estudiante selecciona cancelar solamente la semana.	3.4 Retornar al paso 4.
Prioridad	Crítico.
Prototipo	[Anexo 8]

Tabla 2.7 Caso de uso: Gestionar Pase.

Caso de uso	
CU-2	Autenticar.
Propósito	Permitir al usuario el acceso al sistema.
Actores	Usuario (inicia).
Precondiciones	
Poscondiciones	La autenticación es realizada de forma correcta.
Resumen: el CU se inicia cuando el Usuario introduce sus datos en el sistema para autenticarse; y en este, según los datos introducidos le son asignados sus privilegios y en caso de que no sean correctos se le niega el acceso mostrando un mensaje de error.	
Referencias	R1.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Usuario introduce su nombre de usuario y la contraseña.	1.1 El sistema verifica que el usuario y la contraseña sean válidos. 1.2 Si los datos están correctos y el usuario está registrado, el sistema le asigna sus privilegios según su rol y le da acceso al sistema.
Curso Alterno de los eventos.	
	1.1 Si el sistema no identifica los datos del usuario, emite un mensaje denegando el acceso.
Prioridad	Crítico.
Prototipo	[Anexo 9]

Tabla 2.8 Caso de uso: Autenticar.

Caso de uso	
CU-3	Enviar Inquietud.
Propósito	Garantizar que todo aquel que entre a la aplicación pueda enviar una inquietud.
Actores	Invitado (inicia).
Precondiciones	
Poscondiciones	Se envía una inquietud.
Resumen: el CU se inicia cuando el Invitado entra a la aplicación e introduce la inquietud que desea enviar y su usuario o correo electrónico. El sistema muestra un mensaje de confirmación de envío.	
Referencias	R13.
Curso normal de los eventos.	

Capítulo 2 Características del Sistema

Acción del actor	Respuesta del sistema
1. El Invitado introduce la inquietud y su nombre de usuario o correo electrónico.	1.1 El sistema verifica que los campos no se encuentren vacíos. 1.2 El sistema envía un correo hacia el administrador encargado de atender las inquietudes. 1.3 El sistema muestra un mensaje de confirmación de envío al invitado.
Curso Alterno de los eventos.	
	1.1 Si el sistema encuentra algún campo vacío, muestra un mensaje de error.
Prioridad	Opcional.
Prototipo	[Anexo 10]

Tabla 2.9 Caso de uso: Enviar Inquietud.

Caso de uso	
CU-4	Ver rutas.
Propósito	Garantizar que todo aquel que entre a la aplicación pueda ver el listado de rutas existentes.
Actores	Invitado (inicia).
Precondiciones	
Poscondiciones	Es mostrado el listado con todas las rutas existentes.
Resumen: el CU se inicia cuando el Invitado entra a la aplicación y selecciona la opción Listado de Rutas. El sistema le muestra todas las rutas existentes.	
Referencias	R8.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Invitado selecciona la opción Listado de Rutas.	1.1 El sistema muestra los listados de las rutas existentes, los puntos de salida UCI, Puntos de Destino/Regreso, los municipios a los que pertenecen y la parada intermedia.
Prioridad	Opcional.
Prototipo	[Anexo 11]

Tabla 2.10 Caso de uso: Ver rutas.

Caso de uso	
CU-5	Ver puntos de salida.
Propósito	Garantiza que el invitado vea el punto de salida de los ómnibus de una determinada ruta.
Actores	Invitado (inicia).
Precondiciones	
Poscondiciones	Es mostrado el punto de salida de los ómnibus de la ruta seleccionada.
Resumen: el CU se inicia cuando el Invitado entra a la aplicación y selecciona la opción MAPA: Puntos de salida UCI. El sistema muestra en el mapa de la UCI el lugar exacto desde donde salen los ómnibus de la ruta seleccionada.	

Capítulo 2 Características del Sistema

Referencias	R11.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Invitado selecciona la opción MAPA: Puntos de salida UCI.	1.1 El sistema muestra una interfaz con el mapa de la UCI, con la opción de elegir la ruta para saber el punto de salida de los ómnibus.
2. El Invitado selecciona la ruta para saber el Punto de Salida UCI.	2.1 El Sistema le señala en el mapa, el lugar de donde salen los ómnibus de la ruta seleccionada.
Prioridad	Opcional.
Prototipo	[Anexo 12]

Tabla 2.11 Caso de uso: Ver puntos de salida.

Caso de uso	
CU-6	Buscar estudiantes.
Propósito	Garantiza que los directivos busquen a los estudiantes que deseen.
Actores	Directivo (inicia).
Precondiciones	El Directivo debe haberse autenticado.
Poscondiciones	Es mostrado el/los estudiantes con las características deseadas por el Directivo.
Resumen: el Caso de uso comienza cuando el Directivo selecciona la opción Buscar Estudiantes. El sistema muestra el formulario que le permite al Directivo realizar la búsqueda.	
Referencias	R4.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Caso de uso comienza cuando un Directivo selecciona la opción Buscar estudiante(s).	1.1 El sistema le muestra una interfaz con un formulario para que introduzca los datos del estudiante buscado.
2. El Directivo introduce los datos pedidos o una parte de ellos.	2.1 El sistema realiza una búsqueda en la BD de los estudiantes que reservaron con las características especificadas por el Directivo. 2.2 El sistema muestra una interfaz con los resultados de la búsqueda realizada.
Curso Alternativo de los eventos.	
	2.2 Si el sistema no encuentra ningún estudiante con dichas características, muestra un mensaje.
Prioridad	Secundario.
Prototipo	[Anexo 13]

Tabla 2.12 Caso de uso: Buscar estudiantes.

Caso de uso	
CU-7	Ver reportes.
Propósito	Garantiza que se obtenga información general de los estudiantes que salieron de pase según las rutas, las facultades y los años.
Actores	Directivo (inicia).
Precondiciones	El Directivo debe haberse autenticado.

Capítulo 2 Características del Sistema

Poscondiciones	Es mostrado el listado de la cantidad de estudiantes que reservaron por ruta, por año o por facultad.
Resumen: el Caso de uso comienza cuando el Directivo selecciona la opción Reportes. El sistema muestra las opciones de ver los estudiantes que hasta el momento han reservado pase por cada ruta, facultad o año. El caso de uso finaliza cuando el Directivo selecciona una de estas opciones.	
Referencias	R14.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Directivo decide ver Reporte de los estudiantes y selecciona la opción.	1.1 El sistema muestra los tres tipos de reporte que existen.
2. El Directivo selecciona el reporte que desea ver. (Por: ruta, facultad o año)	2.1 El sistema ejecuta una de las siguientes acciones: Rutas, ir a la sección "Reportes por rutas". Facultad, ir a la sección "Reportes por facultad". Año, ir a la sección "Reportes por año".
Sección "Reportes por Ruta"	
Acción del actor	Respuesta del Sistema
	1.1 El sistema muestra una interfaz con la cantidad de estudiantes que reservaron pase de ida y de regreso, según las rutas.
2 El Directivo selecciona una cantidad para ver los datos específicos de los estudiantes que reservaron en una ruta determinada.	2.1 El sistema le muestra el nombre, número de solapín y grupo de los estudiantes que saldrán y/o entrarán de pase, en esa ruta.
Sección "Reportes por Facultad"	
Acción del actor	Respuesta del Sistema
	1.1 El sistema muestra una interfaz con la cantidad de estudiantes que reservaron pase de ida, de vuelta, solo de ida y solo de vuelta, de cada facultad.
2. El Directivo selecciona una facultad de la cual quiere conocer los estudiantes que salieron de pase.	2.1 El sistema muestra una interfaz con los estudiantes de la facultad seleccionada.
Sección "Reportes por Año"	
Acción del actor	Respuesta del Sistema
	1.1 El sistema muestra una interfaz con la cantidad de estudiantes que reservaron pase de ida y de regreso, según los años.
Prioridad	Secundario.
Prototipo	[Anexo 14]

Tabla 2.13 Caso de uso: Ver Reportes.

Caso de uso	
CU-8	Imprimir boletines.
Propósito	Garantizar que cada estudiante que haya reservado el pase tenga su boletín de entrada y/o de salida para abordar el transporte que brinda la escuela.

Capítulo 2 Características del Sistema

Actores	Directivo (inicia).	
Precondiciones	El Directivo debe haberse autenticado.	
Poscondiciones	Se generan los boletines de pase.	
Resumen: el caso de uso se inicia cuando el Directivo (que normalmente es el Decano), selecciona la opción Impresión de Boletines. El sistema muestra el formulario a llenar. El Directivo llena los campos pedidos y selecciona Generar Boletines en PDF.		
Referencias	R3.	
Curso normal de los eventos.		
Acción del actor	Respuesta del sistema	
1. El Directivo selecciona la opción de Impresión de Boletines.	1.1 El sistema verifica si está en el tiempo de impresión 1.2 El sistema le muestra una interfaz que contiene el formulario que tiene que llenar para la impresión de boletines.	
2. El Directivo llena dicho formulario y va a Generar Boletines en PDF.	2.1 EL sistema genera los boletines en PDF y le brinda la opción de imprimirlos.	
3. El Directivo selecciona el botón de impresión.	3.1 El sistema imprime los boletines.	
Curso Alternativo de los eventos.		
	1.2 El sistema muestra un mensaje que diga que no se encuentra en tiempo de impresión.	
	3.1 El sistema muestra un mensaje que diga que no encuentra dispositivo de impresión.	
Prioridad	Crítico.	
Prototipo	[Anexo 15]	

Tabla 2.14 Caso de uso: Imprimir boletines.

Caso de uso		
CU-9	Modificar horarios.	
Propósito	Garantiza que se puedan modificar los horarios de impresión de boletines y de gestionar el pase.	
Actores	Administrador (inicia).	
Precondiciones	El Administrador debe haberse autenticado.	
Poscondiciones	Se actualicen los horarios de las distintas acciones.	
Resumen: el caso de uso comienza cuando el Administrador va a la opción de Modificar Horarios. El sistema muestra los horarios existentes. El Administrador selecciona el que desea modificar y lo modifica.		
Referencias	R10.	
Curso normal de los eventos.		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción Modificar Horarios.	1.1 El sistema muestra una interfaz con los tipos de horarios a modificar (Horario de impresión y Horario de reservación)	
2. El Administrador selecciona el horario que desea modificar.	2.1 El sistema busca el horario y muestra los datos en pantalla.	

Capítulo 2 Características del Sistema

3. El Administrador actualiza los campos.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2 El nuevo horario se registra en el sistema.
Curso Alterno de los eventos.	
	3.1 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Prioridad	Auxiliar.
Prototipo	[Anexo 16]

Tabla 2.15 Caso de uso: Modificar horarios.

Caso de uso	
CU-10	Gestionar usuario.
Propósito	Garantizar que se agreguen, modifiquen y eliminen usuarios del sistema.
Actores	Administrador (inicia).
Precondiciones	El Administrador debe haberse autenticado.
Poscondiciones	El Administrador debe haber agregado, modificado o eliminado un usuario.
Resumen: el caso de uso se inicia cuando el Administrador accede al sistema y solicita Gestionar Usuario. Inmediatamente se visualiza una pantalla donde el sistema brinda la posibilidad de efectuar las operaciones de agregar, modificar o eliminar el usuario. Finaliza una vez que el Administrador realiza una de las operaciones.	
Referencias	R7.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1 El Administrador decide gestionar usuario y selecciona una de las operaciones a realizar (agregar, modificar, eliminar).	1.1. El sistema ejecuta una de las siguientes acciones: Agregar, ir a la sección "Agregar". Modificar, ir a la sección "Modificar". Eliminar, ir a la sección "Eliminar".
Sección "Agregar Usuario"	
Acción del actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz donde se introduce el nombre de usuario nuevo y el privilegio que se quiere.
3 El Administrador selecciona Verificar.	3.1 El sistema verifica si el nombre de usuario introducido no es estudiante. 3.2 Si no lo es, muestra sus datos.
4 El Administrador presiona la acción de Agregar.	4.1 El sistema registra las actualizaciones realizadas, y muestra un mensaje de confirmación.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.1 Si no es un profesor muestra un mensaje de error.

Capítulo 2 Características del Sistema

Sección “Modificar Usuario”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema busca los usuarios y los muestra en pantalla.
3 El Administrador realiza las actualizaciones deseadas.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2. El sistema actualiza la información.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.1 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección “Eliminar Usuario”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz con los usuarios del sistema.
3. El Administrador selecciona el usuario que desea eliminar y presiona eliminar.	3.1 El sistema muestra una interfaz con el listado actualizado de los usuarios.
Prioridad	Secundario.
Prototipo	[Anexo 17]

Tabla 2.16 Caso de uso: Gestionar usuario.

Caso de uso	
CU-11	Gestionar puntos de salida.
Propósito	Garantizar que se agreguen, modifiquen y eliminen puntos de salida del sistema.
Actores	Administrador (inicia).
Precondiciones	El Administrador debe haberse autenticado.
Poscondiciones	El Administrador debe haber agregado, modificado o eliminado un punto de salida.
Resumen: el caso de uso se inicia cuando el Administrador accede al sistema y solicita Gestionar puntos de salida. Inmediatamente se visualiza una pantalla donde el sistema brinda la posibilidad de efectuar las operaciones de agregar, modificar o eliminar los puntos de salida. Finaliza una vez que el Administrador realiza una de las operaciones.	
Referencias	R11.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1 El Administrador decide gestionar puntos de salida y selecciona una de las operaciones a realizar (agregar, modificar, eliminar).	1.1. El sistema ejecuta una de las siguientes acciones: Agregar, ir a la sección “Agregar”. Modificar, ir a la sección “Modificar”. Eliminar, ir a la sección “Eliminar”.
Sección “Agregar puntos de salida”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz donde se

Capítulo 2 Características del Sistema

	introduce el nombre del punto de salida deseado.
3 El Administrador introduce el punto de salida deseado.	3.1 El sistema verifica si el punto de salida no existe. 3.2 El sistema muestra un mensaje confirmando que se ha introducido el punto de salida.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.2 Si el punto de salida existe el sistema muestra un mensaje de error.
Sección “Modificar puntos de salida”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema busca los puntos de salidas existentes y los muestra en pantalla.
3. El Administrador realiza las actualizaciones deseadas.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2. El sistema actualiza la información.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.1 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección “Eliminar puntos de salida”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz con los puntos de salidas del sistema.
3 El Administrador selecciona el punto de salida que desea eliminar y presiona eliminar.	3.1 El sistema muestra una interfaz con el listado actualizado de los puntos de salidas.
Prioridad	Secundario.
Prototipo	[Anexo 18]

Tabla 2.17 Caso de uso: Gestionar puntos de salida.

Caso de uso	
CU-12	Gestionar municipios.
Propósito	Garantizar que se agreguen, modifiquen y eliminen municipios.
Actores	Administrador (inicia).
Precondiciones	El Administrador debe haberse autenticado.
Poscondiciones	El Administrador debe haber agregado, modificado o eliminado un municipio.
Resumen: el caso de uso se inicia cuando el Administrador accede al sistema y solicita Gestionar municipios. Inmediatamente se visualiza una pantalla donde el sistema brinda la posibilidad de efectuar las operaciones de agregar, modificar o eliminar municipio. Finaliza una vez que el Administrador realiza una de las operaciones.	
Referencias	R9.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema

Capítulo 2 Características del Sistema

1 El Administrador decide gestionar municipios y selecciona una de las operaciones a realizar (agregar, modificar, eliminar).	1.1 El sistema ejecuta una de las siguientes acciones: Agregar, ir a la sección “Agregar”. Modificar, ir a la sección “Modificar”. Eliminar, ir a la sección “Eliminar”.
Sección “Agregar municipio”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz donde se introduce el nombre del municipio que desea.
3 El Administrador introduce el municipio deseado.	3.1 El sistema verifica si el municipio no existe. 3.2 El sistema muestra un mensaje confirmando que se ha introducido el municipio correctamente.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.2 Si el municipio existe el sistema muestra un mensaje de error.
Sección “Modificar municipio”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema busca los municipios existentes y los muestra en pantalla.
3. El Administrador realiza las actualizaciones deseadas.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2. El sistema actualiza la información.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.1 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección “Eliminar municipio”	
Acción del actor	Respuesta del Sistema
	2.1 El sistema busca los municipios existentes y los muestra en la pantalla.
3 El Administrador selecciona el municipio que desea eliminar y presiona eliminar.	3.1 El sistema muestra una interfaz con el listado actualizado de los municipios.
Prioridad	Secundario.
Prototipo	[Anexo 19]

Tabla 2.18 Caso de uso: Gestionar municipios.

Caso de uso	
CU-13	Gestionar rutas.
Propósito	Garantizar que se agreguen, modifiquen y eliminen rutas.
Actores	Administrador (inicia).
Precondiciones	El Administrador debe haberse autenticado.
Poscondiciones	El Administrador debe haber agregado, modificado o eliminado una ruta.

Capítulo 2 Características del Sistema

Resumen: el caso de uso se inicia cuando el Administrador accede al sistema y solicita Gestionar rutas. Inmediatamente se visualiza una pantalla donde el sistema brinda la posibilidad de efectuar las operaciones de agregar, modificar o eliminar rutas. Finaliza una vez que el Administrador realiza una de las operaciones.	
Referencias	R8.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1 El Administrador decide gestionar rutas y selecciona una de las operaciones a realizar (agregar, modificar, eliminar).	1.1. El sistema ejecuta una de las siguientes acciones: Agregar, ir a la sección "Agregar". Modificar, ir a la sección "Modificar". Eliminar, ir a la sección "Eliminar".
Sección "Agregar ruta"	
Acción del actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz donde se introduce el nombre de la ruta que desea.
3 El Administrador introduce la ruta deseada.	3.1 El sistema verifica si la ruta no existe. 3.2 El sistema muestra un mensaje confirmando que se ha introducido la ruta correctamente.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.2 Si la ruta existe el sistema muestra un mensaje de error.
Sección "Modificar ruta"	
Acción del actor	Respuesta del Sistema
	2.1 El sistema busca los municipios existentes y los muestra en pantalla.
3. El Administrador realiza las actualizaciones deseadas.	3.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 3.2. El sistema actualiza la información.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	3.1 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección "Eliminar ruta"	
Acción del actor	Respuesta del Sistema
	2.1 El sistema busca las rutas existentes y las muestra en la pantalla.
3 El Administrador selecciona la ruta que desea eliminar y presiona eliminar.	3.1 El sistema muestra una interfaz con el listado actualizado de las rutas.
Prioridad	Secundario.
Prototipo	[Anexo 20]

Tabla 2.19 Caso de uso: Gestionar rutas.

Capítulo 2 Características del Sistema

Caso de uso	
CU-14	Reservar comida.
Propósito	Garantizar dar baja de la planificación alimenticia para el fin de semana a los estudiantes reservados en dicha semana.
Actores	Reloj (inicia).
Precondiciones	Que la fecha del sistema sea sábado a las 9:00 AM.
Poscondiciones	Son actualizadas las Bases de Datos de los comedores con los listados de estudiantes que no tienen comida asignada para el fin de semana en curso.
Resumen: el Caso de Uso se inicia cuando llega una fecha y una hora determinada, es activado por el Reloj y la función es garantizar que se elabore la comida para todos los estudiantes que no reservaron transporte en la semana correspondiente.	
Referencias	R5.
Curso normal de los eventos.	
Acción del actor	Respuesta del sistema
1. El Reloj marca las 9:00 AM del sábado.	1.1 Se ejecuta una tarea programada para que el sistema actualice en las Bases de Datos de los comedores y de baja de la planificación alimenticia para el fin de semana a los estudiantes reservados en dicha semana.
Prioridad	Crítico.
Prototipo	

Tabla 2.20 Caso de uso: Reservar comida.

2.6 Conclusiones.

En el presente capítulo comenzó a desarrollarse una propuesta de solución al problema planteado, obteniéndose a partir del análisis de los procesos del negocio un listado con las funcionalidades que debe tener el software final, las cuales fueron presentadas mediante el diagrama de casos de uso. Finalmente fueron descritos paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Ya se puede comenzar a desarrollar el sistema, siempre cumpliendo con los requerimientos establecidos en este capítulo.

Capítulo 3 Análisis y Diseño del Sistema

3.1 Introducción.

En el presente capítulo se realiza una traducción de los requisitos a una especificación que describe como realizar la implementación del sistema. Con esta finalidad son modelados los artefactos que ayudan a manejar las complicaciones que implica la construcción de aplicaciones Web. Para ello los componentes de la aplicación son tratados como clases y son representados a través de los diagramas Web los cuales detallan la interacción en las distintas páginas Web. Es además estructurada la información que se requiere persista a través del diseño de la base de datos. Por último son abordados los principios de diseño seguidos en el desarrollo de la aplicación.

3.2 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. Incluida 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. [28]

En la construcción del modelo de análisis 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. [28]

3.2.1 Diagrama de clases del análisis.

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos de un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de las aplicaciones. [28]

Capítulo 3 Análisis y Diseño del Sistema

A continuación se muestran los diagramas de clases del análisis de los casos de uso críticos, para el resto de los casos de uso ver [Anexo 21, 22, 23, 24, 25, 26, 27, 28, 29, 30].

3.2.1.1 Diagrama de clases del caso de uso “Gestionar Pase”.

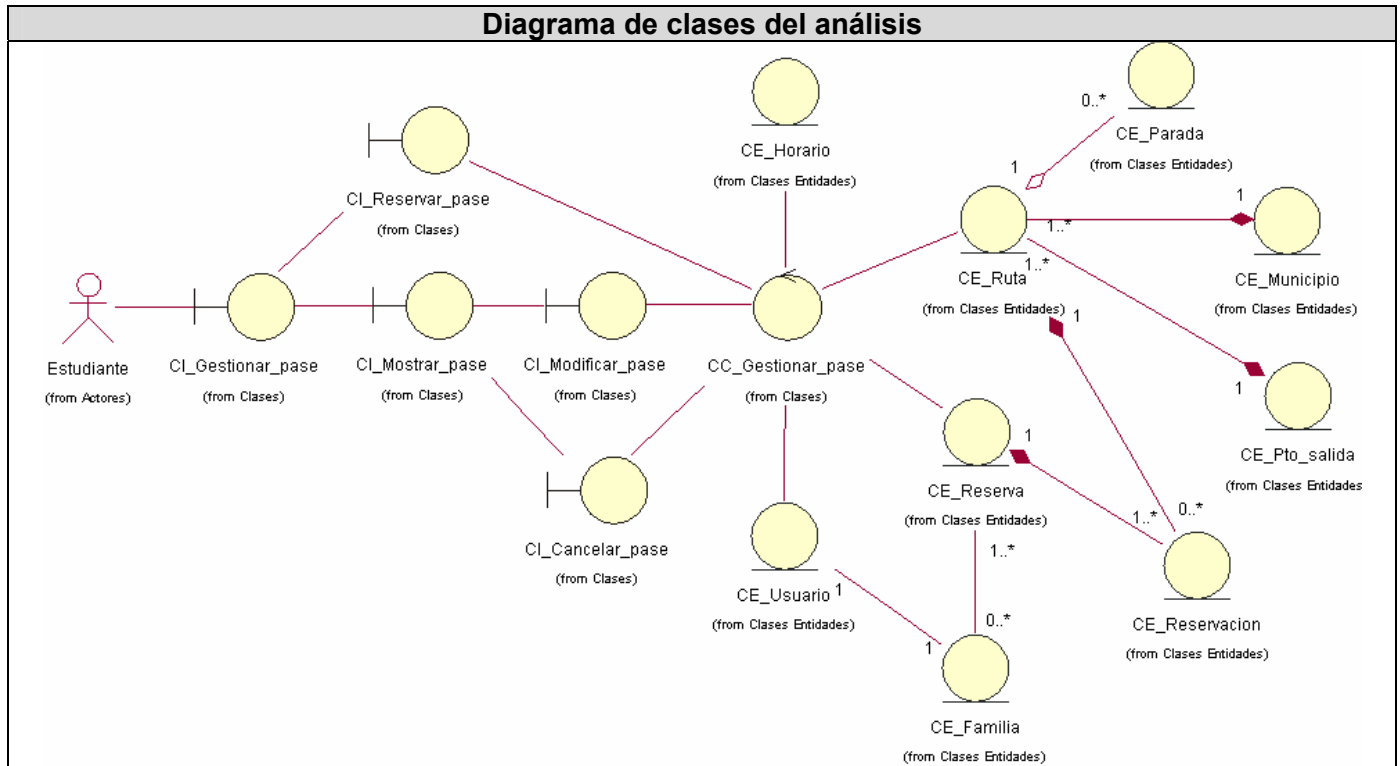


Figura 3.1 Diagrama de clases del análisis de “Gestionar Pase”.

3.2.1.2 Diagrama de clases del caso de uso “Autenticar”.

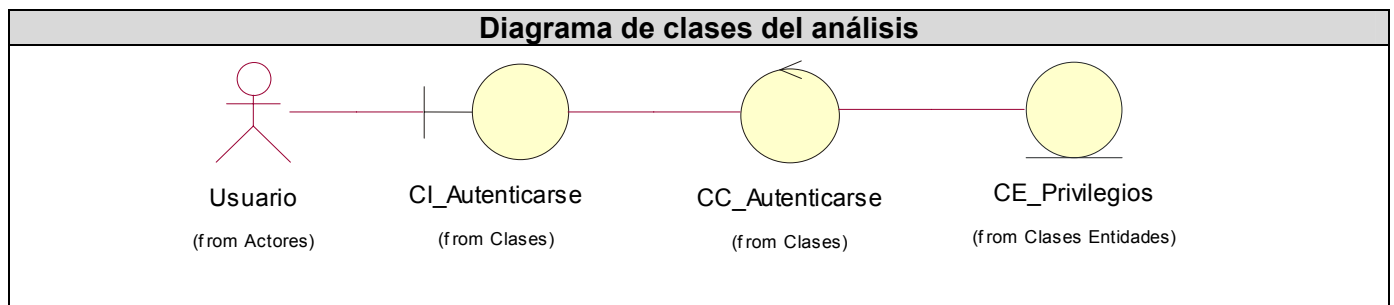


Figura 3.2 Diagrama de clases del análisis de “Autenticar”.

Capítulo 3 Análisis y Diseño del Sistema

3.2.1.3 Diagrama de clases del caso de uso “Imprimir boletines”.

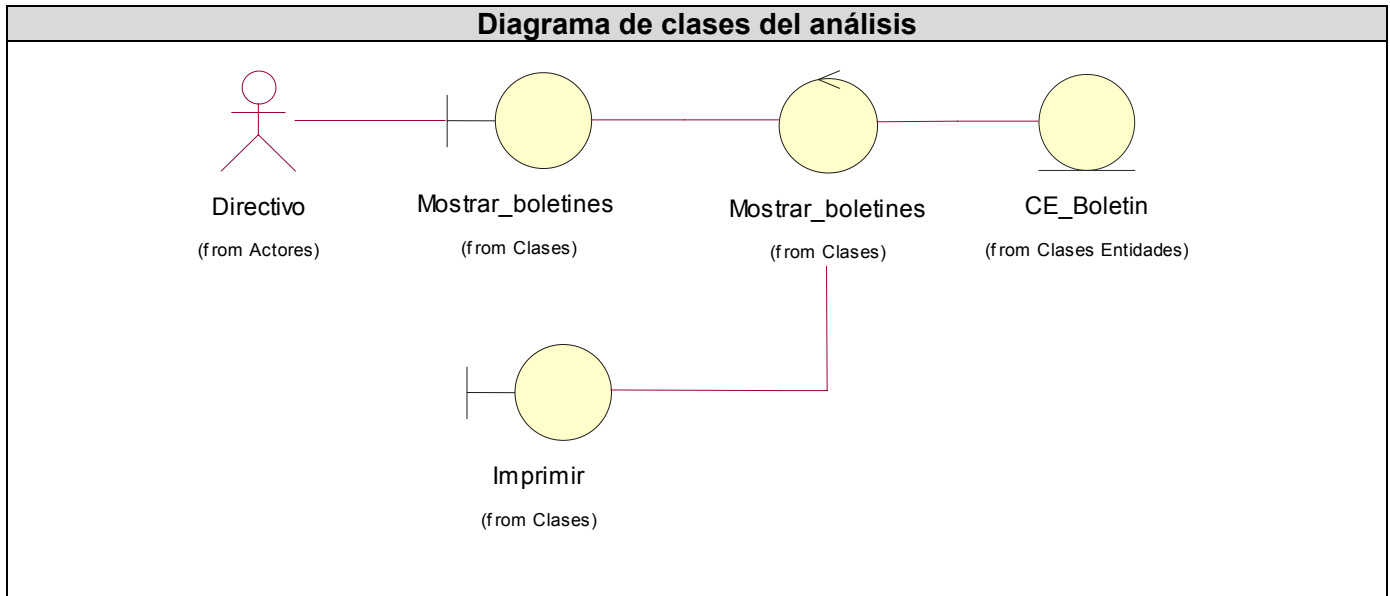


Figura 3.3 Diagrama de clases del análisis de "Imprimir boletines".

3.3 Modelo de diseño.

El modelo de diseño es un modelo de objetos que describe la relación física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto a otras restricciones relacionadas con el entorno de implementación tienen impacto en el sistema a desarrollar. Una entrada esencial en el diseño es el resultado obtenido en el análisis, más bien, el modelo de análisis que proporciona una comprensión detallada de los requisitos.

En el modelo del diseño, los casos de usos son realizados por las clases del diseño y sus objetos. Esto es representado a través de colaboraciones en el modelo del diseño y denota la realización de casos de uso del diseño. [29]

3.3.1 Clases Base.

Los casos de uso del sistema definidos en el capítulo anterior, deben encajar en la arquitectura cuando se llevan a cabo, mientras que la arquitectura debe permitir el desarrollo de los casos de uso requeridos

Capítulo 3 Análisis y Diseño del Sistema

ahora y en el futuro. De esa manera, la arquitectura debe diseñarse para permitir que el sistema evolucione, que los desarrolladores puedan progresar hasta obtener una visión común, que se organice el desarrollo del software y que se fomente la reutilización.

Teniendo en cuenta las posibilidades que ofrece el patrón arquitectónico Modelo-Vista-Controlador, descrito en el Capítulo 1 se decide hacer uso del mismo para definir el sistema de clases base de la aplicación, pues se persigue facilidad de mantenimiento, escalabilidad, rapidez de desarrollo e independencia entre las distintas capas del sistema para facilitar su futura evolución.

3.3.1.1 Diagrama de las clases base.

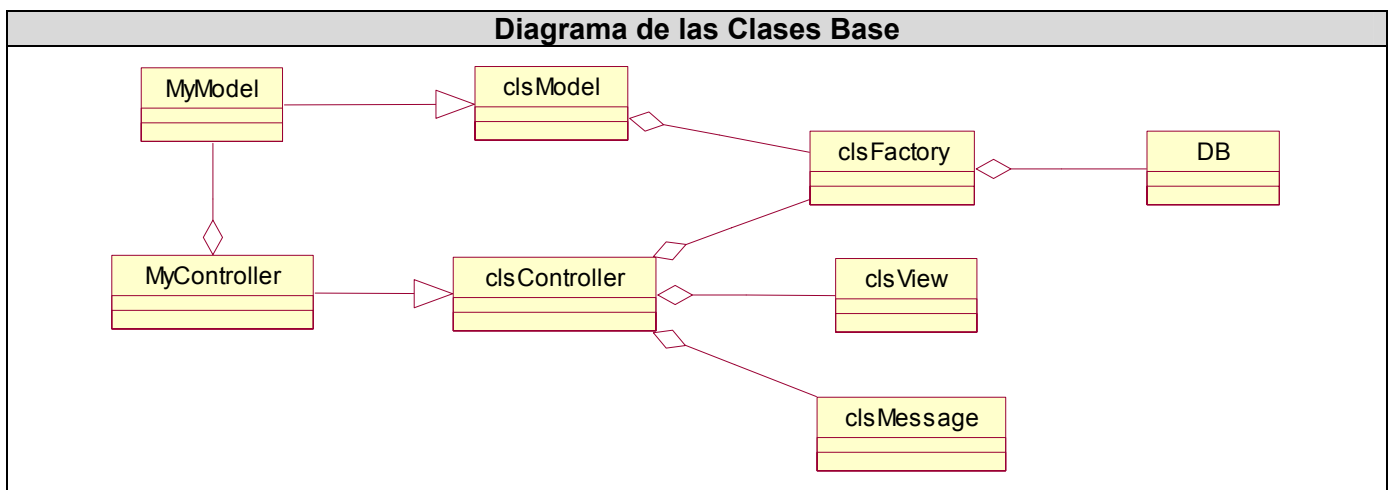


Figura 3.4 Diagrama de las clases base.

Capítulo 3 Análisis y Diseño del Sistema

3.3.1.2 Diagrama de secuencia.

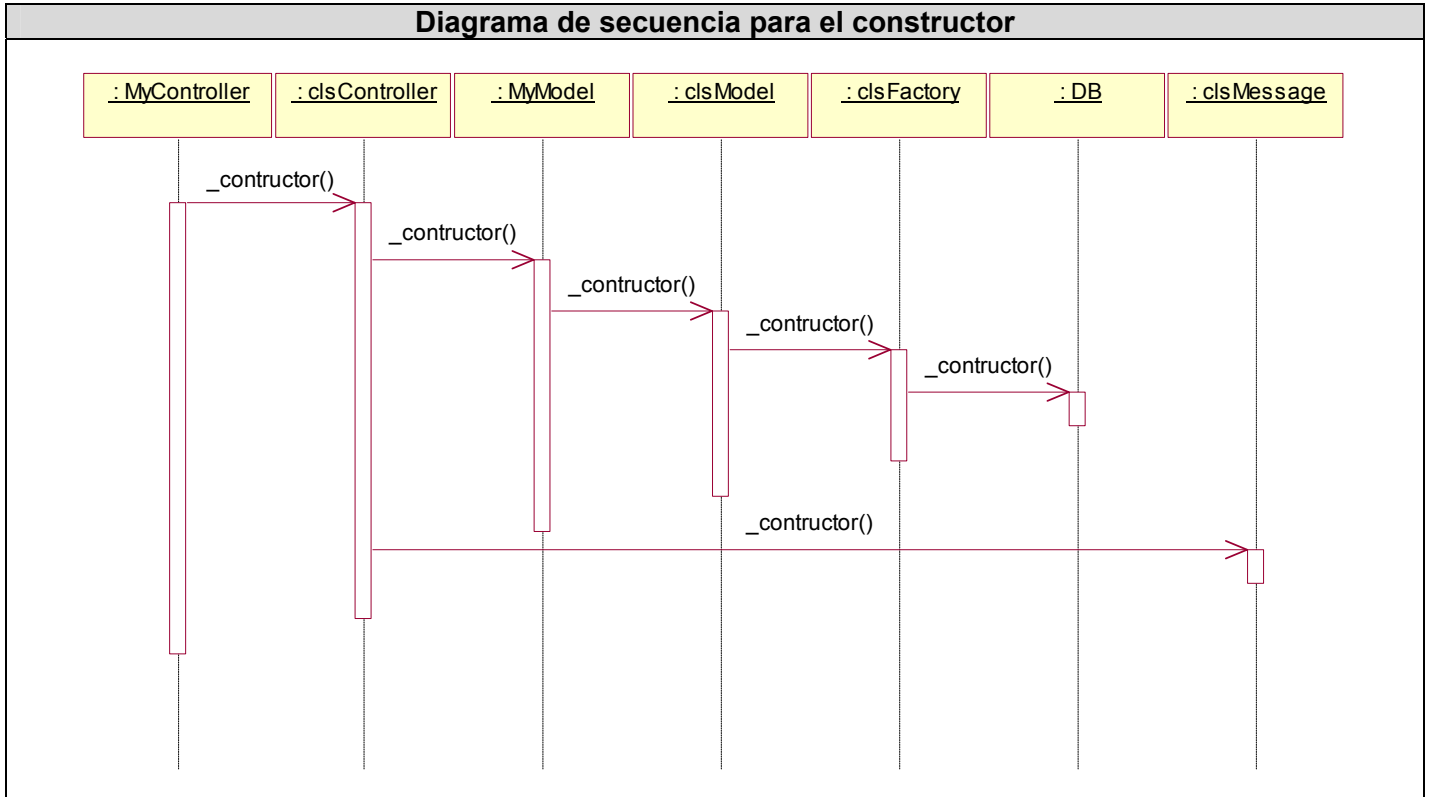


Figura 3.5 Diagrama de secuencia para el constructor.

3.3.2 Diagrama de clases del Diseño Web.

En las aplicaciones Web, donde es más importante la modelación de la lógica y el estado del negocio que los detalles de presentación, un diagrama de clases presenta las páginas, los enlaces entre estas (donde cada página lógica puede ser representada como una clase), todo el código que irán creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente.

El diagrama de clases Web, fue definido, 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 críticos, para el resto de los casos de uso ver [\[Anexo 31, 32, 33, 34, 35, 36, 37, 38, 39, 40\]](#).

Capítulo 3 Análisis y Diseño del Sistema

3.3.2.1 Caso de uso “Gestionar Pase”.

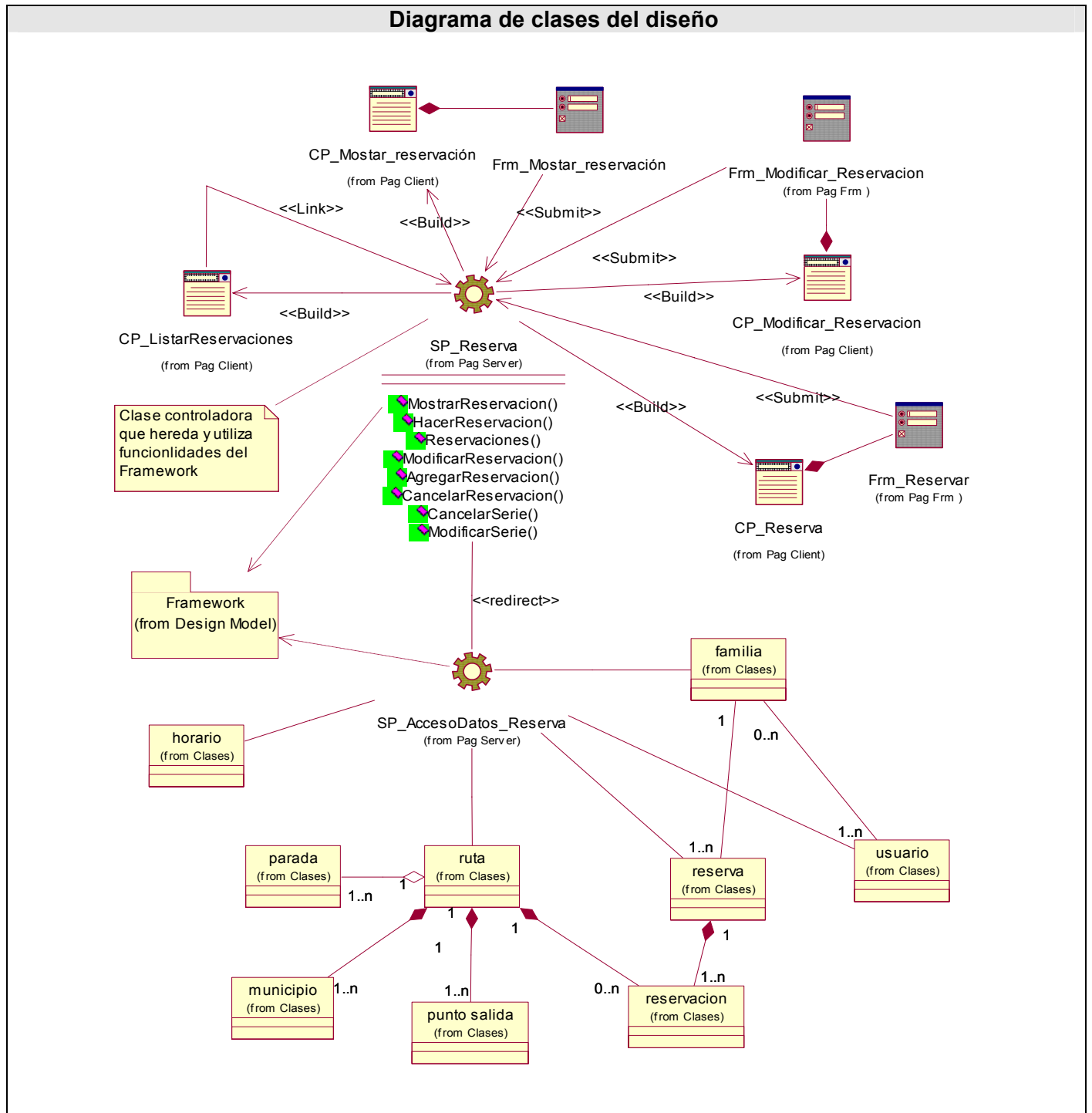


Figura 3.6 Diagrama de clases del diseño de “Gestionar Pase”.

Capítulo 3 Análisis y Diseño del Sistema

3.3.2.2 Caso de uso “Autenticar”.

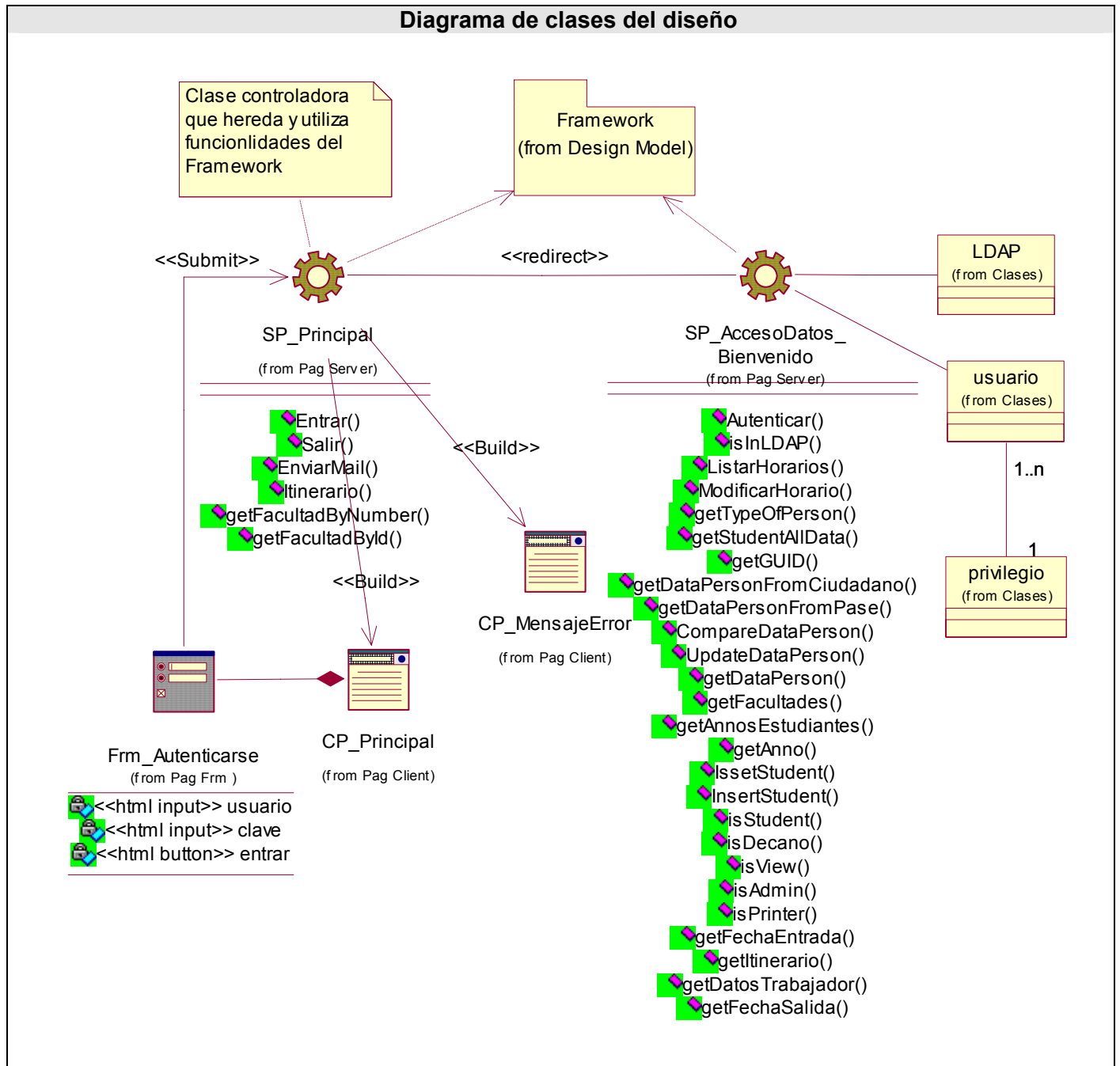


Figura 3.7 Diagrama de clases del diseño de “Autenticar”.

Capítulo 3 Análisis y Diseño del Sistema

3.3.2.3 Caso de uso “Imprimir boletines”.

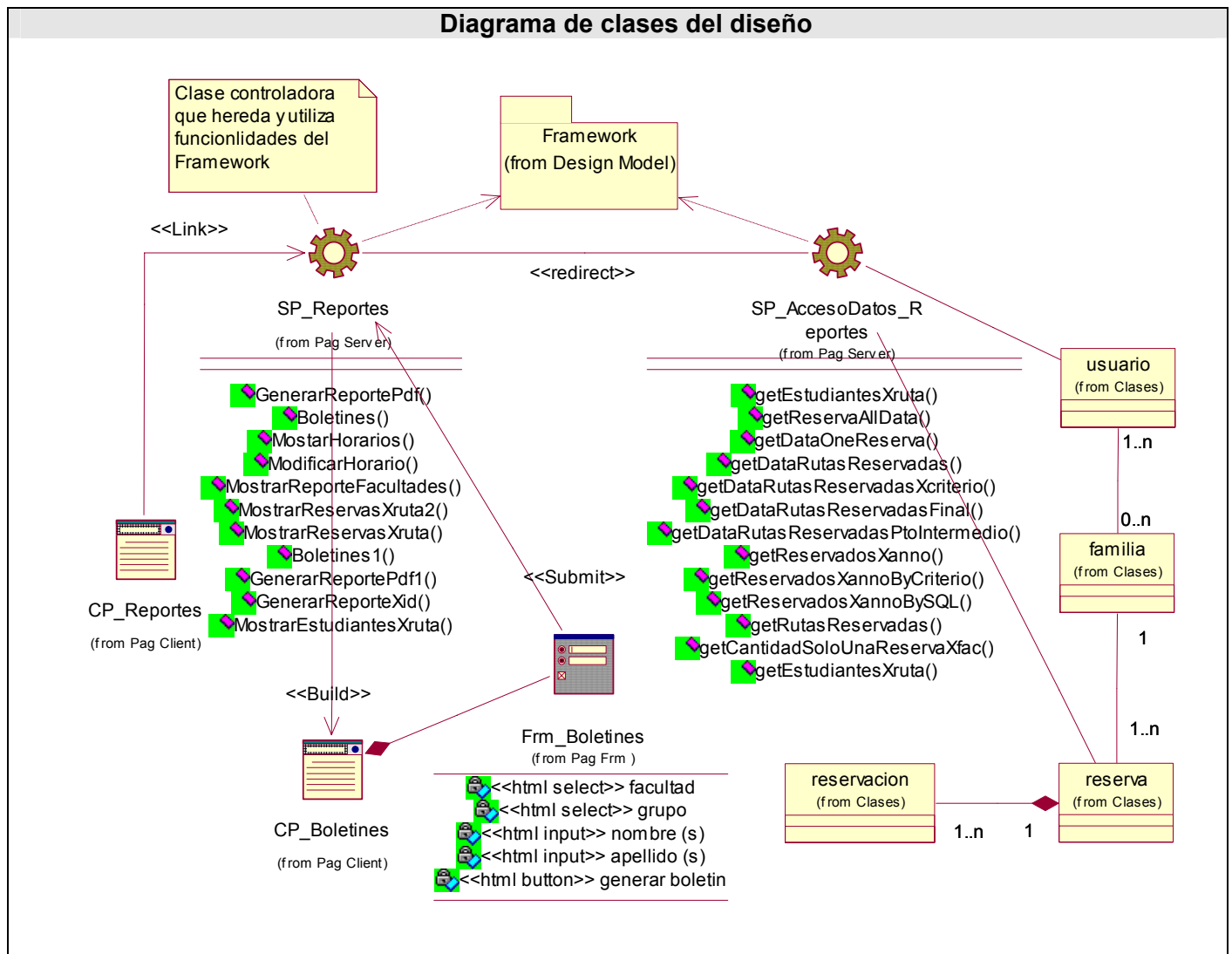


Figura 3.8 Diagrama de clases del diseño de "Imprimir boletines".

3.4 Diseño de la base de datos.

Para una correcta realización del diseño de la base de datos, se ha utilizado el diagrama de clases persistentes así como el modelo de datos que están basados en el modelamiento de las clases del epígrafe anterior. Algunas de estas clases representan los datos que se obtienen y almacenan durante los procesos de la aplicación, estas son las que pueden ser modeladas a través de un diagrama de clases

Capítulo 3 Análisis y Diseño del Sistema

persistentes, lo que permite a su vez ver la relación existente entre los datos manejados en la aplicación, para de esta forma completar el modelamiento de la lógica del negocio del sistema. Por su parte el modelo de datos describe la representación lógica y física de dichos datos persistentes del sistema.

3.4.1 Diagrama de clases persistentes.

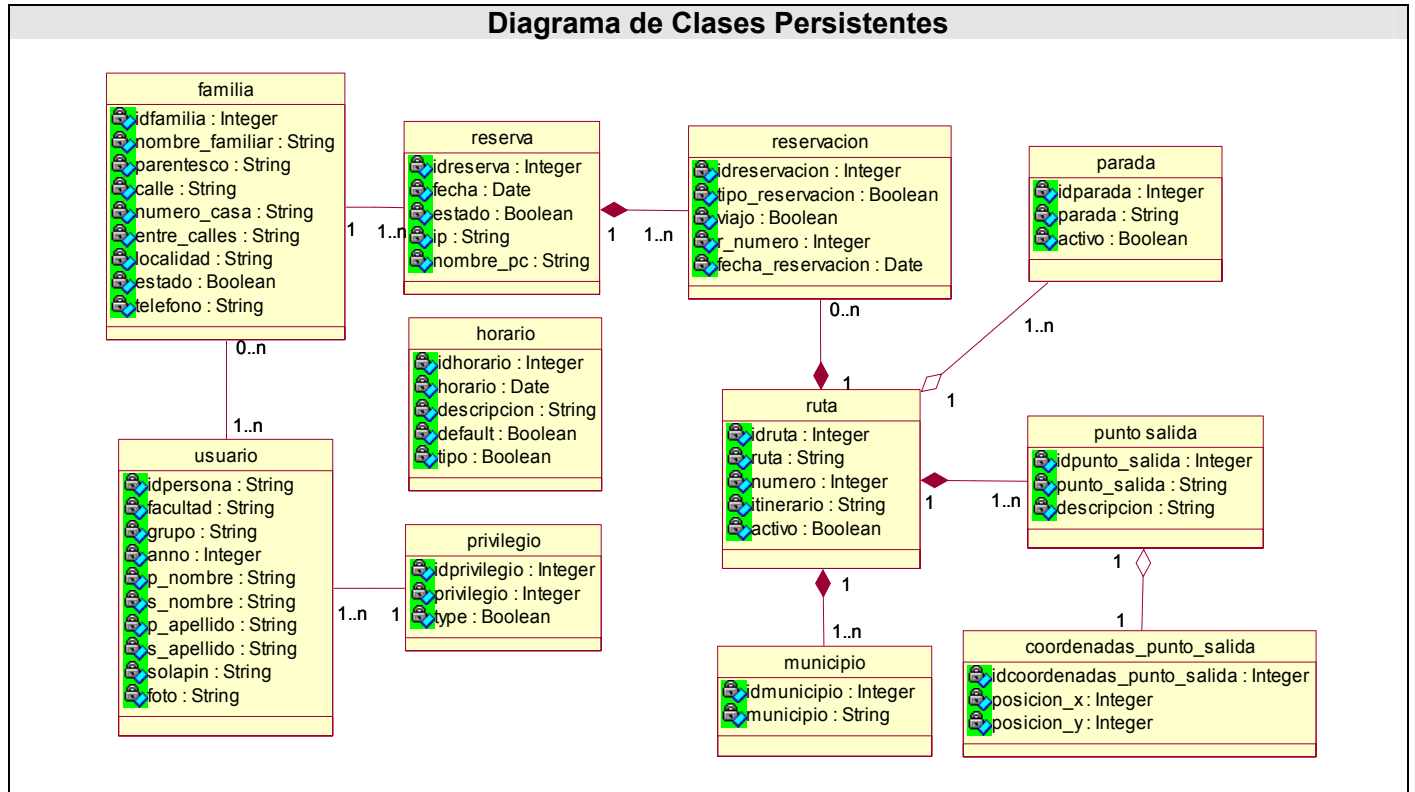


Figura 3.9 Diagrama de clases persistentes.

3.4.2 Modelo de datos.

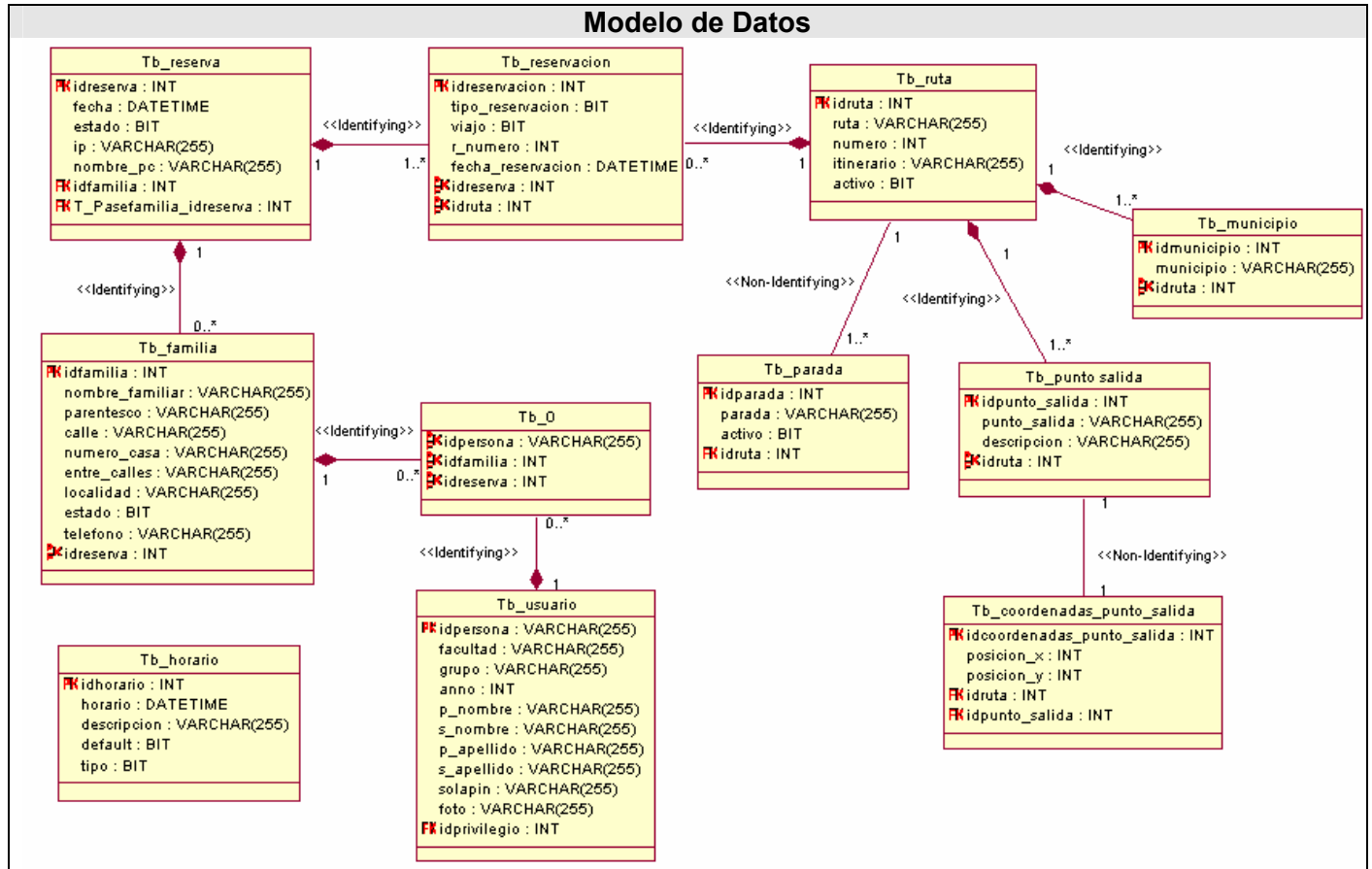


Figura 3.10 Modelo de datos.

3.5 Principios de diseño aplicados.

El correcto diseño de la interfaz de una pequeña, grande o mediana aplicación, la forma en que es presentado al usuario final el conjunto de los reportes, la elaboración de una completa y detallada ayuda y el tratamiento de los errores son los que definen el éxito o el fracaso que tenga en el futuro cualquier aplicación. A continuación son descritos los principios de diseño utilizados para el desarrollo de la presente aplicación. [22]

Capítulo 3 Análisis y Diseño del Sistema

3.5.1 Interfaz de usuario.

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”. [22]

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.

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.
- Obtención en todo momento de información de retroalimentación.
- Diseño de cuadros de mensajes que informen al usuario cada una de las acciones llevadas a cabo por él.
- Previsiones de errores y manejo de errores de entrada de datos por parte del usuario.
- Lograr que los usuarios deshagan las acciones de forma fácil.
- Interfaz sencilla para que el usuario en ningún momento se sienta desorientado.

Una de las características fundamentales de una aplicación es la ventaja que ofrecen las interfaces Web sobre las interfaces de escritorio. Las interfaces Web:

- Proponen un ambiente mucho más amigable.
- Conducen a un aprendizaje por parte del usuario mucho más rápido y fácil.

Además de estos principios se tuvieron en cuenta las siguientes características:

- Usar la misma tipografía, forma y estilos en todas las páginas.

Capítulo 3 Análisis y Diseño del Sistema

- La facilidad de poder navegar desde cualquier punto a otro dentro de la aplicación.
- Se tuvo presente el ancho de banda a la hora de realizar la carga de las páginas de la aplicación y a la hora de la elección de las imágenes que en ella se muestran.
- La simplicidad y consistencia, favoreciendo así la usabilidad de la aplicación.
- Se mantuvo estabilidad y uniformidad en el diseño.

Fueron además utilizadas las hojas de estilo (css) para guardar la configuración del diseño en todas las páginas de una u otra vista (estudiantes o directivos), para los botones y las líneas fueron utilizados estos estilos, eliminando así el número considerable de imágenes que demoran la carga y presentación de las páginas de la aplicación. Los formularios de entrada fueron situados de forma tal que ocupen el centro superior y las entradas de dichos formularios fueron organizadas según su importancia.

La funcionalidad se estructuró de forma tal que sean realizadas varias operaciones en una misma página, de esta forma el usuario no tiene la obligación de moverse excesivamente dentro de la aplicación para realizar el completamiento de una operación.

3.5.2 Formato de salida de los reporte.

La aplicación genera reportes que permiten llevar un control total de los estudiantes que reservan transporte. Estos reportes son obtenidos según las necesidades del usuario. Los reportes han sido diseñados haciendo uso de letra claramente legible y colores claros en el fondo, evitando recargar la página y lograr calidad y nitidez a la hora de presentar la información.

Cada reporte tiene un encabezado que lo identifica y lo describe brevemente, luego se muestra la información consultada de forma legible y organizada. Ejemplos de los diferentes reportes son mostrados. Ver. [\[Anexo 14\]](#)

3.5.3 Tratamiento de errores.

En el sistema propuesto se evitan y minimizan los posibles errores, con el fin de garantizar la integridad y confiabilidad que en son mostrados y registrados los datos del mismo. Los mensajes de error emitidos por el sistema son de fácil comprensión para los usuarios, cuando se introduce información en un formulario y carece de datos, se dispara un cuadro de alerta indicando el campo o dato faltante, muy similar ocurre

Capítulo 3 Análisis y Diseño del Sistema

cuando el formato de los datos introducidos no es el correcto, como por ejemplo el formato de fecha (diferente al formato inglés mm/dd/YYYY), cuando se introducen datos erróneos como por ejemplo caracteres en campos donde deben ir solo números, se disparan ventanas de alertas informándole al usuario lo más claro posible.

3.5.4 Seguridad.

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

3.6 Conclusiones.

En este capítulo fueron desarrollados los diagramas de clases y el diseño de la base de datos de la aplicación. Fueron descritos además los principios del diseño seguidos, específicamente: estándares de la interfaz, concepción y tratamiento de errores, y seguridad.

En este capítulo fueron tratadas además las concepciones para el análisis y diseño de la aplicación, fueron confeccionados los diagramas de clases tanto del análisis como del diseño. Fueron descritas cada una de las clases empleadas en el desarrollo de la aplicación así como los métodos que las componen, se describieron además las tablas de la base de datos que componen la aplicación.

Capítulo 4 Implementación del Sistema

4.1 Introducción.

En el presente capítulo son representados los diagramas de despliegue y componentes. Es representada la distribución física del sistema en términos de cómo es distribuida la funcionalidad entre los nodos de cómputo, los cuales operan sobre la aplicación. Son modelados además la organización y dependencia entre los distintos componentes del sistema. Se modela asimismo en el presente capítulo los agregados que existen dentro de los nodos, las bibliotecas y archivos necesarios para el correcto funcionamiento del sistema.

4.2 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. [28]

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 este 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. [28]

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 aplicación a desarrollar:

Capítulo 4 Implementación del Sistema

4.2.1 Diagrama de despliegue.

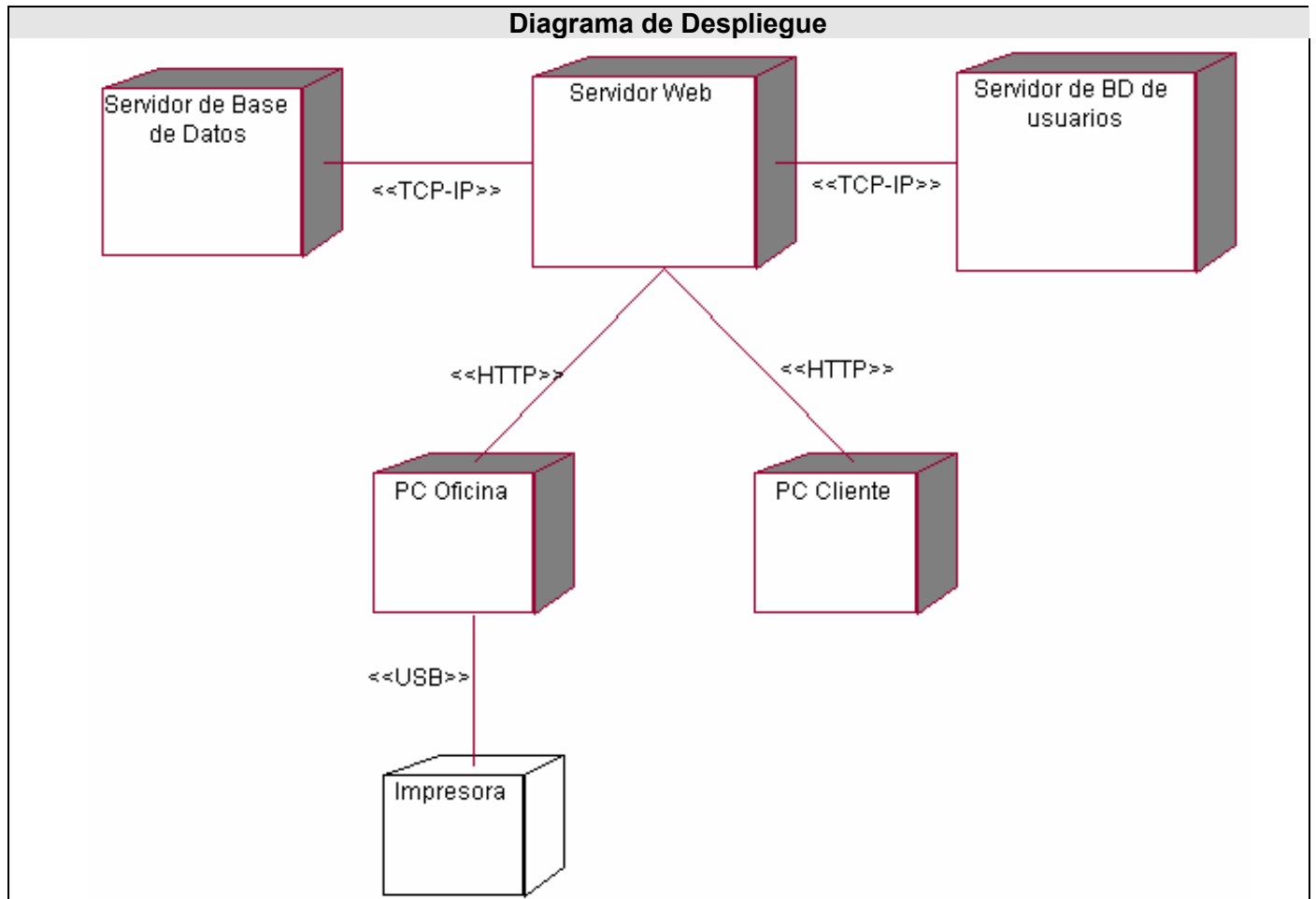


Figura 4.1 Diagrama de despliegue.

4.3 Diagramas de componentes.

Los Diagramas de componentes son unos de los dos tipos de diagramas que aparecen cuando se modelan los aspectos físicos de los sistemas Orientados a Objetos. Un diagrama de componentes muestra la organización y dependencias entre un conjunto de componentes. Los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema. Esto implica modelar las cosas físicas que residen en un nodo de cómputo, tales como ejecutables, bibliotecas,

Capítulo 4 Implementación del Sistema

archivos y documentos. Los diagramas de componentes son fundamentalmente diagramas de clases que se centran en los componentes de un sistema. [28]

Los diagramas de componentes no solo son importantes para visualizar, especificar y documentar sistemas basados en componentes sino también para construir sistemas ejecutables mediante ingeniería inversa o directa. [28]

4.3.1 Diagrama de componentes “Acceso a Datos”.

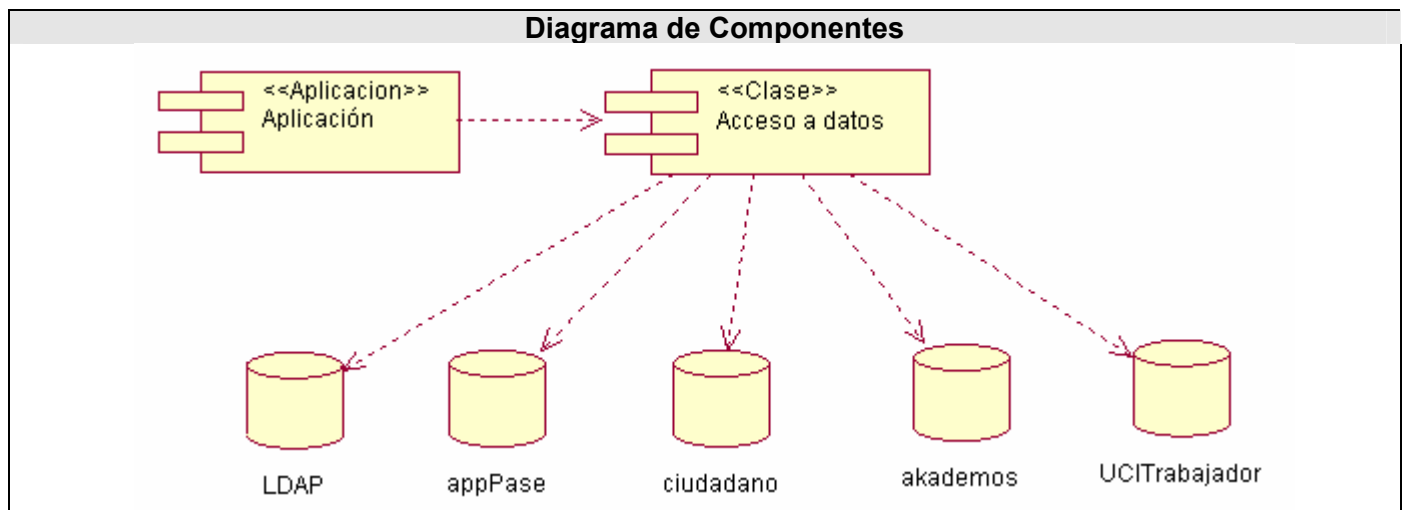


Figura 4.2 Diagrama de componentes “Acceso a datos”.

Capítulo 4 Implementación del Sistema

4.3.2 Diagrama de componentes “Componentes Web o código ejecutable”.

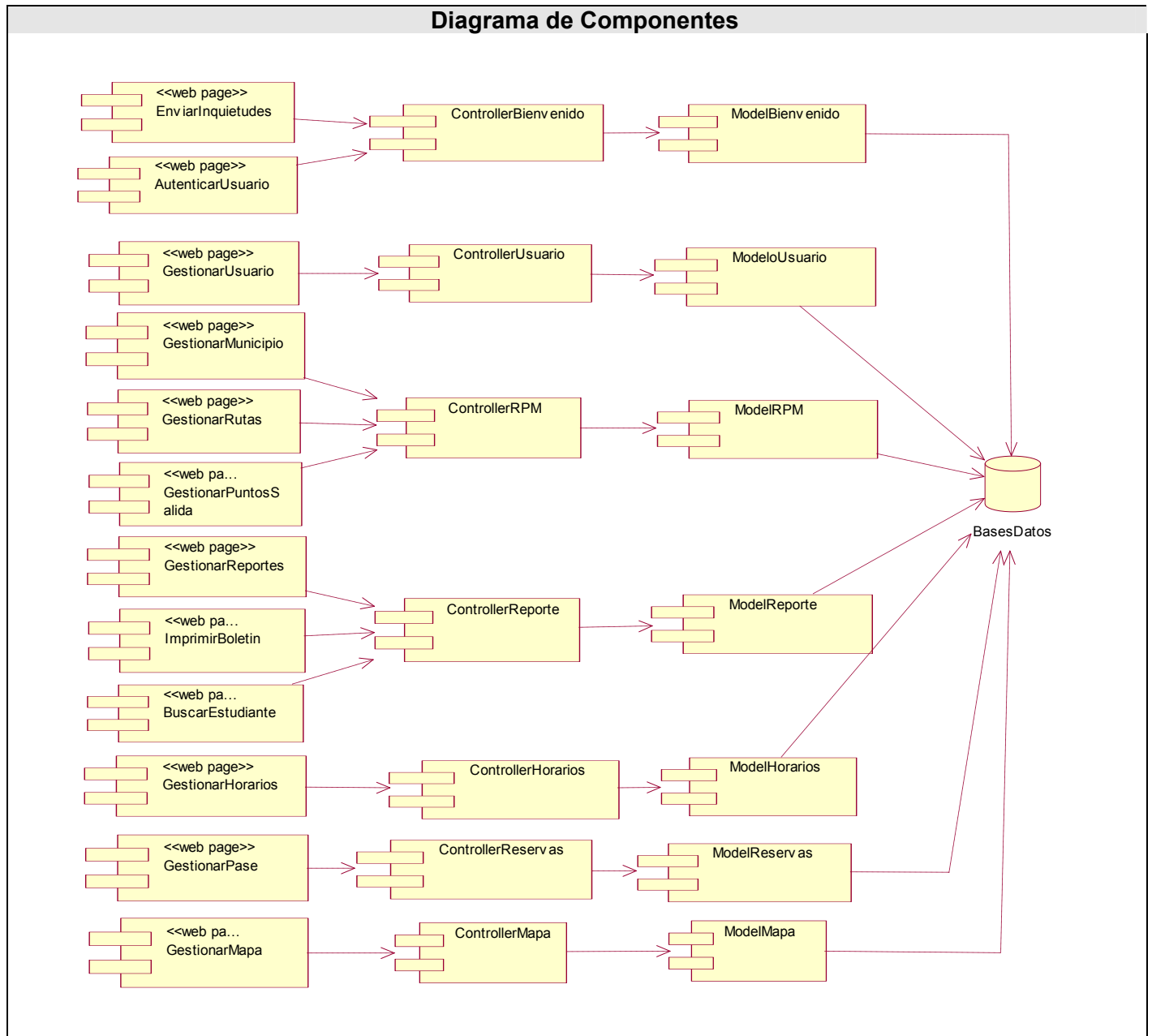


Figura 4.3 Diagrama de componentes “Componentes Web o código ejecutable”.

Capítulo 4 Implementación del Sistema

4.3.3 Diagrama de componentes “Código Fuente”.

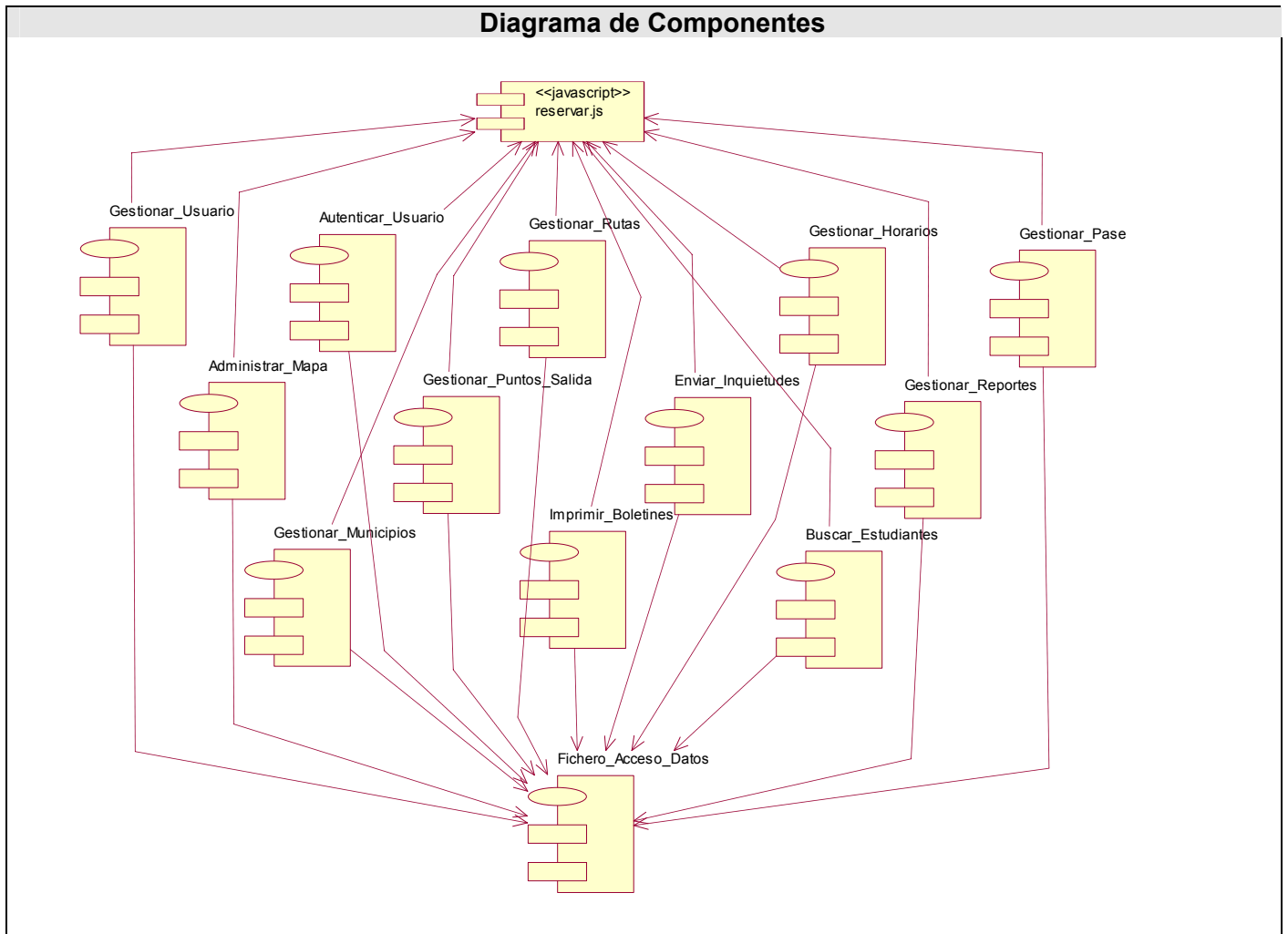


Figura 4.4 Diagrama de componentes “Código fuente”.

4.4 Conclusiones.

En este capítulo son tratados algunos conceptos básicos para que el lector pueda familiarizarse con el sistema en términos de implementación. Son modelados además los diagramas de despliegue y componentes de la aplicación. Es mostrada la organización y dependencia existente entre los componentes que forman al sistema. Es mostrada la topología de hardware sobre la que estará corriendo el sistema.

Capítulo 5 Estudio de Factibilidad

5.1 Introducción.

Para la realización de todo proyecto es de considerable importancia el análisis del costo y los beneficios que el mismo debe reportar una vez finalizado. Como resultado de este análisis es obtenido el tiempo de desarrollo en meses del proyecto que se llevará a cabo, el costo y la cantidad de personas necesarias para el desarrollo del mismo.

En este capítulo es descrita la estimación de costo del sistema, los beneficios que el mismo aportará y el tiempo en que será desarrollado.

5.2 Planificación Basada en Puntos de Casos de Uso.

El cálculo de estimación mediante en análisis de Puntos de Casos de Uso, es un método propuesto por Gustav Karner en su tesis en 1993 (Universidad de Linkoping, Suecia) y supervisado por Ivar Jacobson de Objectory AB. Fue analizado posteriormente en muchos otros estudios, como la tesis de Kirsten Ribu (Universidad de Oslo, Noruega) en el 2001 y hasta estos días refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a un cierto número de factores que lo afectan, para de esta forma finalmente contabilizar el tiempo total del proyecto a partir de dichos factores. [30]

5.2.1 Cálculo de Puntos de Casos de Uso sin ajustar.

$$UUCP = UAW + UUCW$$

Donde: UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Capítulo 5 Estudio de Factibilidad

- Para calcular UAW: Para realizar el cálculo del peso de los actores sin ajustar se realiza un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos como se muestra en la siguiente tabla.

Tipo	Descripción	Peso	Cant * peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0 * 1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0 * 2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	4 * 3
Total			12

Tabla 5.1 Factor de peso de los actores sin ajustar.

Después de obtener este valor se pasa a calcular el factor de peso de los casos de uso sin ajustar, realizando un análisis de la cantidad de casos de uso presentes en el sistema y el nivel de complejidad de cada uno de ellos como se muestra en la siguiente tabla:

- Para calcular UUCW:

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	7 * 5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	6 * 10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	1 * 15
Total			110

Tabla 5.2 Factor de peso de los casos de uso sin ajustar.

Luego: $UUCP = 12 + 110$

$$UUCP = 122$$

5.2.2 Cálculo de los puntos de casos de uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde: UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Capítulo 5 Estudio de Factibilidad

- Para Calcular TCF:

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

Este coeficiente puede ser calculado mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. A continuación es mostrado en la tabla 5.2 el peso de cada uno de estos factores en nuestro sistema acompañado de un breve comentario.

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido.	2	5	10
T2	Objetivos de performance o tiempo de respuesta.	1	1	1
T3	Eficiencia del usuario final.	1	5	5
T4	Procesamiento interno complejo.	1	2	2
T5	El código debe ser reutilizable.	1	4	4
T6	Facilidad de instalación.	0.5	4	2
T7	Facilidad de uso.	0.5	5	2.5
T8	Portabilidad.	2	4	8
T9	Facilidad de cambio.	1	4	4
T10	Concurrencia.	1	5	5
T11	Incluye objetivos especiales de seguridad.	1	4	4
T12	Provee acceso directo a terceras partes.	1	3	3
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	2	2
Total				52.5

Tabla 5.3 Factor de complejidad técnica.

Luego: $\text{TCF} = 0.6 + 0.01 * 52.5$

$$\text{TCF} = 1.125$$

- Para Calcular EF:

$$\text{EF} = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

Capítulo 5 Estudio de Factibilidad

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	6
E2	Experiencia en la aplicación.	0.5	4	2
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	4	2
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	2	4
E7	Personal part-time.	-1	3	-3
E8	Dificultad del lenguaje de programación.	-1	3	-3
Total				17

Tabla 5.4 Factor de ambiente.

Luego: $EF = 1.4 - 0.03 * 17$

$$EF = 0.89$$

Finalmente $UCP = 122 * 1.125 * 0.89$

$UCP = 122.1525 \approx 122.15$ estos son los puntos de casos de uso ajustados.

5.2.3 Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP * CF$$

Donde: E: Esfuerzo estimado en horas hombre.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

- Para calcular CF:

Total_{EF} = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

CF = 20 horas-hombre (si Total_{EF} ≤ 2)

CF = 28 horas-hombre (si Total_{EF} = 3 ó Total_{EF} = 4)

CF = abandonar o cambiar proyecto (si Total_{EF} ≥ 5)

Capítulo 5 Estudio de Factibilidad

O sea, para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por encima de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Como $Total_{EF} = 1 + 0$

$Total_{EF} = 1$

$CF = 20$ horas-hombre (porque $Total_{EF} \leq 2$)

Luego $E = 122.15 * 20$ horas-hombre

$E = 2443$ horas-hombre

5.2.4 Calcular el esfuerzo de todo el proyecto (ET).

Debe tenerse en cuenta que el cálculo anterior brinda una estimación del esfuerzo en horas – hombres contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso. Finalmente, para una estimación mucho mas completa de la duración total del proyecto, se debe agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello se debe tener en cuenta el siguiente criterio, el cual estadísticamente es considerado bastante aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, teniendo en cuenta la siguiente aproximación.

Actividad	Porcentaje	Horas-Hombres
Análisis.	10%	610.75
Diseño.	20%	1221.5
Implementación.	40%	2443
Pruebas.	15%	916.125
Sobrecarga. (otras actividades)	15%	916.125
Total.	100%	6107.5

Tabla 5.5 Distribución del esfuerzo.

Capítulo 5 Estudio de Factibilidad

Como el valor de esfuerzo calculado representa el esfuerzo del FT implementación, por comparación salen el resto de los esfuerzo y la suma de ellos es el **esfuerzo total (E_T)**.

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 24 días hábiles; la cantidad de horas que puede trabajar una persona en 1 mes es 192 horas.

Si $E_T = 6107.5$ **horas-hombre** y por cada 192 horas se tiene 1 mes eso daría un $E_T = 31.8$ **mes-hombre**. Esto quiere decir que un hombre puede realizar el sistema analizado en un período de 31.8 meses.

En caso de que el equipo esté compuesto por 5 personas y ambas realizan el mismo esfuerzo, entonces el problema analizado puede terminarse en aproximadamente 6 meses.

5.2.5 Costo total del proyecto.

La cantidad de de hombres (CH) involucrados en el desarrollo del proyecto es 5 y se asume como salario promedio mensual (CHM) \$180.00.

$$\text{Costo Total} = \text{CHM} * (E_T / \text{CH})$$

$$\text{Costo Total} = 180.00 * 6$$

$$\text{Costo Total} = \$ 1080$$

De los resultados obtenidos se interpreta que con 5 hombres trabajando en el proyecto, el mismo se desarrolla en 6 meses y su costo total se estima que sea \$1080.

5.3 Beneficios tangibles e intangibles.

El Módulo pases programados incorporado al Sistema de Reservación de Transporte no es un producto con fines comerciales, su principal objetivo es resolver los problemas existentes en el proceso de reservación de transporte para el pase estudiantil en la UCI.

Por tanto, el beneficio fundamental es contar con la comodidad y las facilidades que proporciona una reprogramación del módulo de reservas del Sistema de Reservación de Transporte para permitir a los estudiantes realizar una planificación de su reservación de pase.

Capítulo 5 Estudio de Factibilidad

Por tanto, los beneficios inmediatos son generalmente intangibles:

- ◆ El estudiante podrá reservar pase para la fecha que desee sin tener la necesidad de esperar a encontrarse en esa semana.
- ◆ El estudiante podrá realizar una reservación periódica especificando la frecuencia y el intervalo en el que se mantendrá vigente dicha reservación.
- ◆ El estudiante podrá visualizar todas las reservaciones hechas por él y modificarlas o cancelarlas (mientras sea tiempo de reservar y cancelar respectivamente).

5.4 Análisis de costos y beneficios.

Desarrollar un producto informático resulta costoso, justificar entonces su realización depende de los beneficios que reportaría su implantación y utilización.

Como ya se ha mencionado en el epígrafe anterior, el módulo que se propone aportaría beneficios inmediatos intangibles de suma importancia, pues una vez implantado, contribuirá a aumentar la eficiencia del Sistema de Reservación de Transporte, al proporcionar comodidad pues facilita que los estudiantes puedan programar la planificación de su pase como estimen conveniente. Por otro lado se tiene que la tecnología utilizada para el desarrollo de la herramienta es libre y no hay que incurrir en gastos en el pago de licencias de uso.

Por tanto, analizando el costo del proyecto, los numerosos beneficios que reporta, se puede concluir que su implementación es realmente factible.

5.4 Conclusiones.

En este capítulo se le ha realizado un estudio de factibilidad económica al sistema propuesto, que ha permitido arribar a la conclusión de que la herramienta reportará beneficios significativos e importantes para que los estudiantes, en la UCI, puedan programar la planificación de su reservación de pase. Lo que indica que es factible implementar la herramienta propuesta.

Conclusiones

Llegado este punto se espera que el documento haya servido para la comprensión teórica de la situación problemática existente y su solución, así como el desarrollo de las diferentes etapas de la aplicación usando la metodología RUP.

El desarrollo de este trabajo, ha permitido llegar a la conclusión de que se alcanzó, satisfactoriamente, el objetivo propuesto: implementar un módulo que permita la programación del pase por parte de los estudiantes de la UCI mediante el uso de herramientas de Software Libre que admitan la integración de dicho módulo con el “Sistema de Reservación de Transporte”. Además:

- ✓ Se definieron y fundamentaron los principales elementos teórico - metodológicos para el análisis del problema a resolver.
- ✓ Se desarrolló la documentación del Sistema de Reservación de Transporte.
- ✓ Se desarrolló el análisis y diseño de los procesos relacionados con el módulo de pases programados.
- ✓ Se implementó el módulo de pases programados con las características definidas en el análisis y diseño para este módulo.

Recomendaciones

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

- ◆ Poner a prueba el módulo implementado durante un período de tiempo significativo, para comprobar su desempeño y que sus funcionalidades se correspondan con la realidad de lo que se desea.
- ◆ Continuar el estudio del sistema en general con el objetivo de añadir nuevas funcionalidades.

Referencias bibliográficas

- [1]. 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 [Fecha de consulta 7 de Abril 2007].
- [2]. Peñalver, J. "Ventaja e inconvenientes de las aplicaciones Web". Disponible en <http://www.avidos.net/blogold/aplicaciones-web/> [Fecha de consulta 8 de Mayo 2007].
- [3]. Valle, J. "Definición de la arquitectura Cliente/Servidor". Disponible en <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml#algunos> [Fecha de consulta 02 de Abril 2007].
- [4]. Cuenca, C. "Arquitectura del servidor Apache". Disponible en <http://www.desarrolloweb.com/articulos/1112.php> [Fecha de consulta 22 de Enero 2007].
- [5]. Santana, P." Corta historia de PHP". Disponible en <http://www.pecesama.net/php/intro.php?PHPSESSID=4b24cb9c215ecb154469593e27b56707> [Fecha de acceso 17 de Febrero 2007].
- [6]. Castillo, C. "¿Cómo funciona PHP?". Disponible en <http://www.tejedoresdelweb.com/307/article-1067.html> [Fecha de consulta 27 de Marzo 2007].
- [7]. Van Ders, C. "¿Qué es el PHP?". Disponible en <http://www.maestrosdelweb.com/editorial/phpintro/> [Fecha de consulta 17 de Mayo 2007].
- [8]. GNUWin. "Apache Web Server". Disponible en <http://gnuwin.epfl.ch/apps/Apache/es/index.html> [Fecha de consulta 13 de Febrero 2007].

Referencias bibliográficas

- [9]. Garret, J. "AJAX un nuevo acercamiento a Aplicaciones Web". Disponible en <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php> [Fecha de consulta 02 de Abril 2006].
- [10]. Zulueta, R. "AJAX: ¿Moda pasajera o el verdadero futuro?". Disponible en <http://www.metodus.com/articulos/ajax/index.jsp> [Fecha de consulta 27 de Enero 2007].
- [11]. Melo, L." AJAX acelera el desarrollo de interfaces Web en las aplicaciones interactivas. Representa una amplia colección de las últimas tendencias en tecnologías de Internet". Disponible en <http://www.idg.es/computerworld/articulo.asp?id=179064> [Fecha de consulta 24 de Febrero 2007].
- [12]. Anónimo. "Patrones de Diseño. 2007. Disponible en http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o#Objetivos_de_los_patrones [Fecha de consulta 22 de Abril 2007].
- [13]. Anónimo. "Modelo Vista Controlador". Disponible en http://es.wikipedia.org/wiki/Modelo_Vista_Controlador [Fecha de consulta 15 de Mayo 2007].
- [14]. Raga, C. "Base de datos". Disponible en <http://www.monografias.com/trabajos7/bada/bada.shtml#dis> [Fecha de consulta 15 de Enero 2007].
- [15]. Anónimo. "Breve historia de PostgreSQL". Disponible en <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/x56.html> [Fecha de consulta 14 de Febrero 2007].
- [16]. Lotero, H. "Introducción y características principales de PostgreSQL". Disponible en http://72.14.205.104/search?q=cache:FeafU6PTLQJ:ganimides.ucm.cl/aurrutia/doc_ppt/pgsql.ppt+PostgreSQL+%2B+andrew+%2B+chen+%2B+interprete&hl=es&ct=clnk&cd=6&gl=cu [Fecha de consulta 18 de Abril 2006].
- [17]. Jacobson, I y Booch, G y Rumbaugh, J. "El Lenguaje Unificado de Modelado". España. Prentice Hall. 2002. 570.

Referencias bibliográficas

- [18]. Alvarez, M. "Dreamweaver". Disponible en <http://www.desarrolloweb.com/articulos/332.php> [Fecha de consulta 15 de Abril 2007].
- [19]. Alvarez, M. "Evaluando Zend Studio". Disponible en <http://www.maestrosdelweb.com/editorial/zendstudio/> [Fecha de consulta 17 de Mayo 2007].
- [20]. Anónimo. "Rational Rose Enterprise". Disponible en http://www-306.ibm.com/software/info/ecatalog/es_ES/products/M221280M46834Z27.html?&S_TACT=none&S_CMP=none [Fecha de consulta 18 de Abril 2007].
- [21]. López, A. "Herramienta PgAdmin". Disponible en <http://www.ulfix.net/index.php?option=content&task=view&id=407> [Fecha de consulta 16 de Abril 2007].
- [22]. 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.
- [23]. Universidad de Sevilla. "Modelado del Negocio (Diagrama de actividades)". Disponible en <http://www.lsi.us.es/docencia/get.php?id=1877> [Fecha de consulta 17 de Enero 2007].
- [24]. Vilalta, J. "UML. Guía visual". 2001. Disponible en <ftp://10.0.0.22/Documentation/UML/UML-Guia-Visual.pdf> [Fecha de consulta 17 de Abril 2007].
- [25]. Anónimo. "Trabajador del Negocio". Disponible en http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=128&Itemid=294 [Fecha de consulta 23 de Enero 2007].
- [26]. Fernández, L. "Gestión de los Requisitos de un Proyecto Software". Disponible en <http://www.monografias.com/trabajos41/requisitos-software/requisitos-software.shtml> [Fecha de consulta 10 de Mayo 2007].
- [27]. Fernando, M. "AMIR-ST: propuesta de una aproximación metodológica para la ingeniería de Requisitos de Sistemas Telemáticos". Disponible en

Referencias bibliográficas

http://72.14.205.104/search?q=cache:SmhVsqv8_O8J:www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r5_2_art5_c.pdf+captura+de+requisitos+funcional&hl=es&ct=clnk&cd=7&gl=cu [Fecha de consulta 25 de

Enero 2007].

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

[29]. Anónimo. “Modelado y Diseño orientado a Objetos”. Disponible en

<http://html.rincondelvago.com/modelado-y-diseno-orientado-a-objetos.html> [Fecha de consulta 10 de

Enero 2007].

[30]. Badani, S. “Métricas de estimación de tamaño. Puntos de casos de uso”. 2002. Disponible en

http://72.14.209.104/search?q=cache:nfXRYhessoJ:ieic.ucentral.cl/ftp/material/apuntes/ieic61/Diseno/Metodo_PCU.doc+puntos+de+casos+de+uso&hl=es&ct=clnk&cd=5&gl=cu

Bibliografía

AJAX un nuevo acercamiento a aplicaciones Web, abril 28 del 2007. Disponible en: <<http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>> [Fecha de consulta 31 abril 2007].

Clases de Ingeniería del Software I, curso 2006-2007, UCI.

Hernández, Rolando A. y Coello, Sayda. *El Paradigma Cuantitativo de la Investigación Científica*. Noviembre 2002, UCI.

Introducción a php. Disponible en: <www.ciberteca.net/webmaster/php> [Fecha de consulta 24 febrero 2007].

Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Primera Edición por Prentice Hall, Hispanoamericana S.A. 1999.

Larman, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda Edición por Prentice Hall.

Matos, Rosa María. *Introducción al trabajo con base de datos*. Asignatura de Sistemas de Gestión de base de datos.

PostgreSQL 8.1.x. Disponible en: <<http://www.postgresql.cl/>> [Fecha de consulta 26 febrero 2007].

Peralta, Mario. *Estimación del esfuerzo basada en casos de uso*. Centro de Ingeniería del Software e Ingeniería del Conocimiento, Buenos Aires, Argentina.

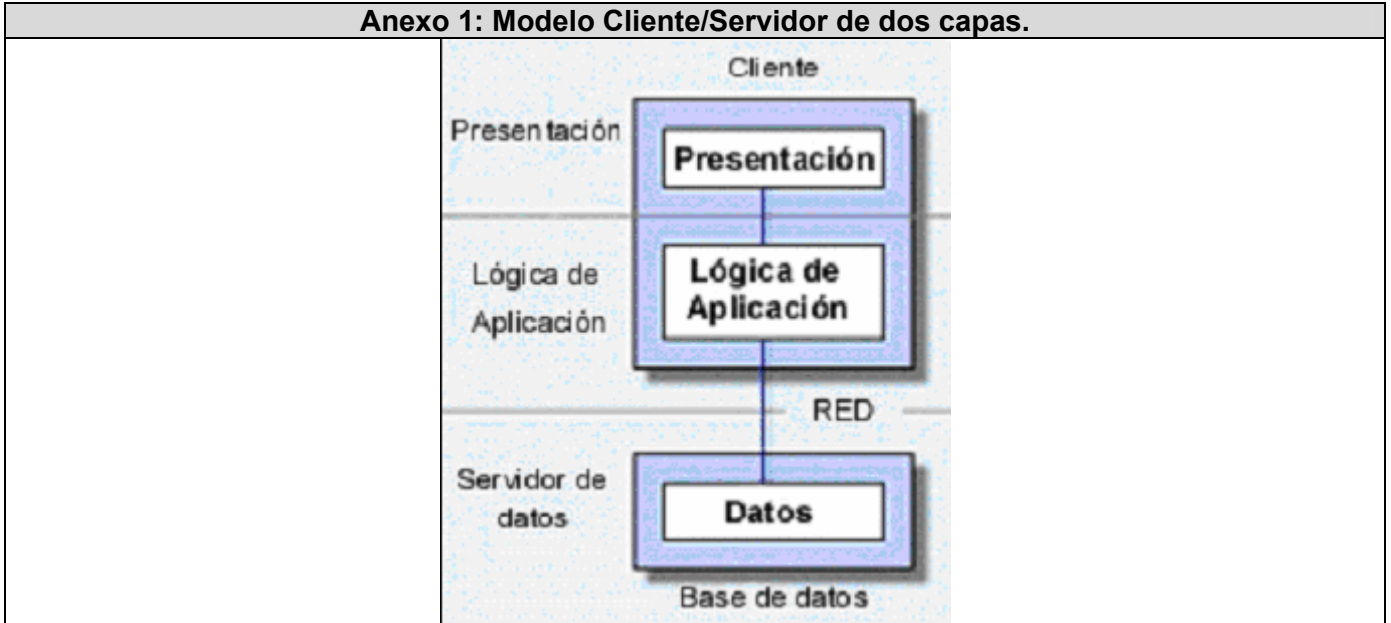
Quatrani, Terry. *Visual Modeling with Rational Rose 2000 and UML*, Publisher Addison Wesley, Second Edition October 19, 1999.

Tutorial de PostgreSQL. Disponible en:

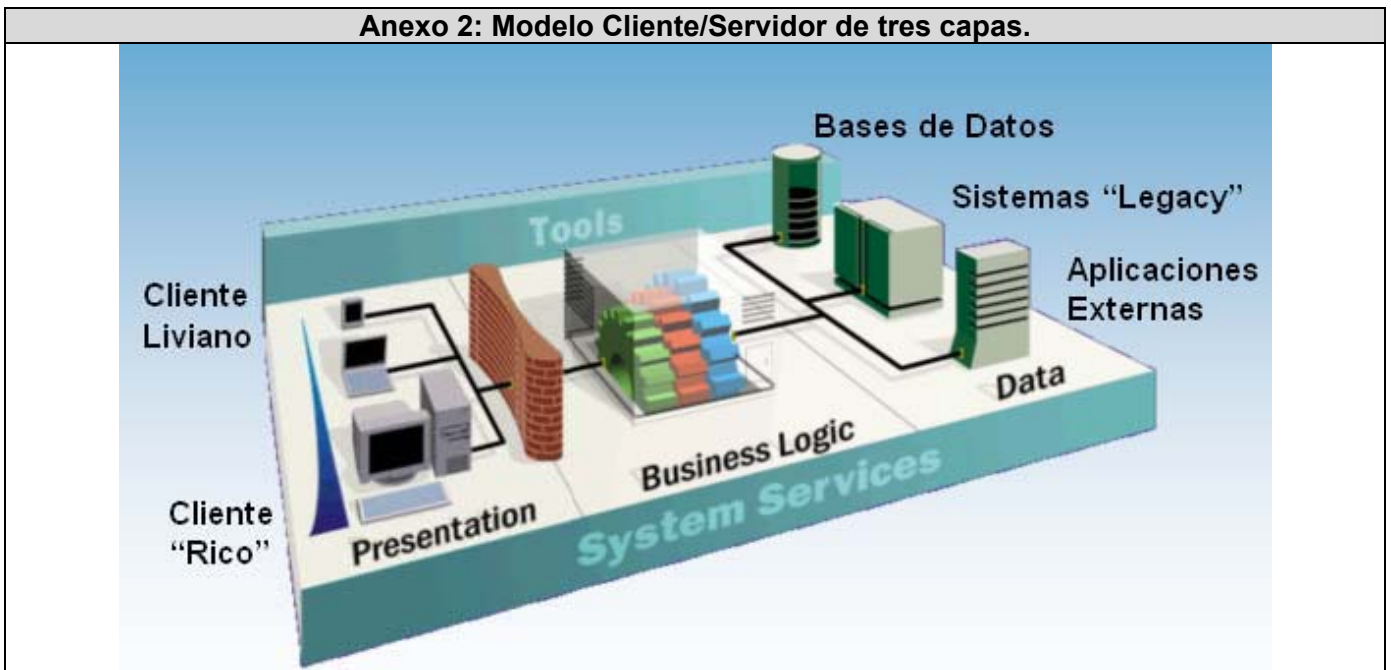
<<http://es.tldp.org/Postgresqls/web/navegable/tutorial/tutorial.html>> [Fecha de consulta 26 febrero 2007].

Anexos

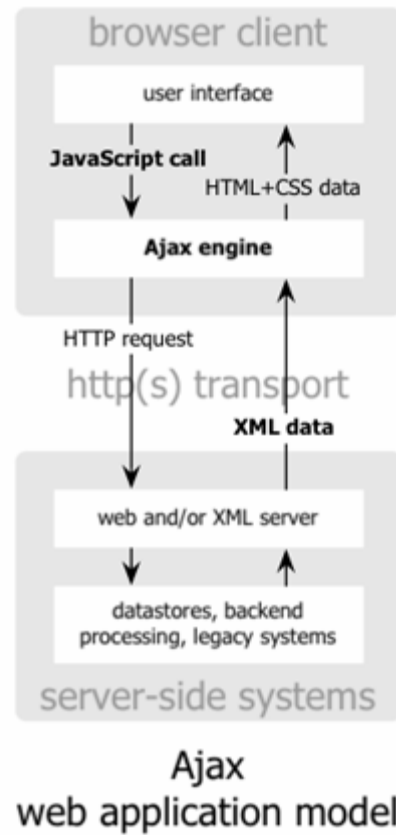
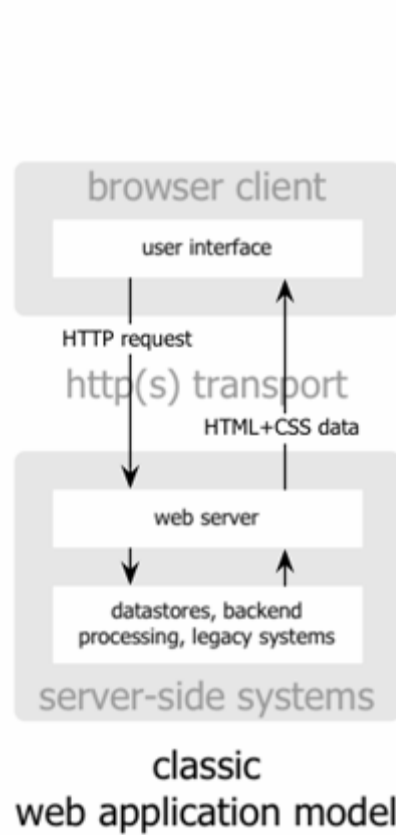
Anexo 1: Modelo Cliente/Servidor de dos capas.



Anexo 2: Modelo Cliente/Servidor de tres capas.

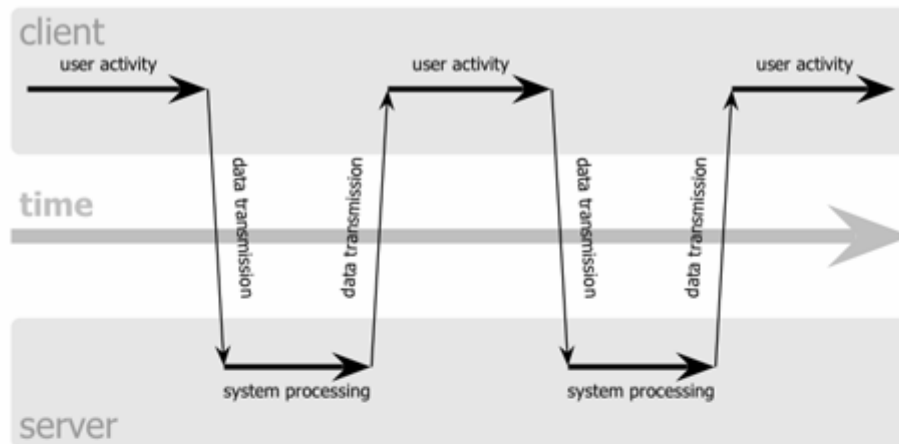


Anexo 3: Modelo tradicional para aplicaciones Web (a la izquierda), comparando con el modelo AJAX (a la derecha).

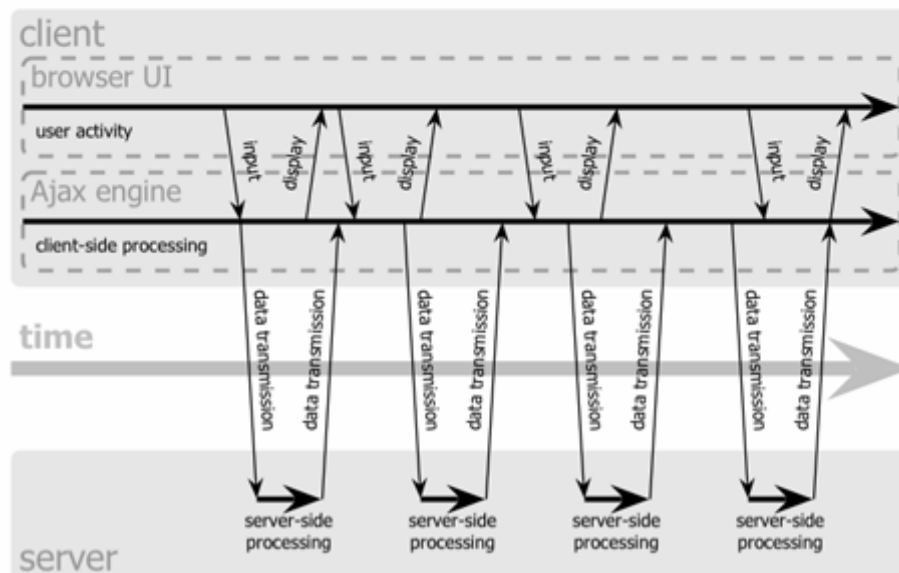


Anexo 4: Muestra el patrón de interacción sincrónica de una aplicación Web tradicional (arriba) comparada con el patrón asincrónico de una aplicación AJAX (debajo).

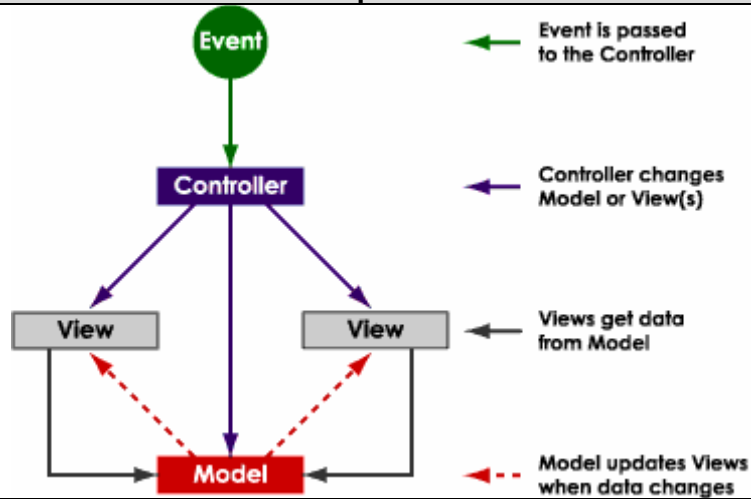
classic web application model (synchronous)



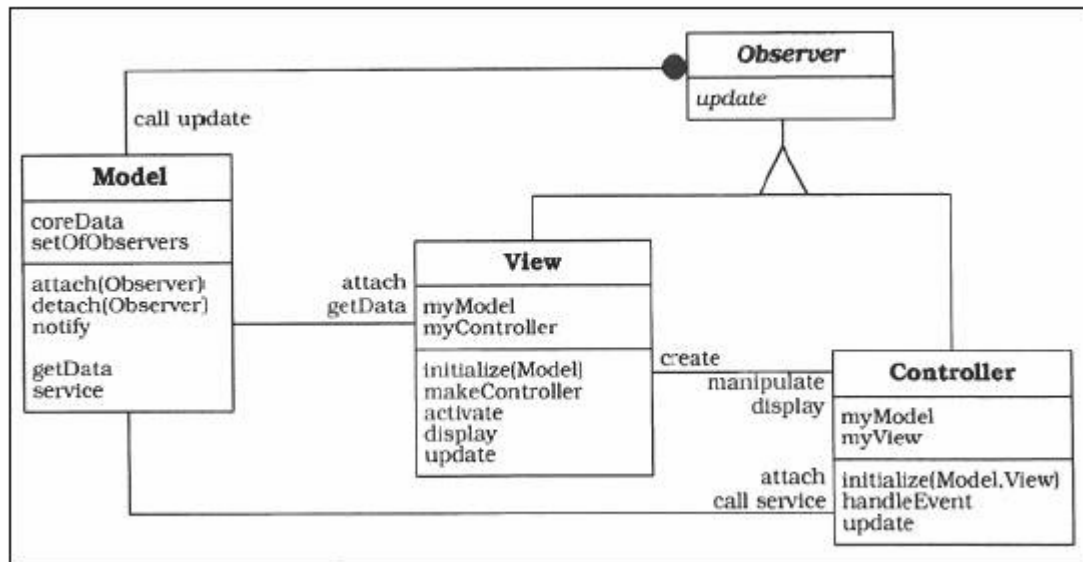
Ajax web application model (asynchronous)



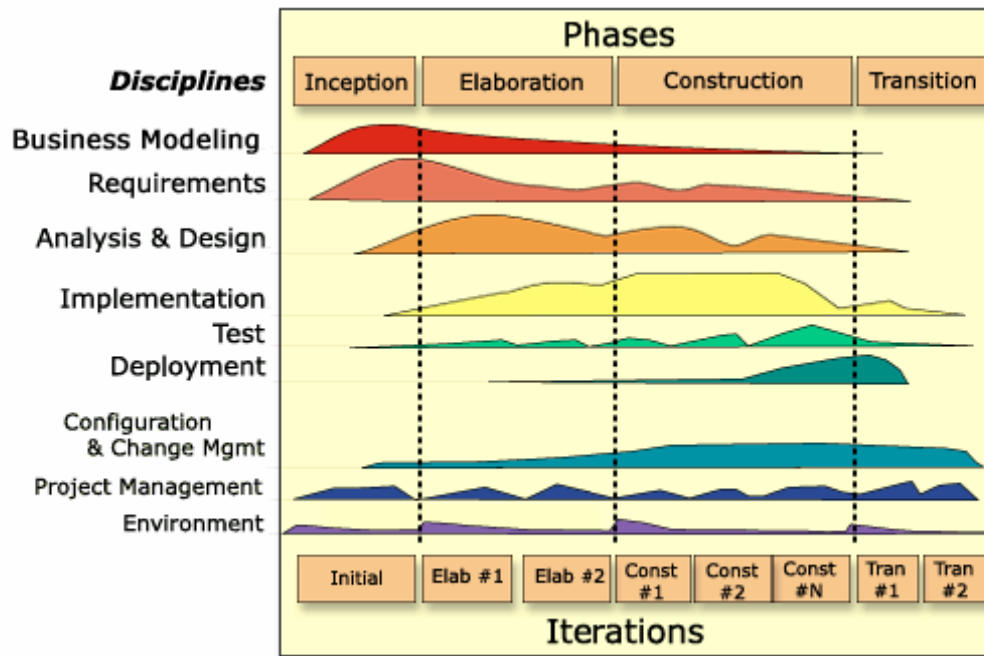
Anexo 5: Funcionamiento del patrón Modelo-Vista-Controlador.




Anexo 6: Estructura del Modelo-Vista-Controlador.




Anexo 7: Flujos de Trabajo de RUP.



Anexo 8: Prototipo Gestionar Pase.




Reservación de transporte
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Reservar
Mis Reservaciones

Reservar salida/entrada de pase



Nombre: Iris Alemán Betancourt

Usuario: ialeman

No. Solapín: 51136

IP: 10.31.18.19 - pase-prueba.uci.cu

Reserva de ida

No reservar

Reservar

Municipio: [seleccione]

Ruta: [-]

Punto: [-]

No periódico

Periódico

Frecuencia

Semanal Repetir cada semanas

Intervalo de Repetición

Comienzo:

Finalizar después de: repeticiones

Finalizar el:

Reserva de regreso

No reservar

Reservar

Municipio: [seleccione]

Ruta: [-]

Punto: [-]

Fecha:

No periódico

Periódico

Datos de la persona que va a visitar

Familiar: Magda **Parentesco:** tía

Calle 27 **No.** 1042 **Apto**

Entre 8 y 10 **Localidad** Plaza **Teléfono:**

Familiar:

Calle:

Entre:

Teléfono:

Parentesco:

No: **Apto:**

Localidad:

Reservar
Limpiar
Salir

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 9: Prototipo Autenticar.



Reservación de transporte

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Bienvenidos al sistema de reservación de transporte

Usuario:

Clave:

Las reservaciones de transporte para la salida del fin de semana se pueden realizar a partir del **Lunes** a la **01:00am** hasta el **Viernes** a las **8:30am**.

A partir del **Viernes** a las **8:30am** y hasta el **Sábado** a las **9:00am** sólo será posible cancelar una reservación previamente realizada.

Una vez reservada una solicitud de transportación y pasado el horario de posible cancelación, es **obligatorio** utilizar el transporte reservado, en caso contrario se **retirará el derecho a reservar transporte** por un período de tiempo especificado por el decano, o cualquier otra medida que se estime conveniente.

Horarios

Salida: Sábado a las 6:00 PM
Regreso: Domingo a las 09:00 PM

[Listado de rutas](#)
[MAPA: Puntos de salida UCI](#)

¿Problemas para entrar al sistema ó reservar?

Correo/Usuario:

Problema:

Contáctenos | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 10: Prototipo Enviar Inquietud.



[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Bienvenidos al sistema de reservación de transporte

Usuario:
 Clave:

Las reservaciones de transporte para la salida del fin de semana se pueden realizar a partir del **Lunes** a la **01:00am** hasta el **Viernes** a las **8:30am**.

A partir del **Viernes** a las **8:30am** y hasta el **Sábado** a las **9:00am** sólo será posible cancelar una reservación previamente realizada.

Una vez reservada una solicitud de transportación y pasado el horario de posible cancelación, es **obligatorio** utilizar el transporte reservado, en caso contrario se **retirará el derecho a reservar transporte** por un período de tiempo especificado por el decano, o cualquier otra medida que se estime conveniente.

Horarios

Salida: Sábado a las 6:00 PM
 Regreso: Domingo a las 09:00 PM

[Listado de rutas](#)
[MAPA: Puntos de salida UCI](#)

¿Problemas para entrar al sistema ó reservar?

Correo/Usuario:
 Problema:

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 11: Prototipo Ver rutas.



[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

RUTAS E ITINERARIOS DE LOS OMNIBUS

No. Ruta	Punto de Salida UCI	Pto Destino/Regreso	Municipio	Única parada Intermedia	Itinerario
1	Areas Deportivas	9na y E (Vedado)	Plaza de la Revolucion		
2	Areas Deportivas	Entronque de Alberro	Cotorro	100 y Boyeros	
3	Areas Deportivas	Entronque de Alberro	Cotorro	100 y Calzada de Bejucal (Puente Calabazar)	
4	Areas Deportivas	Feria Agropecuaria	Boyeros		
5	Areas Deportivas	Lonja del Comercio	Habana Vieja		

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 12: Prototipo Ver puntos de salida.

UCI **Reservación de transporte**
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Mapa de la UCI Seleccione Ruta de Ida para ver Punto Salida UCI:
[--Seleccione--]

Contáctenos | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 13: Prototipo Buscar estudiantes.



Reservación de transporte

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

- Inicio**
 - Inicio
- Boletines**
 - Impresión de Boletines
- Reportes**
 - Por Rutas
 - Por Años
 - Por Facultades
- Búsquedas**
 - Buscar estudiante(s)
- Salir

Búsqueda avanzada de estudiantes

Grupos:

Tipo de Pase :

Todo:

Solapín:

Nombre:

Primer Apellido:

Segundo Apellido:

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 14: Prototipo Ver reportes.



[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

Inicio

[Inicio](#)

Boletines

[Impresión de Boletines](#)

Reportes

[Por Rutas](#)

[Por Años](#)

[Por Facultades](#)

Búsquedas

[Buscar estudiante\(s\)](#)

[Salir](#)

Reporte por Facultades

	Reservaciones IDA	Reservaciones REGRESO	Sólo IDA	Sólo REGRESO
Facultad 1	0	0	0	0
Facultad 2	0	0	0	0
Facultad 3	0	0	0	0
Facultad 4	0	0	0	0
Facultad 5	0	0	0	0
Facultad 6	0	0	0	0
Facultad 7	0	0	0	0
Facultad 8	0	0	0	0
Facultad 9	0	0	0	0
Facultad 10	0	0	0	0
Totales	0	0	0	0

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 15: Prototipo Imprimir Boletines.



Reservación de transporte

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

- Inicio**
 - Inicio
- Boletines**
 - Impresión de Boletines
- Reportes**
 - Por Rutas
 - Por Años
 - Por Facultades
- Búsquedas**
 - Buscar estudiante(s)
- Salir

Asistente de Impresión de Boletines

Grupo:

Nombre(s):

Apellido(s):

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 16: Prototipo Modificar horarios.



[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

Inicio

[Inicio](#)

Boletines

[Impresión de Boletines](#)

Búsquedas

[Buscar estudiante\(s\)](#)

Reportes

[Por Rutas](#)

[Por Años](#)

[Por Facultades](#)

Usuarios

[Agregar usuarios](#)

[Listar usuarios](#)

Rutas

[Listado de rutas](#)

[Agregar ruta](#)

Municipios

[Agregar municipio](#)

[Modificar municipio](#)

Horario

[Modificar horarios](#)

Puntos de salida

[Insert punto de salida](#)

[Editar punto de salida](#)

[Salir](#)

Listado de Horarios

No.	Horarios	Descripción	Editar Horarios
7	2006-10-28 17:30:00	Fin de Impresion Boletines	
6	2006-10-27 09:00:00	Comienzo de Impresion Boletines	
5	2006-10-23 00:00:01	Inicio de reservacion	
4	2006-10-28 08:00:00	Fin de cancelacion	
3	2006-10-29 21:00:00	Entrada	
2	2006-10-28 18:00:00	Salida	
1	2006-10-27 08:30:00	Fin de reservacion	

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 17: Prototipo Gestionar usuario.



Reservación de transporte

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

Inicio
Inicio

Boletines
Impresión de Boletines

Búsquedas
Buscar estudiante(s)

Reportes
Por Rutas
Por Años
Por Facultades

Usuarios
Agregar usuarios
Listar usuarios

Rutas
Listado de rutas
Agregar ruta


Municipios
Agregar municipio
Modificar municipio

Horario
Modificar horarios

Puntos de salida
Insert punto de salida
Editar punto de salida

Salir


Listado de usuarios

	<u>Nombres</u>	<u>Apellidos</u>	<u>Area</u>	<u>Privilegio</u>	
	TOMAS	LOPEZ JIMENEZ	A0000	Solo Ver	✘
	MANUEL ALEJANDRO	GIL MARTIN	D30000	Solo Ver	✘
	IRKA	ROSELLO LANDA	J1000	Solo Ver	✘

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 18: Prototipo Gestionar puntos de salida.



Reservación de transporte

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

- Inicio**
 - Inicio
- Boletines**
 - Impresión de Boletines
- Búsquedas**
 - Buscar estudiante(s)
- Reportes**
 - Por Rutas
 - Por Años
 - Por Facultades
- Usuarios**
 - Agregar usuarios
 - Listar usuarios
- Rutas**
 - Listado de rutas
 - Agregar ruta
- Municipios**
 - Agregar municipio
 - Modificar municipio
- Horario**
 - Modificar horarios
- Puntos de salida**
 - Insert punto de salida
 - Editar punto de salida
- Salir

Punto:

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 19: Prototipo Gestionar municipios.



Reservación de transporte

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

- Inicio**
 - Inicio
- Boletines**
 - Impresión de Boletines
- Búsquedas**
 - Buscar estudiante(s)
- Reportes**
 - Por Rutas
 - Por Años
 - Por Facultades
- Usuarios**
 - Agregar usuarios
 - Listar usuarios
- Rutas**
 - Listado de rutas
 - Agregar ruta
- Municipios**
 - Agregar municipio
 - Modificar municipio
- Horario**
 - Modificar horarios
- Puntos de salida**
 - Insert punto de salida
 - Editar punto de salida
- Salir

Agregar Municipio

Municipio:

Agregar

[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

Anexo 20: Prototipo Gestionar rutas.



Reservación de transporte

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

[Inicio](#) | [Intranet](#) | [Foro de la Comunidad](#) | [Inter-Nos](#) | [Guía telefónica](#) | [WikiProd](#) | [FEU](#)

Menú de opciones

- Inicio**
 - Inicio
- Boletines**
 - Impresión de Boletines
- Búsquedas**
 - Buscar estudiante(s)
- Reportes**
 - Por Rutas
 - Por Años
 - Por Facultades
- Usuarios**
 - Agregar usuarios
 - Listar usuarios
- Rutas**
 - Listado de rutas
 - Agregar ruta
- Municipios**
 - Agregar municipio
 - Modificar municipio
- Horario**
 - Modificar horarios
- Puntos de salida**
 - Insert punto de salida
 - Editar punto de salida
- Salir

Agregar ruta

No. Ruta:

Ruta:

Municipio: Habana del Este

Parada intermedia:

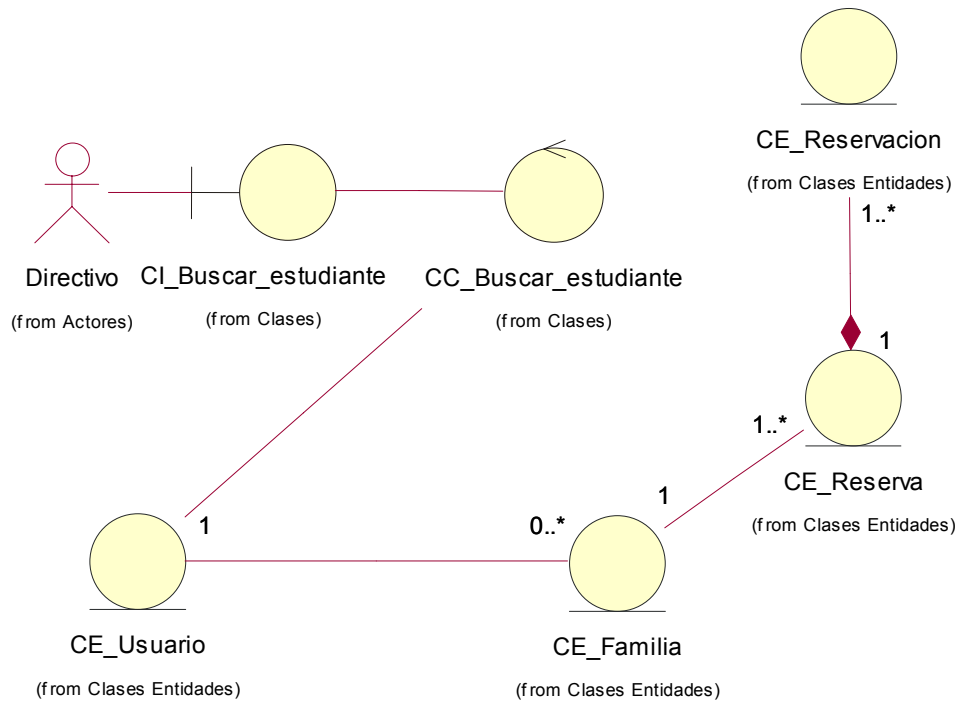
Itinerario:

Punto Salida UCI: Areas Deportivas()

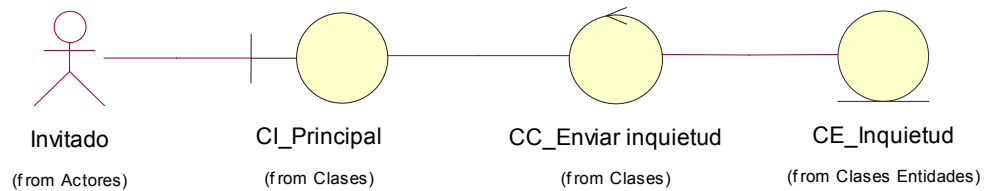
[Contáctenos](#) | [Intranet UCI](#) | [Créditos](#)

Copyright © 2006 por Universidad de las Ciencias Informáticas. Todos los derechos reservados.

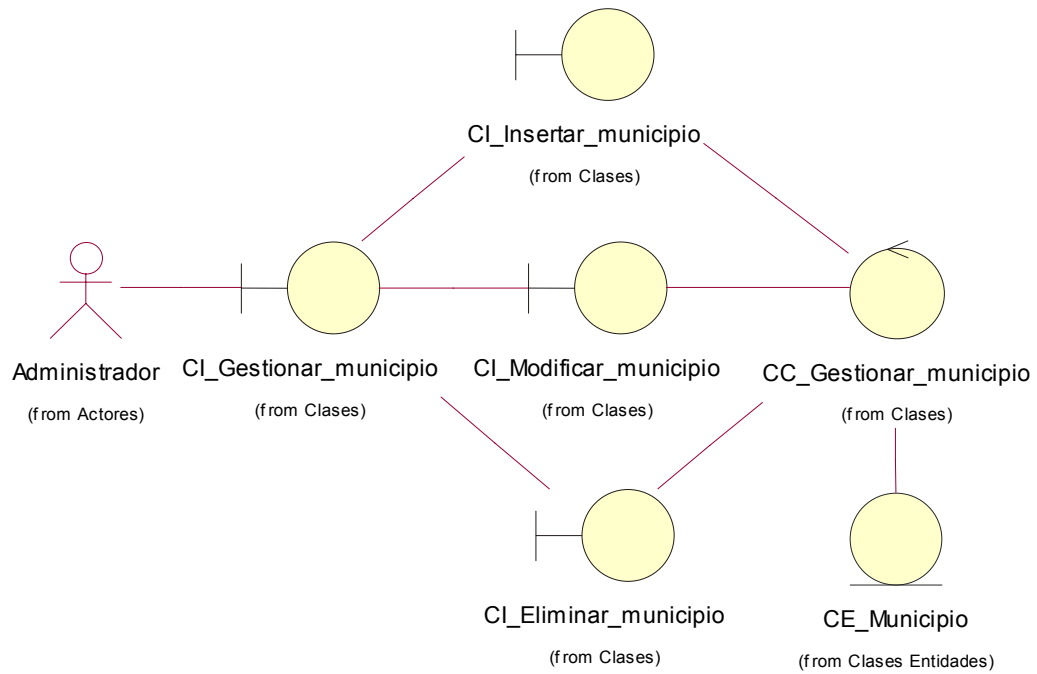
Anexo 21: Diagrama de clases del análisis "Buscar Estudiante".



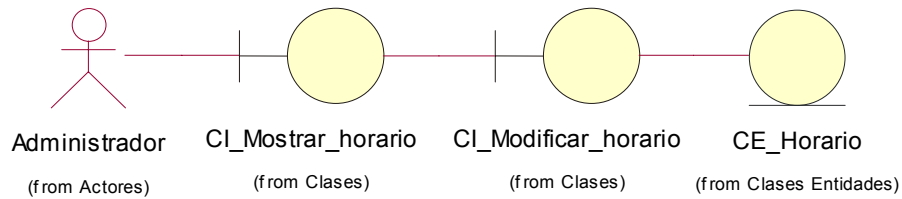
Anexo 22: Diagrama de clases del análisis "Enviar Inquietud".



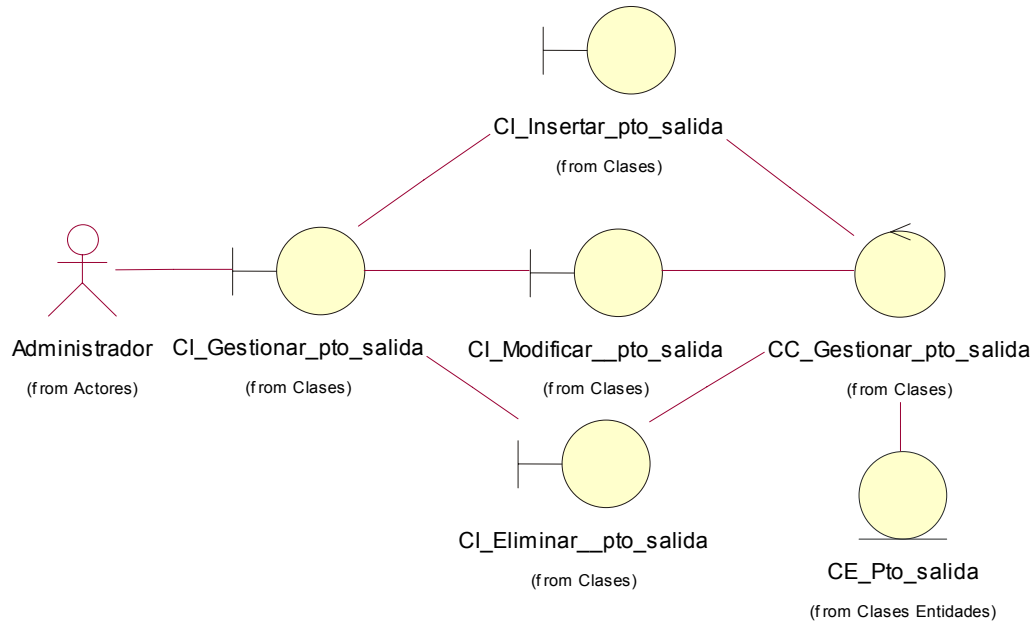
Anexo 23: Diagrama de clases del análisis "Gestionar municipios".



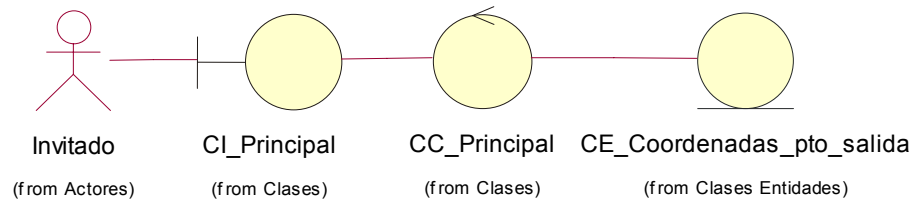
Anexo 24: Diagrama de clases del análisis "Modificar Horarios".



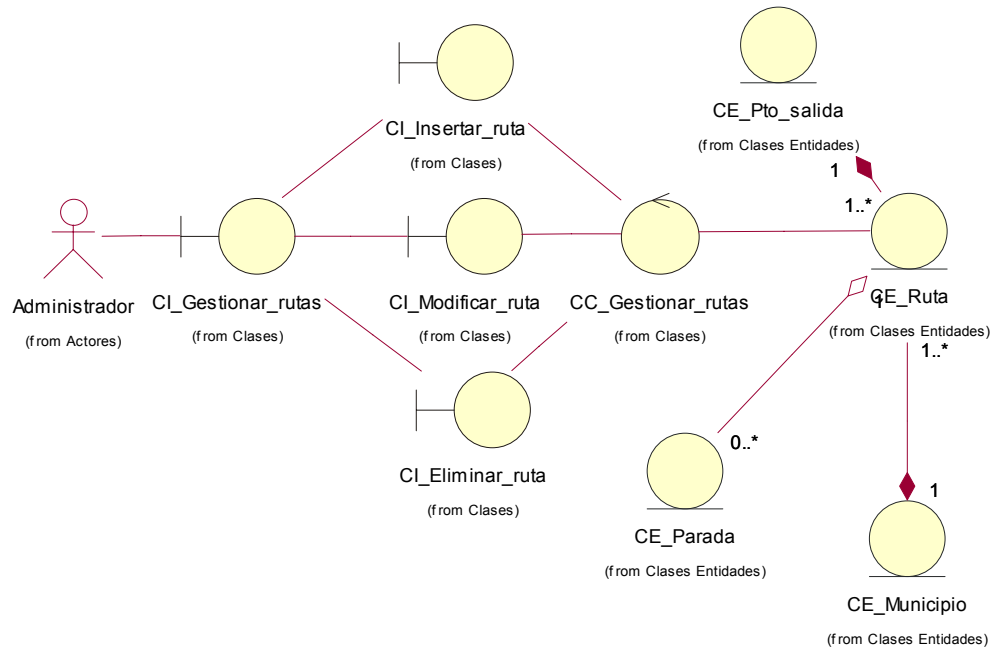
Anexo 25: Diagrama de clases del análisis "Gestionar puntos de salida".



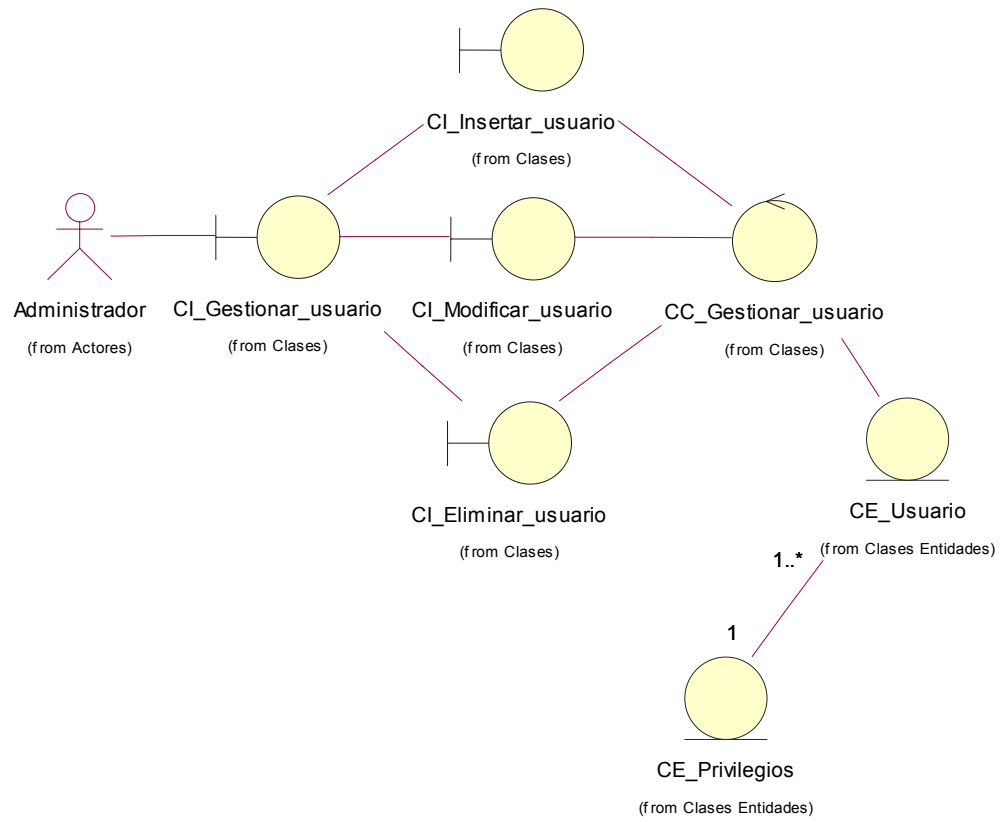
Anexo 26: Diagrama de clases del análisis "Ver puntos de salida".



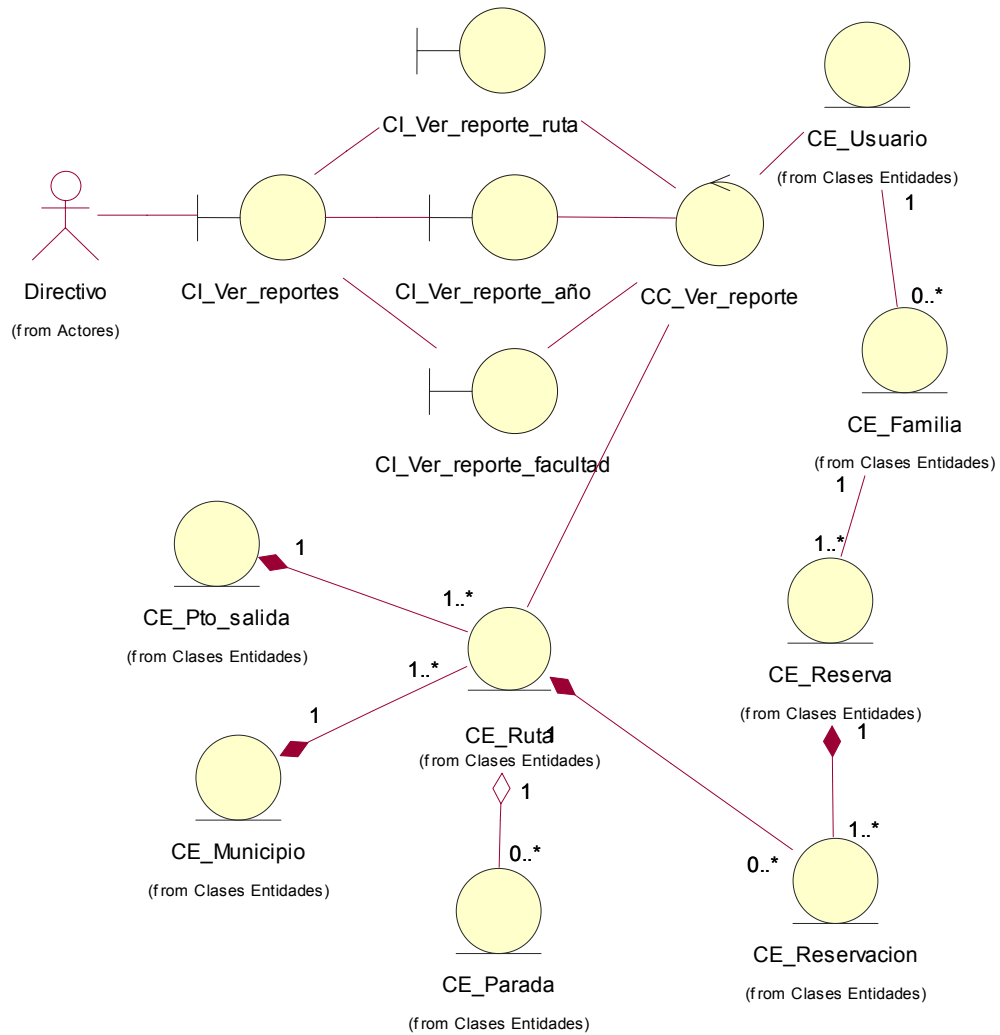
Anexo 27: Diagrama de clases del análisis "Gestionar rutas".



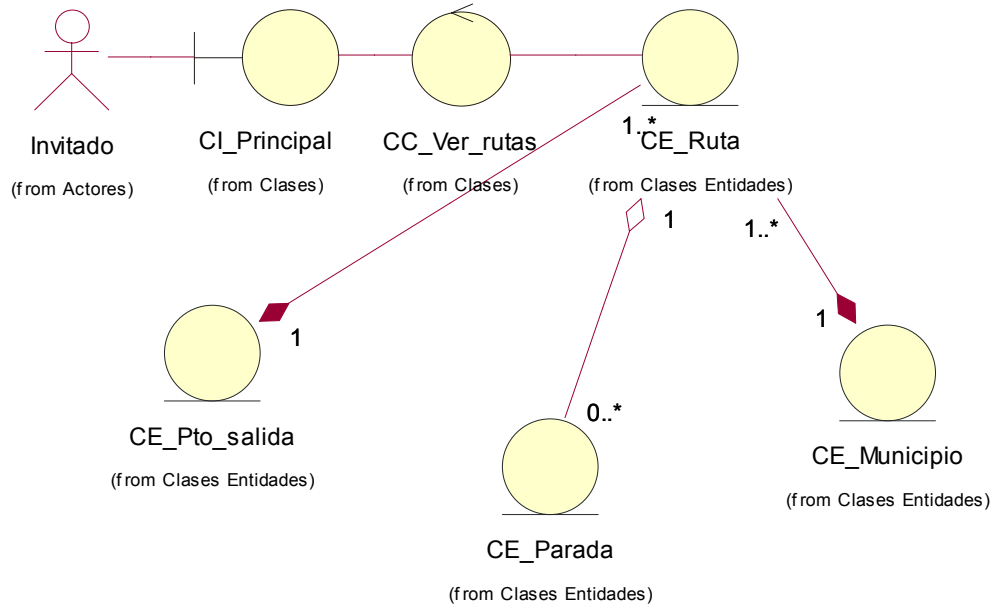
Anexo 28: Diagrama de clases del análisis "Gestionar Usuario".



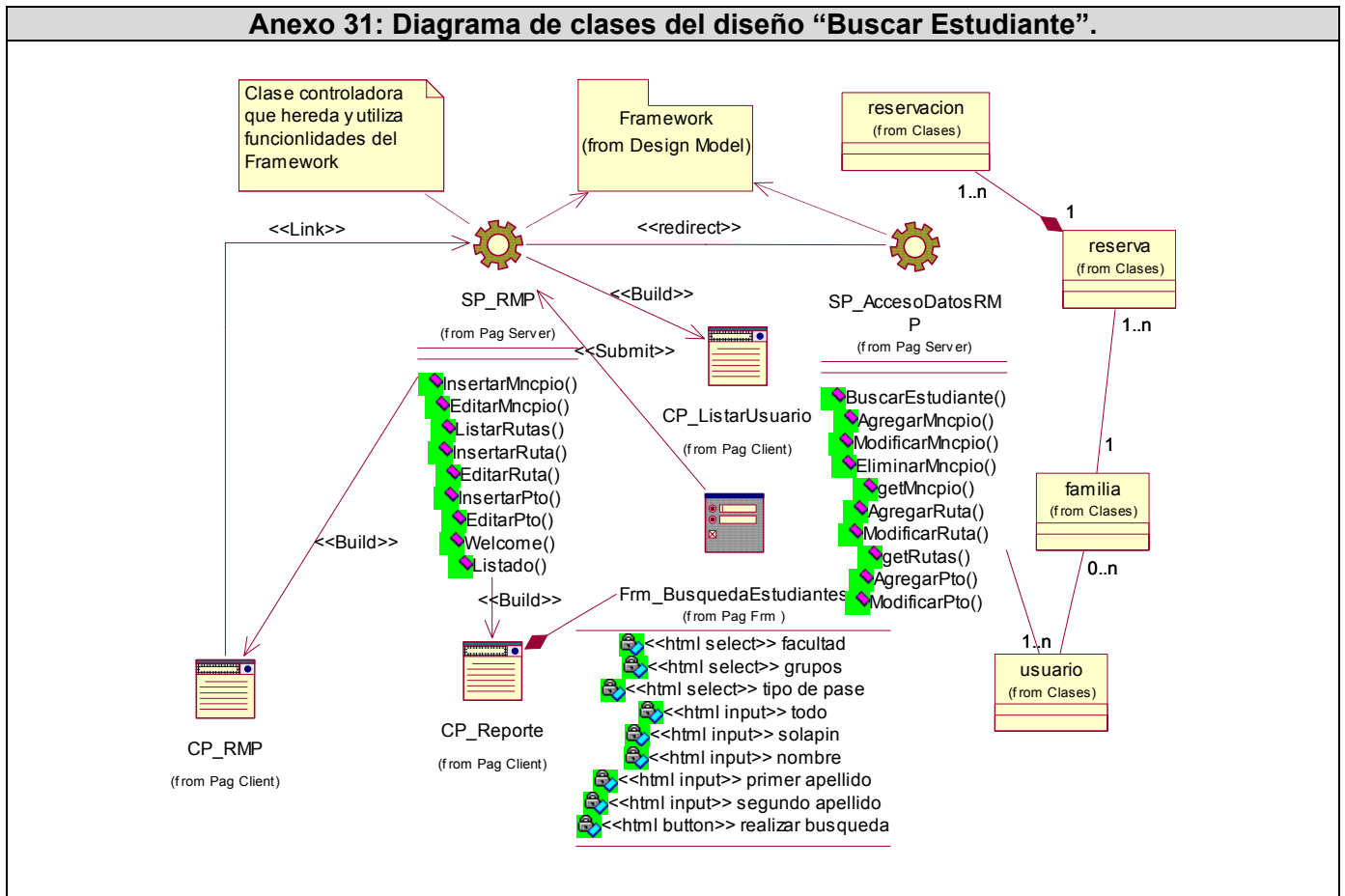
Anexo 29: Diagrama de clases del análisis “Ver reportes”.



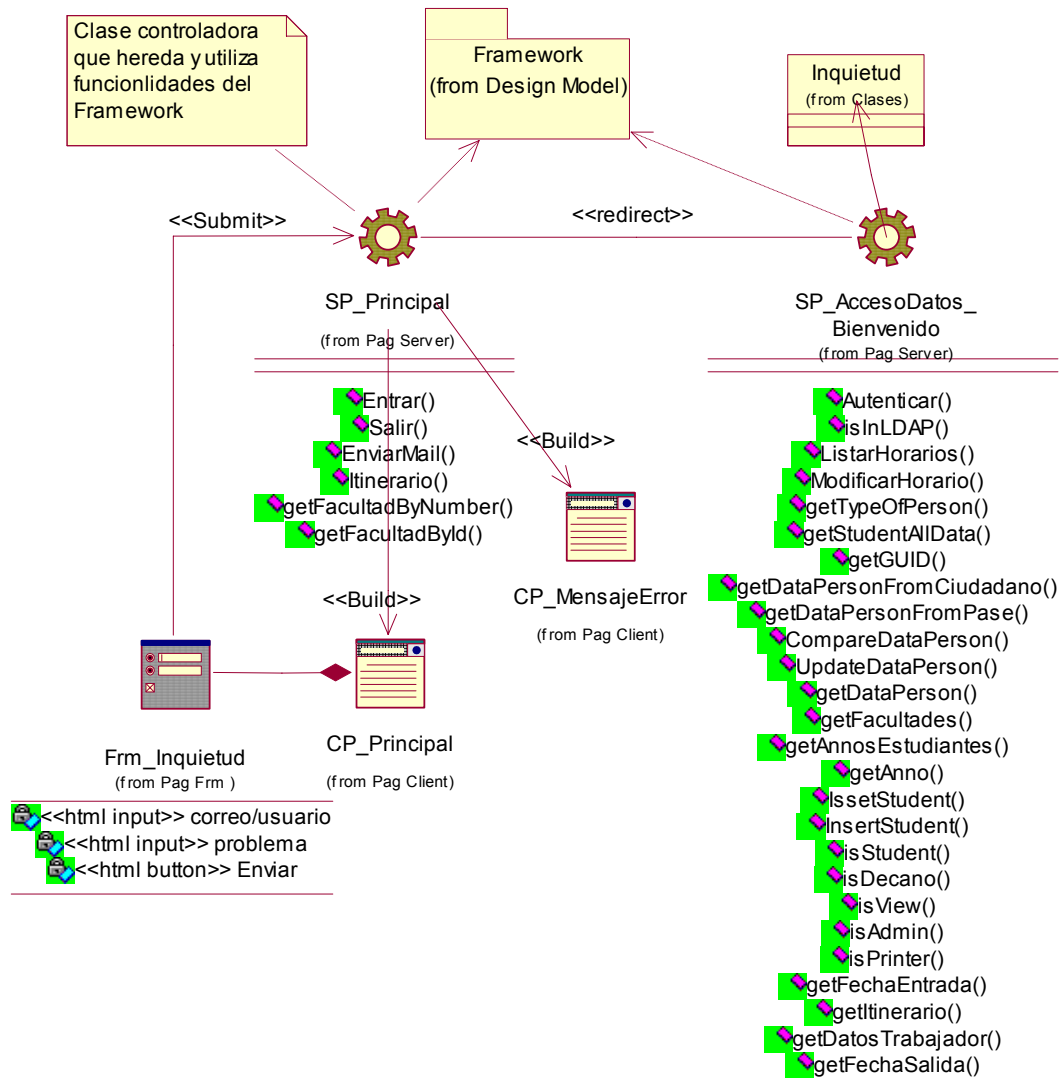
Anexo 30: Diagrama de clases del análisis "Ver rutas".



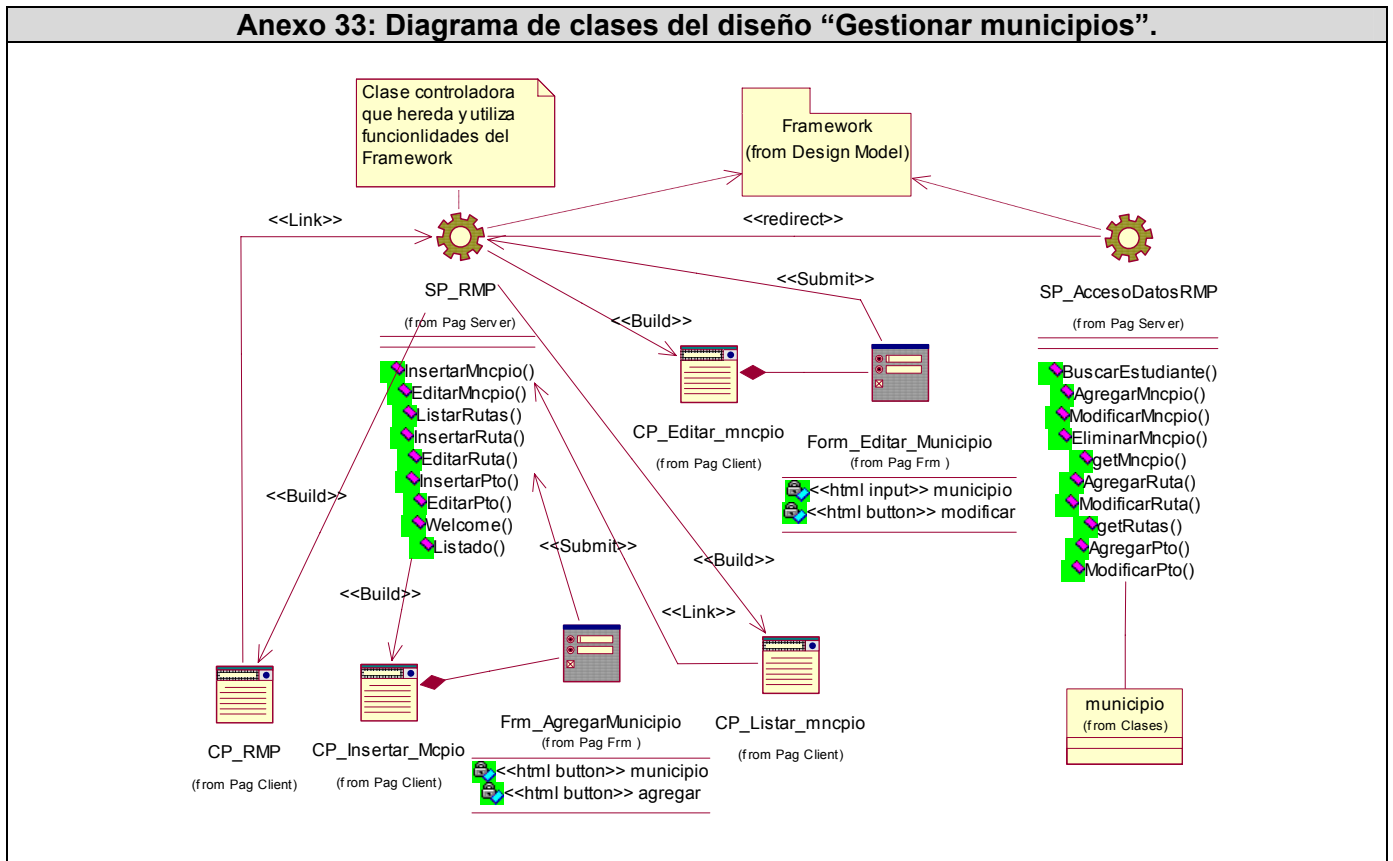
Anexo 31: Diagrama de clases del diseño "Buscar Estudiante".



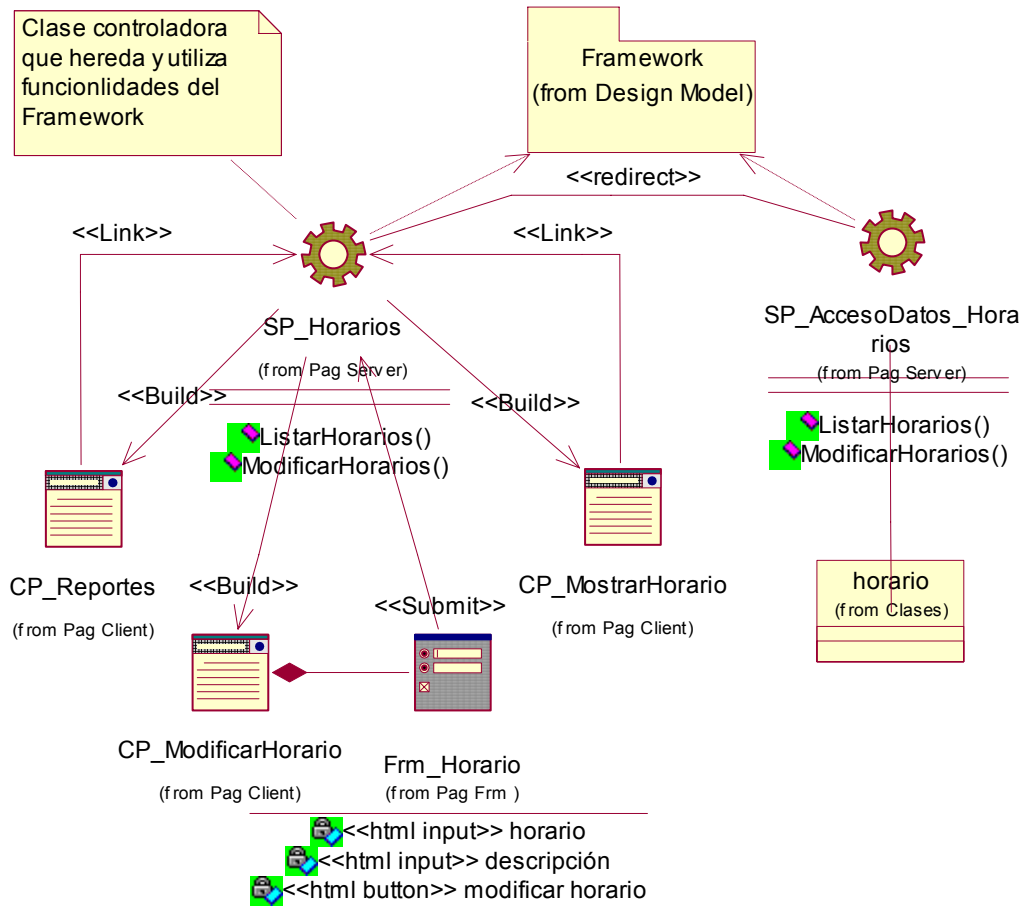
Anexo 32: Diagrama de clases del diseño "Enviar Inquietud".



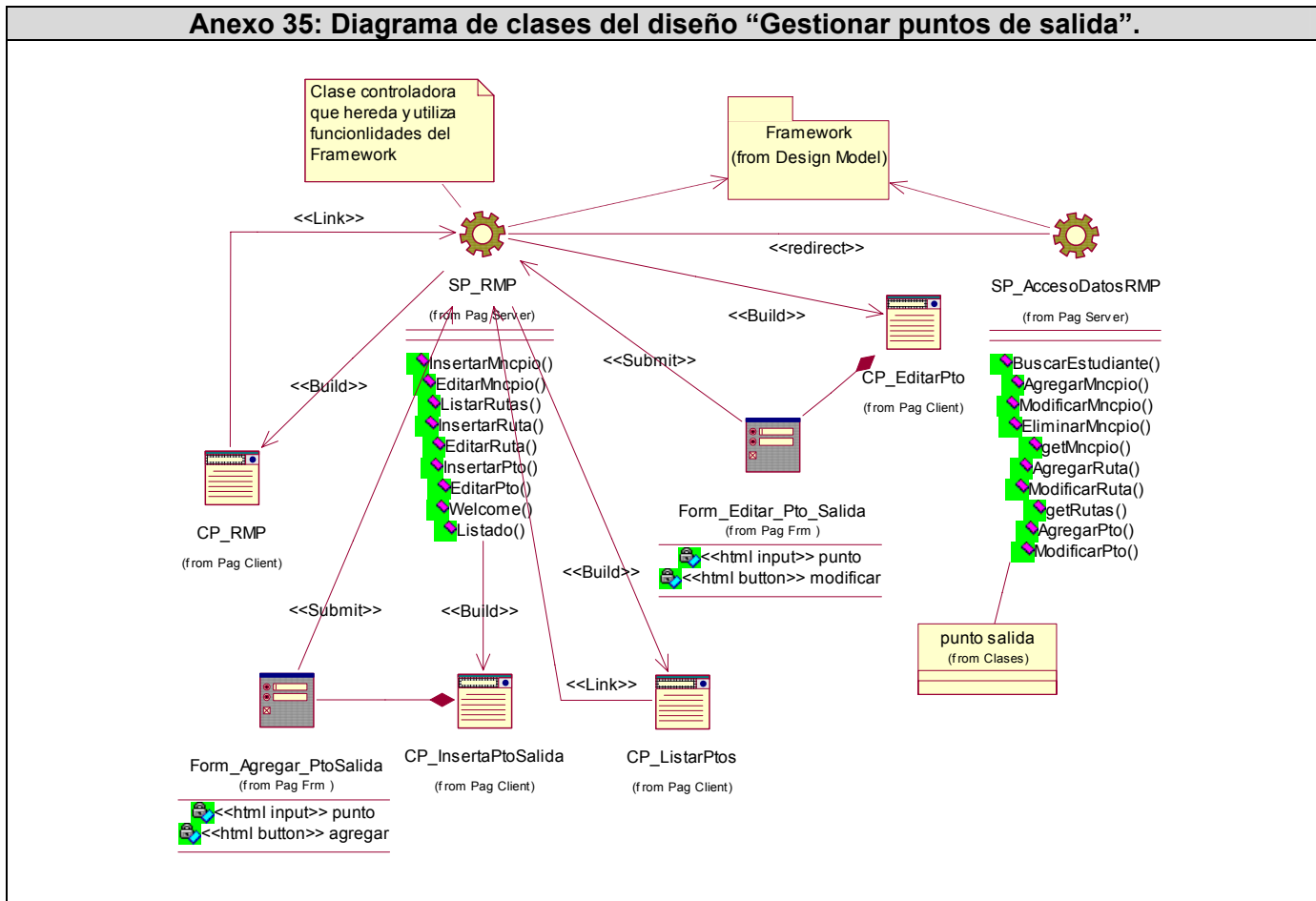
Anexo 33: Diagrama de clases del diseño "Gestionar municipios".



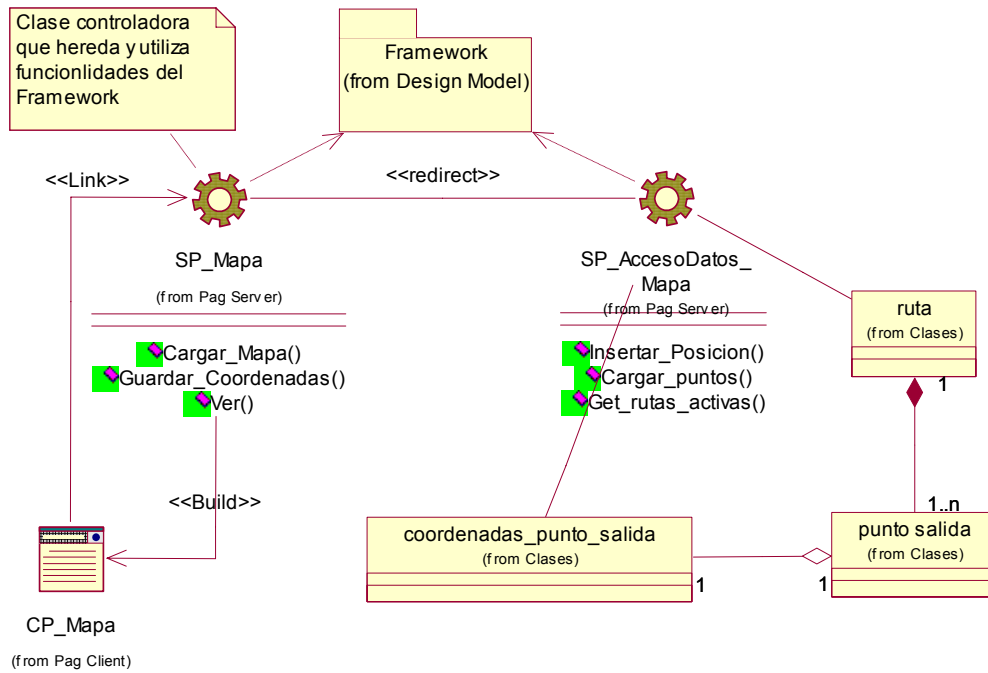
Anexo 34: Diagrama de clases del diseño "Modificar Horarios".



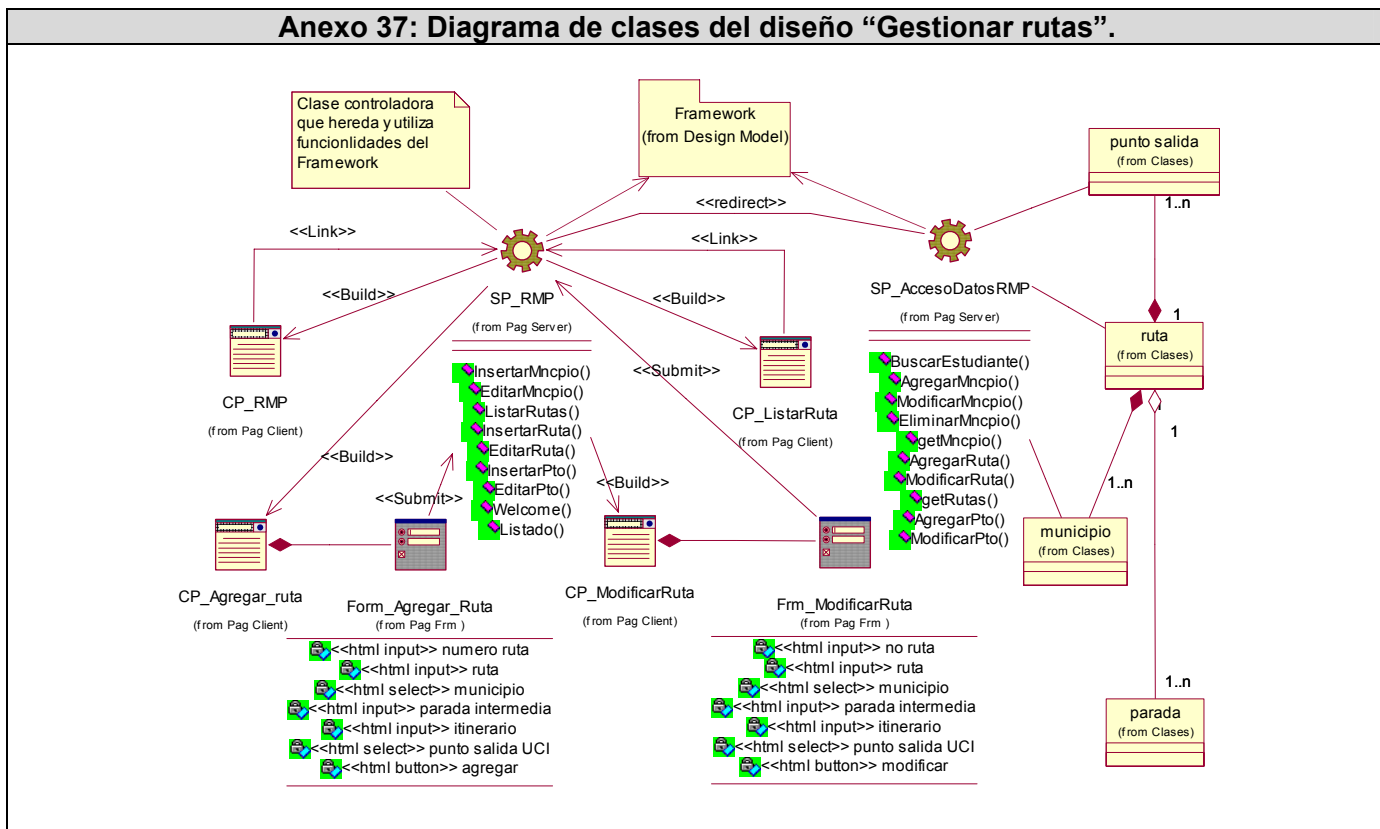
Anexo 35: Diagrama de clases del diseño “Gestionar puntos de salida”.



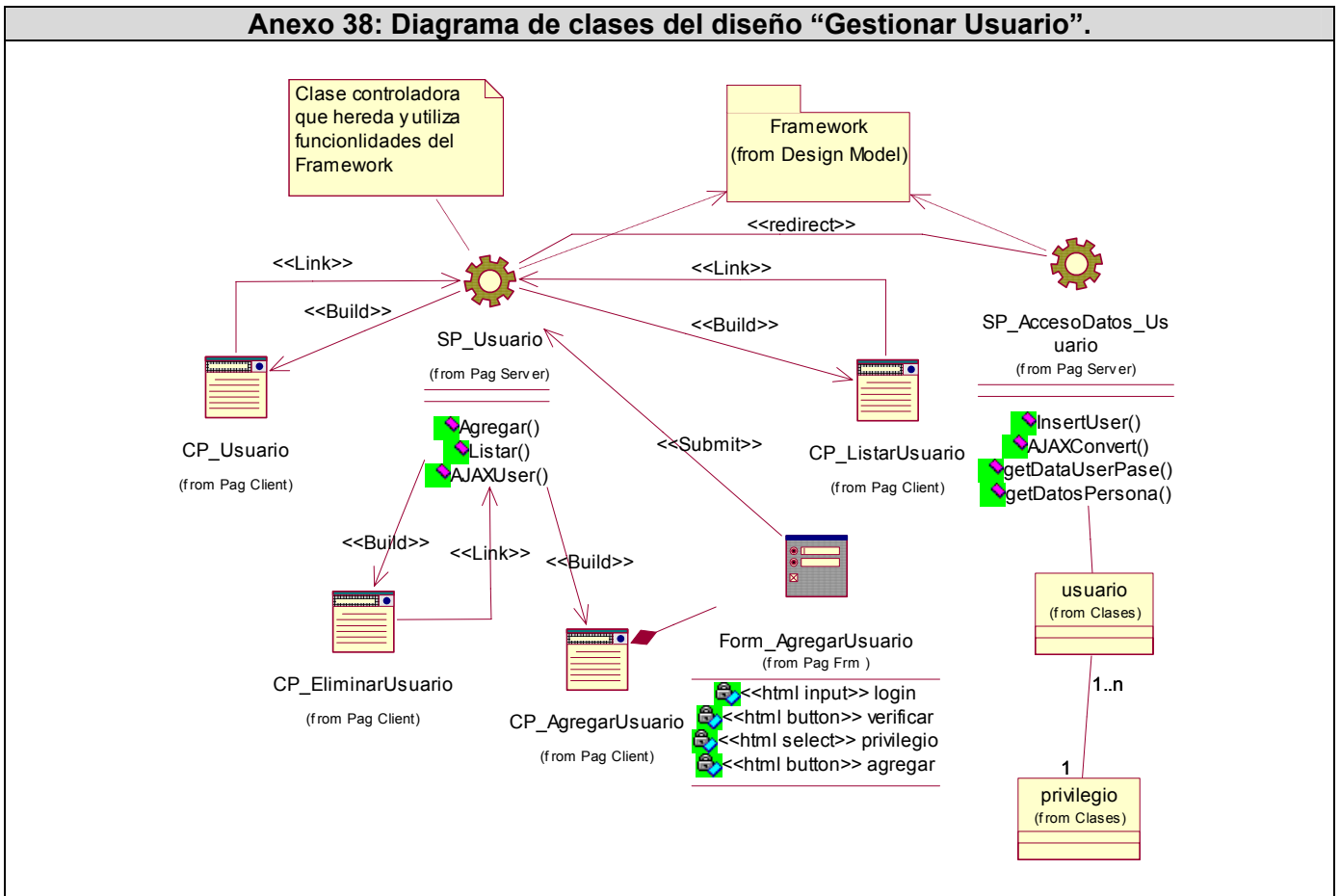
Anexo 36: Diagrama de clases del diseño “Ver puntos de salida”.



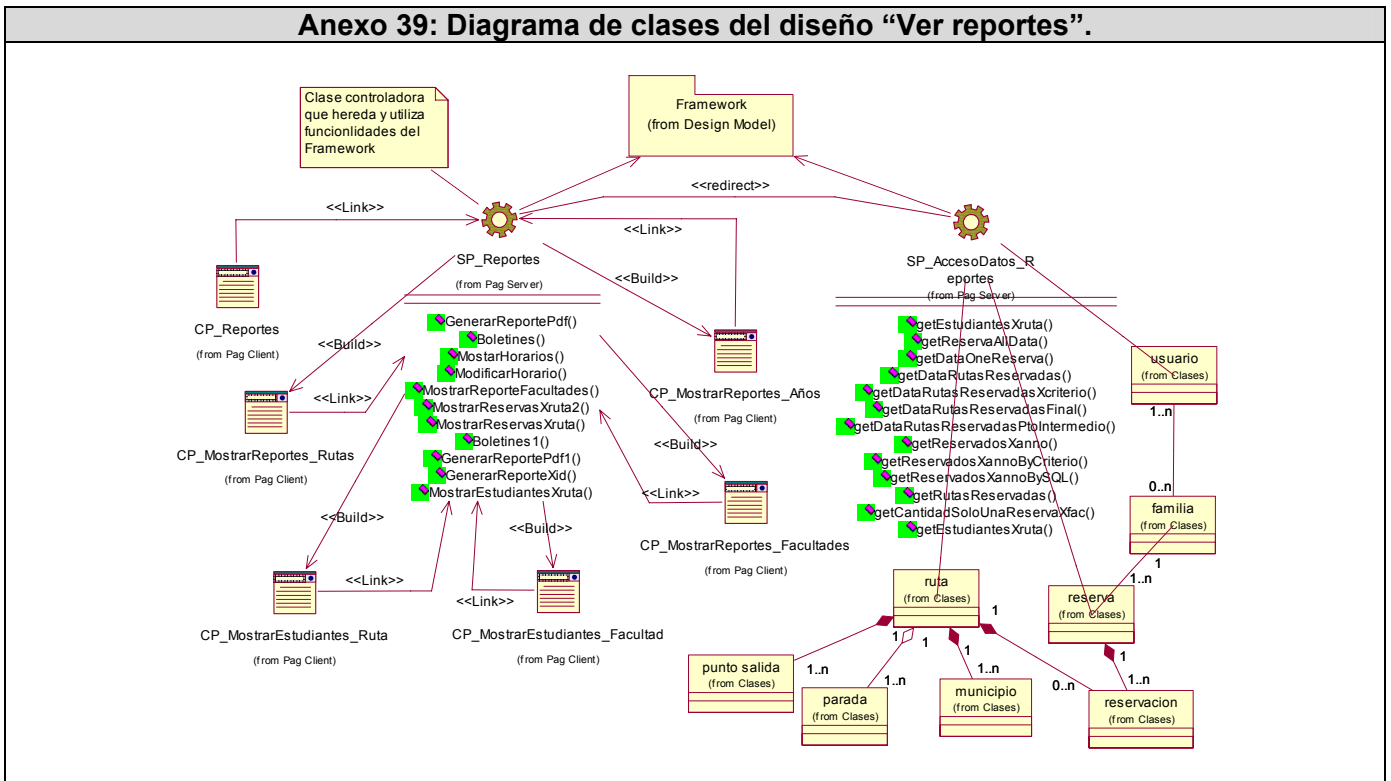
Anexo 37: Diagrama de clases del diseño “Gestionar rutas”.



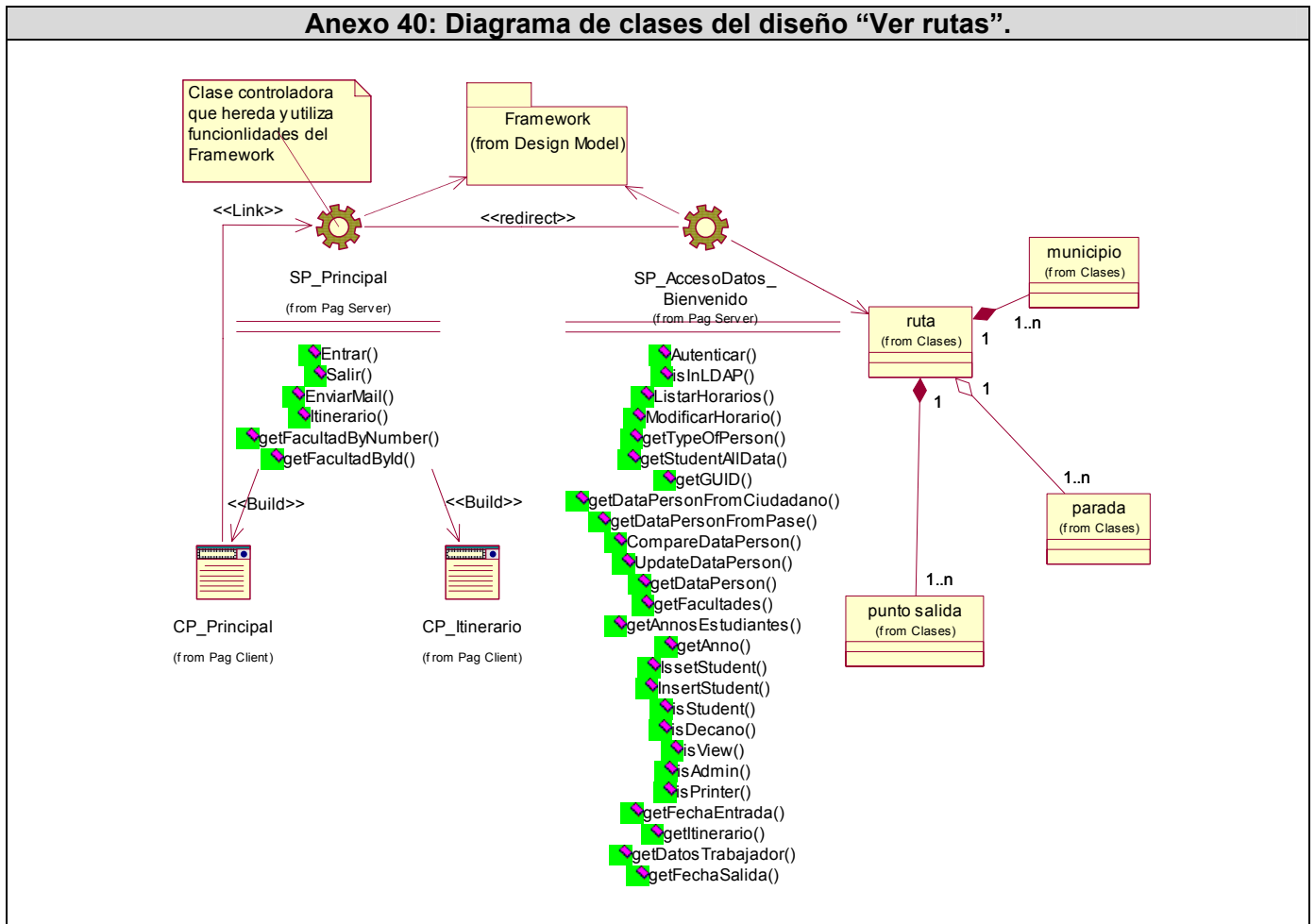
Anexo 38: Diagrama de clases del diseño "Gestionar Usuario".



Anexo 39: Diagrama de clases del diseño "Ver reportes".



Anexo 40: Diagrama de clases del diseño "Ver rutas".



Glosario de Términos

- Administrador: es la persona que posee privilegios para determinadas funcionalidades del sistema.
- 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.
- AJAX: **A**synchronous **J**avaScript **A**nd **X**ML.
- ASP: *Active Server Pages*. Es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). Con ASP se pueden combinar páginas HTML, *scripts* y objetos COM. Con el objetivo de crear aplicaciones potentes. Se caracterizan por su fácil desarrollo y mantenimiento.
- 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.
- CASE: *Computer Aided Software Engineering*.
- CGI: *Common Gateway Interface*.
- COCOMO: Modelo para la estimación de costos de productos informáticos.
- CUN: *Caso de uso del negocio*.
- CUS: *Caso de uso del sistema*.
- DHTML: *Dynamic HTML*.
- Directivo: es la persona que posee privilegios para determinadas funcionalidades del sistema.
- HTML: *HyperText Markup Language*. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. 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.
- HTTP: *HyperText Transfer Protocol*. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- Herramientas CASE: herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.

- **Hardware:** componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.
- **Internet:** sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.
- **Invitado:** es la persona que posee los privilegios mínimos en el funcionamiento del sistema.
- **Jefe de Consejo de Estado:** es la persona encargada de asignar la cantidad de ómnibus necesarios para la salida y entrada de estudiantes cada fin de semana de la universidad.
- **Jefe de Transporte:** es la persona encargada de realizar según la información que le suministra cada secretaria, la reservación de transporte según la necesidad de ese fin de semana.
- **JSP: *Java Server Pages.*** Es la tecnología para generar páginas Web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java. La tecnología JSP, o de JavaServer Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página Web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático.
- **Linux:** es el nombre de un núcleo, pero se suele denominar con este nombre a un sistema operativo de libre distribución software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.
- **Macromedia Dreamweaver MX:** herramienta para el desarrollo de aplicaciones Web de Macromedia. Combina en un único entorno de desarrollo accesible y potente las reconocidas herramientas de presentación visual de Dreamweaver, las características de rápido desarrollo de aplicaciones Web de Dreamweaver UltraDev y ColdFusion Studio, y el extenso soporte de edición de código de HomeSite. Ofrece una completa solución abierta para las tecnologías Web y estándares de hoy, incluyendo la accesibilidad y servicios Web.
- **Microsoft:** compañía que manufactura los sistemas de operación DOS y Windows.
- **MySQL:** es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves

foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

- MVC: *Modelo Vista Controlador*.
- PC: *Personal Computer*.
- PDO: *PHP Data Objects*.
- Perl: *Practical Extraction and Report Language*. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.
- PHP: *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 comprobar que cada estudiante salga de pase en el ómnibus que le corresponde.
- RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.
- Secretaria: es la persona encargada de llevar el control en su facultad de los estudiantes que salen de pase, así como reportar la cantidad de estudiantes por ruta que saldrán de pase para contribuir a la reservación del transporte.
- Software: programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.
- SQL: *Structured Query Language*. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Asocia características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.
- Sitio Web: es un conjunto de páginas Web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.
- SGBD: *Sistema de Gestión de Bases de Datos*. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.
- 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 (WWW): red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.
- XML: *Extensible Markup Language*. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.