



**Universidad de las Ciencias Informáticas
Facultad 1**



Sistema de Reservación de Pase Masivo

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

Autores: Yohana Baró Montenegro

Reynaldo Hernández Rodríguez

Tutor: Ing. Isuel Méndez Roldán

Ciudad de La Habana. Junio de 2007

A mis padres que siempre confiaron en mí, por haberme brindado su apoyo incondicional, y por guiarme durante tantos años.

Reynaldo

...A Dunia y Pablo, mis padres, por ser las personas más importantes en mi vida, por estar siempre a mi lado apoyándome, ayudándome en todo y a pesar de todo; por demostrarme que no hay nada imposible en la vida...

Yohana

Agradecimientos.

A mis padres, por ser mis primeros maestros. No se que hubiera hecho sin ustedes.

A mis profes de la vida: Julia, Esther Ulloa, Rosa Mirian, Annia, Arguelles, que moldearon el camino y confiaron en mí.

A mis amigos y vecinos de Jovellanos, gracias por pensar siempre en mí y dedicarme sus mejores pensamientos.

A mis amigos y compañeros de aula, por hacer que estos 5 años sean inolvidables.

A Jaime, por enseñarme a volar.

Yohana

A mis padres.

A mis familiares en general.

A todos los amigos y compañeros que de una forma u otra dieron su aporte para la realización de este trabajo.

A mis hermanos, por todo su cariño.

A Yoly, por haberme acompañado durante todos estos años de la carrera, por su amor, comprensión, y por todos los momentos que hemos vivido.

Reynaldo

A la Revolución, por habernos dado la posibilidad de estudiar y obtener los conocimientos necesarios para realizar este trabajo.

Resumen.

El presente proyecto consiste en la confección de una herramienta que sea capaz de agilizar la gestión del proceso de reservaciones de pase masivo en la Universidad de las Ciencias Informáticas (UCI).

La automatización de dicho proceso se hace necesaria debido al proyecto de Informatización de la UCI y a lo difícil de distribuir los viajeros de forma manual, ya que su número va en aumento.

Además es de gran interés de la universidad lograr una integración entre los diferentes servicios que existen así como un mejor aprovechamiento de la infraestructura informática del centro, garantizando de esta forma un adecuado flujo de la información y un rápido acceso a la misma por parte de los usuarios interesados.

Previamente no se ha elaborado ningún otro sistema para estas funciones. Es por ello que el presente está diseñado para hacer un óptimo uso de los servicios y recursos existentes en la UCI y lograr una completa integración con los sistemas de los cuales solicita información necesaria.

Índice	
INTRODUCCIÓN.	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.	6
Introducción.	6
1.1 Sistemas Automatizados de Reservas.	6
1.1.1 Historia de CRS.	7
1.1.2 De los CRS a los GDS	8
1.2 Sistemas Automatizados de reservas que existen.	10
1.3 Descripción actual de cómo se lleva el negocio.	12
1.3.1 Reglas del Negocio.	13
1.4 Propuesta de Solución: Sistema de Reservación de Pase Masivo.	13
1.5 Marco de trabajo y herramientas de apoyo para la elaboración.	14
1.5.1 Aplicaciones Web.	15
1.5.2 Arquitectura Cliente/Servidor.	16
1.5.3 PHP.	17
1.5.4 PostgreSQL.	18
1.5.5 Servidor Web Apache.	19
1.5.6 Java Script.	20
1.5.7 Ajax.	20
1.5.8 RUP.	21
1.5.9 UML.	23
1.5.10 Rational Rose.	24
1.5.11 Zend Studio.	25
1.5.12 CASE Studio.	26
1.5.13 CakePHP.	26
Conclusiones.	27
CAPITULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.	28
Introducción.	28
2.1 Modelo del negocio propuesto.	28
2.1.1 Actores / Trabajadores del negocio.	28
2.2 Descripción del Proceso del Negocio.	30
2.2.1 Diagrama de CU del Negocio.	30
2.2.2 Descripción de los CUN.	30
2.3 Diagrama del Modelo de Objetos del Negocio.	33

2.4	Especificación de los requerimientos de Software. -----	33
2.4.1	Requerimientos Funcionales. -----	33
2.4.2	Requerimientos No Funcionales. -----	36
2.5	Descripción de los procesos a automatizar. -----	38
2.6	Modelo del Sistema. -----	39
2.6.1	Actores del Sistema. -----	39
2.6.1	Diagrama de Casos de Uso del Sistema. -----	41
2.6.2	Descripción de los Casos de Uso del Sistema. -----	42
	Conclusiones. -----	63
CAPITULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA. -----		64
	Introducción. -----	64
3.1	Patrones. -----	64
3.1.1	Patrón de arquitectura. -----	65
3.2	Modelo de Diseño. -----	65
3.2.1	Diagramas de clases del diseño. -----	65
3.3	Diseño de la base de datos. -----	75
3.3.1	Diagrama de clases persistentes. -----	75
3.3.1	Modelo de datos. -----	77
3.4	Principios de Diseño. -----	78
3.4.1	Estándares de la Interfaz de la aplicación. -----	78
3.4.2	Tratamiento de excepciones. -----	79
3.5	Modelo de implementación. -----	79
3.5.1	Diagrama de despliegue. -----	79
3.5.2	Diagrama de componentes. -----	80
	Conclusiones. -----	80
CAPITULO 4: ESTUDIO DE FACTIBILIDAD. -----		81
	Introducción. -----	81
4.1	El Método de análisis de Puntos de Casos de Uso. -----	81
1.	Cálculo de Puntos de Casos de Uso sin ajustar. -----	82
2.	Cálculo de Puntos de Casos de Uso ajustados. -----	83
3.	Estimación del esfuerzo -----	85
4.2	Beneficios tangibles e intangibles. -----	86
4.3	Análisis de costo / beneficio. -----	87
	Conclusiones. -----	87

CONCLUSIONES. -----	88
RECOMENDACIONES. -----	89
GLOSARIO DE TÉRMINOS. -----	90
REFERENCIAS BIBLIOGRÁFICAS. -----	92
ANEXOS. -----	94
Anexo I. Diagramas de secuencia. -----	94
Anexo II. Diagrama de Despliegue. -----	99
Anexo III. Diagrama de Componentes.-----	100

Índice de Figuras.

FIGURA 2. 1 ACTORES/ TRABAJADORES DEL NEGOCIO.	29
FIGURA 2. 2 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.	30
FIGURA 2. 3 DIAGRAMA DE ACTIVIDAD GESTIÓN PASE MASIVO.....	31
FIGURA 2. 4 DIAGRAMA DE ACTIVIDAD REALIZAR DISTRIBUCIÓN.....	32
FIGURA 2. 5 MODELO DE OBJETOS DEL SOFTWARE.	33
FIGURA 2. 6 ACTORES DEL SISTEMA.	40
FIGURA 2. 7 DIAGRAMA DE CASOS DE USO DEL SISTEMA.	41
FIGURA 3. 1 DIAGRAMA DE CLASES DEL DISEÑO " AUTENTICAR USUARIO"	66
FIGURA 3. 2 DIAGRAMA DE CLASES DEL DISEÑO " GESTIONAR RESERVACIÓN ESTÁNDAR"	67
FIGURA 3. 3 DIAGRAMA DE CLASE DEL DISEÑO "GESTIONAR RESERVACIÓN CON PERMISO"	68
FIGURA 3. 4 DIAGRAMA DE CLASE DEL DISEÑO "GESTIONAR DISTRIBUCIÓN"	69
FIGURA 3. 5 DIAGRAMA DE CLASE DEL DISEÑO "GESTIONAR SALIDA"	70
FIGURA 3. 6 DIAGRAMA DE CLASE DISEÑO "GESTIONAR TRANSPORTACIÓN"	71
FIGURA 3. 7 DIAGRAMA DE CLASE DEL DISEÑO "GESTIONAR TRANSPORTE"	72
FIGURA 3. 8 DIAGRAMA DE CLASE DEL DISEÑO "GESTIONAR VIAJE"	73
FIGURA 3. 9 DIAGRAMA DE CLASE DEL DISEÑO "CONSULTAR INFORMACIÓN"	74
FIGURA 3. 10 DIAGRAMA DE CLASE DEL DISEÑO "GESTIONAR USUARIO"	75
FIGURA 3. 11 DIAGRAMA DE SECUENCIA " GESTIONAR TRANSPORTACIÓN"	94
FIGURA 3. 12 DIAGRAMA DE SECUENCIA " GESTIONAR TRANSPORTE"	95
FIGURA 3. 13 DIAGRAMA DE SECUENCIA " GESTIONAR VIAJE"	96
FIGURA 3. 14 DIAGRAMA DE SECUENCIA " GESTIONAR SALIDA"	97
FIGURA 3. 15 DIAGRAMA DE SECUENCIA " GESTIONAR DISTRIBUCIÓN"	98
FIGURA 3. 16 DIAGRAMA DE CLASES PERSISTENTES.....	76
FIGURA 3. 17 MODELO DE DATOS.....	77
FIGURA 3. 18 DIAGRAMA DE DESPLIEGUE.	99
FIGURA 3. 19 DIAGRAMA DE COMPONENTES.	100

Índice de Tablas.

TABLA 1. ACTORES Y TRABAJADORES DEL NEGOCIO PROPUESTO.	29
TABLA 2. DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.	40
TABLA 3. DESCRIPCIÓN CUS ´ AUTENTICAR USUARIO´	42
TABLA 4. DESCRIPCIÓN CUS ´ GESTIONAR RESERVACIÓN´	43
TABLA 5. DESCRIPCIÓN CUS ´ GESTIONAR SOLICITUD DE RESERVACIÓN´	45
TABLA 6. DESCRIPCIÓN CUS ´ CONSULTAR INFORMACIÓN´	46
TABLA 7. DESCRIPCIÓN CUS ´ GESTIONAR SALIDA´	47
TABLA 8. DESCRIPCIÓN CUS ´ GESTIONAR VIAJE	49
TABLA 9. DESCRIPCIÓN CUS ´ GESTIONAR TRANSPORTACIÓN´	51
TABLA 10. DESCRIPCIÓN CUS ´ GESTIONAR TRANSPORTE´	54
TABLA 11. DESCRIPCIÓN CUS ´ GESTIONAR DISTRIBUCIÓN´	57
TABLA 12. DESCRIPCIÓN CUS ´ GESTIONAR USUARIO´	59
TABLA 13. DESCRIPCIÓN CUS ´ GESTIONAR BOLETINES´	62
TABLA 14. FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR.	82
TABLA 15. FACTOR DE PESO DE LOS CU SIN AJUSTAR.	82
TABLA 16. FACTOR DE COMPLEJIDAD TÉCNICA.	83
TABLA 17. FACTOR DE AMBIENTE.	84
TABLA 18. VALOR DEL ESFUERZO.	85

Introducción.

Hoy en el mundo nos encontramos, con que el uso de las nuevas tecnologías de la información y las comunicaciones se hace indispensable en las actividades de la vida cotidiana. La cual en la sociedad actual está alcanzando un lugar notable e irrefutable, y por tanto ha hecho que el hombre cambie sus formas de adquirir y transmitir los conocimientos. Nuestro país, además de estar inmerso en una verdadera revolución educacional, se ha volcado a la labor de automatizar al máximo la sociedad cubana.

En la Universidad de las Ciencias Informáticas (UCI), en correspondencia con el actual desarrollo y puesta en marcha de proyectos en el ámbito de la informática y las telecomunicaciones, se están desarrollando varios proyectos que pretenden automatizar al máximo todos los procesos, no solo relacionados con la actividad educacional dentro de la universidad, sino en el entorno de la sociedad misma la cual será el punto de partida para la informatización de la sociedad cubana.

Con este propósito se desea que toda la gestión docente y social esté integrada y sea de fácil acceso para cualquier ciudadano.

El trabajo “Sistema de Reservación de Pase Masivo” está vinculado a la distribución de los estudiantes y profesores de la universidad en el transporte que se brinda para las salidas masivas que se efectúan en la misma.

Actualmente, la UCI en dos fechas diferentes de cada curso escolar planifica los pases masivos para que estudiantes y profesores viajen hacia sus provincias de origen, desde donde también deben regresar.

Una vez hecha la planificación el tramitador negocia con la agencia central las cifras y las fechas (de ida y regreso) y estas se publican en la Intranet para conocimiento de todos los interesados.

Para poder salir en el transporte se exige un ticket o pasaje en el que se especifican el nombre y apellidos, carne de identidad, fecha, número del transporte (tren o guagua), hora de salida, destino y una asignación coche-asientos, etc.

Al analizar dicho proceso, se detectaron problemas que frenan el óptimo funcionamiento de este servicio. Por ejemplo, la gestión de salidas y entradas de los usuarios es una tarea engorrosa, ya que, hasta el momento el proceso se lleva a cabo de forma manual o sea no existe un sistema capaz de gestionar todo el flujo de información sobre aquellas personas que desean viajar, lo que trae consigo pérdida de tiempo y dificultades en el trabajo.

Al no existir un control preciso de la información que se maneja, no es posible detectar los usuarios que no viajan luego de haberseles reservado, ya sea porque no se encuentran en la universidad por algún problema o porque hayan viajado por sus propios medios. Esto provoca que no se puedan optimizar los recursos de transporte que se ponen a disposición de los usuarios.

Además no es posible que los usuarios puedan reservar para otra provincia que no sea la de origen.

De lo anteriormente expuesto, se deriva la necesidad de un sistema que automatice el proceso de reservación de transporte para el pase masivo en la UCI.

Problema Científico

¿Los procesos de gestión de transportes para el pase masivo en la UCI se agilizarán con la implementación de un sistema automatizado?

Objetivo General

Desarrollar un sistema automatizado que permita agilizar la gestión del proceso de reservación de transporte para el pase masivo en la Universidad de las Ciencias Informáticas.

Objetivos Específicos:

- Diseñar un módulo que permita a los usuarios reservar sus viajes.
- Diseñar un módulo que brinde las funcionalidades administrativas del sistema.
- Implementar las funcionalidades priorizadas del sistema.

Objeto de Estudio

El proceso de reservación de transporte para el pase masivo en la Universidad de las Ciencias Informáticas.

Campo de Acción

Universidad de las Ciencias Informáticas.

Hipótesis

La automatización de los procesos implicados en la reservación del transporte para el pase masivo, agilizará el servicio de transportación en la Universidad de las Ciencias Informáticas.

Tareas de la Investigación:

- Estudiar el marco contextual (proceso de reservaciones de pase masivo) y diagnosticar el problema.
- Realizar un estudio teórico y tendencial sobre las aplicaciones Web, las herramientas, técnicas y metodología que se utilizarán.
- Realizar el proceso de modelado del software para la elaboración del sistema.
- Realizar un estudio de factibilidad.
- Implementar las funcionalidades priorizadas de la aplicación.

Este trabajo ha sido organizado de la siguiente manera:

Capítulo 1: Fundamentación Teórica. En este capítulo se abordará el estado actual de desarrollo de las aplicaciones de gestión de reservaciones en el ámbito mundial, nacional y en la Universidad. Se describirá brevemente como se lleva a cabo el proceso de reservación en la universidad y se planteará una propuesta de solución para resolver el problema detectado. Además se presentará la propuesta de herramientas y tecnologías a utilizar para el desarrollo de la aplicación.

Capítulo 2: Descripción de la Solución Propuesta. En este capítulo se modela el negocio propuesto, identificándose los actores, trabajadores y los casos de uso correspondientes. Se plantean los requerimientos funcionales y no funcionales de la aplicación a desarrollar y se describe la solución propuesta en términos de casos de uso de sistema.

Capítulo 3: Construcción de la Solución Propuesta. En el presente capítulo se modelan los artefactos que se recogen a partir del flujo trabajo de diseño e implementación. Utilizando las extensiones de UML para Web, se definen los diagramas de clases del diseño, se definen los diagramas de clases persistentes, el modelo de datos, el diagrama de despliegue y el diagrama de componentes. Además se explican los principios de diseño seguidos para la confección de la interfaz y el tratamiento que se le da a los errores que puedan surgir.

Capítulo 4: Estudio de Factibilidad. Es un estudio de factibilidad de la construcción del sistema, observando los beneficios tangibles e intangibles y analizando los costos de desarrollo de la propuesta.



Fundamentación Teórica.

Introducción.

En este capítulo se abordará el estado actual de desarrollo de las aplicaciones de gestión de reservaciones en el ámbito mundial, nacional y en la Universidad. Se describirá brevemente como se lleva a cabo el proceso de reservación en la universidad y se planteará una propuesta de solución para resolver el problema detectado. Además se presentará la propuesta de herramientas y tecnologías a utilizar para el desarrollo de la aplicación.

1.1 Sistemas Automatizados de Reservas.

Actualmente en el mundo se ha avanzado mucho con respecto a los Sistemas Automatizados de Reservas con el fin de obtener las ventajas competitivas que el uso de esta herramienta permite generar, convirtiéndose Internet en la plataforma y el instrumento dominante que se utiliza para la promoción y la distribución de estos.

Las ventajas competitivas que ofrece para las empresas del sector del transporte, así como para las dedicadas a la prestación de servicios turísticos, un sistema automatizado de reservas on line se traducen, entre otras, en un aumento de la calidad de los servicios prestados a sus clientes.

Donde algunos de los principales servicios que se ofrecen a través de estos sistemas consisten en poner a disposición del usuario: **información** sobre todos los servicios disponibles; **reservas**; listas de espera de plazas en vuelos, barcos y hoteles; **tarifas** confidenciales y tarifas normales; **confeccionar** agendas, cambios y anulaciones; **emisión** de billetes y de 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 un ahorro operativo y de papel); **chip-card** (tarjeta electrónica con soporte magnético del estilo de la Visa).

1.1.1 Historia de CRS.

Con anterioridad a la aparición de los sistemas automatizados de reservas, las agencias de viajes tenían que navegar por las listas de rutas y precios que les suministraban las diversas compañías aéreas, hasta encontrar la opción más acorde con las preferencias de los clientes. Una vez escogida ésta, la agencia de viajes se ponía en contacto con la línea aérea (telefónicamente o por télex) para reservar una plaza.

Este sistema basado en el manejo manual de reservas de plazas resultaba ineficaz y costoso, tanto para las agencias de viajes como para las compañías aéreas, y por otro lado, el cliente nunca podía saber si la opción escogida era o no la mejor, ya que era prácticamente imposible disponer de toda la información sobre los horarios de todas las líneas.

Ante esta situación American Airlines (AA) decide a principios de los sesenta, desarrollar un sistema para automatizar la gestión de reservas de plazas. Nace así Sabre, el primer CRS (Computerized Reservation Systems). Al principio, la finalidad de Sabre era la de gestionar toda la información que AA generaba diariamente. Se trataba de desarrollar un sistema informatizado que permitiera gestionar no sólo los datos relativos a la logística de las naves (agendas de catering y reparación, horarios del personal de vuelo, etc.), sino también los datos de reservas de plazas realizadas por los clientes.

A mediados de los setenta AA instaló en algunas agencias de viajes, terminales conectados a su ordenador central para que fueran las propias agencias las que realizaran las reservas desde sus terminales, consiguiendo de esta forma simplificar el proceso y reducir costes. Surge así una nueva industria, la de la distribución electrónica de información sobre vuelos, en la que los CRS o sistemas automatizados de reserva, constituyeron el primer estadio en la realización de actividades de e-business en el sector turístico

El siguiente paso consistió en incluir en Sabre información sobre las tarifas y vuelos de otras compañías aéreas competidoras de AA, ya que las agencias de viajes se mostraban reticentes a incorporar otros terminales conectados a los CRS desarrollados por otras compañías, las cuales no tardaron en reaccionar ante la gran amenaza que Sabre representaba para su supervivencia. Así United Airlines diseñó Apollo; el consorcio europeo entre: British Airways, Alitalia, KLM y Swissair desarrolló Galileo y, finalmente, el formado por Lufthansa, Air France e Iberia desarrolló el sistema Amadeus.

Los primeros CRS, por lo tanto, consistían en un inventario centralizado de los asientos disponibles en cada vuelo. A través de terminales periféricos situados en los puntos de venta, agencias de viajes y oficinas propias de las compañías aéreas, los usuarios podían acceder directamente a dicho inventario de plazas disponibles, rutas y tarifas de diversas compañías.

1.1.2 De los CRS a los GDS

En la década de los setenta los CRS dejaron de ser únicamente un canal de distribución para reservar billetes de avión y se incorporaron otros servicios propios de las agencias de viajes, tales como reservas hoteleras, coches de alquiler, paquetes turísticos, cruceros, e incluso información general de los destinos, como horarios comerciales, eventos culturales y espectáculos, meteorología, etc.

Este proceso supuso la incorporación de otros proveedores que veían en estos sistemas un buen canal para que las agencias de viajes pudieran acceder a sus productos con más facilidad y

economía. Surgen de esta forma los GDS (Global Distribution Systems), como auténticos sistemas globales de distribución ya que integran las actividades y procesos de las diversas cadenas de valor de los diferentes proveedores de servicios turísticos (compañías aéreas, ferroviarias y marítimas, agencias de viajes, hoteles, etc.).

De esta manera, los CRS / GDS se pueden definir como sistemas informáticos que conectan a los proveedores de productos turísticos con sus distribuidores, que se convierten a su vez en los principales usuarios (agencias de viajes). Actualmente los GDS engloban consorcios de proveedores de productos y servicios turísticos y suministran información en tiempo real a cientos de compañías de transporte (aéreo, marítimo, ferroviario e incluso de alquiler de coches) y decenas de miles de hoteles. Sus terminales se encuentran distribuidos por las agencias de viajes y configuran una gigantesca red mundial dedicada a la venta de productos turísticos.

Con la aparición del comercio electrónico se le a permitido al cliente realizar las reservas que le interesen desde su casa o lugar de trabajo sin necesidad de desplazarse a la agencia de viajes. Los principales GDS tienen ya su propia página web para poder ofrecer viajes al usuario final desde internet.

- ABACUS <http://www.abacus.com.sg/>
- AMADEUS <http://www.amadeus.net/>
- GALILEO <http://www.galileo.com/>
- SABRE <http://www.sabre.com/>
- WORLDSPAN http://www.worldspan.com/map/init_index.asp

1.1.3 Cuba y los Sistemas Automatizados de Reserva.

Específicamente en Cuba aprovechando el desarrollo del polo turístico y la introducción de las nuevas tecnologías se a comenzado ha trabajar en función de la informatización de los servicios de esta rama de la economía.

Donde la mayoría de nuestros hoteles y establecimientos de servicios del sector turístico cuentan con sitios Web publicados en Internet y cuentan con el servicio de reservaciones on-line, ejemplo de ello tenemos:

- Para reservar vuelos <http://www.cubana.cu>
- Para la reservación de paquetes turísticos <http://www.gocubaplus.com>
- Para la renta de autos de turismo <http://www.cubacar.cubanacan.cu> y <http://www.transturentacar.cu>
- Para servicios de taxis <http://www.transturtaxionline.cu>
- Para la reserva de [hoteles](http://www.hotelnacionaldecuba.cu) <http://www.hotelnacionaldecuba.cu>

1.2 Sistemas Automatizados de reservas que existen.

En la actualidad y como una de las consecuencias del desarrollo alcanzado por la humanidad, las personas tienen menos tiempo dada la agitada vida que se lleva. Es por esto, y gracias al desarrollo de las TIC, que en el mundo está tomando cada vez más auge el uso de la red para gestionar ciertas actividades que tomaría demasiado tiempo llevarlas a cabo personalmente. Las reservaciones online han ganado mucha aceptación ya que se les permite a los usuarios, desde cualquier lugar que esté acceder a los diferentes servicios que se ofertan.

Sistema de Reservación de eDreams. (www.edreams.es)

(Software dedicado a la reserva de servicios, entre ellos, reserva de vuelos). Permite que el usuario reserve viaje de ida y vuelta, del origen y destino que escoja, así como las fechas y horarios en las que quiere viajar. Posibilita al cliente la ventaja de escoger viajar en vuelos directos o sea sin hacer escalas hasta el destino final y con las compañías aéreas de su preferencia. Facilita información sobre el viaje para que el cliente sepa las posibilidades de reservación que tiene de forma general. Ofrece al pasajero varias formas de pago, entre las que se encuentra: tarjeta, por transferencia y en efectivo. El cliente puede elegir la forma de pago que más le convenga. Además de que puede realizar la compra on line o vía telefónica.

Para confirmar la solicitud de viaje del cliente, se le envía un correo electrónico al e-mail especificado por este y no es válida la reservación hasta que no se confirme la solicitud mediante esta misma vía.

Sistema de Reservación de Cubana de Aviación. (www.cubana.cu)

(Software dedicado a la reserva de vuelos específicamente). Este software de Cubana de Aviación, S.A. posibilita al reservar el billete de viaje seleccionar la ciudad origen y destino, la fecha y clase (económica, club tropical (1ra)) en la cual desea viajar el cliente, además puede solicitar el asiento que desee ocupar en la aeronave (asiento de fumador, no fumador, ventanilla, pasillo, número y fila).

Ofrece a sus clientes información sobre la aerolínea además de facilitar las ventas directas de pasaje. Las ventas directas comprenden la reserva y el pago. El pago deberá efectuarse usando una tarjeta de crédito Visa ó MasterCard; enviando notificación al pasajero vía e-mail si la transacción fue exitosa.

Sistema de Reservación de Transporte, UCI. (pase.uci.cu)

Este sistema informático permite la reservación de los pases de fin de semana de los estudiantes en la Universidad de las Ciencias Informáticas hacia los diferentes municipios de La Habana.

Para reservar el pase, el estudiante solicita hasta una fecha determinada, salir el fin de semana. De esta manera el sistema permite al estudiante reservar ida y regreso en el transporte de la universidad, atendiendo al municipio y ruta en la que quiera viajar este. Si el estudiante no es de La Habana debe introducir el nombre del familiar, el parentesco, la dirección y el teléfono de la persona que va a visitar. Haciéndose luego la asignación del transporte (ómnibus). Todos estos trámites son totalmente gratis.

En este momento en la Universidad de la Ciencias Informáticas no existe un sistema que se encargue de gestionar toda la información con respecto a las reservaciones hechas por los usuarios para las salidas de pase masivo en la universidad, es por ello que se plantea la necesidad de desarrollar un sistema que le de solución a esta problemática.

Para esto se ha decidido no utilizar ninguno de los sistemas que existen en el mundo ni en la universidad como los antes mencionados, debido a que cada uno de ellos tienen características particulares, es decir, cada uno ha sido implementado con el fin de satisfacer necesidades particulares de la compañía o institución (UCI) a la que pertenecen.

1.3 Descripción actual de cómo se lleva el negocio.

La dirección de Transporte de la UCI recibe la información de un nuevo pase masivo y comienza a gestionar, teniendo en cuenta la cantidad total de viajeros, los transportes necesarios para el pase.

Una vez que obtiene esta información, distribuye los pasajeros, y si algún pasajero desea, saber la fecha y hora de salida y regreso debe esperar a que se publique en la Intranet de la Universidad los listados de transporte.

En caso de que exista algún error o disconformidad los pasajeros pueden reclamar a través de, el correo electrónico, el teléfono o acudiendo personalmente a la Dirección de Transporte.

1.3.1 Reglas del Negocio.

Se identificaron las siguientes reglas que debe seguir la aplicación que se desarrolle, a fin de respetar y garantizar las restricciones que existen en el negocio:

1. Cualquier persona de la universidad puede obtener información acerca de los viajes, las transportaciones y de las personas que viajaran.
2. Los únicos que pueden reservar son los estudiantes, profesores internos y directivos que residan en la universidad.
3. Los trabajadores del centro no pueden reservar a no ser que estén autorizados por los directivos de la universidad encargados.
4. El Especialista Superior de la Dirección de Transporte es el único que puede planificar, cambiar y actualizar las salidas.
5. Cada salida tendrá dos viajes: uno de ida y otro de regreso.
6. Los usuarios pueden reservar una sola vez por viaje.

1.4 Propuesta de Solución: Sistema de Reservación de Pase Masivo.

Se propone la elaboración de un sistema informático que brinde soporte a los procesos de la dirección de transporte, incluyendo además una serie de funcionalidades de administración del sistema, que permitan gestionar toda la información con la que este trabaja y que brinden la

mayor flexibilidad posible, con vista a su adaptación a cualquier tipo de situación que pueda presentarse.

El Sistema de Reservación de Pase Masivo, para todas sus acciones consulta la información contenida en la base de datos central que es la que proporciona los datos de las personas que reservarán.

Además de las funcionalidades que soportan todo el flujo de procesos que ocurren en la dirección de transporte en cuanto a la reservación de los viajes, la solución propuesta brinda la posibilidad de adaptarse a cualquier disposición diferente que exista en cuanto a quienes pueden ser las personas que reserven y hacia donde, lo cual resulta actualmente todo un proceso engorroso.

1.5 Marco de trabajo y herramientas de apoyo para la elaboración.

Para la elaboración de la propuesta de solución que plantea esta investigación se desarrollará una aplicación Web, lo que permite que el sistema pueda ser utilizado desde distintos lugares y sin más requerimientos que una computadora con navegador Web y conexión a la red.

Se usará PHP como lenguaje de programación del lado del servidor, utilizando el Zend Studio como herramienta de desarrollo, PostgreSQL como Sistema Gestor de Bases de Datos y Apache como servidor de aplicaciones Web, todo esto por las potencialidades que ofrecen, así como por formar parte del grupo de software de código abierto, solución por la que se aboga en la UCI.

Se utilizará JavaScript del lado del cliente para lograr la interactividad con el usuario en el navegador y específicamente la técnica Ajax, ya que la misma permite actualizar parte de una página en cualquier momento, dándole a los usuarios una respuesta instantánea a sus ingresos y consultas.

Para controlar, y planificar la propuesta que presenta este trabajo y por las características y facilidades que aporta a todo el proceso, se empleará una metodología RUP, usando como herramienta el Rational Rose y el lenguaje UML.

1.5.1 Aplicaciones Web.

Como resultado del desarrollo y evolución del mundo, surgió “una red gigante que agrupa miles de redes de computadoras distribuidas por toda la superficie del globo conocida como **Internet**.

La cual nace específicamente a mediados de la década de los setenta, bajo los auspicios de DARPA, la Agencia de Proyectos Avanzados para la Defensa de Estados Unidos. DARPA inició un programa de investigación de técnicas y tecnologías para unir diversas redes de conmutación de paquetes, permitiendo así a los ordenadores conectados a estas redes comunicarse entre sí de forma fácil y transparente. A través de el conjunto de protocolos TCP/IP esta red conecta actualmente a millones de computadoras en el mundo.

Uno de los servicios más importantes y más usados de Internet es el **World Wide Web** (del inglés, *Telaraña Mundial*), *la Web* o *WWW*, que constituye el universo de información accesible a través de Internet, fuente inagotable del conocimiento humano. Esta información se visualiza de manera gráfica e interactiva haciendo uso del sistema de hipertexto.

Este servicio, de igual forma, ha evolucionado. Estos últimos se conocen como **aplicaciones Web**, y son implementados por grupos de desarrollo de software.

Con las aplicaciones Web el usuario no sólo recibe páginas del servidor como respuesta a su solicitud, sino que puede también enviar información de regreso a través de formularios. Están diseñadas para interactuar con bases de datos con el fin de recoger, almacenar, organizar y distribuir información, creando herramientas poderosas a ser utilizadas en la administración consistente de la información.

1.5.2 Arquitectura Cliente/Servidor.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. (3)

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

Los **clientes** realizan generalmente funciones como:

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

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

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

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo. (4)

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

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

1.5.3 PHP.

PHP (acrónimo de Hypertext Preprocessor) es un lenguaje del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado usado para la creación de contenido dinámico para sitios web. Puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades tan extraordinarias que se ha convertido en el favorito de millones de programadores en todo el mundo. (5)

PHP fue creado por Rasmus Lerdorf a finales de 1994, aunque no hubo una versión utilizable por otros usuarios hasta principios de 1995. Esta primera versión se llamó, Personal Home Page Tools. A lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales. (6)

El crecimiento de PHP ha sido exponencial, ya que cada día son muchísimas más las páginas Web que lo utilizan para su funcionamiento, según las estadísticas, PHP se utiliza en

más de 10 millones de páginas, y cada mes realiza un aumento del 15%. Actualmente es la gran tendencia en el mundo de Internet. (7)

Corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, además de que:

- Es **software libre**, lo que implica menos costes y servidores más baratos que otras alternativas.
- Es **muy rápido**. Su integración con base de datos como MySQL, MS SQL, Oracle, Informix, PostgreSQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su **librería estándar** es realmente **amplia**, lo que permite reducir los llamados 'costes ocultos', uno de los principales defectos de ASP.
- PHP es suficientemente **versátil** y **potente** como para hacer tanto aplicaciones grandes que necesiten acceder a recursos a bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario relleno por el usuario.
- PHP tiene una de las **comunidades más grandes en Internet**, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.
- PHP esta revolucionando Internet a la vez que evoluciona a pasos gigantescos.

1.5.4 PostgreSQL.

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos derivado del paquete Postgres escrito en Berkeley. Es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas

por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, PERL, tcl y python).

Es capaz de manejar una gran cantidad de datos, permite gran cantidad de accesos simultáneos de los usuarios, y sin embargo es un sistema de Bases de Datos que es relativamente pequeño y rápido. Además está disponible sin coste alguno. (8)

1.5.5 Servidor Web Apache.

Un servidor de páginas Web es un programa que permite acceder a páginas Web alojadas en un ordenador. Hoy en día Apache es el servidor web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad, surgió en abril de 1996 y ya en julio del 2002 era utilizado por el 57% de los sitios Web de Internet.

Tiene capacidad para servir páginas tanto de contenido estático, para lo que nos serviría sencillamente un viejo ordenador 486, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información, es muy potente y altamente configurable.

Los servidores Web suministran páginas web a los navegadores que lo solicitan. En términos más técnicos, los servidores Web soportan el Protocolo de Transferencia de Hipertexto como HTTP (HyperText Transfer Protocol), el estándar de Internet para comunicaciones web. Usando HTTP, un servidor Web envía páginas web en HTML y Common Gateway Interface (CGI), así como otros tipos de scripts a los navegadores o browsers cuando éstos los requieren. Cuando un usuario hace clic sobre un enlace a una página web, se envía una solicitud al servidor web para localizar los datos nombrados por ese enlace. El servidor web recibe esta solicitud y suministra los datos que le han sido solicitados o bien devuelve un mensaje de error. (9)

1.5.6 Java Script.

JavaScript es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Se utiliza embebido en el código HTML, entre las tags <script> y </script>.

Es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias Java Script contenidas en una página HTML y ejecutarlas adecuadamente.

Java Script es un lenguaje orientado a objetos. El modelo de objetos de Java Script está reducido y simplificado, pero incluye los elementos necesarios para que los Scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

Cuando un usuario pincha sobre un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante JavaScript se pueden desarrollar Scripts que ejecuten acciones en respuesta a estos eventos. En este caso actúa como un lenguaje orientado a eventos. (10)

1.5.7 Ajax.

AJAX es el acrónimo inglés para Asynchronous JavaScript and XML (JavaScript y XML asíncrono). Es una técnica de desarrollo web que genera aplicaciones web interactivas combinando:

- Document Object Model (DOM) para visualizar dinámicamente e interactuar con la información presentada.
- XML, XSLT para intercambiar y manipular datos.
- CSS para definir el aspecto (look and feel) del documento.
- JSON y JSON-RPC pueden ser alternativas a XML/XSLT

- XMLHttpRequest para recuperar datos de forma asincrónica.
- Javascript como nexo de unión de todas estas tecnologías.

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. (11)

Entre algunas de las características de Ajax se puede mencionar:

- Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente.
- Estas aplicaciones tienen un aspecto (look and feel) muy similar a las aplicaciones de escritorio tradicionales sin depender de plugins o características específicas de los navegadores.
- Se reduce el tamaño de la información intercambiada.
- Muchas micro-peticiones, pero el flujo de datos global es inferior.
- Se libera de procesamiento a la parte servidora (se realiza en la parte cliente)
- AJAX actualiza porciones de la página en vez de la página completa.

1.5.8 RUP.

Durante el proceso de desarrollo de software empleamos la metodología RUP (12) (Rational Unified Process, Proceso Unificado de Rational). RUP hace énfasis en la adopción de las mejores prácticas del desarrollo de software, como una manera de reducir los riesgos inherentes en el desarrollo de una nueva aplicación de software, de esta manera logramos

resultados más predecibles unificando nuestro equipo con procesos comunes que mejoran la comunicación y un mejor entendimiento de todas las tareas y responsabilidades.

Principales características de **RUP**:

- Iterativo e Incremental.
- Dirigido por los Casos de Uso.
- Centrado en la Arquitectura.

Iterativo e Incremental:

Pequeños proyectos que incorporan incrementalmente nueva funcionalidad y cuyo desarrollo es una iteración.

- Obtiene un Sistema Robusto
- Reduce el Riesgo de tener un mal producto
- Reduce el Riesgo de no obtener el producto en el tiempo previsto
- Permite atacar problemas con requisitos incompletos.

Dirigido por los Casos de Uso:

Servicios que un actor requiere del sistema y le proporcionan un resultado.

- Proporcionan los Requisitos Funcionales del Sistema.
- Describen toda la funcionalidad del Sistema.

- Cambios en Requisitos de un Caso de Uso fácil detectar las clases y componentes que afectan.

Centrado en la Arquitectura:

- Casos de Uso describen la *Funcionalidad* del Sistema
- Arquitectura define la *Forma* del Sistema
- Se describe mediante *Vistas* que incorporan el 5-10% de los casos de uso más relevantes.

1.5.9 UML.

Para modelar el análisis y el diseño del software se utiliza el lenguaje UML que es el más utilizado mundialmente.

UML (Unified Modeling Language, Lenguaje Unificado de Modelación) (13), es el resultado del trabajo hecho por Grady Booch, James Rumbaugh e Ivar Jacobson. (14)

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos del sistema de un Software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

Además tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).

- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

1.5.10 Rational Rose.

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

Racional Rose domina el mercado de herramientas para el análisis, modelamiento, diseño y construcción orientada a objetos, tiene todas las características que los desarrolladores, analistas, y arquitectos exigen – soporte UML incomparable, desarrollo basado en componentes

con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada.

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 visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. (15)

1.5.11 Zend Studio.

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web en lenguaje PHP.

Este programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor.

Contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que vayamos creando. (16)

1.5.12 CASE Studio.

Es una herramienta profesional con la que se puede diseñar las bases de datos, facilitando herramientas para la creación de diagramas de relación, modelado de datos y gestión de estructuras.

Tiene soporte para trabajar con una amplia variedad de formatos de base de datos (Oracle, SQL, MySQL, PostgreSQL, Access, etc.) y permite además generar scripts SQL, aplicar procesos de retroingeniería (reverse engineering) a las bases de datos, usar plantillas de diseño personalizables y crear detallados informes en HTML y RTF.

A través de los diagramas de relación se puede tener una visión más clara del contenido y estructura de la base de datos, facilitando la gestión y mantenimiento de la misma. (17)

1.5.13 CakePHP.

CakePHP es un framework para PHP de código abierto para el rápido desarrollo de aplicaciones. Es una estructura de librerías, clases y estructuras run-time que le permite a los programadores trabajar de una manera estructurada y rápida sin la pérdida de flexibilidad. (18)

Algunas de las características de CakePHP que lo hacen nuestra mejor opción para desarrollar esta aplicación son:

- Tiene una comunidad muy activa en la cual encontrar apoyo.
- Compatibilidad con PHP 4 y PHP 5.
- CRUD integrado para la interacción de la base de datos y consultas simplificadas.
- Scaffolding para aplicaciones (AMB de prueba simple).
- Arquitectura Modelo Vista Controlador (MVC).

- Validación incorporada.
- Seguridad, sesión, y componentes que manejan peticiones.
- Listas flexibles del control de acceso.
- Desinfección de datos.
- View Caching flexible.
- Trabaja desde cualquier subdirectorio de la web, con poca o ninguna configuración de Apache.

Conclusiones.

Se ha comentado el estado actual de desarrollo de sistemas automatizados de reservación en el ámbito mundial, nacional y en la Universidad, exponiendo ejemplos de algunos sistemas automatizados de gestión de reservaciones que existen actualmente, planteándose además una propuesta de solución a la problemática que existe en la universidad.

También realizamos un análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, fundamentándose las elecciones de cada una de ellas.



Capítulo 2: Descripción de la Solución Propuesta.

Introducción.

En el presente capítulo se modela el negocio propuesto, identificándose los actores, trabajadores y los casos de uso correspondientes. Se plantean los requerimientos funcionales y no funcionales de la aplicación a desarrollar y se describe la solución propuesta en términos de casos de uso del sistema.

2.1 Modelo del negocio propuesto.

El modelo del negocio se realiza con el propósito de comprender las características y actividades que se llevan a cabo en el contexto a automatizar. En él, se modelan los clientes y trabajadores del negocio y su interacción con los procesos y objetos que lo componen.

2.1.1 Actores / Trabajadores del negocio.

Un **actor** del negocio es algo o alguien fuera del sistema que interactúa con este, o sea, son todas aquellas personas o sistemas que obtienen un resultado de valor de uno o varios procesos del negocio.

Los **trabajadores** del negocio son aquellas personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos, pero no obtienen ningún resultado de valor.

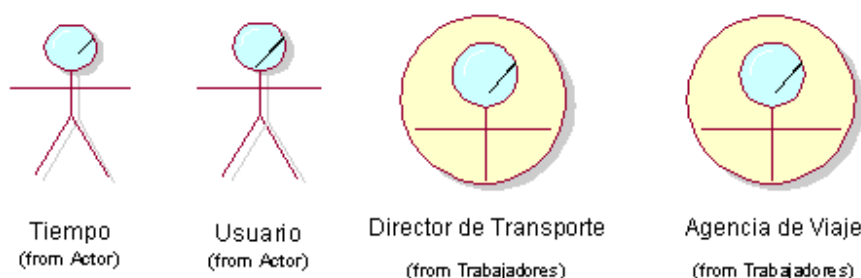


Figura 2. 1 Actores/ Trabajadores del Negocio.

Los actores y trabajadores del negocio propuesto son los siguientes:

Tabla 1. Actores y trabajadores del negocio propuesto.

Actores del Negocio	Justificación
Tiempo	Es un actor ficticio, que ejecuta las tareas programadas.
Usuario	Son todas aquellas personas que se benefician con el negocio, ya sea para viajar o para obtener información y reportes de este.
Trabajadores del Negocio	Justificación
Director de Transporte	Es el encargado de comenzar a gestionar el pase masivo, obtener la cantidad total de viajeros y así solicitar a la agencia los transportes necesarios para luego realizar la distribución.
Agencia de Viajes	Es la agencia a la cual se le compran los pasajes precisos para los viajeros

2.2 Descripción del Proceso del Negocio.

2.2.1 Diagrama de CU del Negocio.

Llegado el momento de realizar el Pase Masivo en la UCI, comienza la Distribución del Pase Masivo, donde se pueden diferenciar dos procesos principales: la Gestión del Pase Masivo, que es el proceso donde se reservan los pasajes, y la Distribución, que es donde se ubican los viajeros por medio de transporte.

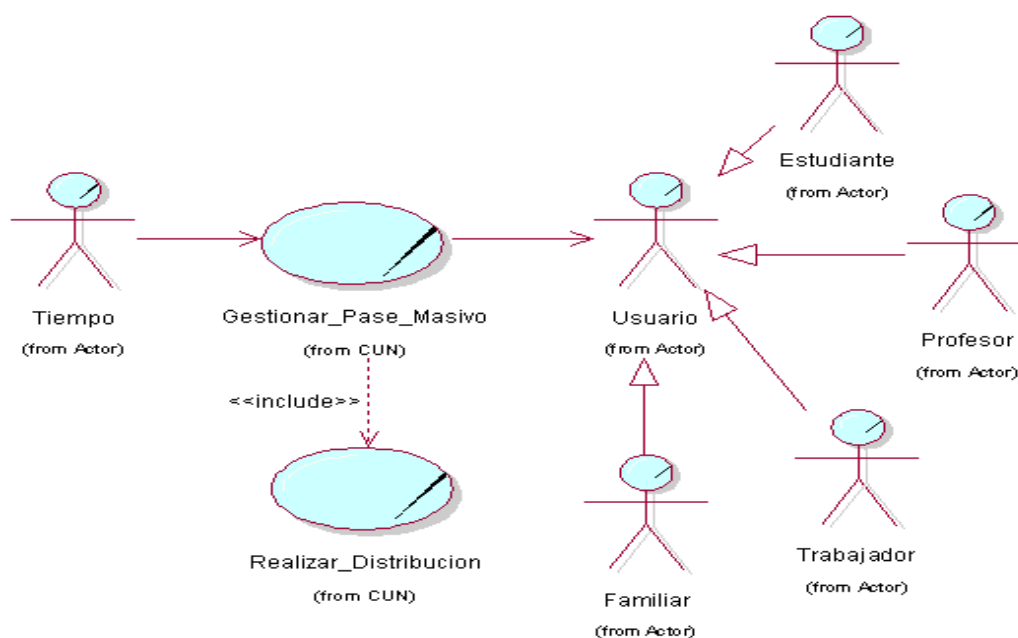


Figura 2. 2 Diagrama de Casos de Uso del Negocio.

2.2.2 Descripción de los CUN.

Gestión del Pase Masivo.

Cuando se necesita gestionar un Pase Masivo la Dirección de Transporte UCI solicita al Departamento de Recursos Humanos los datos de las personas a viajar de la UCI. Con esta lista

se obtiene la cantidad de viajeros en total y con este dato la Dirección de Transporte tramita la compra de los pasajes con la Agencia de Viajes que corresponda. Dicha Agencia envía un listado de asientos que son los que corresponden a los pasajes comprados, también vienen los datos referentes a nombre del Transporte, Medio de Transporte, destino, fecha y hora de salida de cada Transporte, en un Pase Masivo pueden existir varios Transportes que tengan diferente fecha de salida.

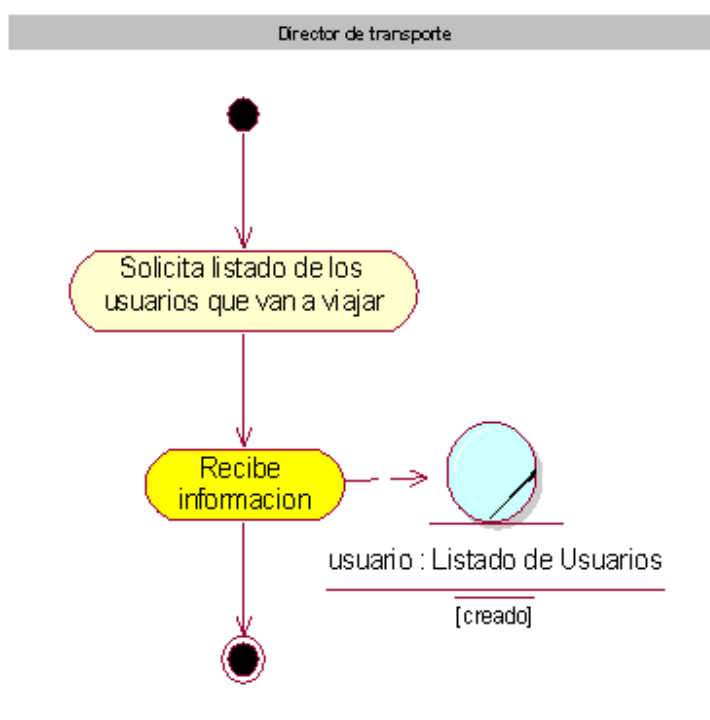


Figura 2. 3 Diagrama de Actividad Gestión pase Masivo.

Realizar Distribución.

Una vez obtenidos todos los datos del transporte la Dirección de Transporte comienza la Distribución según los criterios que deba seguir, cada Pase Masivo cuenta con sus restricciones.

En caso de que algún viajero desee cambiar su destino debe notificarlo a la Dirección de Transporte con antelación y esta hace el cambio de ser posible. Luego de terminada la Distribución se procede a imprimir los Boletines que serán entregados a los viajeros.

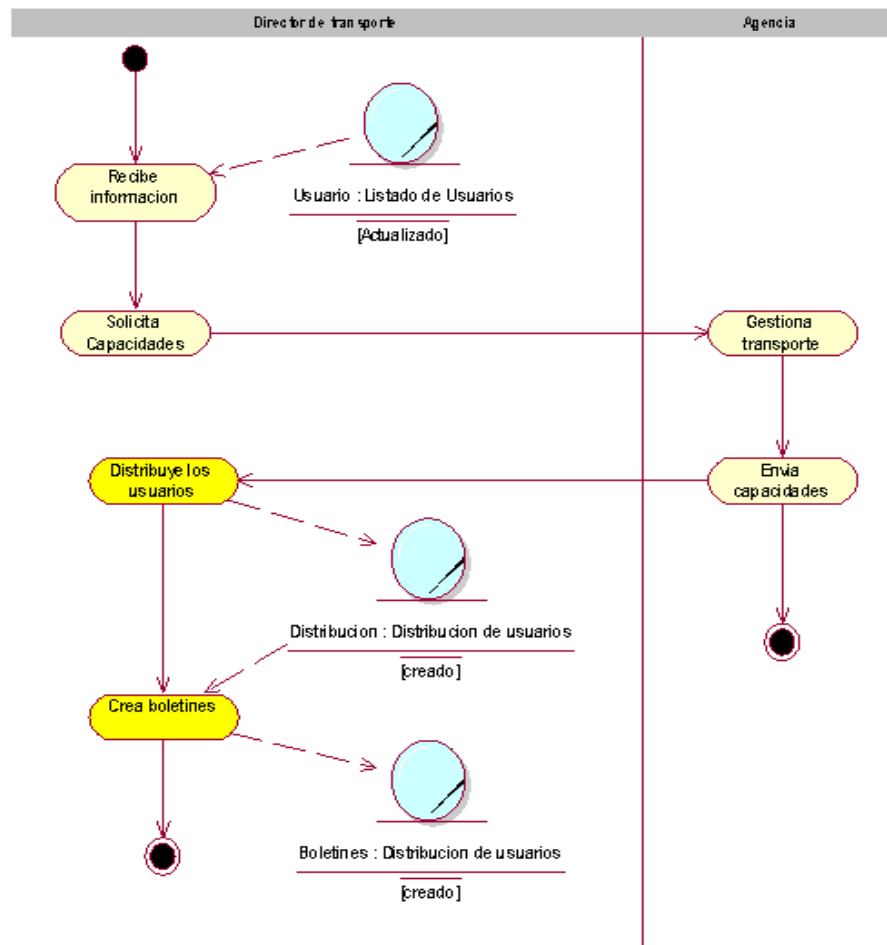


Figura 2. 4 Diagrama de Actividad Realizar Distribución.

2.3 Diagrama del Modelo de Objetos del Negocio.

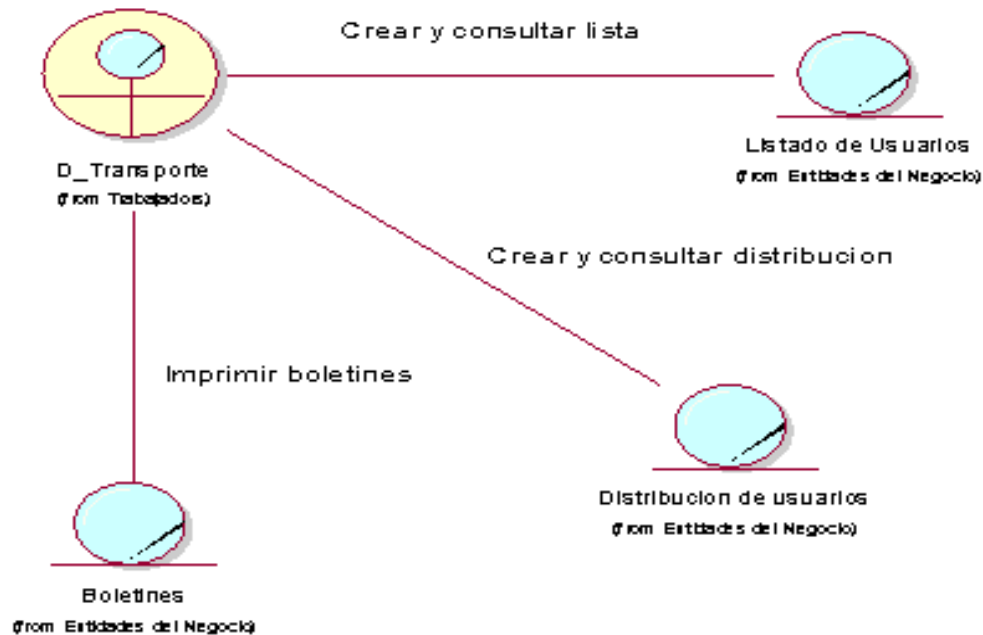


Figura 2. 5 Modelo de Objetos del Software.

2.4 Especificación de los requerimientos de Software.

2.4.1 Requerimientos Funcionales.

RF1- Autenticar Usuario.

- Introducir nombre de usuario y contraseña de Dominio UCI.
- Validar los datos introducidos.
- Otorgar los accesos al usuario.

RF2- Reservación de Viajes.

- Seleccionar la opción de reservar viaje de ida o regreso.

- Verificar el plazo establecido para reservar.
- Elegir provincia y municipio al que desea viajar.
- Adicionar a la base de datos la nueva reservación.
- Mostrar mensaje de confirmación.

RF3- Modificar una Reservación.

- Mostrar las reservaciones hechas por el usuario.
- Verificar el plazo establecido para modificar.
- Modificar los parámetros de interés.
- Actualizar la base de datos.
- Mostrar mensaje de confirmación.

RF4-Cancelar una reservación.

- Mostrar las reservaciones hechas por el usuario.
- Verificar el plazo establecido para cancelar.
- Marcar la reservación como cancelada en la base de datos.
- Mostrar mensaje de confirmación.

RF-5 Gestionar Solicitud de Reservación.

- Mostrar formulario a llenar.
- Guardar información.
- Permitir modificar información.
- Cancelar solicitud.

RF6- Gestionar Usuario.

- Introducir identificador (nombre, usuario UCI, # carne, # solapín)
- Buscar un usuario.

- Mostrar datos del usuario.
- Mostrar las reservaciones hechas por el usuario.
 - Modificar la reservación.
 - Cancelar la reservación.
 - Mostrar mensaje de confirmación.

RF7- Consultar Información.

- Introducir un identificador (nombre, usuario UCI, # carne, # solapín)
- Mostar datos del usuario.

RF8- Gestionar Datos Viaje.

- Introducir fecha de inicio/ fin de reservación.
- Determinar tipo de salida. (estándar o permiso)
- Determinar tipo de viaje. (ida, regreso o ambos)
- Introducir datos del viaje y la transportación.
 - Fecha salida/regreso.
 - Destino.
 - Tipo de transporte. (guagua, tren, otros)
 - Cantidad de asientos por transporte.

RF9- Gestionar Distribución.

- Determinar tipo de distribución. (provincia o municipio)
- Mostrar las capacidades y las reservaciones hechas por selección.
- Realizar distribución con la lista de reservados determinada.
- Mostrar lista hecha y dar la opción de aceptar o no.
- Permitir enviar un correo al momento o en otro momento determinado a los usuarios con los datos del transporte en que viajará.

RF10- Autorizar Solicitud.

- Mostar lista de Solicitudes de reservaciones hechas por los jefes de áreas.
- Permitir seleccionar aquellas que son aceptadas.
- Enviar un correo informando la decisión tomada con respecto a la solicitud.

RF11- Gestionar Boletines.

- Crear nuevo estilo.
- Guardar estilo.
- Cargar estilo.
- Eliminar estilo.
- Generar Documento.
- Seleccionar Distribución a Imprimir.
- Imprimir selección.

2.4.2 Requerimientos No Funcionales.

- Apariencia o Interfaz Externa.
 - Diseño encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.
 - El sistema debe tener una interfaz sencilla, intuitiva, amigable y mantener el formato de las páginas de los otros módulos. En general, fácil de usar y agradable a la vista del usuario.
- Usabilidad.
 - El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

- Portabilidad.
 - Permitir que el sistema se ejecute sobre el Sistema Operativo Linux, Windows 98 o superior.
- Rendimiento.
 - Estará implementado sobre una tecnología Web, facilitando su uso a través de la red.
 - Tiempo de respuesta rápido. El sistema debe tener un tiempo de respuesta rápido ante cualquier solicitud del usuario.
- Seguridad.
 - Establecer los distintos roles que establezcan las acciones que pueden realizar los usuarios.
 - Transmisión de datos por la red a través de un protocolo seguro.
 - Verificación sobre acciones irreversibles (por ejemplo las eliminaciones).
 - Garantizar que la información sea modificada y vista únicamente por quien tenga permiso para esto.
- Confiabilidad.
 - La información manejada por el sistema estará protegida de acceso no autorizado.
 - Validar las entradas de los usuarios.
 - Mantenimiento. El sistema debe estar bien documentado, de forma tal que el tiempo de mantenimiento sea mínimo.

- Preciso en la información. El sistema debe ser preciso en la información que le suministra al usuario para evitar cualquier tipo de error.
- Integridad.
 - La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes. Deberán existir mecanismos de chequeo de integridad.
- Disponibilidad.
 - Los usuarios autorizados tendrán acceso a la información en todo momento.
 - Deberá existir una estrategia de replicación que permita, de manera transparente para el usuario final, balancear la carga de acceso entre múltiples servidores aumentando los tiempos de respuesta y facilitar la recuperación inmediata del sistema si falla uno de ellos.
 - Disponible todo el tiempo. El sistema debe estar disponible todo el tiempo para trabajar de forma tal que se pueda acceder las 24 h.
- Funcionalidad.
 - Mínima cantidad de páginas para ejecutar todas las funciones posibles (preferentemente que estén relacionadas).

2.5 Descripción de los procesos a automatizar.

Con vista a resolver el problema que se presenta en la Reservación de Pase Masivo en la UCI se propuso el desarrollo de un sistema que automatizara este proceso.

Para ello dicho sistema sería capaz de permitir que los usuarios del sistema reserven los viajes de ida y regreso hacia sus provincias, que los Jefes de área puedan hacer solicitudes de reservación para sus trabajadores; además debe permitir también al trabajador que lo operaría (Director de Transporte) obtener los datos de las personas a viajar tales como Nombre(s) y Apellido(s), #Carné de Identidad, destino (Municipio), entre otros. Asimismo debe permitir al trabajador del sistema obtener reportes sobre los viajeros en cuanto a cantidad de estos por provincia y municipio o ambos, por tipo, por facultad, grupo y permitir hacer una búsqueda de viajeros y cambiar los datos de las reservaciones hechas de ser necesario. Luego debe permitirle entrar los datos correspondientes a los transportes disponibles, Nombre del transporte, Cantidad de medios que lo componen y destino (Provincia), y de sus medios la cantidad de asientos y cuales son, su Nombre y destino (Municipio). Una vez hecho esto el sistema debe aceptar los requisitos dados por el trabajador para realizar la Distribución según los mismos.

Por otra parte debe ser posible cambiar los datos de la Distribución por el Trabajador una vez terminada esta, obtener reportes sobre la misma y añadir nuevos viajeros en cualquier momento. Como punto final de la aplicación debe permitir generar los Boletines e imprimirlos, esta también es una tarea del Trabajador del sistema.

2.6 Modelo del Sistema.

En este epígrafe se muestra el modelo del sistema. Para ello se identifican los actores del sistema y se especifican los casos de uso del sistema.

2.6.1 Actores del Sistema.

Los actores de un sistema son agentes externos, es decir, aquellas personas o sistemas que interactúan con él. En la siguiente figura se puede observar la representación UML de los actores del sistema propuesto.

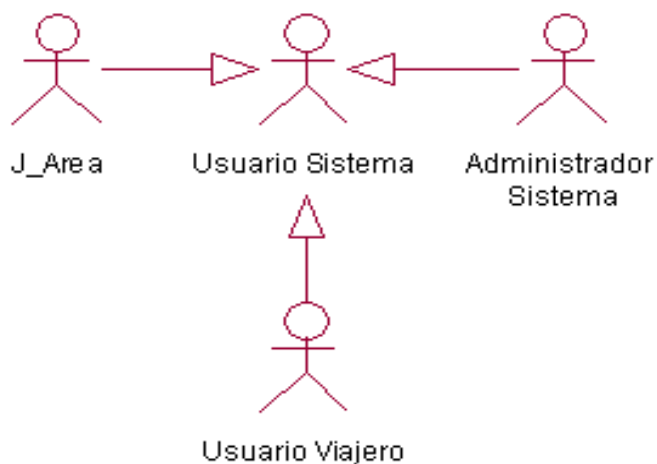


Figura 2. 6 Actores del Sistema.

La siguiente tabla define los actores del sistema y su justificación.

Tabla 2. Descripción de los actores del sistema.

Actor	Descripción
Usuario Sistema	Es la generalización que se establece entre los Usuarios Viajeros, jefes de área y administrador del sistema, puesto que todos deben autenticarse en el sistema y todos pueden consultar información.
Usuario Viajero	Son todos aquellos usuarios que pueden acceder al caso de uso gestionar reservación.(estudiantes, profesores, administrador del sistema)
Administrador del Sistema	Persona encargada de usar el sistema para Gestionar las Distribuciones, gestionar datos de los viajes, Autorizar solicitudes, generar boletines e imprimirlos.
Jefe de Área	Es la persona que le corresponde enviar la solicitud de viaje al director de transporte, de algún trabajador que desee viajar en la transportación de la universidad.

2.6.1 Diagrama de Casos de Uso del Sistema.

Los casos de uso son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación, lo cual se muestra en el siguiente diagrama.

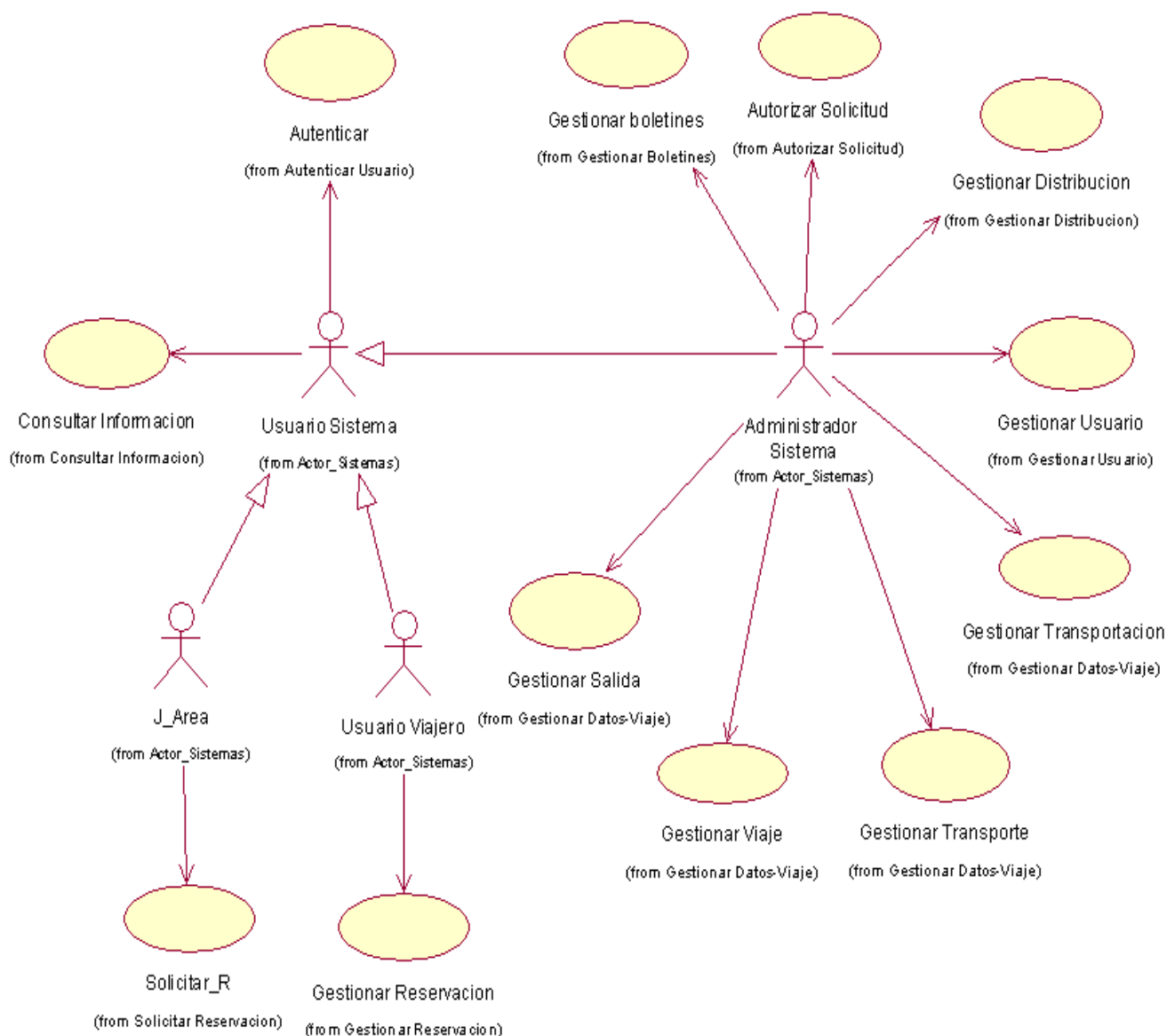


Figura 2. 7 Diagrama de Casos de Uso del Sistema.

2.6.2 Descripción de los Casos de Uso del Sistema.

Tabla 3. Descripción CUS ‘ Autenticar Usuario’

Caso de Uso:	Autenticar Usuario.	
Propósito:	Conocer en todo momento quien accede al sistema.	
Actores:	Usuario	
Resumen:	Inicia cuando el usuario ingresa sus datos en el sistema (Nombre de usuario y contraseña del dominio UCI). Se le informa si los datos introducidos son correctos o no, y si esta en algún grupo restrictivo; y termina el CU.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
2. Introduce los datos y acepta la operación.	1. Muestra los controles que permiten al usuario introducir su nombre de usuario y contraseña de dominio UCI. 2.1 Valida la identidad del usuario. 2.2 El sistema muestra la pagina o las paginas a las que el usuario tiene acceso.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	2.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.	
Poscondiciones:		

Tabla 4. Descripción CUS ‘ Gestionar Reservación’

Caso de Uso:	Gestionar Reservación.	
Propósito:	Permitir al usuario registrar, modificar y cancelar reservaciones.	
Actores:	Usuario Viajero.	
Resumen:	El caso de uso se inicia cuando el usuario viajero accede al sistema y solicita efectuar las operaciones de crear, modificar o eliminar un viaje. Finaliza una vez que el usuario viajero realiza una de las operaciones.	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Gestionar Reservación.	1.1 Muestra las opciones para gestionar las reservaciones: crear, modificar y eliminar.	
2. Selecciona una de las opciones relacionadas con la gestión de reservación: a) Solicita crear una reservación. Ver sección Crear Reservación. b) Solicita modificar una reservación. Ver sección Modificar Reservación. c) Solicita eliminar una reservación. Ver sección Eliminar Reservación.		
Sección Crear Reservación		
Flujo Normal de Eventos		
	2.1 Muestra los datos disponibles sobre el usuario y los formularios para entrar los datos referentes a la	

	reservación.(provincia, municipio)
3. Introduce los datos.	3.1 Registra la reservación y muestra un mensaje de confirmación de la operación.
Sección Modificar Reservación	
Flujo Normal de Eventos	
	2.1 Muestra todas las reservaciones hechas por el usuario.
3. Escoge la reservación que desea modificar.	3.1 Muestra los campos que pueden ser modificados.
4. Introduce los datos.	4.1 Guarda la nueva información modificando la reservación y muestra un mensaje de confirmación.
Sección Eliminar Reservación	
	2.1 Muestra todas las reservaciones hechas por el usuario.
3. Escoge cual desea eliminar.	3.1 Muestra un mensaje “¿Desea eliminar esta reservación?”.
4. Acepta eliminar. Ver FA1.	4.1 Elimina la reservación.
Flujo alternativo 1	
4. No acepta.	4.1 Regresa al paso 2.1.

Tabla 5. Descripción CUS ‘ Gestionar Solicitud de Reservación ’

Caso de Uso:	Gestionar Solicitud de Reservación.	
Propósito:	Permitir a un trabajador hacer una solicitud para reservar mediante el Sistema de Transporte de Reservación de Pase Masivo.	
Actores:	Jefe de Área.	
Resumen:	El trabajador hace una solicitud de reservación que genera un registro, el cual este puede modificar para poder tramitar la solicitud. Se activa el CU Autorizar Reservación.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El Jefe de área selecciona la opción de realizar una solicitud de reservación.</p> <p>2. El Jefe de área introduce la información necesaria para la solicitud de reservación.</p>	<p>1.1 El sistema muestra un formulario para completar la solicitud.</p> <p>1.2 El sistema verifica el cumplimiento de las políticas determinadas (tiempo requerido, etc.). Si está en tiempo.</p> <p>2.1 El sistema verifica que los datos introducidos sean correctos.</p> <p>2.2 El sistema registra la solicitud, lo marca como pendiente para ser autorizado, e invoca el CU Autorizar solicitud.</p>	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	2.2 El sistema emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.	

Poscondiciones:	

Tabla 6. Descripción CUS ‘ Consultar Información’

Caso de Uso:	Consultar Información.	
Propósito:	Obtener datos de los viajeros.	
Actores:	Usuario Sistema	
Resumen:	El CU inicia cuando el usuario introduce un identificador de usuario (CI, # solapín, nombre, apellido, etc.) y el sistema devuelve los datos del usuario que tenga guardados.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
2. Introduce dato (usuario dominio, CI, # solapín, nombre y apellidos).	1. Muestra el formulario a llenar para realizar la búsqueda. 2.1 Hace la búsqueda en la BD. 2.2 Si encuentra coincidencia. Muestra toda la información que tenga sobre el usuario. Ver FA1.	
Flujo Alternativo 1		
Acción del Actor	Respuesta del Sistema	
	2.2 Muestra un mensaje de error y regresa al paso 1	
Poscondiciones:		

Tabla 7. Descripción CUS ' Gestionar Salida '

Caso de Uso:	Gestionar Salida.	
Propósito	Garantizar que los datos de cada Salida se registren, se modifiquen y se eliminen del sistema.	
Actores:	Administrador	
Resumen:	El caso de uso se inicia cuando el administrador accede al sistema y solicita Gestionar Salida, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de crear, modificar o eliminar una de ellas. Finaliza una vez que el administrador realiza una de las operaciones.	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Gestionar Salida.	1.1 Muestra las opciones para gestionar salida: crear, modificar y eliminar.	
2. Selecciona una de las opciones relacionadas con la gestión de salida: a) Solicita crear una salida. Ver sección Crear Salida. b) Solicita modificar una salida. Ver sección Modificar Salida. c) Solicita eliminar una salida. Ver sección Eliminar Salida.		
Sección Crear Salida'		
Flujo Normal de Eventos		
	2.1 Muestra los formularios para entrar los datos referentes a la salida.	
3. Introduce los datos.	3.1 Valida que los datos sean los	

	correctos para la operación. Ver FA1.
	3.2 Guarda la información creando una salida.
Flujo alternativo 1	
	3.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección Modificar Salida	
Flujo Normal de Eventos	
<p>3. Escoge la salida que desea modificar.</p> <p>4. Introduce los datos que desee modificar de la salida seleccionada.</p>	<p>2.1 Muestra todas las salidas que están creadas.</p> <p>3.1 Muestra los campos que pueden ser modificados.</p> <p>4.1 Valida que los datos sean los correctos para la operación. Ver FA1.</p> <p>4.2 Guarda la nueva información modificando la salida.</p>
Flujo alternativo 1	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección Eliminar Salida	
Flujo Normal de Eventos	
	2.1 Muestra todas las salidas creadas.
3. Escoge cual desea eliminar.	3.1 Muestra un mensaje “Desea

	eliminar esta salida”.
4. Acepta eliminar. Ver FA1.	3.3 Elimina la salida.
Flujo alternativo 1	
4. No acepta eliminar la salida.	4.1 Regresa al paso 2.1.
Poscondiciones:	

Tabla 8. Descripción CUS ´ Gestionar Viaje

Caso de uso:	Gestionar Viaje.
Propósito	Garantizar que los datos de cada Viaje se registren, se modifiquen y se eliminen del sistema.
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador accede al sistema y solicita Gestionar Viaje, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de crear, modificar o eliminar una de ellas. Finaliza una vez que el administrador realiza una de las operaciones.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de Gestionar Viaje.	1.1 Muestra las opciones para gestionar viaje: crear, modificar y eliminar.
2. Selecciona una de las opciones relacionadas con la gestión de viaje: a) Solicita crear un viaje. Ver sección Crear Viaje. b) Solicita modificar un viaje. Ver sección	

<p>Modificar Viaje.</p> <p>c) Solicita eliminar un viaje. Ver sección Eliminar Viaje.</p>	
Sección Crear Viaje	
Flujo Normal de Eventos	
	2.1 Muestra las salidas disponibles para crearle viaje.
3. Selecciona la salida a la cual desea crearle un viaje.	3.1 Muestra los formularios para entrar los datos referentes al viaje.
4. Introduce los datos.	4.1 Valida que los datos sean los correctos para la operación. Ver FA1.
	4.2 Guarda la información creando un viaje.
Flujo Alternativo 1	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección Modificar Viaje	
Flujo Normal de Eventos	
	2.1 Muestra todos los viajes que están creados.
3 Escoge el viaje que desea modificar.	3.1 Muestra los campos que pueden ser modificados.
4 Introduce los datos que desee modificar del viaje seleccionado.	4.1 Valida que los datos sean los correctos para la operación. Ver

	FA1.
	4.2 Guarda la nueva información modificando el viaje.
Flujo alternativo 1	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección Eliminar Viaje	
Flujo Normal de Eventos	
	2.1 Muestra todos los viajes creados.
3. Selecciona cual desea eliminar.	3.1 Muestra un mensaje “Desea eliminar este viaje”.
4. Acepta eliminar. Ver FA1.	4.1 Elimina el viaje.
Flujo alternativo 1	
4. No acepta eliminar el viaje.	4.1 Regresa al paso 2.1.
Poscondiciones:	

Tabla 9. Descripción CUS ´ Gestionar Transportación´

Caso de uso:	Gestionar Transportación.
Propósito	Garantizar que los datos de cada Transportación se registren, se modifiquen y se eliminen del sistema.
Actores:	Administrador

Resumen:	El caso de uso se inicia cuando el administrador accede al sistema y solicita Gestionar Transportación, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de crear, modificar o eliminar una de ellas. Finaliza una vez que el administrador realiza una de las operaciones.	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Gestionar Transportación.	1.1 Muestra las opciones para gestionar transportación: crear, modificar y eliminar.	
2. Selecciona una de las opciones relacionadas con la gestión de transportación: a) Solicita crear una transportación. Ver sección Crear Transportación. b) Solicita modificar una transportación. Ver sección Modificar Transportación. c) Solicita eliminar una transportación. Ver sección Eliminar Transportación.		
Sección Crear Transportación		
Flujo Normal de Eventos		

<p>3. Selecciona el viaje al cual desea crearle la transportación.</p> <p>4. Introduce los datos.</p>	<p>2.1 Muestra todos los viajes creados.</p> <p>3.1 Muestra los formularios para entrar los datos referentes a la transportación.</p> <p>4.1 Valida que los datos sean los correctos para la operación. Ver FA1.</p> <p>4.2 Guarda la información creando una transportación.</p>
<p>Flujo alternativo 1</p>	
	<p>4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.</p>
<p>Sección Modificar Transportación</p>	
<p>Flujo Normal de Eventos</p>	
<p>3. Escoge la transportación que desea modificar.</p> <p>4. Introduce los datos que desea modificar de la transportación seleccionada.</p>	<p>2.1 Muestra todas las transportaciones creadas.</p> <p>3.1 Muestra los campos que pueden ser modificados.</p> <p>4.1 Valida que los datos sean los correctos para la operación. Ver FA1.</p> <p>4.2 Guarda la nueva información modificando la transportación.</p>
<p>Flujo alternativo 1</p>	

	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección Eliminar Salida	
Flujo Normal de Eventos	
3. Selecciona cual desea eliminar. 4. Acepta eliminar. Ver FA1.	2.1 Muestra todas las transportaciones creadas. 3.1 Muestra un mensaje “Desea eliminar esta transportación”. 4.1 Elimina la salida.
Flujo alternativo 1	
4. No acepta eliminar la transportación.	4.1 Regresa al paso 2.1
Poscondiciones:	

Tabla 10. Descripción CUS ‘ Gestionar Transporte ’

Caso de uso:	Gestionar Transporte
Propósito	Garantizar que los datos de cada Transporte se registren, se modifiquen y se eliminen del sistema.
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador accede al sistema y solicita Gestionar Transporte, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de crear, modificar o eliminar una de ellas. Finaliza una vez que el administrador realiza una de las operaciones.

Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de Gestionar Transporte. 2. Selecciona una de las opciones relacionadas con la gestión de transporte: a) Solicita crear un viaje. Ver sección Crear Viaje. b) Solicita modificar un viaje. Ver sección Modificar Viaje. c) Solicita eliminar un viaje. Ver sección Eliminar Viaje.	1.1 Muestra las opciones para gestionar transporte: crear, modificar y eliminar.
Sección Crear Transporte	
Flujo Normal de Eventos	
	2.1 Muestra las todas las transportaciones.
3. Selecciona la transportación a la cual desea crearle un transporte.	3.1 Muestra los formularios para entrar los datos referentes al transporte.
4. Introduce los datos.	4.1 Valida que los datos sean los correctos para la operación. Ver FA1.
	4.2 Guarda la información creando un transporte.
Flujo Alternativo 1	
	4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.
Sección Modificar Transporte	

Flujo Normal de Eventos	
<p>3. Selecciona el transporte que desea modificar.</p> <p>4. Introduce los datos que desea modificar del transporte seleccionado.</p>	<p>2.1 Muestra todos los transportes creados.</p> <p>4.3 Muestra los campos que pueden ser modificados.</p> <p>4.1 Valida que los datos sean los correctos para la operación. Ver FA1.</p> <p>4.2 Guarda la nueva información modificando el transporte.</p>
Flujo alternativo 1	
	<p>4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.</p>
Sección Eliminar Transporte	
Flujo Normal de Eventos	
<p>3. Selecciona cual desea eliminar.</p> <p>4. Acepta eliminar. Ver FA1.</p>	<p>2.1 Muestra todos los transportes creados.</p> <p>3.1 Muestra un mensaje “Desea eliminar este transporte”.</p> <p>4.1 Elimina el transporte.</p>
Flujo alternativo 1	
<p>4. No acepta eliminar el transporte.</p>	<p>4.1 Regresa al paso 2.1</p>
Poscondiciones:	

Tabla 11. Descripción CUS ‘ Gestionar Distribución’

Caso de Uso:	Gestionar Distribución.	
Propósito	Asignarles transporte a los usuarios que hayan reservado.	
Actores:	Administrador	
Resumen:	El caso de uso se inicia cuando el administrador accede al sistema y solicita Gestionar Distribución, a partir de aquí el sistema brinda la posibilidad de asignarles transporte a los usuarios que previamente habían reservado para un viaje.	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona Gestionar Distribución.	1.1 Muestra las opciones para gestionar distribución: Distribución inicial, por transporte o deshacer distribución.	
2. Selecciona una de las opciones relacionadas con la gestión de distribución: a) Distribución inicial. <i>Ver sección Distribución inicial.</i> b) Distribución por transporte. <i>Ver sección Distribución por transporte.</i> c) Deshacer distribución. <i>Ver sección Deshacer distribución.</i>		
Sección Distribución Inicial		
	2.1 Muestra la cantidad de reservados, las capacidades, los distribuidos y lo que faltan por distribuir por provincia. 2.2 Distribuye a los viajeros en los transportes según municipio para el cual halla reservado.	

	2.3 Muestra el mensaje “Distribución con éxito”. Ver FA1
Sección Distribución por Transporte	
<p>3. Selecciona el transporte que va llenar.</p> <p>4. Selecciona el municipio con que va a llenar el transporte.</p>	<p>2.1 Muestra todos los transportes de la provincia con su municipio de destino, las reservaciones, la capacidad total y la cantidad de distribuidos.</p> <p>3.1 Le brinda la posibilidad de llenar el transporte con el municipio que desee.</p> <p>4.1 Distribuye a las personas de ese municipio el en transporte.</p> <p>4.2 Muestra el mensaje de “Distribución con éxito”. Ver FA1.</p>
Sección Deshacer distribución	
	3.1 Deshace la distribución de todas las personas de la provincia, dejando a esta lista para una nueva distribución.
Flujo alternativo 1	
	3.2 Muestra el mensaje “La distribución ha fallado” y redirecciona a la página anterior
Poscondiciones:	

Tabla 12. Descripción CUS ‘ Gestionar Usuario ‘

Caso de Uso:	Gestionar usuario.	
Propósito	Permitir al Administrador del sistema hacer búsqueda de usuarios, además de crear, modificar y eliminar una reservación.	
Actores:	Administrador.	
Resumen:	El caso de uso se inicia cuando el administrador accede al sistema y solicita Gestionar Usuario, a partir de aquí el sistema brinda la posibilidad de efectuar las operaciones de buscar, crear, modificar o eliminar reservaciones. Finaliza una vez que el administrador realiza una de las operaciones.	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Gestionar Usuario.	1.1 Muestra las opciones para gestionar usuario: buscar usuarios; crear, modificar y eliminar reservaciones.	
2. Selecciona una de las opciones relacionadas con la gestión de usuario: a) Solicita buscar algún usuario a partir de datos previos. <i>Ver sección Buscar Usuario.</i> b) Solicita crear una reservación. <i>Ver sección Crear Reservación.</i> c) Solicita modificar una reservación. <i>Ver sección Modificar Reservación.</i> d) Solicita eliminar una reservación. <i>Ver sección Eliminar Reservación.</i>		
Sección Buscar usuario.		
Flujo Normal de Eventos		

<p>3. Introduce dato (usuario dominio, CI, # solapín, nombre y apellidos).</p>	<p>2.1 Muestra formulario a llenar para realizar la búsqueda. 3.1 Hace la búsqueda en la BD. 3.2 Si encuentra coincidencia. Muestra toda la información que tenga sobre el usuario. Ver FA1.</p>
<p>Sección Crear Reservación</p>	
<p style="text-align: center;">Flujo Normal de Eventos</p>	
<p>3. Introduce los datos.</p>	<p>2.1 Muestra los datos disponibles sobre el usuario y los formularios para entrar los datos referentes a la reservación. (provincia, municipio) 3.1 Verifica que los datos sean correctos. Ver FA1 3.2 Registra la reservación y muestra un mensaje de confirmación de la operación.</p>
<p>Sección Modificar Reservación</p>	
<p style="text-align: center;">Flujo Normal de Eventos</p>	

<p>3. Escoge la reservación que desea modificar.</p> <p>4. Introduce los datos que desea modificar de la reservación seleccionada.</p>	<p>2.1 Muestra todas las reservaciones hechas por el usuario.</p> <p>3.1 Muestra los campos que pueden ser modificados.</p> <p>4.1 Verifica que los datos introducidos sean correctos. Ver FA1</p> <p>4.2 Guarda la nueva información modificando la reservación.</p>
Sección Eliminar Salida	
Flujo Normal de Eventos	
<p>3. Escoge cual desea eliminar.</p> <p>4. Acepta eliminar. Ver FA2.</p>	<p>2.1 Muestra todas las reservaciones hechas por el usuario.</p> <p>3.1 Muestra un mensaje “Desea eliminar esta reservación”.</p> <p>4.3 Elimina la reservación.</p>
Flujo alternativo 1	
	<p>4.2 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos.</p>
Flujo alternativo 2	
<p>4. No acepta.</p>	<p>4.1 Regresa al paso 2.1.</p>
<p>Poscondiciones:</p>	

Tabla 13. Descripción CUS ´ Gestionar Boletines ´

Caso de Uso:	Gestionar Boletines	
Propósito:	Lograr un diseño para los boletines de viaje e imprimir boletines.	
Actores:	Administrador.	
Resumen:	El caso de uso se inicia cuando el administrador accede al sistema y solicita Gestionar Boletines, a partir de aquí el sistema brinda la posibilidad de cargar estilos de boletines ya creados, seleccionar la distribución que desea imprimir, guardar en forma de documento protegido y luego imprimir en el momento que se necesite.	
Precondiciones:	Distribución de viajeros.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción de Estilo de Boletín.	1.1 Muestra una lista con estilos de boletines ya creados.	
2. Selecciona el estilo que desea utilizar.	2.1 Muestra el listado de distribuciones previamente hechas.	
3. Selecciona la distribución que desea imprimir.	3.1 Guarda en forma de documento protegido la selección.	
4. Selecciona la opción de Imprimir.	3.2 Muestra los documentos que se encuentran guardados.	
5. Selecciona el documento que desea imprimir.	5.1 Imprime el documento.	
Poscondiciones:		

Conclusiones.

En el presente capítulo se ha desarrollado la propuesta de solución, obteniéndose los casos de uso del sistema y las funciones que debe tener el mismo partiendo de un análisis profundo de los procesos del negocio. Basándose en los requerimientos y funciones que han sido considerados a lo largo del capítulo se comienza a elaborar la solución al problema. Además se ha descrito la forma en que el sistema dará solución al problema en cuestión.



Capítulo 3: Construcción de la Solución Propuesta.

Introducción.

En el presente capítulo se modelan los artefactos que se recogen a partir del flujo trabajo de diseño e implementación. Utilizando las extensiones de UML para Web, se definen los diagramas de clases del diseño, se definen los diagramas de clases persistentes, el modelo de datos, el diagrama de despliegue y el diagrama de componentes. Además se explican los principios de diseño seguidos para la confección de la interfaz y el tratamiento que se le da a los errores que puedan surgir.

3.1 Patrones.

“En la terminología de objetos, el patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas. (...) Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software. Todo lo contrario: intentan codificar el conocimiento, las expresiones y los principios ya existentes: cuanto más trillados y generalizados, tanto mejor. (19)

3.1.1 Patrón de arquitectura.

Se usó el patrón de arquitectura de software MVC (Modelo Vista Controlador) cuyo propósito es crear un contrato para manejar la persistencia de una información ocultando la implementación de esta persistencia. Este patrón es útil para abstraer y encapsular los accesos, gestionar las conexiones a la base de datos y obtener los datos almacenados. Para ello se utilizó el framework de php, CakePHP.

Este patrón de arquitectura separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos; donde, la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos y el modelo es el modelo de datos.

3.2 Modelo de Diseño.

El modelo de diseño es un modelo de objeto que describe la realización física de los casos de uso, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tiene impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación. (12)

3.2.1 Diagramas de clases del diseño.

El diagrama de clases del diseño para Web, difiere un poco del resto de las aplicaciones que se construyen, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir: modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente.

La forma tradicional de modelar clases no es aplicable a la hora de diseñar una aplicación Web, es por esto que Rational creó una extensión para UML que se adapta a la arquitectura de

este tipo de sistemas. Donde se modelan las páginas, sus enlaces y todo el contenido dinámico de crear las páginas del lado del servidor y el contenido dinámico en el lado del cliente; siendo estos los artefactos necesarios que se modelan para que el desarrollador los implemente luego y obtener así el producto final.

A continuación se muestran los diagramas de clases del diseño para la realización de los casos de uso del sistema.

Diagramas de Clases del Diseño.

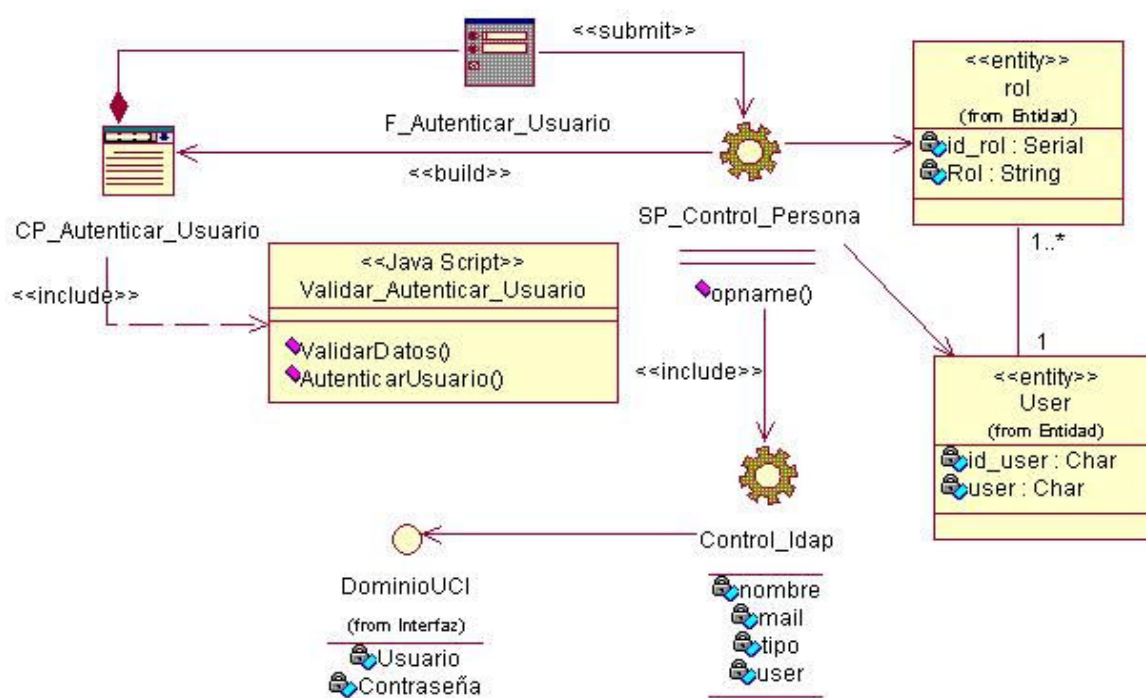


Figura 3. 1 Diagrama de Clases del Diseño " Autenticar Usuario"

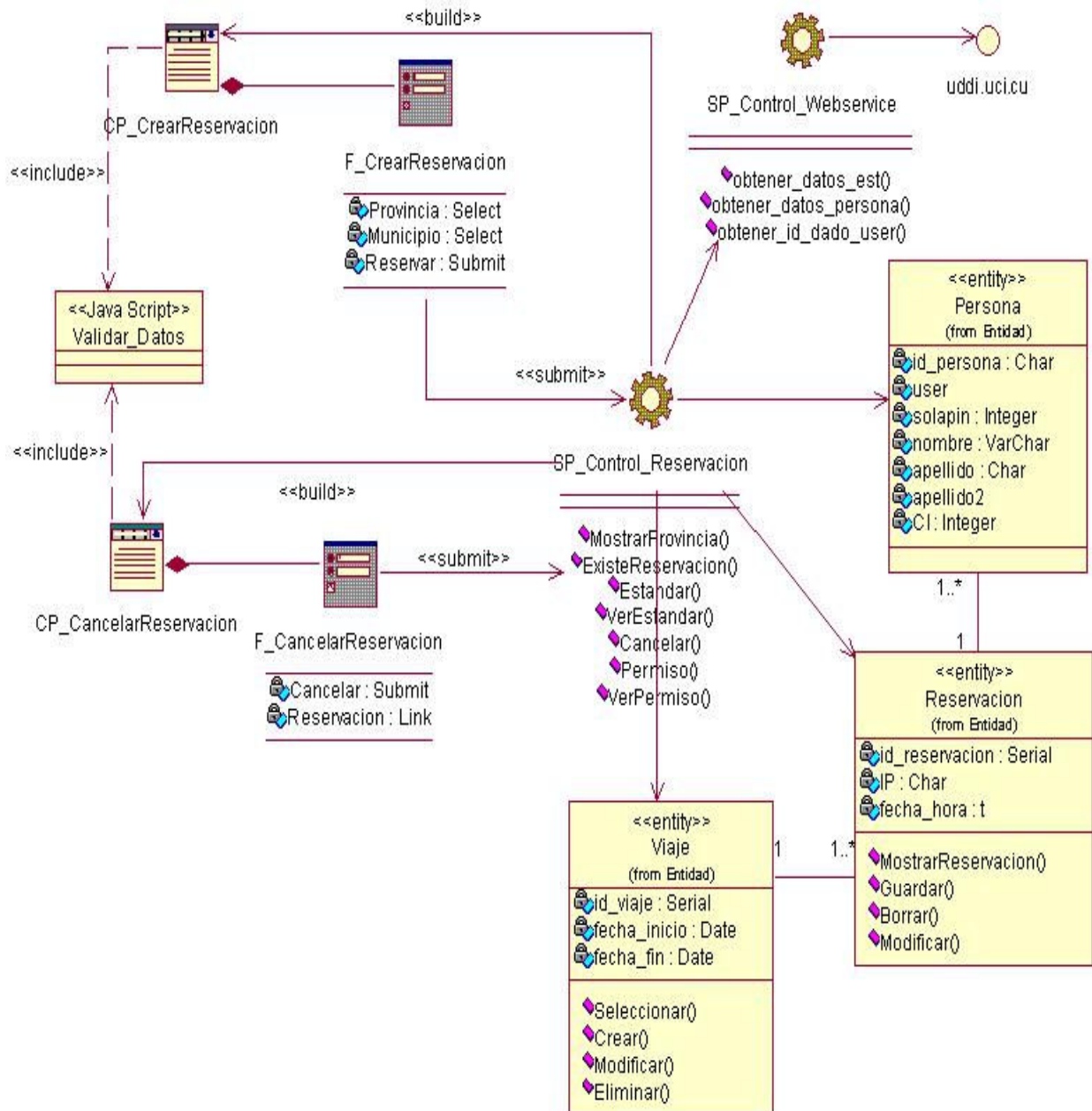


Figura 3. 2 Diagrama de Clases del Diseño " Gestionar Reservación Estándar"

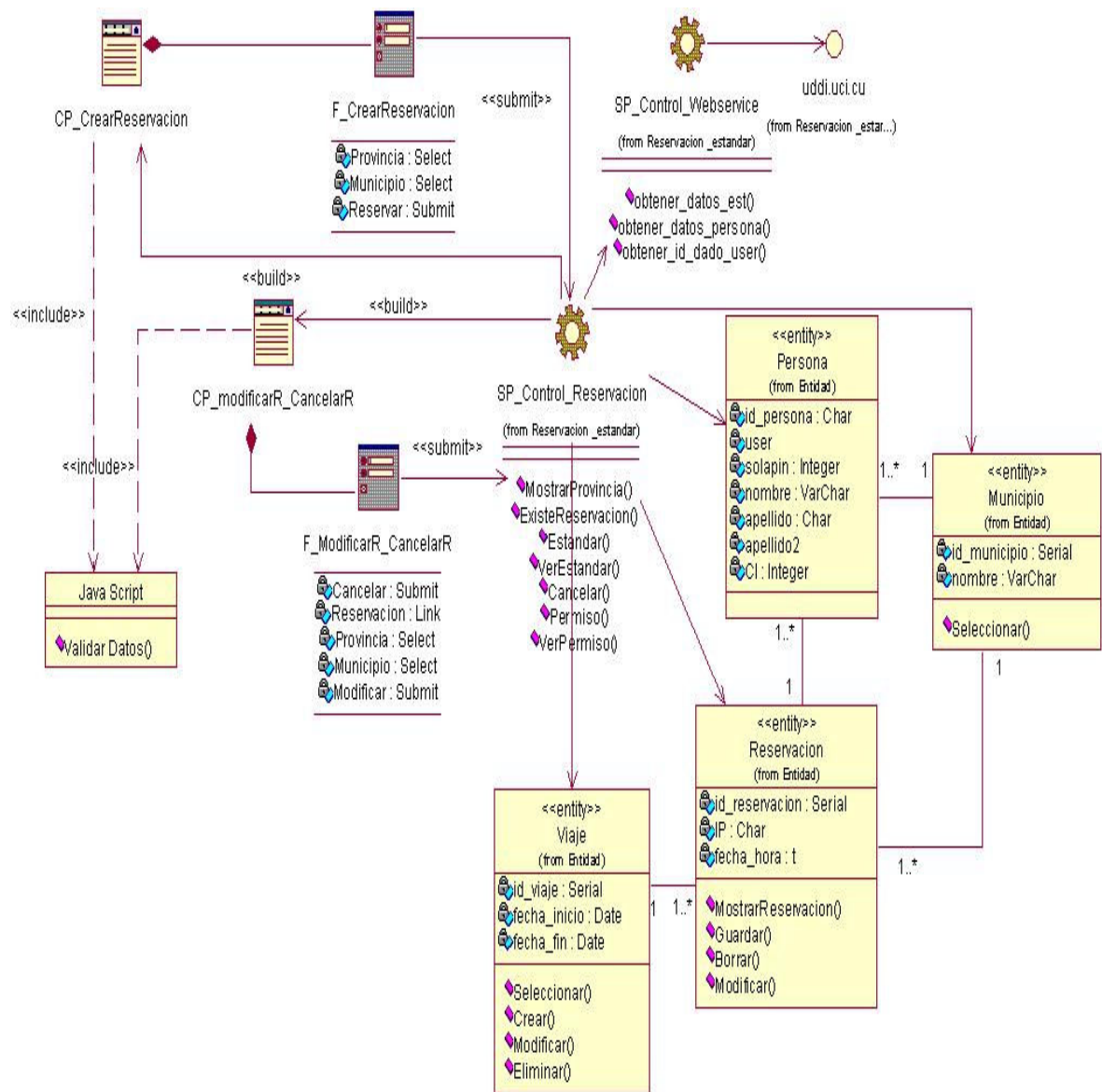


Figura 3. 3 Diagrama de Clase del Diseño "Gestionar Reservación con Permiso"

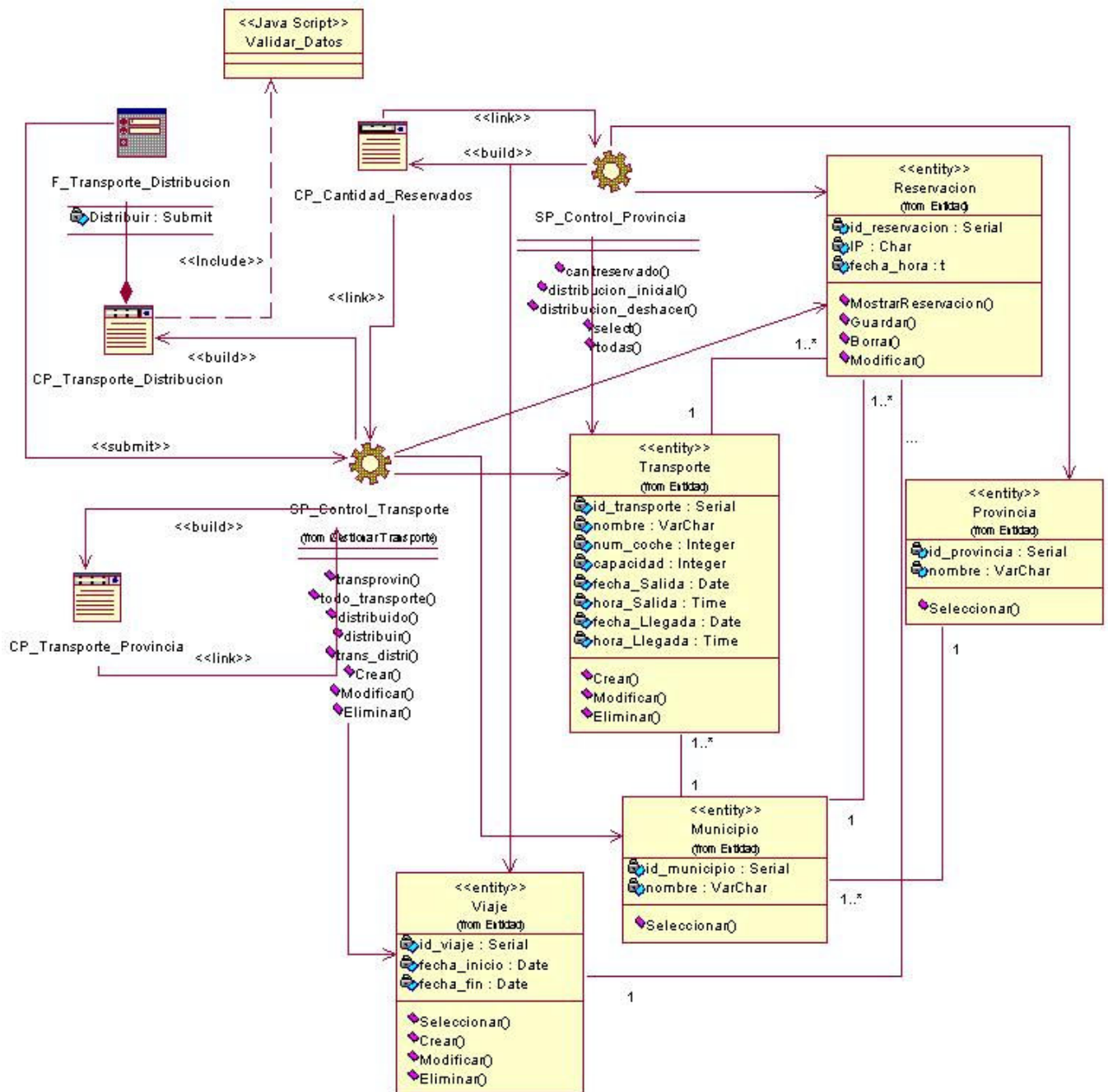


Figura 3. 4 Diagrama de Clase del Diseño "Gestionar Distribución"

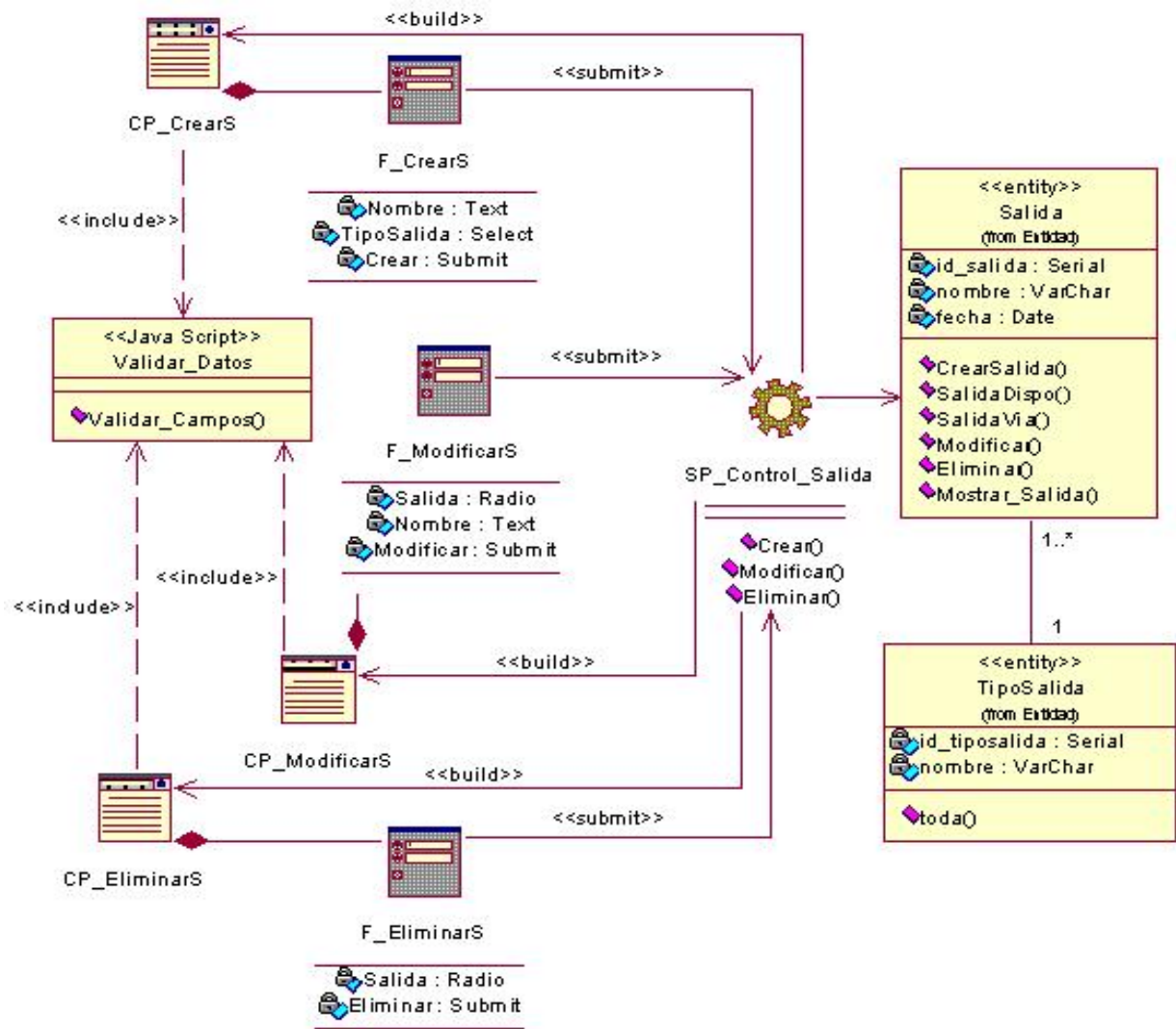


Figura 3. 5 Diagrama de Clase del Diseño "Gestionar Salida"

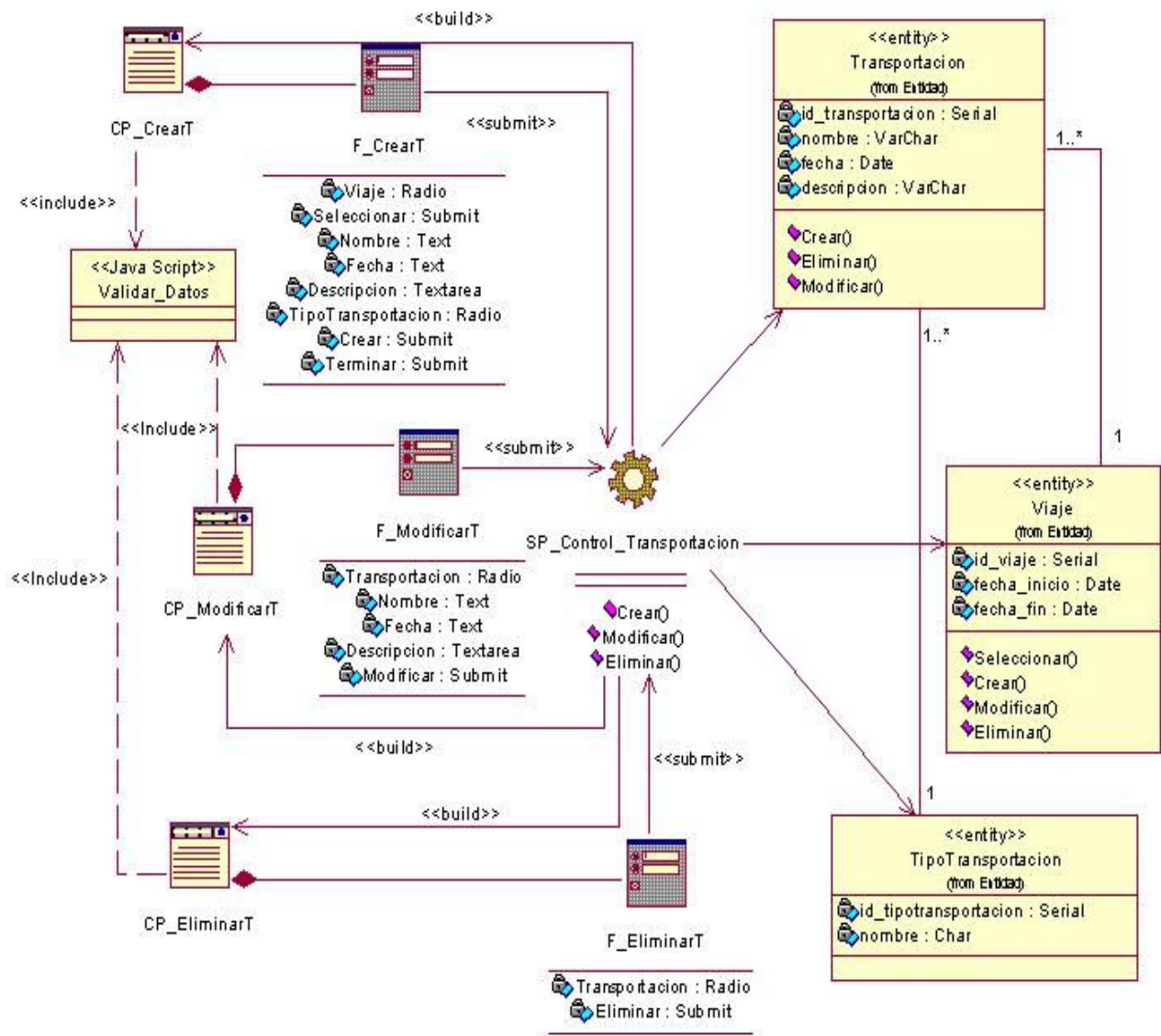


Figura 3. 6 Diagrama de Clase Diseño "Gestionar Transportación"

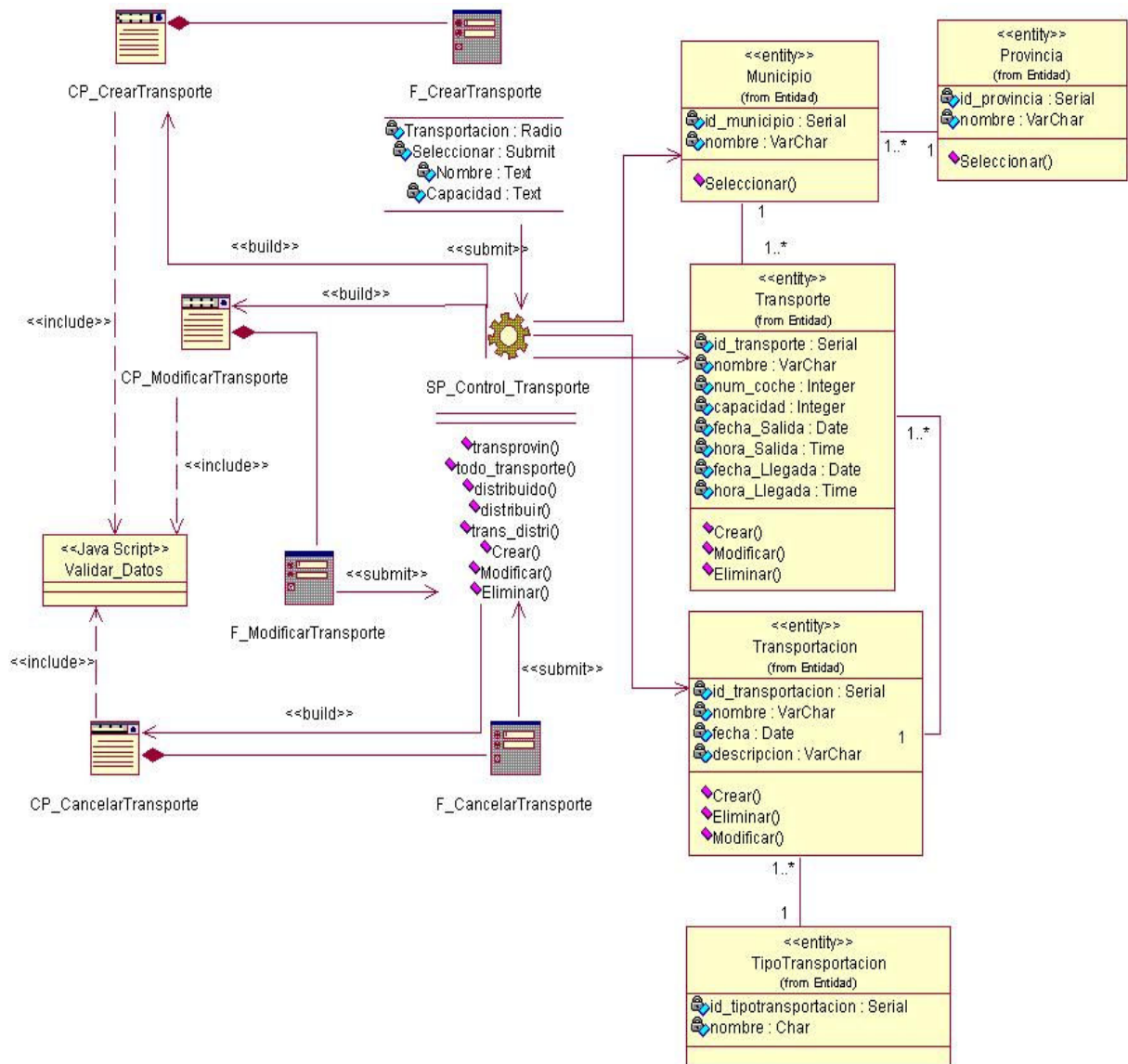


Figura 3. 7 Diagrama de Clase del Diseño "Gestionar Transporte"

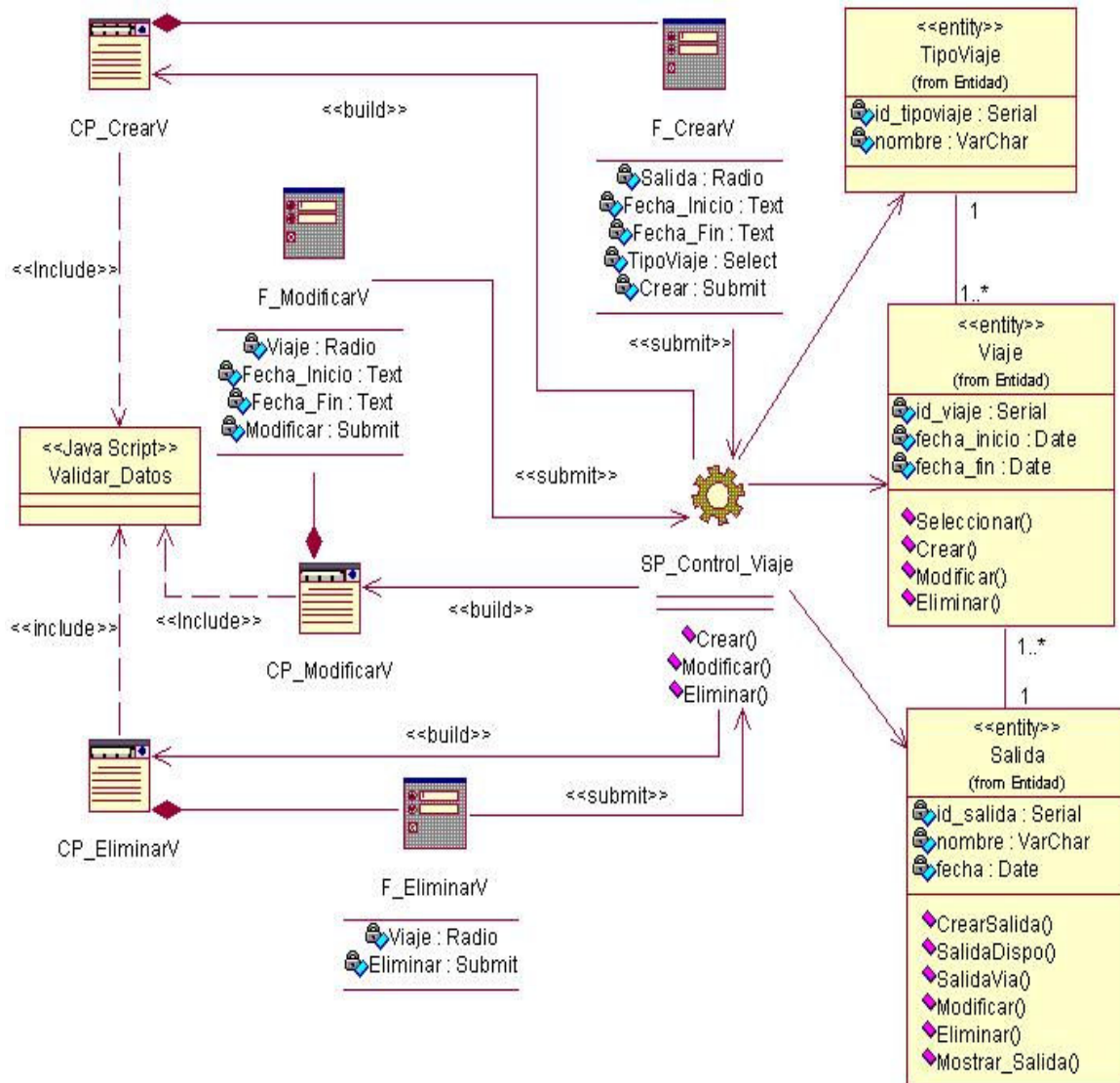


Figura 3. 8 Diagrama de Clase del Diseño "Gestionar Viaje"

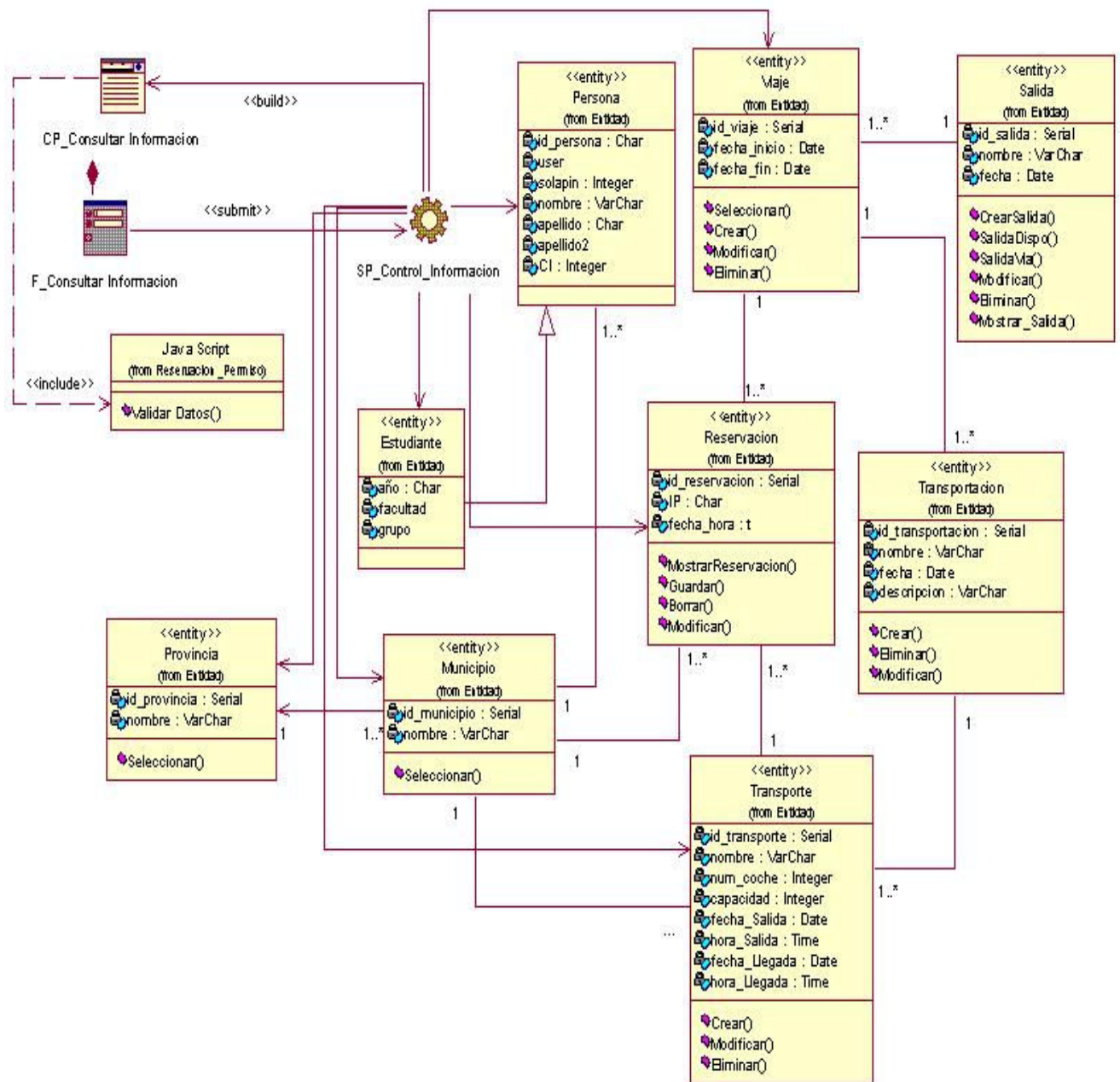


Figura 3. 9 Diagrama de Clase del Diseño "Consultar Información"

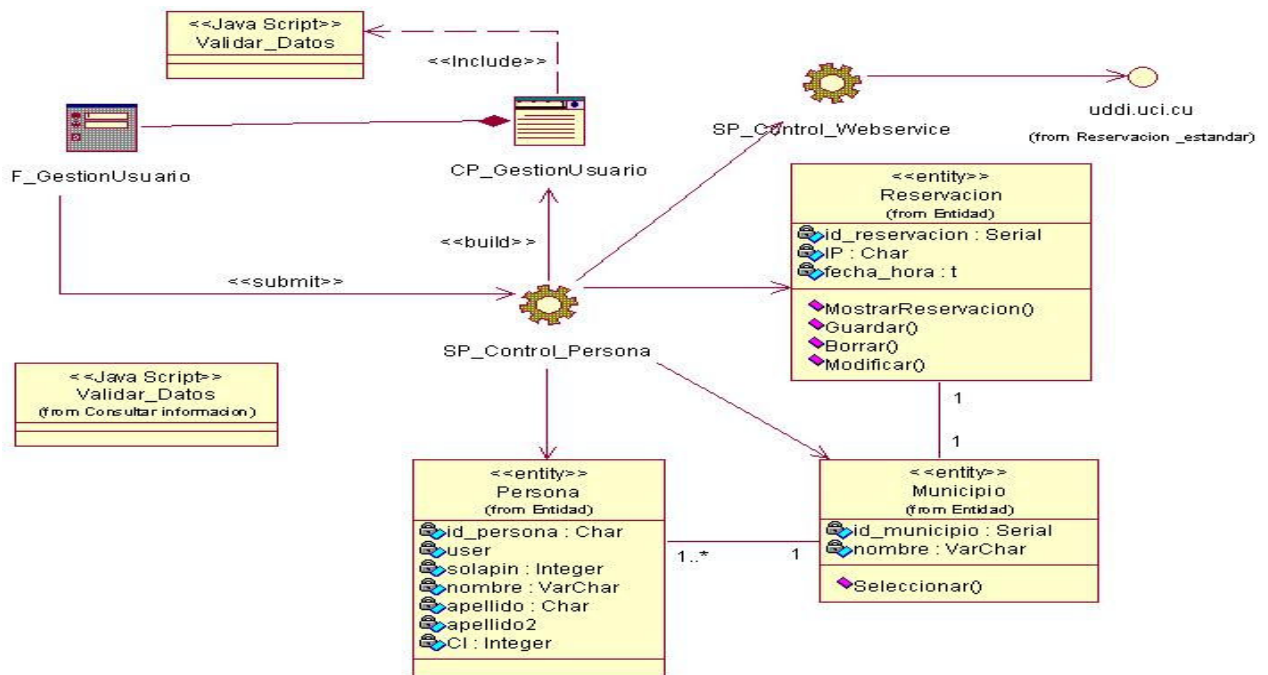


Figura 3. 10 Diagrama de Clase del Diseño "Gestionar Usuario"

3.2.2 Diagramas de secuencia.

Los diagramas de secuencia son diagramas de interacción que hacen énfasis en la ordenación temporal de los mensajes. (Ver Anexo I. Diagramas de secuencia.)

3.3 Diseño de la base de datos.

Para diseñar la base de datos del sistema, se utiliza el diagrama de clases persistentes y el modelo de datos, que están basados en la modelación de las clases del epígrafe anterior.

3.3.1 Diagrama de clases persistentes.

Las clases persistentes son las clases que necesitan ser capaz de guardar su estado en un medio permanente, la necesidad de guardar su estado esta dado por el almacenamiento físico

permanente de la información de la clase, para la copia de seguridad en caso de caída del sistema, o para el intercambio de información. Las clases que representan los datos que se obtienen y almacenan durante los procesos de la aplicación, son las que se pueden modelar a través de un diagrama de clases persistentes.

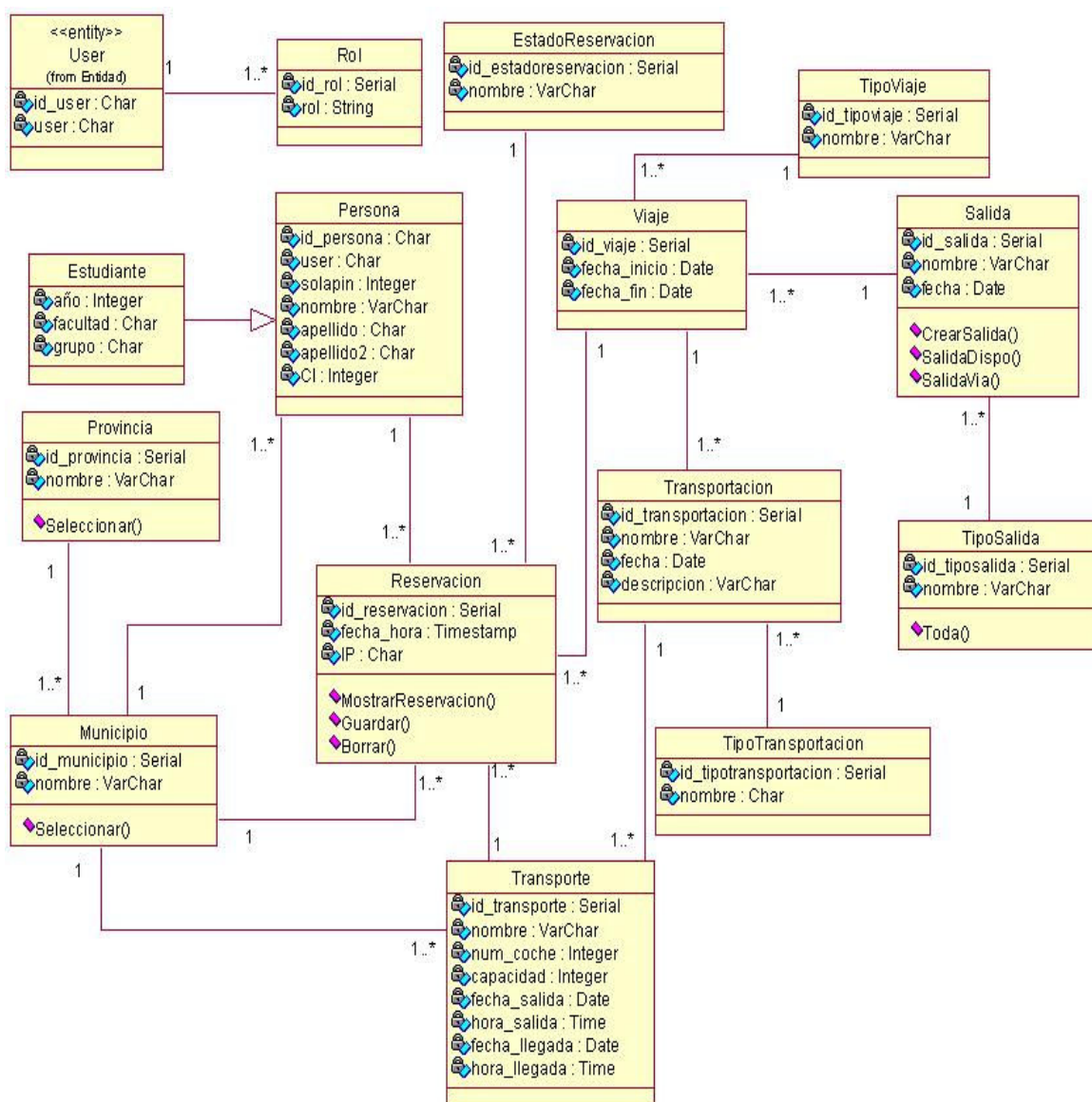


Figura 3. 11 Diagrama de Clases Persistentes.

3.3.1 Modelo de datos.

El modelo de los datos describe la representación lógica y física de datos persistentes en el sistema. También incluye cualquier conducta definida en la base de datos.

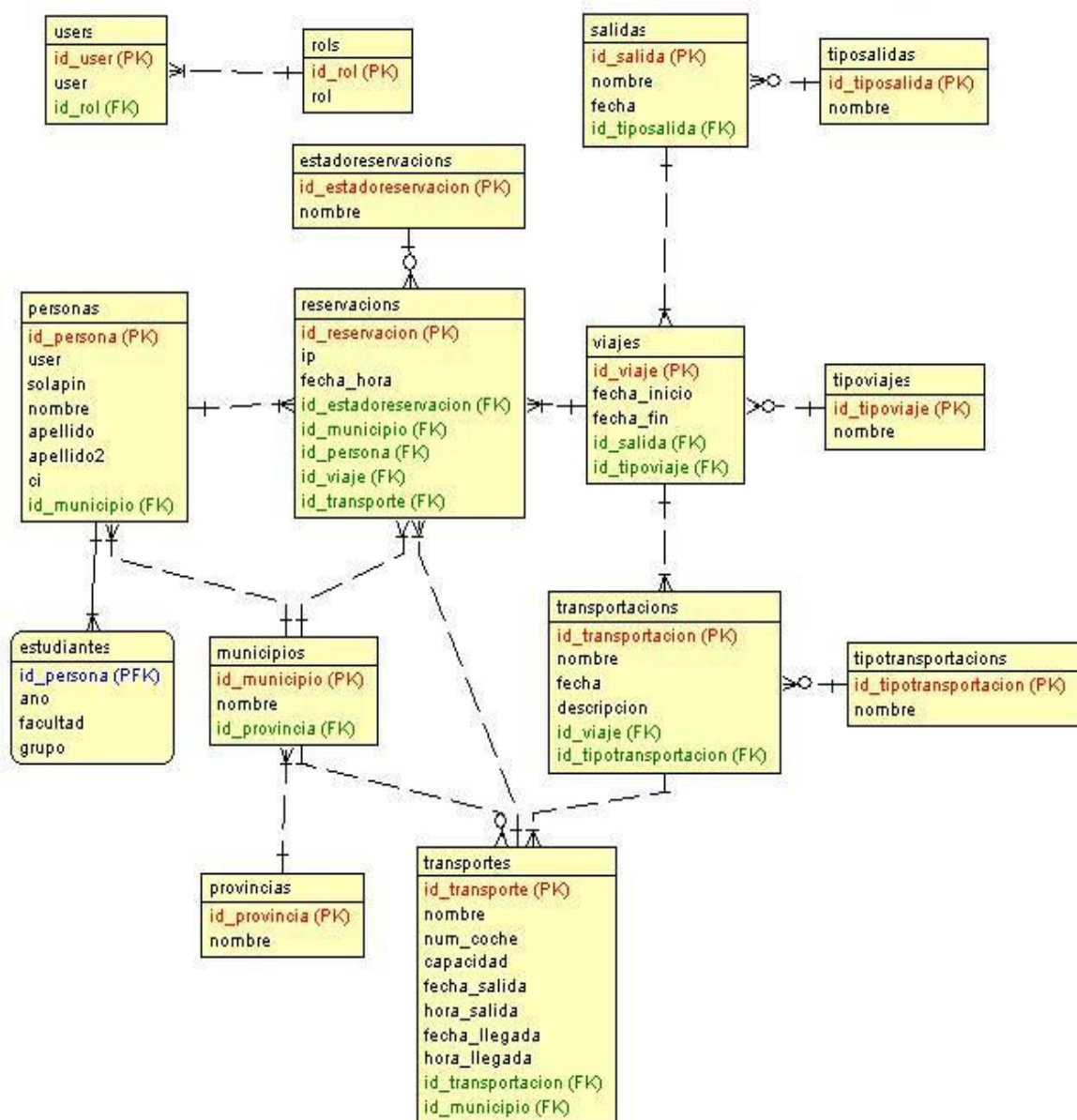


Figura 3. 12 Modelo de Datos.

3.4 Principios de Diseño.

El diseño de la interfaz de una aplicación, el tratamiento de excepciones y la concepción de la ayuda tiene gran influencia en el éxito o fracaso de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo del sistema en cuestión.

3.4.1 Estándares de la Interfaz de la aplicación.

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema, se puede definir como: “el conjunto de trabajos y pasos que seguirá el usuario, durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que realizará, así como las respuestas que el sistema dará”. (20)

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

La interfaz del usuario es el medio por el cual este interactúa con el sistema, por lo que esta debe ser lo mas amigable posible y lograr que se sienta identificado con la misma.

Para el diseño de la interfaz del sistema se tuvieron en cuenta aspectos necesarios, que garanticen la comodidad por parte del usuario, teniendo presente la organización de la información que se muestra y su distribución en la pantalla. Los elementos que se repiten en varias pantallas son ubicados en el mismo lugar, permitiéndole al usuario acostumbrarse al ambiente y que este no se vea desorientado. Estas solo contienen información necesaria para el usuario, evitando que estén sobrecargadas. Las pantallas son uniformes logrando un balance de los elementos que la componen.

Se utilizo una misma tipografía, forma y estilo en todas las páginas, para los botones y las líneas se utilizaron también estos estilos, disminuyendo así el número de imágenes que demoren la presentación de la página.

3.4.2 Tratamiento de excepciones.

El tratamiento de errores posibilita el buen funcionamiento de una aplicación dándole una mejor apariencia ante los clientes. En el sistema se controla la entrada de datos del usuario a nivel de interfaz. Cuando se produce un error por la entrada incorrecta de un valor suministrado por el usuario se le señala en la pantalla donde se encuentra para que sea rectificado. Cuando se introduce información en un formulario y faltan datos, sale un mensaje de alerta indicando el campo o dato que falta. Similar ocurre cuando introduce información errónea en un campo numérico.

Las excepciones que ocurren internamente en la aplicación son capturadas por el sistema y mostradas como mensajes de error al usuario. Estos mensajes aparecen en otra ventana, son específicos y a la vez concisos, además entendibles para el usuario.

Para prevenir errores por parte del usuario, solo se brindan las opciones mínimas necesarias, a la hora de efectuar cualquier operación, por ejemplo, se deshabilitan determinados botones si el usuario no tiene acceso a las paginas correspondientes.

3.5 Modelo de implementación.

Los **diagramas de despliegue** y **componentes** conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

3.5.1 Diagrama de despliegue.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. (12)

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 diagramas 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 objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes.

El diagrama de despliegue muestra la topología del hardware sobre el que se ejecuta el sistema. (Ver Anexo II. Diagrama de Despliegue.)

3.5.2 Diagrama de componentes.

El diagrama de componentes incluye a los componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico. (Ver Anexo III. Diagrama de Componentes.)

Conclusiones.

En este capítulo se representaron las diferentes partes de la solución propuesta a través de diferentes diagramas utilizando las extensiones de UML para modelar los elementos relativos a la aplicación.

Se obtuvo el diagrama de clases del diseño donde se representaron las clases y sus asociaciones. Se definió el diseño de la base de datos, creando para ello el diagrama de clases persistente y a partir de este el modelo de datos. Además se confeccionó el diagrama de despliegue y de componentes.

También se presentó la concepción del tratamiento de errores y se plantearon los principios de diseño de la aplicación Web.



Capítulo 4: Estudio de Factibilidad.

Introducción.

En el presente capítulo se realiza una estimación de los costos que generará el desarrollo de la aplicación usando el método de Puntos de Casos de Uso y se analizan los beneficios para determinar la factibilidad de la implantación del Sistema de Reservación de Transporte para el Pase Masivo en la UCI.

4.1 El Método de análisis de Puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente 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 finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

Finalmente se hace una estimación del costo monetario del proyecto teniendo en cuenta el tiempo estimado de desarrollo, la cantidad de personas involucradas y el salario de estas personas.

1. Cálculo de Puntos de Casos de Uso sin ajustar.

$$UUCP = UAW + UUCW$$

UUCP: Puntos de Casos de Uso sin ajustar.

$$UUCP = 14 + 140 = 154$$

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

1.1 Factor de Peso de los Actores sin ajustar. (UAW)

Tabla 14. Factor de Peso de los Actores sin Ajustar.

Actores	Factor Peso
Usuario Sistema	Complejo(3)
Usuario Viajero	Complejo(3)
Administrador del Sistema	Complejo(3)
Jefe de Área	Complejo(3)
Web Service	Simple(1)
Dominio UCI	Simple(1)

$$UAW = (3 * 4) + (1 * 2) = 12 + 2 = 14$$

1.2 Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Tabla 15. Factor de Peso de los CU sin Ajustar.

Casos de Uso	Factor de Peso
Autenticar Usuario	Simple(5)
Gestionar Boletines	Medio(10)
Gestionar Solicitud	Medio(10)
Consultar Información	Medio(10)
Gestionar Salida	Complejo(15)

Gestionar Viaje	Complejo(15)
Gestionar Transportación	Complejo(15)
Gestionar Transporte	Complejo(15)
Gestionar Distribución	Complejo(15)
Gestionar Usuario	Complejo(15)
Gestionar Reservación	Complejo(15)

$$UUCW = 5 + (3 * 10) + (7 * 15) = 5 + 30 + 105 = 140$$

2. Cálculo de Puntos de Casos de Uso ajustados.

$$UCP = UUCP * TCF * EF$$

$$UCP = 154 * 0.775 * 0.545 = 65,04575$$

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

2.1 Factor de complejidad técnica (TCF)

Tabla 16. Factor de Complejidad Técnica.

Factor	Descripción	Peso	Valor A.	$P_i * V_{Ai}$
T1	Sistema Distribuido	2	0	0
T2	Objetivos de performance o tiempo de respuesta	1	2	2
T3	Eficiencia del usuario final	1	2	2
T4	Procesamiento interno complejo	1	1	1
T5	El código debe ser reutilizable	1	0	0

T6	Facilidad de instalación	0.5	3	1,5
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	1	2
T9	Facilidad de cambio	1	2	2
T10	Concurrencia	1	1	1
T11	Incluye objetos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requiere facilidades especiales de entrenamiento a usuario	1	1	1

$$TCF = 0.6 + 0.01 * \sum(Pi * VAi) = 0.6 + 0.01 * 17.15 = 0.775$$

2.2 Factor de ambiente (EF)

Tabla 17. Factor de Ambiente.

Factor	Descripción	Peso	Valor A.	<i>Pi * VAi</i>
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7,5
E2	Experiencia en la aplicación	0.5	5	2,5
E3	Experiencia en OO	1	3	3
E4	Capacidad del analista líder	0.5	1	0,5

E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	5	10
E7	Personal part-time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	0	0

$$EF = 1.4 - 0.03 * \sum(Pi * VAi) = 1.4 - 0.03 * 28.5 = \mathbf{0.545}$$

3. Estimación del esfuerzo

$$E = UCP * CF$$

$$E = \mathbf{65.88505} * 20 \text{ horas – hombre}$$

$$E = \mathbf{1317.701 \text{ horas – hombre}}$$

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

Si además se considera que este esfuerzo representa un porcentaje del esfuerzo total del proyecto, de acuerdo a los valores porcentuales de la siguiente tabla, se obtiene (el valor de esfuerzo calculado representa el esfuerzo del FT implementación):

Tabla 18. Valor del Esfuerzo.

Actividad	% Esfuerzo	Valor Esfuerzo
Análisis	10%	325,22875
Diseño	20%	650,4575
Implementación	40%	1300,915
Prueba	15%	487,843125
Sobrecarga	15%	487,843125

Total	100%	3252,2875
-------	------	-----------

Suponiendo que una persona trabaje 8 horas por día, y una semana tiene como promedio 5 días laborables; la cantidad de horas que puede trabajar una persona en una semana es 40 horas.

Si el esfuerzo total es de 3252,2875 horas-hombre y por cada 192 horas se tiene un mes de trabajo, eso daría un tiempo de desarrollo de 16,9389974 mes-hombre.

Si en el equipo hay 2 personas y todas realizan el mismo esfuerzo entonces el problema analizado puede terminarse en aproximadamente 8.57878255 meses.

Si el salario mínimo de estas personas por mes fuera de \$150; entonces el costo total del proyecto sería de \$ 2540,84961.

4.2 Beneficios tangibles e intangibles.

El Sistema de Reservación de Transporte para el Pase Masivo en la UCI no es un software con fines comerciales, su principal objetivo es ayudar a agilizar los procesos de reservación de transporte en la Universidad. Por tanto, los beneficios inmediatos son mayormente intangibles.

Con la implantación de dicho sistema en la UCI, disminuye considerablemente los gastos por concepto de papel e imprenta para la elaboración de las listas de usuarios reservados y los reportes estadísticos.

Al lograr un mayor control de los procesos, se minimizan las pérdidas materiales, en específico de los recursos de transporte que la Universidad pone a disposición para cada uno de estos viajes.

Se logra una disminución considerable del tiempo que emplea el Director de Transporte en elaborar las listas de usuarios que viajarán. Se le facilita también a éste el trabajo con

estadísticas y la obtención de reportes diversos, lo que elimina contenido de trabajo y hace más confiable los resultados al eliminarse el error humano.

Hay un incremento considerable del bienestar y la protección del usuario, que ahora puede realizar las reservaciones con el menor esfuerzo y puede acceder mejor a las informaciones sobre las reservaciones de los viajes.

4.3 Análisis de costo / beneficio.

El Sistema de Reservación de Transporte para el Pase Masivo no requiere inversión en software pues todas las herramientas que se han empleado en su desarrollo son libres y de código abierto.

El sistema está orientado al usuario, es de fácil aprendizaje, por lo que no reporta grandes gastos por concepto de entrenamiento de los trabajadores del negocio.

No son necesarios los gastos por concepto de tecnología pues la Universidad cuenta con la necesaria para la implantación de la aplicación. Sólo se precisa garantizar la conectividad con la red local en las áreas de la Dirección de Transporte donde se gestiona el proceso de reservaciones.

Por todo ello se considera que es factible el desarrollo de la aplicación y que el esfuerzo de desarrollo (2 hombres Mes) está plenamente justificado.

Conclusiones.

En este capítulo se describió el estudio de factibilidad realizado al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado.

La herramienta propuesta reportará beneficios importantes para el desarrollo de esta actividad en la UCI, al contribuir a mejorar todos los servicios y procesos que se realizan aquí, lo que indica que es factible implementar la herramienta propuesta.

Conclusiones.

A partir de la investigación realizada para la elaboración de este sistema utilizando el Proceso Unificado de Desarrollo de Software (RUP), para lograr una mejor comprensión de los requisitos de la aplicación y formalización de los mismos, se arribó a las siguientes conclusiones:

- Se alcanzó, satisfactoriamente, el objetivo propuesto: Desarrollar un sistema automatizado que permitiera agilizar la gestión del proceso de reservación de transporte para el pase masivo en la universidad.
- Se demostró la eficacia de los lenguajes y tecnologías utilizadas para el desarrollo del sistema.
- Se realizó una base de datos, donde se almacena toda la información necesaria que se genera de los procesos de reservación, para de esta forma garantizar la veracidad y centralización de la misma.
- Se realizó el análisis, diseño e implementación de las funcionalidades priorizadas del sistema.
- Se siguió los principios básicos de diseño descritos para el desarrollo del sistema.
- Se logró seguridad y protección de los datos consecuente con el nivel de seguridad requerido.

Recomendaciones.

Se recomienda:

- Poner a prueba el sistema durante un período de tiempo, para comprobar su desempeño.
- Implementar las funcionalidades no priorizadas en esta versión del sistema.
- Confeccionar la ayuda de la aplicación, ya que pueden existir muchos usuarios que no tengan mucho nivel en el uso de las tecnologías y se encuentren desorientados en determinados momentos.
- Agregar el sistema de Reservación de Pase Masivo como un módulo al Sistema de Servicios Generales.

Glosario de Términos.

- **CSS:** Siglas en Inglés de Cascading Style Sheets (Hojas de Estilo en Cascada).
- **UML:** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.
- **URL:** Acrónimo de Universal Resource Locator (localizador universal de recursos), método de identificación de documentos o lugares en Internet, que se utiliza principalmente en World Wide Web (WWW). Un URL es una cadena de caracteres que identifica el tipo de documento, la computadora, el directorio y los subdirectorios en donde se encuentra el documento y su nombre.
- **Web Services:** Aplicación que realiza un cometido y que puede formar parte de otros servicios para formar un servicio más completo. La comunicación hacia y desde el Webservice se realiza con XML. Permite una llamada a una funcionalidad localizada en un servidor remoto.
- **PHP:** acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.
- **Navegador web** o **explorador web:** es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.

- **HTML:** acrónimo inglés de **HyperText Markup Language** (lenguaje de marcas hipertextuales) es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.
- **Hipertexto:** es un paradigma en la interfaz del usuario cuyo fin es el de presentar documentos que puedan, "bifurcarse o ejecutarse cuando sea solicitado". La forma más habitual de hipertexto en documentos es la de hipervínculos.

Referencias Bibliográficas.

1. **Sklar, David.** *Learning PHP 5*. s.l. : O'Reilly, 2004. pág. 368.
2. **Alshanetsky, Iliia.** *PHP|Architect's Guide to PHP Security*. [ed.] Martin Streicher. First Edition. Canada : Marco Tabini & Associates Inc., 2005.
3. **Guillermo, Jose Valle.** Definicion arquitectura cliente servidor. *Monografias.com*. [En línea] 2005.
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
4. **Anónimo.** Monografias.com. *Monografias.com*. [En línea] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
5. **Gutmans, Andi, Sæther, Stig Bakken y Rethans, Derick.** *PHP 5 Power Programming*. s.l. : Prentice Hall PTR, 2004. pág. 720.
6. **adictos, PHP.** Historia de PHP. *PHP adictos*. [En línea] 30 de enero de 2004.
<http://www.phpadictos.com/?secc=leer&id=62>.
7. **Ciberaula.** Master PHP.Soluciones para el desarrollo web. *Ciberaula*. [En línea]
http://www.ciberaula.com/curso/masterphp/que_es/.
8. **Wikipedia.** PostgreSQL. *Wikipedia. La Enciclopedia Libre*. [En línea] <http://es.wikipedia.org/wiki/PostgreSQL..>
9. **González, Cesar Hernández.** Módulo Alojamiento del Sistema.Tesis para optar por el Título de Ingeniero en Informática. [En línea] 2006.
10. **Desarrolloweb.** Qué es Javascript. *Desarrolloweb.com*. [En línea]
<http://www.desarrolloweb.com/articulos/25.php>.
11. **Mellado, Javier.** Ajax. *Ajax Hispano*. [En línea] 17 de Enero, de 2006. <http://www.ajaxhispano.com/que-es-ajax.html>.
12. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. s.l. : Addison Wesley., 2000.
13. **Laman, Craig.** *UML y patrones.Introducción al análisis y diseño orientado a objetos*. Mexico : Prentice Hall, 1999. pág. 536.

14. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas.* Mexico : s.n., 2000. pág. 448.
15. **Soluciones, Grupo.** Rational Rose Enterprise. *GSINNOVA.Grupo Soluciones.* [En línea] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
16. **Company, Zend The PHP.** Zend Studio: The Proven PHP Development Environment. *Zend The PHP Company.* [En línea] http://www.zend.com/products/zend_studio.
17. **Charonware.** Data Modeler - database design tool. *Charonware.* [En línea] <http://www.casestudio.com>.
18. **CakePHP.** The Manual. *CakePHP.* [En línea] <http://manual.cakephp.org/chapter/intro>.
19. **Laman, Craig.** *UML y Patrones.Introducción a1 análisis y diseño orientado a objetos.* Mexico : s.n., 1999. pág. 536.
20. **Webstudio.** PHP Avanzado. *www.zonaphp.com.* [En línea] 30 de Abril de 2003. <http://www.zonaphp.com/%C2%BFque-esperar-de-php5//>.
21. **Dondo, Agustín.** ¿Por qué elegir PHP? *www.programacion.net.* [En línea] 3 de febrero de 2002. <http://www.programacion.net/php/articulo/porquephp/>.
22. **Quintero, Beatriz.** La agencia de viaje mas grande del mundo. *Informativos TELECINCO.com.* [En línea] 28 de julio de 2006. http://www.informativos.telecinco.es/internet/viajes/bajo_coste/dn_29325.htm.
23. **World, Comunicaciones.** El comercio electrónico crece en España más de un 16%. *Comunicaciones World.* [En línea] 25 de 07 de 2006. <http://www.idg.es/comunicaciones/noticia.asp?id=49558>.
24. **García, Alfredo Jiménez, y otros.** *Turismo. Desempeño y futuro.* 2006.
25. **Parra, A y Matos, M.** Sistema Automatizado para la Gestión de Información de la Unión de Jóvenes Comunistas. [En línea] 2005.

Anexos.

Anexo I. Diagramas de secuencia.

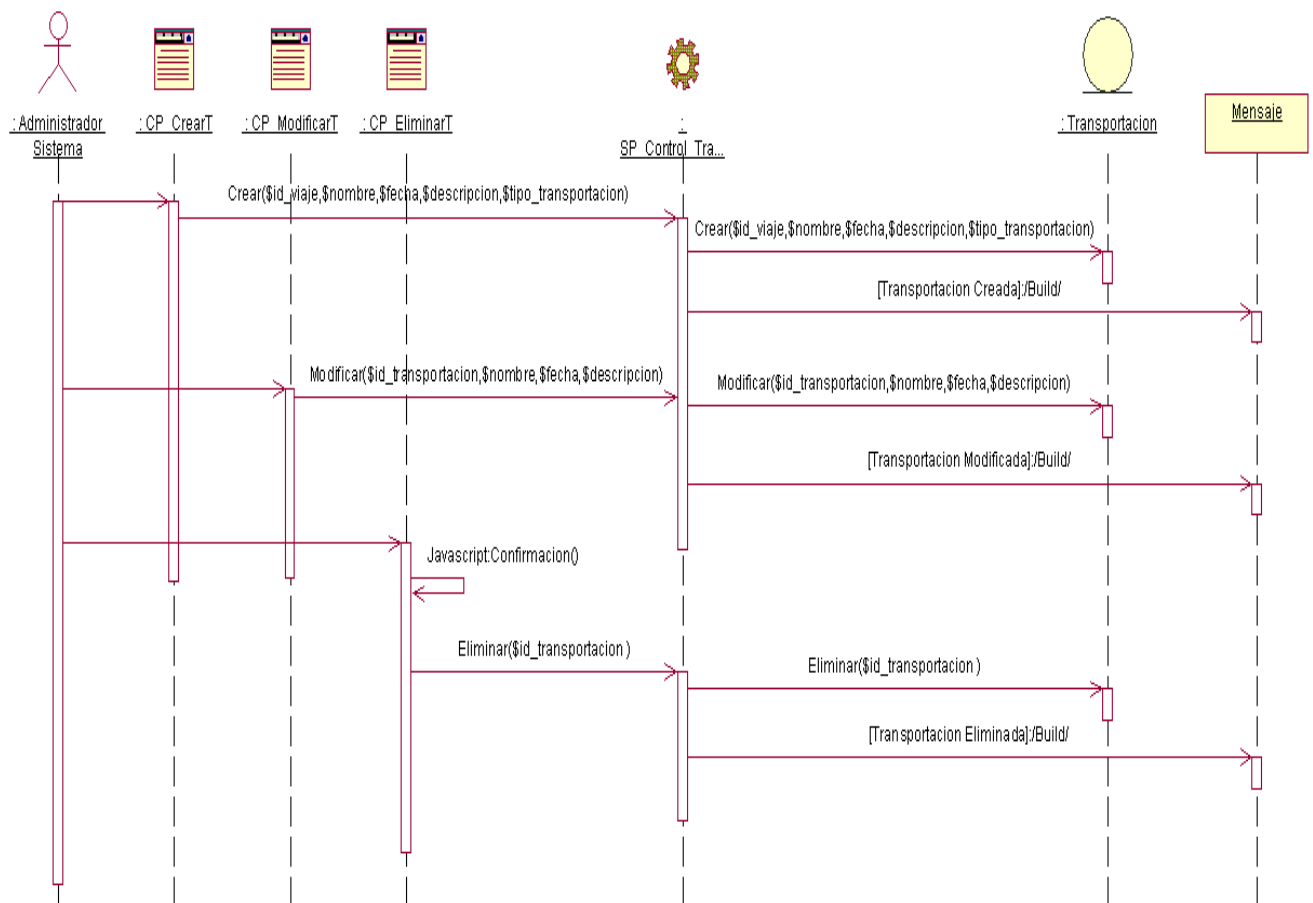


Figura 3. 13 Diagrama de Secuencia " Gestionar Transportación"

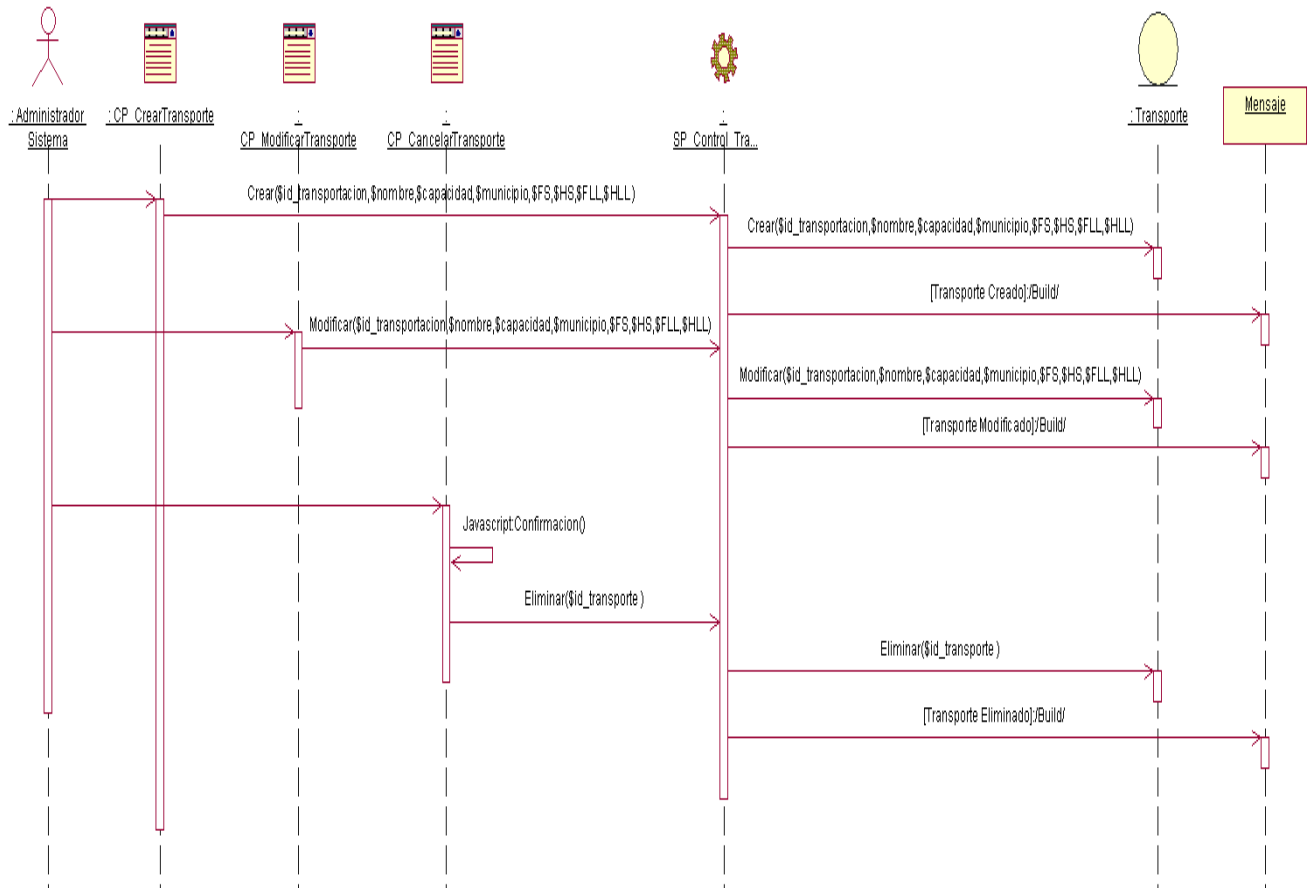


Figura 3. 14 Diagrama de Secuencia " Gestionar Transporte"

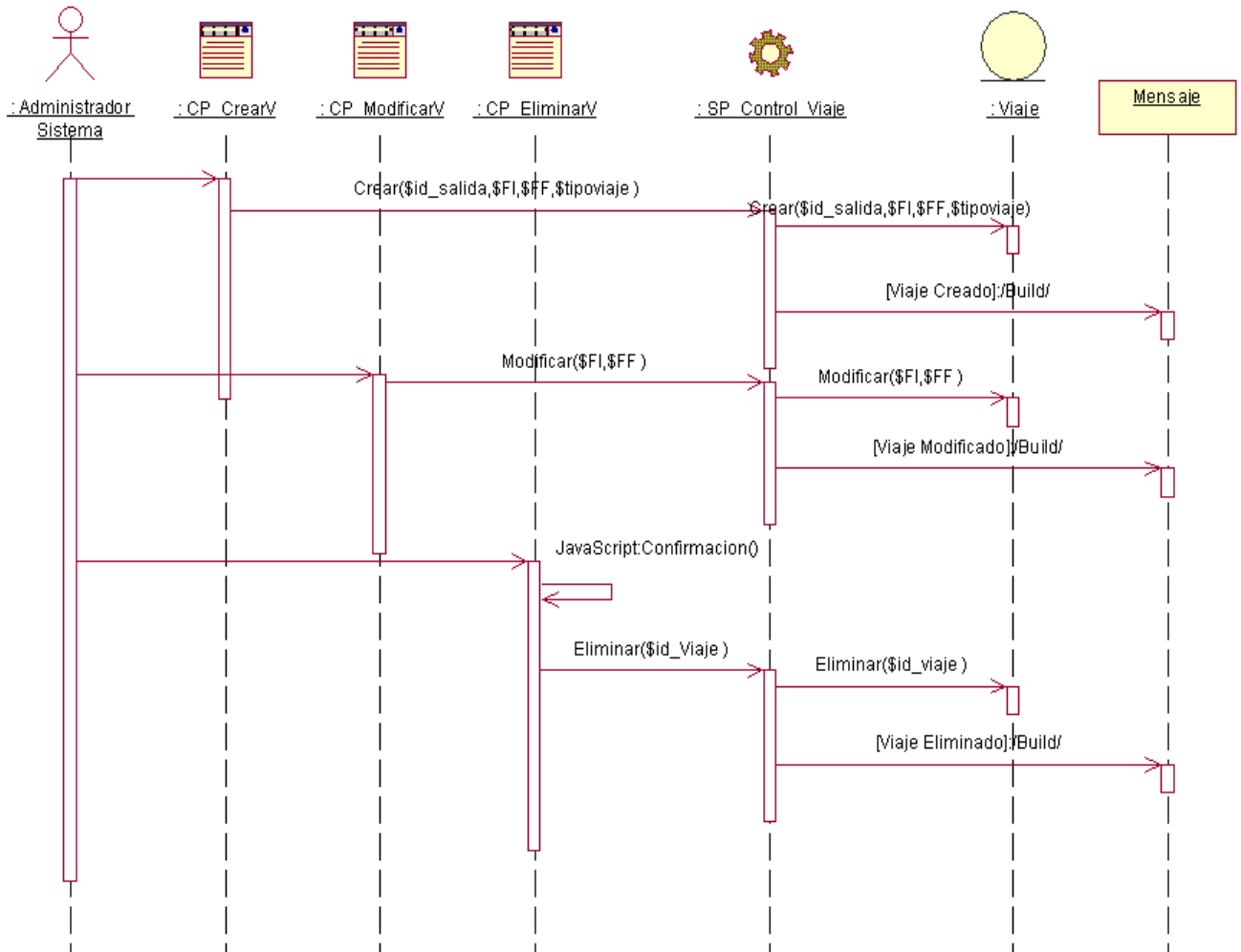


Figura 3. 15 Diagrama de Secuencia " Gestionar Viaje"

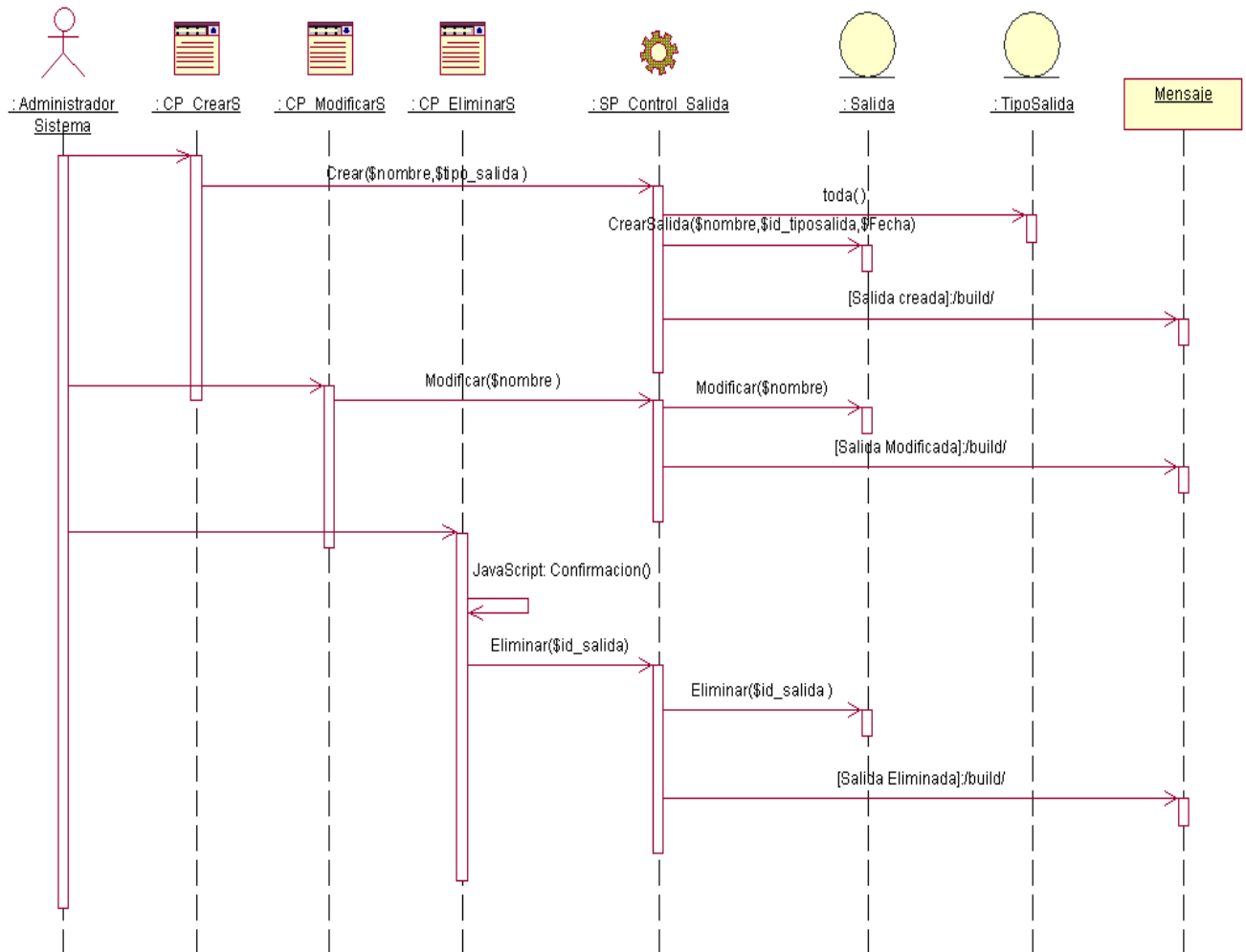


Figura 3. 16 Diagrama de Secuencia " Gestionar Salida"

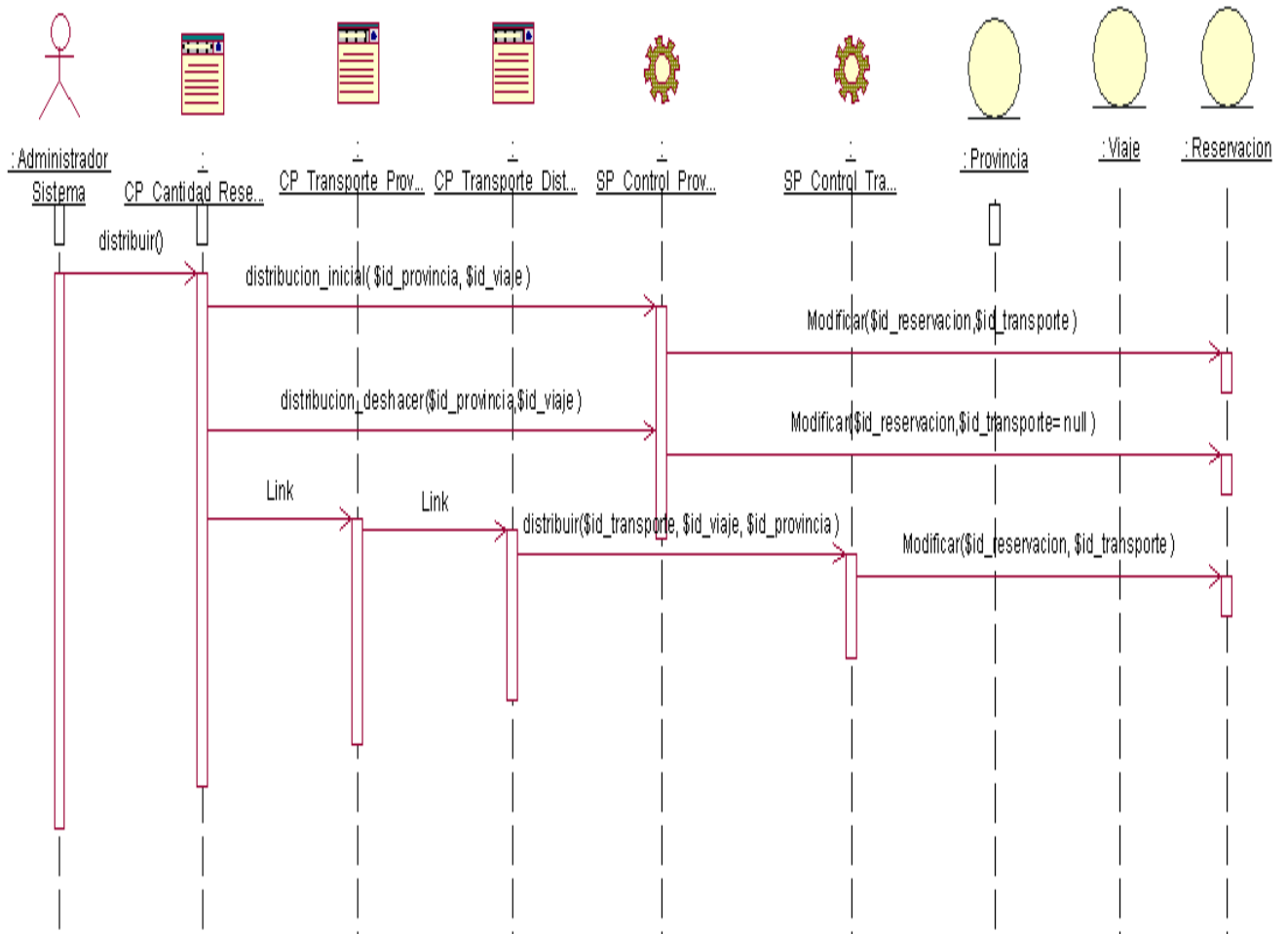


Figura 3. 17 Diagrama de Secuencia " Gestionar Distribución"

Anexo II. Diagrama de Despliegue.

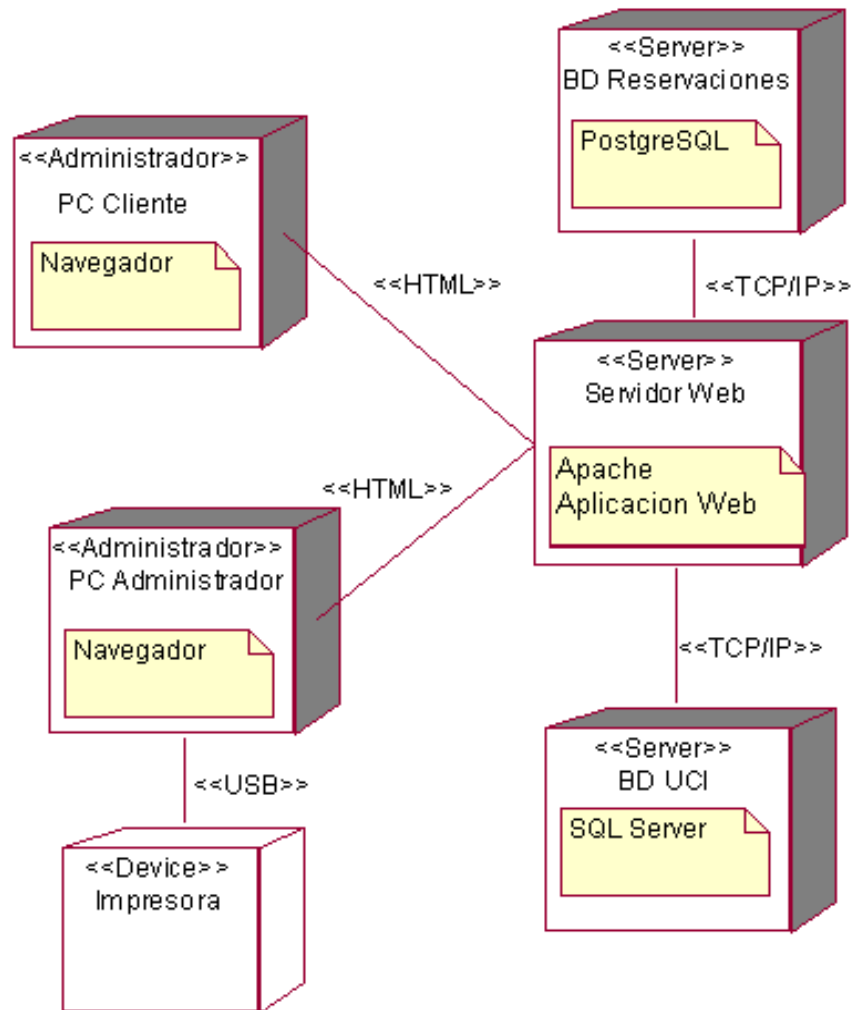


Figura 3. 18 Diagrama de Despliegue.

Anexo III. Diagrama de Componentes.

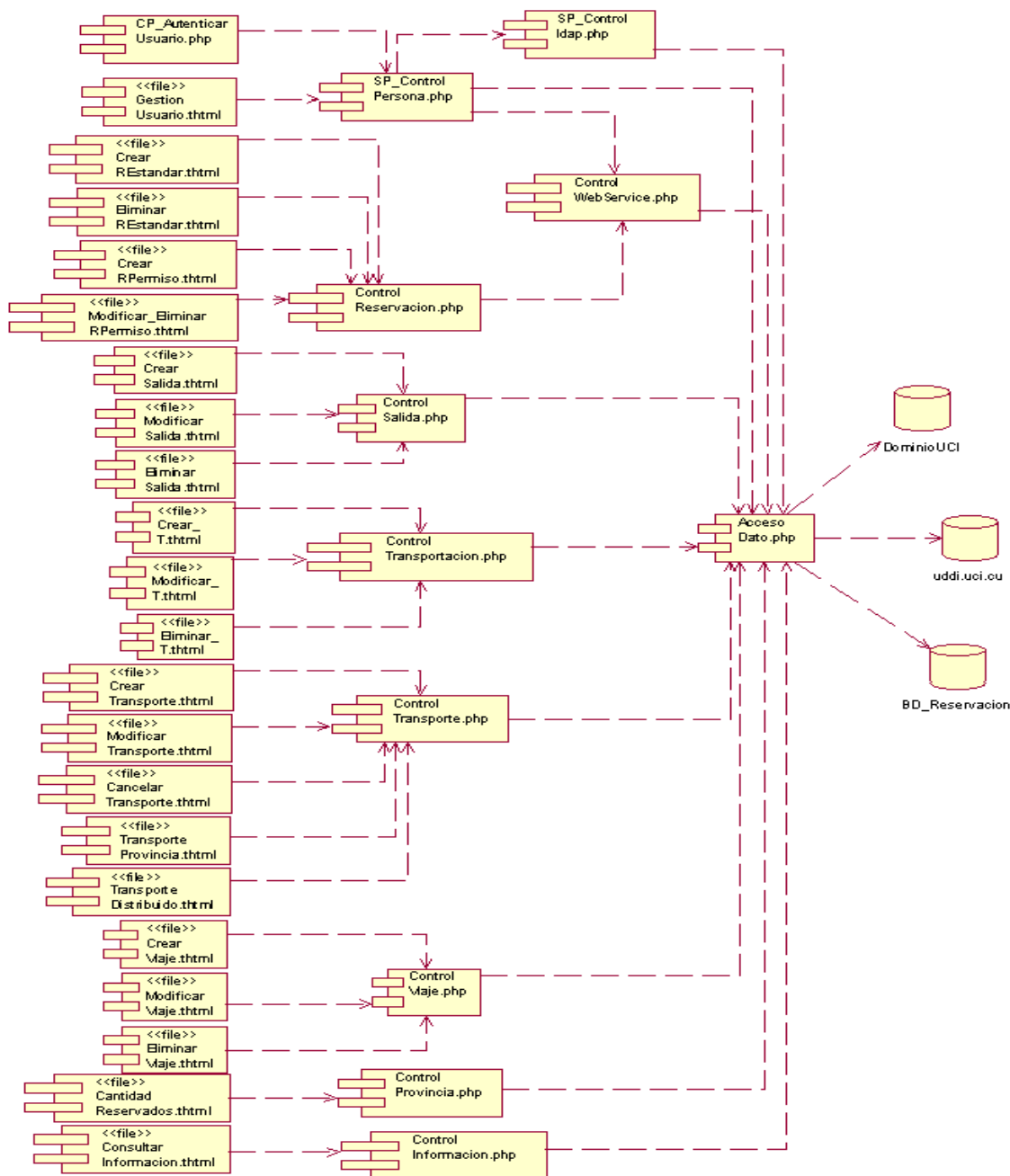


Figura 3. 19 Diagrama de Componentes.

DECLARACION DE AUTORIA

Nosotros: **Yohana Baró Montenegro** y **Reynaldo Hernández Rodríguez** nos declaramos como únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que haga uso de este trabajo de la manera que estime conveniente.

Y para que así conste firmamos la presente a los ____ días del mes de _____ de 2007.

Firma de los Autores

Firma del Tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA.

El Trabajo de Diploma titulado **Sistema de Reservación de Pase Masivo** fue desarrollado en la Universidad de las Ciencias Informáticas. Esta entidad 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 entidad los beneficios siguientes:

Y para que así conste, se firma la presente a los ____ días del mes de _____ de 2007.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Sistema de Reservación de Pase Masivo**

Autores: Yohana Baró Montenegro y Reynaldo Hernández Rodríguez

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de

Firma

Fecha