




Universidad de las Ciencias Informáticas

Facultad 1

A semi-transparent image of a blue bus is centered in the background. The bus is a modern, long-distance model with a large windshield and a prominent front grille. The text is overlaid on this image.

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

Título: Análisis y Diseño de un Sistema de
Transportación Nacional

Autores: Franklin Yannier Garmendia López
Yanet Rodríguez Medina

Tutor: Ing. Alexander Rodríguez Mompíe

Ciudad de la Habana, 5 de junio del 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores del presente trabajo de tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2008.

Franklin Yannier Garmendia López

Yanet Rodríguez Medina

Firma del Autor

Firma del Autor

Ing. Alexander Rodríguez Mompíe

Firma del Tutor

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Análisis y Diseño de un Sistema de Transportación Nacional

Autores: Franklin Yannier Garmendia López, Yanet Rodríguez Medina

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

Los estudiantes desarrollaron el Diseño de una Aplicación Web para el Sistema de Transportación Nacional de la Universidad de las Ciencias Informáticas. El trabajo fue realizado con muy alta independencia y responsabilidad por parte de los diplomantes, los cuales demostraron en todo momento estar preparados para asumir y acometer correctamente el trabajo propuesto. La creatividad se vio demostrada por la calidad del documento y el tiempo dedicado. Además demuestran su laboriosidad y responsabilidad a la hora de asumir una tarea. La comunicación con el tutor fue positiva y constante, mostrando interés en aspectos relacionados con el desarrollo del trabajo.

La alta calidad científico-técnica, se evidencia en la investigación realizada, demostrando un alto dominio de los conceptos actuales del desarrollo de software. Los mismos, apoyándose en una gran variedad de bibliografía, fueron capaces de realizar profundos análisis y llegar a conclusiones que proveen al trabajo de un alto valor científico.


El documento presentado tiene una estructura adecuada, hace un uso correcto del lenguaje y refleja de manera clara y concisa todas las etapas desarrolladas en la investigación. El trabajo contiene resultados que poseen valor para ser presentados en eventos y talleres científicos relacionados.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de Excelente ____ puntos.

Ing. Alexander Rodríguez Mompíe

Firma

Fecha



“Hoy estamos por delante y vamos rápido, y mucho más podremos hacer en un futuro próximo, gracias a la fuerza de la Revolución.”

Fidel Castro Ruz

AGRADECIMIENTOS

Compartidos

A la Revolución y nuestro siempre Comandante Fidel por darnos la oportunidad de estudiar en esta maravillosa universidad, por confiar siempre en nosotros y permitirnos ser profesionales.

A la Universidad de las Ciencias Informáticas, por ser nuestra casa estos 5 años.

A todos los que aportaron tiempo, conocimientos y amor en el desarrollo de este trabajo.

A nuestro tutor Alexander por sus enseñanzas y apoyo durante el desarrollo de este trabajo.

A nuestro cliente Felipe por su amistad y apoyo durante la realización de la tesis.

Yanet

A mi mamá y a mi papá por ser mi fuente de inspiración, por apoyarme y guiarme siempre, sin sus enseñanzas no sería posible, su amor y comprensión hicieron que llegara hasta aquí, mis resultados son por ustedes.

A mis hermanitas del alma Yaimé y Yilena por siempre creer en mí y darme su cariño y amor en todo momento.

A mi novio Frank, por su amor, comprensión y apoyo; nenito: sin ti no hubiera sido posible.

A mis tíos Nélide y Pavón que fueron como mis padres en estos 5 años, ustedes son parte esencial de este triunfo.

A mis abuelos, tíos, primos y toda mi familia en general que me han brindado su cariño, apoyo y confianza.

A mis suegros Franklin, Yayi y Elba por su apoyo, amistad y cariño durante todo este tiempo.

A todos mis amigos y compañeros de la UCI, esos que han sido parte de algún momento de mi vida, a los que están y los que no, siempre me enseñaron y ayudaron mucho.

Franklin

A mis padres por ser mi fuente de inspiración, ejemplo a seguir, por vivir por y para mí, por haber hecho que llegara hasta aquí, por su confianza y amor este triunfo es de ustedes.

A mis hermanos Yandris y Grethel por su cariño y confianza en todo momento.

A mi novia Yane por estar siempre a mi lado, por su amor, cariño y comprensión durante estos 5 años.

A Elba por ser como una madre, por su apoyo y enseñanzas durante el desarrollo de la tesis.

A mis tíos, primos, abuelos y familia en general por haberme apoyado y brindado su confianza y cariño durante todo este tiempo.

A todos mis compañeros que en algún momento de la vida me enseñaron y ayudaron mucho.

DEDICATORIA

Yanet

A mis padres Martha y Rolando por ser las personas más importantes de mi vida, por ayudarme, guiarme y demostrarme que no hay nada imposible en la vida.

A mis hermanas por ser mis amigas, por confiar en mí, son parte de mi corazón.

A mi familia en general por su apoyo, confianza y cariño durante todo este tiempo.

Franklin

A mis padres Gladys y Franklin por su confianza y amor, por estar a mi lado siempre apoyándome, por guiarme durante todos estos años.

A mis hermanos por su confianza y apoyo durante el desarrollo del trabajo.

A toda mi familia que estuvo siempre a mi lado, ayudándome siempre que fue necesario.

RESUMEN

Las Tecnologías de la Informática y las Comunicaciones (TIC) juegan un papel esencial en el desarrollo de la informatización de los diversos sectores de la sociedad. La investigación desarrollada en este trabajo de diploma está relacionada con un sector muy importante dentro de la misma: el transporte. La Universidad de las Ciencias Informáticas (UCI) cuenta con una dirección de transporte, la cual tiene entre sus ramas la reservación del pase masivo, el cual es un proceso bastante engorroso, a partir de lo cual surge la necesidad de diseñar una aplicación web, que agilice y organice dicho trabajo, para lo cual se realiza un estudio de otros sistemas de reservaciones existentes en el mundo, buscando una posible solución.

Para su desarrollo se tiene en cuenta la selección de la metodología, herramienta y tecnologías a utilizar. Además, se hace un análisis detallado de como se lleva a cabo actualmente el proceso del negocio relacionado con la salida masiva de la universidad, y explica de forma detallada como debe funcionar el sistema. También aparecen descritas las etapas de análisis y diseño, mediante la modelación de varios diagramas, para un mejor entendimiento de las funcionalidades que posteriormente se van a implementar. Finalmente se sitúa el estudio de la factibilidad, donde se obtiene una estimación del costo que implica el desarrollo de dicha aplicación.

Palabras claves:

Reservación, aplicación web, modelado, pase masivo, transportación nacional.

ÍNDICE

AGRADECIMIENTOS	VI
DEDICATORIA.....	VIII
RESUMEN.....	VIII
INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	6
1.1 INTRODUCCIÓN.....	6
1.2 CONCEPTOS BÁSICOS RELACIONADOS CON EL DOMINIO DEL PROBLEMA	6
1.2.1 Analista de procesos del negocio.	6
1.2.2 El diseñador del negocio.	6
1.2.3 Analista del sistema.	6
1.2.4 Especificador de requerimientos.	7
1.3 SISTEMAS AUTOMATIZADOS DE RESERVACIÓN EXISTENTES EN EL MUNDO	7
1.3.1 Airqueens.com	7
1.3.2 Transtur.cu.....	8
1.3.3 Reservación del pase de fin de semana (pase.uci.cu).....	9
1.3.4 Pase Masivo.....	9
1.4 TENDENCIAS Y TECNOLOGÍAS ACTUALES	10
1.4.1 Las Tecnologías de la Información y las Comunicaciones (TICs).....	10
1.4.2 ¿Qué es Internet?	11
1.4.3 Aplicaciones web.....	11
1.4.4 Software libre	12
1.4.5 Ambiente de trabajo frameworks	13
1.4.5.1 CakePHP	14
1.5 METODOLOGÍA DE DESARROLLO UTILIZADA	15
1.5.1 Rational Unified Process (RUP)	16
1.6 LENGUAJE DE MODELADO UTILIZADO (UML)	18

1.7	HERRAMIENTA DE MODELADO UTILIZADA	20
1.8	PATRONES	22
1.8.1	Patrón de diseño web.....	22
1.8.2	Patrón de arquitectura.....	24
1.9	LENGUAJE DE PROGRAMACIÓN WEB UTILIZADO.....	25
1.9.1	PHP.....	25
1.10	CONCLUSIONES.....	26
CAPÍTULO 2. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA		27
2.1	INTRODUCCIÓN.....	27
2.2	NEGOCIO	27
2.2.1	Modelo de negocio	27
2.2.2	Reglas del negocio.....	27
2.2.3	Actores y trabajadores del negocio	28
2.2.4	Descripción del negocio.	29
2.2.5	Diagrama de casos de uso del negocio.....	30
2.2.6	Diagrama de actividades del negocio	31
2.2.7	Descripción textual de los casos de uso de negocio.....	33
2.2.8	Modelo de objetos del negocio.....	35
2.3	ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE	35
2.3.1	Requerimientos funcionales	36
2.3.2	Requerimientos no funcionales.	44
2.4	DESCRIPCIÓN DEL SISTEMA PROPUESTO.	47
2.4.1	Descripción de los actores del sistema.....	49
2.4.2	Diagrama de casos de uso del sistema.	50
2.4.3	Descripción de los casos de uso del sistema.	50
2.5	CONCLUSIONES.....	51
CAPÍTULO 3. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....		52
3.1	INTRODUCCIÓN.....	52
3.2	MODELO DE ANÁLISIS.....	52

3.2.1 Clases del análisis.....	52
3.3 MODELO DE DISEÑO.....	55
3.3.1 Clases del diseño.....	55
3.3.2 Descripción de las clases del diseño.....	60
3.3.3 Diagramas de secuencia del diseño.....	60
3.4 PRINCIPIOS DE DISEÑO.....	63
3.4.1 Estándares de la interfaz de aplicación.....	63
3.5 CONCEPCIÓN GENERAL DE LA AYUDA.....	63
3.6 TRATAMIENTO DE EXCEPCIONES.....	63
3.7 DIAGRAMA DE CLASES PERSISTENTES.....	64
3.8 CONCLUSIONES.....	66
CAPÍTULO 4. ESTUDIO DE FACTIBILIDAD.....	67
4.1 INTRODUCCIÓN.....	67
4.2 PLANIFICACIÓN BASADA EN CASOS DE USO.....	67
4.3 BENEFICIOS TANGIBLES E INTANGIBLES.....	74
4.4 ANÁLISIS DE COSTO Y BENEFICIOS.....	74
4.5 CONCLUSIONES.....	75
CONCLUSIONES GENERALES.....	76
RECOMENDACIONES.....	77
REFERENCIAS BIBLIOGRÁFICAS.....	78
BIBLIOGRAFÍAS.....	80
GLOSARIO DE TÉRMINOS.....	81
ANEXOS.....	84

ÍNDICE DE TABLAS

Tabla 2.1 Actores del Negocio.....	28
Tabla 2.2 Trabajadores del Negocio.....	29
Tabla 2.3 Descripción del CUN: “Reservar Transporte”	34
Tabla 2.4 Actores del Sistema.....	50
Tabla 4.1 Factor Peso de los Actores sin ajustar (UAW).	68
Tabla 4.2 Factor Peso de los Casos de Usos sin ajustar (UUCW).....	68
Tabla 4.3 Clasificación del Factor de Complejidad Técnica del Sistema.....	70
Tabla 4.4 Clasificación del Factor de Ambiente del Sistema.....	71
Tabla 4.5 Distribución del esfuerzo.....	72
Tabla 1. CUS Gestionar usuario	84
Tabla 2. CUS Gestionar reglas	85
Tabla 3. CUS Autenticar usuario	86
Tabla 4. CUS Gestionar bloques de transportación.....	86
Tabla 5. CUS Gestionar transportes.....	87
Tabla 6. CUS Gestionar provincias.....	89
Tabla 7. CUS Gestionar reservación	90
Tabla 8. CUS Gestionar viajero	93
Tabla 9. CUS Gestionar distribución.....	95
Tabla 10. CUS Bloquear usuario	96
Tabla 11. CUS Consultar información.....	97
Tabla 12. CUS Gestionar viajes.....	97
Tabla 13. CUS Generar reportes	99
Tabla 14. CUS Generar boletines.....	100
Tabla 15. CUS Gestionar coches	101

ÍNDICE DE FIGURAS

Figura 1.1 Fases e iteraciones de la metodología RUP	17
Figura 2.1 Diagrama de casos de uso del negocio	30
Figura 2.2 Diagrama de actividades del negocio	31
Figura 2.3 Modelo de objeto del negocio	35
Figura 2.4 Diagrama de casos de uso del sistema.	50
Figura 3.1 Diagrama de clases del análisis: CU “Autenticar usuario”	52
Figura 3.2 Diagrama de clases del análisis: CU “Gestionar bloques de transportación”	53
Figura 3.3 Diagrama de clases del análisis: CU “Gestionar viaje”	53
Figura 3.5 Diagrama de clases del análisis: CU “Gestionar reglas”	54
Figura 3.6 Diagrama de clases del análisis: CU “Bloquear usuario”	54
Figura 3.7 Diagrama de clases del análisis: CU “Generar boletines”	54
Figura 3.8 Diagrama de clases del diseño: CU “Autenticar usuario”	55
Figura 3.9 Diagrama de clases del diseño: CU “Gestionar bloque de transportación”	56
Figura 3.10 Diagrama de clases del diseño: CU “Gestionar viaje”	57
Figura 3.11 Diagrama de clases del diseño: CU “Gestionar reglas”	57
Figura 3.12 Diagrama de clases del diseño: CU “Bloquear usuario”	58
Figura 3.13 Diagrama de clases del diseño: CU “Generar boletines”	59
Figura 3.14 Diagrama de secuencia: CU “Autenticar usuario”	60
Figura 3.15 Diagrama de secuencia: CU “Bloquear usuario”	61
Figura 3.16 Diagrama de secuencia: CU “Generar boletines”	62
Figura 3.17 Diagrama de clases persistentes	65

INTRODUCCIÓN

En Cuba se le ha concedido prioridad máxima a la socialización de las Tecnologías de la Informática y las Comunicaciones. Cuando se hace referencia a la informatización de la sociedad cubana, se está hablando de la aplicación ordenada y masiva de las tecnologías de la informática en todos los sectores, para mediante su uso racional y adecuado lograr una mayor generación de riquezas.

El país hace enormes esfuerzos para lograr cada vez más eficacia y eficiencia en todos los procesos, para así aumentar la calidad de vida de sus ciudadanos. La informatización se basa en principios y valores sociales de nuestra sociedad. Entre los servicios al que se le ha dado mayor interés está la informatización de la comunidad incluyendo los servicios comunitarios.

La Universidad de las Ciencias Informáticas no queda ajena a este enorme proceso de informatización de la sociedad cubana por lo que tiene como fundamental misión informatizar las prestaciones necesarias e indispensables para el país.

En la UCI se está llevando a cabo un proceso de informatización de todos los servicios generales que allí se proporcionan, que no son más que aquellas prestaciones que brindan un beneficio en específico a la comunidad y resuelven problemas existentes en el centro. La dirección de transporte brinda varios servicios a la universidad; de todos los procesos que se informatizarán pertenecientes a este sector se pretende dar solución en específico a la parte de “Reservación de Pase Masivo”, de vital importancia en la escuela.

En la actualidad, producto a la gran cantidad de personal que radica en ella, se hace necesario la implantación de un sistema que gestione este proceso como una aplicación que posteriormente se incorpore al sistema general de Servicios Comunitarios.

La reservación del pase masivo es un beneficio que se les brinda a los estudiantes y trabajadores internos. Dicho proceso se lleva a cabo generalmente dos veces al año: en fin de año y al terminar el curso escolar, pero puede utilizarse siempre que sea necesario hacer transportaciones con similares características.

Actualmente en la UCI la reservación de pase masivo es un proceso que se lleva a cabo manualmente casi en su totalidad, por lo que es bastante difícil enfrentar dicha tarea, la Vicerrectoría de formación es la encargada de comenzar dicho proceso, orientándole al especialista de transporte las fechas de salida/entrada en que debe planificar la transportación masiva de la universidad para estudiantes y trabajadores internos. Luego el especialista de transporte solicita a Akademos y a Recursos humanos el listado de todos los el listado de todos los estudiantes y trabajadores internos respectivamente, seguidamente se procede a la confección del plan de transportación para negociarlo con el Ministerio de Transporte (MITRANS), el cual le notifica si puede o no disponer del transporte en dichas fechas. De no poder adquirir los transportes en los días orientados, la Vicerrectoría de formación debe hacer cambios para que el especialista vuelva a hacer la solicitud. Cuando el MITRANS acepta, le asigna a la UCI los transportes que necesite (ómnibus, tren, barco). Después se realizan los listados con la distribución de todos los viajeros de acuerdo a provincia y municipio. Luego se publica mediante el correo electrónico el plan y la distribución efectuada. Llegado el día de la salida se le entrega a un responsable por guagua los listados con la distribución correspondiente.

Anteriormente se hizo una aplicación web para automatizar dicho servicio de reservación del pase masivo, pero fue un proceso incompleto pues no satisface todas las funcionalidades que requiere el sistema para cumplir las necesidades de los clientes.

Debido a las notables dificultades planteadas el **problema científico** al que se dará solución queda expuesto de la forma siguiente:

¿Cómo agilizar y organizar el proceso de reservación del pase masivo en la UCI?

Teniendo en cuenta el problema científico se elaboran las siguientes **preguntas científicas**:

- ¿Se logrará con la selección de las herramientas y metodologías de desarrollo de software, realizar una propuesta de solución, para la futura implementación de una aplicación web?
- ¿Contribuirá el desarrollo del análisis y diseño, a la implementación de un sistema que permita automatizar el proceso de gestión de reservaciones en la UCI?

Partiendo del problema planteado el **objeto de estudio** se enfocará hacia los procesos de reservación de

transporte en la UCI. Además se tendrá en cuenta que la investigación se centra fundamentalmente en la reservación del pase masivo siendo este el **campo de acción**.

Se ha identificado como **objetivo general**: Realizar análisis y diseño de una aplicación web que incluya todas las funcionalidades necesarias para agilizar la gestión del proceso de reservaciones de pase masivo en la UCI, del cual se derivan los siguientes **objetivos específicos**:

- Identificar los procesos del negocio.
- Identificar los requerimientos de la aplicación.
- Diseñar todas las clases que se implementarán.

Como **idea a defender** se formula: Con la selección de las herramientas, metodologías y la buena realización del análisis y diseño se contribuirá al posterior desarrollo de la aplicación web.

Las **Variables de investigación** quedan desglosadas de la siguiente forma:

Variable independiente:

- Análisis y diseño del sistema.

Variables dependientes:

- Organización de los procesos.
- Eficiencia de los procesos.

Para lograr un correcto análisis y diseño de dicho sistema y obtener los resultados requeridos es necesario trazar el siguiente conjunto de **tareas científicas**:

- Realizar entrevistas a personas que están inmersas en el proceso que se pretende informatizar.
- Estudiar y describir los sistemas de reservación de transportes dentro y fuera de la universidad.
- Estudiar el estado del arte de las principales y más actuales tecnologías utilizadas para el diseño de aplicaciones web.

- Seleccionar la metodología y lenguaje de modelado para el análisis y diseño de sistemas informáticos, que facilite la creación y garantice la calidad futura del sistema.
- Seleccionar la herramienta de modelado a utilizar para el desarrollo del trabajo.
- Elaborar la fundamentación teórica de todo el proceso.

Con la puesta en marcha de este diseño se obtendrá como resultado un proceso total, que mejorará en amplio sentido, la forma de manejar la reservación del pase masivo. Mediante el presente trabajo es preciso referirse a los **aportes** que brinda, tal como son: el desarrollo de un robusto análisis y diseño del sistema proporcionará la posterior fabricación de una aplicación web que facilite, agilice y organice el proceso de transportación masiva de estudiantes y trabajadores internos. También permitirá optimizar todos los recursos puestos a disposición, controlar de forma centralizada todo el proceso por el personal autorizado, mejorando así, todo el procedimiento concerniente a las reservaciones.

Los **métodos científicos** empleados en el transcurso de la investigación permitieron observar el problema cuidadosamente y con objetividad. Dentro el método empírico se encuentra la entrevista, de vital importancia y esencial, ya que define qué es lo que debe hacer la posterior aplicación web, identificando las funcionalidades y restricciones requeridas.

Para llevar a cabo esta investigación se hizo un análisis minucioso de la trayectoria histórico - lógico de los sistemas de transportación en la universidad.

Significativa es la utilización del análisis y la síntesis como método teórico ya que se hará un desglose de todo el proceso facilitando su entendimiento, lo que luego nos permite descubrir sus características y relaciones más específicas.

La inducción y la deducción se utilizan en varias fases de la investigación, donde de ideas generales podemos deducir unas más simples y llegar a conclusiones particulares.

El método de la modelación resultó necesario por el gran peso que tienen los modelos en el trabajo, donde la realidad es representada a través de un modelo que guía todo el proceso.

El trabajo se estructura en cinco capítulos:

Capítulo 1: Fundamentación Teórica

Este es un capítulo introductorio donde se muestran los principales conceptos manipulados en el transcurso de la investigación; se abordará además la situación actual fuera y dentro de la UCI. Se hace una breve referencia al estado del arte y a las herramientas y tecnologías utilizadas para dar solución al problema que se presenta.

Capítulo 2: Presentación de la Solución Propuesta

En este capítulo se describe el flujo actual de los procesos involucrados en el campo de acción, haciéndose un análisis profundo de cómo se desarrolla actualmente este proceso. Se abordan cuáles son las características del sistema y se hace una propuesta general de cómo debe funcionar el mismo.

Capítulo 3: Construcción de la Solución Propuesta

Este capítulo se orienta al análisis y diseño del software. Se muestran los diagramas de clases correspondientes, así como otros temas relacionados con el flujo de trabajo.

Capítulo 4: Estudio de Factibilidad

En este capítulo se realiza un estudio de la factibilidad del sistema, logrando obtener un costo aproximado de desarrollo, se tendrán en cuenta las utilidades que brinda el mismo, así como los beneficios tangibles e intangibles proporcionados.



CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se hace un estudio de las tendencias y tecnologías actuales sobre las que se apoyará la propuesta de solución. Se aborda sobre las herramientas seleccionadas a utilizar para el análisis y diseño del sistema, de forma tal que cumpla con los objetivos, teniendo en cuenta el desarrollo del software libre. Se plantean varios aspectos sobre otros sistemas de gestión de transporte existentes en el mundo, se da una breve descripción de cómo se lleva a cabo el proceso actual, y la necesidad e importancia de la informatización del mismo.

1.2 Conceptos básicos relacionados con el dominio del problema

- 1.2.1 Analista de procesos del negocio:** responsable de la arquitectura del negocio por lo que dirige y coordina el proceso de modelamiento del negocio. Decide cuáles son actores y los procesos del negocio y las relaciones entre ellos y cuáles son las reglas de negocio a tener en cuenta.
- 1.2.2 El diseñador del negocio:** Describe los procesos de negocio y como parte de la realización de estos procesos identifica a las entidades y trabajadores del negocio y sus relaciones. Define cuáles son los requerimientos en la automatización.
- 1.2.3 Analista del sistema:** define los alcances del sistema e identifica a los actores y casos de usos que permiten modelar completa y consistentemente el sistema.

1.2.4 Especificador de requerimientos: detalla la especificación de una parte de la funcionalidad del sistema describiendo el aspecto de los requisitos de uno o varios casos de uso y otros requisitos de soporte del software. El especificador de requerimientos puede también ser responsable de un paquete de casos de uso, y mantiene la integridad de ese paquete. Se recomienda que el especificador de los requisitos responsable de un paquete de casos de uso sea también responsable de sus casos de uso y actores contenidos.

1.3 Sistemas automatizados de reservación existentes en el mundo

Nuestra sociedad moderna manifiesta una clara tendencia a los viajes, al comercio internacional y al auge de las comunicaciones. Para ello es preciso mejorar constantemente la infraestructura de transporte y dar respuestas tecnológicas a los problemas que plantean el incremento del volumen de tráfico y la demanda de transporte público. La implementación de un sistema informatizado de transporte ayuda a que una organización logre mejoras continuas en su operación. Las reservas on-line traen consigo múltiples ventajas y un aumento considerable en la calidad de las prestaciones dirigidas a sus clientes. El uso de estos sistemas permite modernizar consecutivamente su misión, estrategias, operaciones y niveles de servicio. Además pone a disposición del usuario información sobre reservas disponibles en vuelos, barcos, autos y ferrocarriles. Permiten observar las diferentes tarifas, confeccionar agendas, hacer cambios y anulaciones, emisión de billetes, tarjetas de embarque y billete electrónico que logra un ahorro operativo y de papel pues el cliente no tiene que llevar ningún documento.

1.3.1 Airqueens.com

Air Queens (AQ) es una empresa argentina con una amplia experiencia en vuelos corporativos con el objetivo de brindar un servicio exclusivo y personalizado dentro del transporte aéreo, dispone de una página Web en Internet con la siguiente dirección: www.airqueens.com. Su pasión por el mundo aeronáutico hace que AQ siempre esté superándose para llegar a ser referente en el sector de la aviación privada, ofreciendo, entre otros servicios, vuelos para viajes de empresas, negocios o placer.

Este sitio está a la disposición del público las 24 horas del día y los 365 días del año. Su prestación permite ganar en tiempo, evita los retrasos de las aerolíneas y brinda una alta comodidad. AQ se centra en hacer llegar los beneficios de alquiler de aviones privados de manera sencilla y eficaz.

Ventajas de AQ:

- Disponibilidad: agilidad en las operaciones.
- Seriedad: máxima puntualidad.
- Privacidad: acceso al avión por zonas exclusivas de los aeropuertos.
- Control: el horario lo fija el cliente (no hay pérdidas de vuelos).
- Exclusividad: catering incluido durante el vuelo.
- Confianza: tripulaciones con alta experiencia.
- Tecnología: sistema de navegación de última generación.
- Seguridad: revisión de las aeronaves cada vuelo, no sólo por horas de vuelo, como las compañías comerciales.

Las reservaciones se realizan a través del llenado de una solicitud de cotizaciones, donde se entran los datos, los precios de las reservas varían de acuerdo a los aspectos introducidos por el viajero, ya sea la cantidad de pasajeros, clase de vuelo, tipo de vuelo y los servicios que se soliciten, teniendo en cuenta que las reservaciones on-line es más barato.

1.3.2 Transtur.cu

Transtur S.A. es un grupo empresarial cubano del transporte turístico, que pone a disposición en todo el país servicios de alquiler de ómnibus, microbuses y taxis, así como renta de autos con y sin chofer y panelería de carga ligera. Brinda además servicios técnicos automotores, comunicaciones y asistencia técnica en la vía. Surge en el 1980 como una nueva organización de transportación del turismo en Cuba. Cuenta con varias sucursales en todo el país. Para realizar la reserva on-line podemos visitar su página web www.transtur.cu, donde nos da una información detallada de los servicios que brinda. Transtur nos da la ventaja de escoger cualquier día del año, a la hora que se desee, y el lugar donde se esté para viajar al destino que se apetezca en una categoría de carro que el pasajero quiera.

1.3.3 Reservación del pase de fin de semana (pase.uci.cu)

Este es un sistema que funciona en la UCI con vista a que los estudiantes puedan hacer sus reservaciones los fines de semana hacia los diferentes municipios de La Habana. Para reservar el pase, el estudiante lo solicita hasta una determinada fecha. Esto permite reservar ida y regreso de acuerdo al municipio y ruta que haya solicitado. Si el estudiante no es de La Habana debe introducir en la aplicación datos como: nombre del familiar, parentesco, dirección y teléfono de la persona a visitar. Una vez realizada la reserva se hace la asignación del ómnibus a cada estudiante y se envía un reporte por correo electrónico con los detalles de la misma.

1.3.4 Pase Masivo

El proceso de reservación de pase masivo en la UCI es una tarea bastante difícil, se lleva a cabo de acuerdo a la planificación de la Vicerrectoría de formación. El especialista de transporte elabora un plan de transportación para coordinar con el MITRANS la asignación de los transportes necesarios de acuerdo a las cifras de viajeros por provincia y municipios, para la posterior distribución de las salidas/entradas. Producto a la complejidad del proceso nace la necesidad de informatizar el mismo para agilizarlo y organizarlo. Se probó hacer un sistema para automatizar dicho procedimiento, pero no cumple las funcionalidades requeridas que satisfagan las necesidades de los clientes.

Es imposible utilizar los sistemas anteriormente explicados pues todos ellos fueron confeccionados con características específicas para dar solución a problemas determinados de cada una de las empresas y compañías. Además no cumplen con las características particulares del proceso que se desea llevar a cabo, es decir no solucionan las necesidades específicas que requiere la elaboración del sistema que se precisa elaborar. A pesar de ser sitios con magníficas funciones y desarrollo no se ajustan a este sistema, ya que difieren en gran sentido de las particularidades que exige el sistema de reservación del pase masivo en la UCI.

1.4 Tendencias y tecnologías actuales

1.4.1 Las Tecnologías de la Información y las Comunicaciones (TICs)

Las Tecnologías de la Información y las Comunicaciones, mejor conocidas como TICs, se encargan del estudio, diseño, desarrollo, fomento y distribución de información, utilizando hardware y software. Las TICs más comunes son los computadores, celulares, la televisión, la radio, los dispositivos portátiles como PDAs, y los blogs, entre otros. Estos sistemas informáticos han transformado de una manera importante nuestra vida cotidiana, ya que modifican las barreras geográficas y de tiempo.

Las TICs tienen varias características interesantes. Una de las más importantes es la interactividad. El usuario de las TICs es sujeto activo, esto es que, puede comunicarse, logrando así una interacción persona-persona y persona-grupo, una opción no existente en el pasado. Además, por primera vez es posible transmitir en un mismo archivo informaciones multisensoriales, es decir, información de audio, video y datos.(FAJARDO 2008)

Entre los beneficios que aporta el empleo de las TICs están:

- Facilitan las comunicaciones.
- Eliminan las barreras de tiempo y espacio.
- Favorecen la colaboración entre distintas entidades.
- Aumentan la producción de bienes y servicios.
- Elevan la calidad de vida de los individuos.
- Aumentan las respuestas innovadoras a los retos del futuro.
- El internet, como herramienta estándar de comunicación, permite un acceso igualitario a la información y al conocimiento.
- Mejora de la administración y gestión de los centros.
- Recursos compartidos.

1.4.2 ¿Qué es Internet?

Internet es llamado como “red de las redes” o “autopista de la información”, significa que todos los ordenadores del mundo que tienen acceso a Internet forman parte de una gran comunidad, donde pueden compartir mensajes, archivos y otros grandes volúmenes de información a través de la red.

Podríamos decir entonces que Internet es una gran red mundial de ordenadores, que se comunican entre sí, porque están unidos a través de conexiones telefónicas o de otros tipos de conexiones y porque utilizan un lenguaje o protocolo común. (*RENa (red escolar nacional) 2005*)

El lenguaje común o protocolo de comunicación que utilizan se conoce como TCP/IP, permitiendo comunicar todo tipo de ordenadores desde los más grandes hasta los portátiles.

Los servicios más usados en Internet son: la transferencia de ficheros, servicios web, el correo electrónico, grupos de noticias, los foros de debate, las conversaciones en línea y servicios de telefonía.

1.4.3 Aplicaciones web

¿Qué es una aplicación web?

Las aplicaciones web son soluciones informáticas que los usuarios utilizan accediendo a un servidor a través de Internet o su red interna (intranet). Como interfaz con la aplicación se utiliza un navegador de Internet. (*SPL Sistemas de Información 2007*)

Las ventajas son múltiples:

- Curva de aprendizaje rápida, el concepto de hipervínculo está muy extendido entre los usuarios.
- Basadas en arquitectura cliente/servidor.
- Los datos centralizados en el servidor.
- Las actualizaciones son inmediatas, ya que no requieren instalación.
- Interfaz web.

- Los usuarios acceden a las aplicaciones mediante un navegador de Internet, por lo que el acceso a la aplicación puede realizarse desde diferentes tipos de máquinas, PCs, Macs, PDAs, etc. (*SPL Sistemas de Información 2007*)

1.4.4 Software libre

Software libre (en inglés free software) es la denominación del software que respeta la libertad de los usuarios y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie. (*Centro Nacional de Tecnologías de Información 2007*)

¿Por que migrar para software libre?

Se ha decidido migrar al software libre debido a los innumerables **beneficios** que trae consigo, explicados a continuación:

Razones económicas

Ahorros importantes al liberarse del pago de licencias y especialmente por la replicación casi gratuita de aplicaciones comunes a toda la administración pública. El muy bajo costo del software permitirá la ejecución de programa.

Independencia tecnológica

El Estado deja de depender de terceros (a menudo transnacionales) para el diseño, desarrollo y mantenimiento de sus sistemas de información, retomando el control total de sus procesos, en particular de los procesos críticos y de alta importancia estratégica para el país.

Control de la información

El acceso al código fuente, la libertad de inspeccionar el funcionamiento del software, la libertad de decidir la manera en que almacenan los datos y la posibilidad de modificar cualquiera de estos aspectos queda en manos del Estado, lo cual le permite el control total de la información y por consiguiente el ejercicio de la soberanía nacional.

Confiabilidad y estabilidad

El software libre realizado por comunidades está sometido a la inspección de un importante número de personas, este número de verificadores es mucho mayor que el del software propietario. Estas personas identifican los problemas, los resuelven, y comparten las soluciones con los demás. Por tal razón los programas libres de las comunidades gozan de gran confiabilidad y estabilidad.

Seguridad

La información que el Estado maneja generalmente es importante y/o confidencial, puede ser muy peligroso que esta información caiga en manos incorrectas. Por esta razón es imprescindible que el Estado pueda verificar que su software no tenga puertas de entrada traseras, voluntarias o accidentales, y que pueda cerrarlas en caso de encontrarlas; tal control sólo es posible con el software libre.

Desarrollo del país

Se genera transferencia tecnológica hacia los actores nacionales productores de software, acelerando el desarrollo endógeno y reforzando la soberanía nacional. (*Centro Nacional de Tecnologías de Información 2007*)

1.4.5 Ambiente de trabajo frameworks

El framework es usado en el desarrollo de sistemas software, no solo en aplicaciones web. Es una estructura de software compuesta por componentes para el desarrollo de una aplicación. Como objetivos principales tiene acelerar el proceso de desarrollo, reutilizar código y promover el uso de patrones. Es un esqueleto para el desarrollo y la implementación de una aplicación.

Proporciona varias **ventajas** como son:

- El programador no necesita definir una estructura de la aplicación, el framework le proporciona un esquema a seguir.
- Facilita encontrar las herramientas necesarias para el desarrollo.
- Ahorra tiempo y trabajo a los desarrolladores.

1.4.5.1 CakePHP

CakePHP es un marco de desarrollo (framework) libre y de código abierto para el desarrollo en PHP. Es una estructura de librerías, clases y una infraestructura en tiempo de ejecución (run-time) para programadores de aplicaciones web, originalmente inspirado en el framework Ruby On Rails. Su principal meta es permitir su trabajo de manera estructurada y rápida, sin pérdida de la flexibilidad.

CakePHP tiene varias características que lo hacen una gran opción como un framework para desarrollo de aplicaciones rápidas y con el menor costo de molestia. Algunas características son:

- Comunidad activa y amigable.
- Licencia flexible.
- Compatibilidad con PHP4 y PHP5.
- CRUD integrado para la interacción con la base de datos (BD) y las preguntas simplificadas.
- Arquitectura Modelo Vista Controlador (MVC).
- Despachador de peticiones con buena vista, URL personalizadas.
- Validación incorporada.
- Plantillas rápidas y flexibles (sintaxis PHP, con helpers).
- Ayudas en vistas para AJAX, Javascript, formularios HTML y más.

- Seguridad, sesiones y componentes para manejo de peticiones.
- Lista de control y acceso flexible.
- Desinfección de datos.
- Caché flexible en vistas.
- Trabaja desde cualquier subdirectorio web del sitio, con poca o ninguna configuración de Apache envuelta.

En términos de Cake el MVC se interpreta de la siguiente forma: el modelo (“Model”) representa una base de datos, tabla ó registro, así como sus relaciones con otras tablas ó registros. Los Modelos (“Models”) contienen las reglas para la validación de datos. Estas reglas son aplicadas cuando en el modelo (“model”) se insertan o actualizan datos. Las vistas en cake están representadas por los archivos “view”, los cuales son archivos de HTML con código PHP incluido. Los controladores (“controllers”) de Cake manejan las peticiones del servidor, toma las entradas del usuario (URL y datos de POST), aplica la lógica de negocios, utiliza los modelos (“models”) para leer y escribir en base de datos y otras rutas, y manda la salida apropiada de datos con los archivos de vistas (“view”).

Para hacer tan fácil como sea posible la organización de una aplicación, Cake usa este patrón no solo para manejar como interactúan los objetos dentro de la aplicación, si no también como son almacenados los archivos.

1.5 Metodología de desarrollo utilizada

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales. Aspectos positivos de las metodologías:

- Son interactivas e incrementales.

- Fácil de dividir el sistema en varios subsistemas independientes.
- Se fomenta la reutilización de componentes.(*Metodologías de Desarrollo de Software* 2007)

1.5.1 Rational Unified Process (RUP)

RUP llamada así por sus siglas en inglés Rational Unified Process, es un proceso para el desarrollo de un proyecto de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como 3 características esenciales están: **dirigido por los casos de uso**: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está **centrado en la arquitectura**: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es **iterativo e incremental**: donde divide el proyecto en miniproyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.(GALLEGO 2007)

La metodología RUP divide en 4 fases el desarrollo del software:

- **Inicio**: el objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**: en esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción**: en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición**: el objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de desarrollo

- Ingeniería de negocios: entiende las necesidades del negocio.
- Requerimientos: Traslada las necesidades del negocio a un sistema automatizado.
- Análisis y diseño: Traslada los requerimientos dentro de la arquitectura de software.

- Implementación: crea un software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: asegura que el comportamiento requerido es el correcto y que todo lo solicitado está presente. (SANCHEZ 2004)

Disciplina de soporte

- Configuración y administración del cambio: guarda todas las versiones del proyecto.
- Administrando el proyecto: administra horarios y recursos.
- Ambiente: administra el ambiente de desarrollo.
- Distribución: hace todo lo necesario para la salida del proyecto.(SANCHEZ 2004)

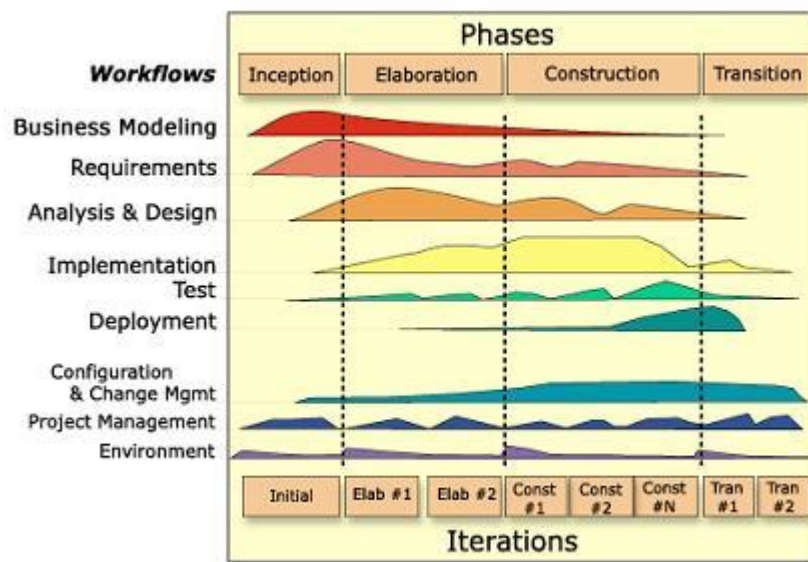


Figura 1.1 Fases e Iteraciones de la Metodología RUP

Los elementos del RUP son:

- **Actividades:** son los procesos que se llegan a determinar en cada iteración.

- **Trabajadores:** vienen a ser las personas o entes involucrados en cada proceso.
- **Artefactos:** un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.(SANCHEZ 2004)

1.6 Lenguaje de modelado utilizado (UML)

La falta de estandarización en la manera de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores. Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño.(ORALLO 2004)

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.

UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.

Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).(ORALLO 2004)

UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén

actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.(ORALLO 2004)

UML es ante todo un lenguaje. Un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Indica cómo crear y leer los modelos, pero no dice cómo crearlos.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.(ORALLO 2004)

Un modelo UML esta compuesto por tres clases de bloques de construcción:

- Elementos: los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: son colecciones de elementos con sus relaciones.(ORALLO 2004)

UML incluye los siguientes diagramas:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.

UML resuelve de forma satisfactoria problemas de desarrollo de software como es su modelado gráfico.

1.7 Herramienta de modelado utilizada

La toma de una acertada decisión a la hora de escoger una herramienta de modelado con UML, puede ser un factor importante para lograr la calidad de un proyecto.

Además de la aproximación metodológica y el lenguaje, uno de los factores determinantes en el éxito de un proyecto de software, lo constituye la acertada selección de las herramientas de trabajo que se utilizarán durante su desarrollo, es por eso que en la actualidad la administración de proyectos atiende cuidadosamente la designación de herramientas como una tarea de vital importancia.

Las herramientas CASE (Computer-Aided Software Engineering), están tomando cada vez más relevancia en la planeación y ejecución de proyectos que involucren sistemas de información, pues suelen inducir a sus usuarios a la correcta utilización de metodologías que le ayudan a llegar con facilidad a los productos de software construidos.

Todas las herramientas CASE prestan soporte a un lenguaje de modelado para acompañar la metodología y es lógico suponer, que un alto porcentaje de ellas soportan UML, teniendo en cuenta la amplia aceptación de este lenguaje y el valor conceptual y visual que proporciona, y su facilidad para extender el lenguaje para representar elementos particulares a determinados tipos de aplicaciones.(JUAN BERNARDO QUINTERO 2005)

Cuando se hace la planificación de la base de datos, la primera etapa del ciclo de vida de las aplicaciones de bases de datos, también se puede escoger una herramienta CASE que permita llevar a cabo el resto de tareas del modo más eficiente y efectivo posible. Una herramienta CASE suele incluir:

- Un diccionario de datos para almacenar información sobre los datos de la aplicación de bases de datos.
- Herramientas de diseño para dar apoyo al análisis de datos.
- Herramientas que permitan desarrollar el modelo de datos corporativo, así como los esquemas conceptual y lógico.
- Herramientas para desarrollar los prototipos de las aplicaciones.

El uso de las herramientas CASE puede mejorar la productividad en el desarrollo de una aplicación de BD. Y por productividad se entiende tanto la eficiencia en el desarrollo, como la efectividad del sistema desarrollado. La eficiencia se refiere al coste, tanto en tiempo como en dinero, de desarrollar la aplicación. La efectividad se refiere al grado en que el sistema satisface las necesidades de los usuarios. Para obtener una buena productividad, subir el nivel de efectividad puede ser más importante que aumentar la eficiencia.(ANDRÉS 2001)

Visual Paradigm (VP)

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. La herramienta está diseñada para una amplia gama de usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios.

Visual Paradigm para UML Enterprise: Es un producto que facilita a las organizaciones visualizar, crear diagramas de diseño, integrar y desplegar aplicaciones empresariales de misión crítica y de bases de datos subyacentes. La herramienta permite manejar grandes y complicadas estructuras de un proyecto.

Visual Paradigm para UML soporta un conjunto de lenguas como son: Java, C++, CORBA IDL, PHP, XML Schema, Ada y Python. Además soporta la generación de código C#, VB.NET, Object Definition Language (ODL), Flash ActionScript, Delphi, Perl, Objective-C, y Ruby. En ingeniería inversa también soporta clases Java, .NET dll y exe, JDBC, y archivos de mapeo Hibernate.

Visual Paradigm emplea una respuesta rápida y bajos requisitos de memoria, lo que permite manejar grandes y complicadas estructuras de proyecto de forma eficiente. Visual Paradigm para UML proporciona a los desarrolladores de software una herramienta vanguardia para crear aplicaciones de elevada calidad, rápidas y más baratas. Proporciona una mejor interfaz gráfica de usuario.

Estas son algunas características:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.

- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs (*Integrated Development Environment*).
- Disponibilidad en múltiples plataformas.
- Ofrece un mecanismo general para la organización de los modelos/subsistemas/capas agrupando elementos de modelado.

1.8 Patrones

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones. Surgieron en el año 1977, inventados por Christopher Alexander.

Cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que puedes usar esa solución un millón de veces más, sin hacer jamás la misma cosa dos veces. (*Arquitectura y Patrones de diseño* 2008)

Categorías de patrones

- **Patrones de arquitectura:** expresan un esquema organizativo estructural fundamental para sistemas software.
- **Patrones de diseño:** expresan esquemas para definir estructuras de diseño con las que se pueden construir sistemas software.
- **Idiomas:** Son patrones de bajo nivel específicos para un lenguaje de programación.

1.8.1 Patrón de diseño web

Grasp

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para indicar la importancia de

captar (grasping) estos principios, si se quiere diseñar eficazmente el software orientado a objetos.(LARMAN 1999)

Los 5 primeros patrones Grasp son:

Experto: es el encargado de asignar responsabilidades al experto en información. Con él se conserva el encapsulamiento de la información. Soporta un bajo acoplamiento, lo que ayuda a tener un sistema más sólido y de fácil mantenimiento. Una ventaja es la distribución del comportamiento del manejo de la información.

Creador: este patrón guía la asignación de responsabilidades que tengan que ver con la creación de objetos. Uno de sus propósitos principales es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Brinda soporte a un bajo acoplamiento, lo cual brinda oportunidades de reutilización y mayor mantenibilidad.

Bajo acoplamiento: asigna una responsabilidad de manera que el acoplamiento permanezca bajo. Es un principio a tener en cuenta en todas las decisiones del diseño: es un objetivo a tener en cuenta en todo momento. Es un patrón evaluativo que aplica un diseñador mientras evalúa todas las decisiones de diseño. Ayuda a asignar responsabilidades de forma que su localización no aumente el acoplamiento y se lleguen a producir resultados negativos producidos por un alto acoplamiento. Soporta clases más independientes. Dentro de sus ventajas están: no afecta cambios en otros componentes, fácil de entender de forma independiente e importante en el tema de la reutilización.

Alta cohesión: asigna responsabilidades de manera que la cohesión permanezca alta. Es un principio a tener en cuenta en todo momento en las decisiones del diseño. Es un patrón evaluativo que permite al desarrollador evaluar decisiones de diseño. Una clase con alta cohesión contiene un número relativamente pequeño de métodos, con funcionalidades grandemente relacionadas y no realiza un gran trabajo. Si la tarea es muy grande colabora con otros objetos para compartir el esfuerzo. Como ventajas se tiene que simplifica el mantenimiento y las mejoras, soporta el bajo acoplamiento a menudo, aumenta la reutilización y sobre todo facilita la comprensión del diseño.

Controlador: asigna la responsabilidad de manejar un mensaje de los eventos del sistema a una clase que represente una de las siguientes operaciones:

- Representa el sistema global (controlador de fachada).
- Empresa u organización global (controlador de fachada).
- Representa algo del mundo real que puede participar en una tarea (controlador de tareas).
- Representa un escenario de caso de uso en el que tiene lugar el evento del sistema (controlador de casos de usos)

Un controlador es un objeto de interfaz que se encarga de manejar un evento del sistema. Un controlador delega trabajo que se necesite hacer en otros objetos, además controla y coordina la actividad. Normalmente el objeto controlador es un objeto del lado del cliente. Dentro de las ventajas se ve el gran potencial que tiene para la reutilización de interfaces y hace una reflexión o razonamiento sobre el estado de los casos de usos.

1.8.2 Patrón de arquitectura

Modelo Vista Controlador

Este es un patrón de arquitectura de software que va a separar los datos de una aplicación, la interfaz del usuario, y la lógica de control en tres componentes diferentes. Este patrón está generalmente en aplicaciones web, donde la vista es la página HTML y el código que provee los datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos (SGBD) y el controlador es la lógica de negocio.

Descripción del patrón:

- **Modelo:** esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

1.9 Lenguaje de programación web utilizado

La programación web parte de las siglas www (World Wide Web). Para confeccionar una página web se debe tener en cuenta 3 conceptos fundamentales:

- URL (Uniform Resource Locators): es el sistema con el cual se puede localizar un recurso dentro de la red, es decir indica el camino hacia el recurso que se busca.
- HTTP (Hypertext Transfer Protocol): es el protocolo encargado de transportar a través de la red la información que contiene una página web.
- HTML (Hypertext Markup Language): es el lenguaje para representar la información que contiene la página web, es decir es el lenguaje para codificar y construir una página web.

1.9.1 PHP

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

PHP es un lenguaje de script interpretado en el lado del servidor, utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).(VALDÉS 2007)

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.

- Capacidad de conexión con la mayoría de los manejadores de BD: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.(VALDÉS 2007)

1.10 Conclusiones

En este capítulo se ha realizado un estudio de las tecnologías informáticas a emplear en la solución del problema, teniendo en cuenta las tendencias de migración al software libre. Se ha abordado sobre los conceptos fundamentales relacionados con el rol que juegan los analistas durante las distintas fases y flujos de desarrollo del software. Se han puesto ejemplos sobre sistemas automatizados de reservación existentes en el ámbito internacional, nacional y en la UCI, dando una breve descripción de la situación problemática y el por qué no utilizar ninguno de los sistemas explicados. Además se han dado a conocer características sobre la metodología, lenguaje y herramienta de modelado, patrones y lenguaje de programación a utilizar en el desarrollo del sistema.



CAPÍTULO 2

PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

En el presente capítulo se modela el negocio siendo este el primer flujo de trabajo durante el proceso de desarrollo de un software; además se identifican actores, trabajadores y casos de usos con sus correspondientes descripciones. Se muestran los requisitos funcionales y no funcionales que se proponen, lo que permite tener una concepción general del sistema y representar a través del diagrama de caso de uso del sistema las relaciones de los actores con los casos de uso.

2.2 Negocio

2.2.1 Modelo de negocio

El modelo del negocio es una técnica para comprender los procesos de negocio de la organización, permitiendo descubrir a más profundidad las necesidades de los clientes. Este modelo provee una descripción de dónde se va a ajustar el sistema de software considerado dentro de la estructura organizacional y de las actividades habituales. Dicho modelo tiene implicado varios objetivos entre los que están:

- Ayudan a comprender los procesos de negocio de la organización, identificar las mejoras potenciales y derivar los requerimientos del sistema que va a soportar la organización.
- Son un vehículo muy eficaz para capturar, organizar y visualizar requerimientos.
- Representan un lenguaje de comunicación eficaz entre clientes y desarrolladores.

2.2.2 Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. (*Fase de Inicio. Modelo del Negocio 2007*)

Se han distinguido las siguientes reglas para lograr garantizar y cumplir las restricciones existentes en el negocio:

1. La Vicerrectoría de formación es responsable de determinar las fechas de salida/entrada.
2. El especialista de transporte es el responsable de solicitar listados de las personas que van a viajar.
3. Akademos y Recursos humanos son los encargados de entregar al especialista de transporte los listados de los estudiantes y trabajadores internos respectivamente.
4. El especialista de transporte es el único que puede elaborar el plan de transportación y luego darlo a conocer a la universidad.
5. Todas las personas de la universidad pueden obtener información sobre las salidas, tipo de transportes y personas que viajarán.
6. Los únicos que pueden reservar son los estudiantes y trabajadores internos.
7. Cada viajero sólo puede hacer una reserva por viaje (ida/regreso).
8. El especialista de transporte es el único autorizado a realizar la distribución de los viajeros en los medios de transporte asignados (ómnibus, tren, barco).

2.2.3 Actores y trabajadores del negocio

Definición de los actores del negocio

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

Actores del negocio	Justificación
Vicerrectoría de formación	Es la oficina externa al negocio que se beneficia con el proceso de reservación de pase.
Akademos	Es el encargado de entregar el listado de todos los estudiantes registrados en la BD.
Recursos humanos	Es el encargado de entregar el listado con todos los trabajadores internos existentes en la UCI.

Tabla 2.1 Actores del Negocio

Definición de los trabajadores del negocio

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

Trabajadores del negocio	Justificación
Especialista de transporte	Es el encargado de solicitar los listados de los estudiantes y trabajadores internos a Akademos y Recursos humanos respectivamente. Elabora el plan de transportaciones desglosado por provincias y municipios, entrega este plan al MITRANS para su aprobación. Una vez aprobado el plan confecciona los listados de la distribución de los viajeros en los medios de transportes disponibles, los cuales entrega el día de la salida/entrada.
MITRANS	Aprueba el plan de transportación de acuerdo a las fechas y la disponibilidad de transporte y asigna los transportes por provincias.

Tabla 2.2 Trabajadores del Negocio

2.2.4 Descripción del negocio

Actualmente el proceso de reservación de pase masivo en la UCI se lleva a cabo de una forma bastante engorrosa, comienza cuando la Vicerrectoría de formación manda al especialista de transporte un documento con las fechas de salida/entrada en que debe planificar la transportación masiva de la universidad para estudiantes y profesores internos, esto lo hace con dos meses de antelación aproximadamente. Seguidamente de el especialista de transporte recibir dicho documento solicita a Akademos el listado de todos los estudiantes, y a Recursos humanos el listado de todos los trabajadores internos, una vez que tenga estos listado procede a la confección del plan de transportación de la universidad, el cual se desglosa por: provincia, municipios, fechas, puntos de concentración, hora, destino y la cantidad de personas por provincias.

Luego el especialista negocia este plan con el Ministerio de Transporte (MITRANS) donde se analizan las fechas y cantidad de viajeros y se le notifica si puede o no disponer del transporte en dicho plazo. De no poder adquirir los transportes en los días orientados para esa cifra de viajeros, el especialista de transporte se lo informa a la Vicerrectoría de formación para que planifique nuevas fechas, la misma hace el cambio pertinente informándoselo nuevamente al especialista para que vuelva a hacer la solicitud. Cuando se logra un acuerdo entre el MITRANS y la UCI en cuanto a los días planificados se le informa a la Vicerrectoría de formación y el MITRANS asigna a la UCI los transportes que necesite (ómnibus, tren ó barco en el caso de la Isla de la Juventud).

Continuamente se realizan los listados con la distribución de todos los viajeros por provincia y municipio en sus respectivos transportes. Luego el especialista publica mediante el correo electrónico el plan y la distribución efectuada. Llegado el día de la salida el especialista de transporte se dirige al lugar de salida entregando a un responsable de cada medio de transporte los listados con la distribución correspondiente. Al final de todo el proceso el especialista de transporte informa a la Vicerrectoría de formación que la salida masiva se llevó a cabo satisfactoriamente, por lo que la Vicerrectoría logra llevar a cabo el pase en las fechas establecidas.

2.2.5 Diagrama de casos de uso del negocio

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.

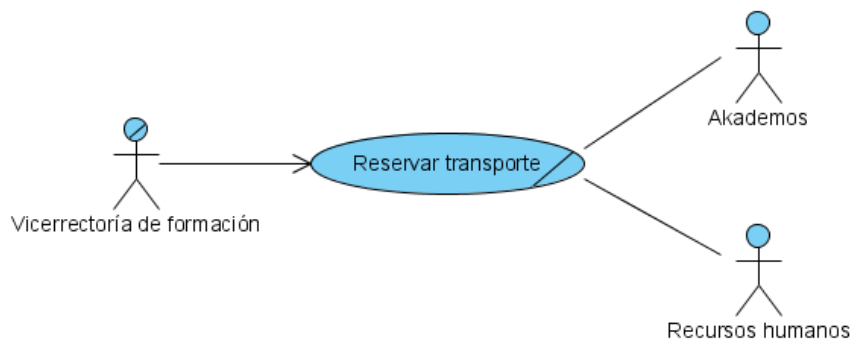


Figura 2.1 Diagrama de Casos de Uso del Negocio

2.2.6 Diagrama de actividades del negocio

Los diagramas de actividades ayudan a describir el detalle de qué es lo que pasa dentro del negocio, y para ello se examinan los roles específicos que juegan las personas (trabajadores del negocio) y las actividades que realizan. Ayudan a identificar qué funciones deberá asumir el producto de software, y quiénes serán los actores del futuro sistema. Describe gráficamente la estructura del flujo de actividades. Es un grafo de actividades que contiene estados en que puede hallarse una actividad.

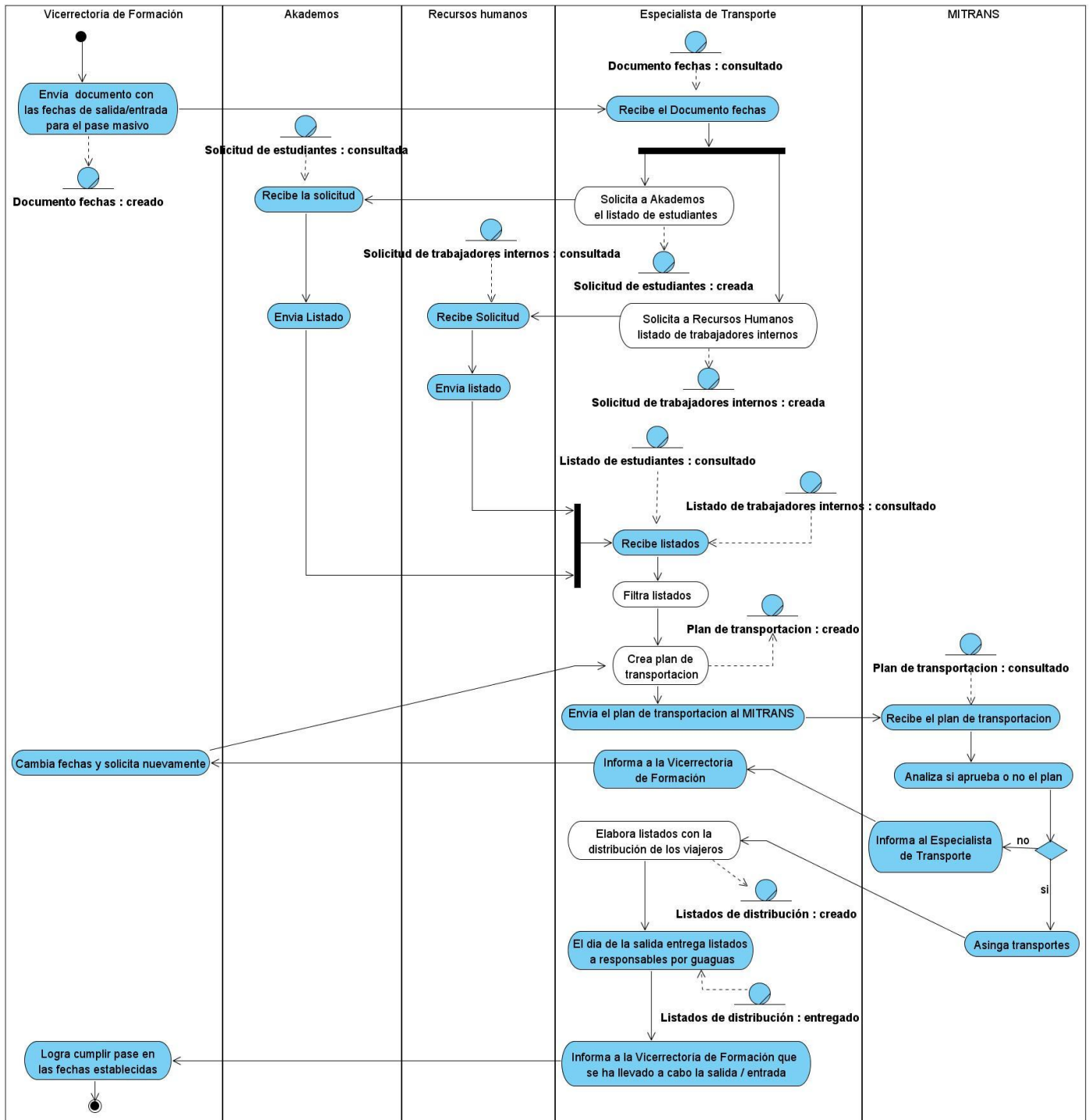


Figura 2.2 Diagrama de Actividades CUN: "Reservar transporte"

2.2.7 Descripción textual de los casos de uso del negocio

Nombre del Caso de Uso	Reservar transporte
Actor	Vicerrectoría de formación
Propósito	Realizar una correcta salida de pase masiva de estudiantes y trabajadores internos en la UCI en las fechas establecidas.
Resumen	El caso de uso comienza cuando la Vicerrectoría de formación envía al especialista de transporte un documento con las fechas para la salida/entrada masiva de estudiantes y trabajadores internos, este pide a Akademos y Recursos humanos los listados de los viajeros respectivamente, con estos datos confecciona el plan de transportación, para luego discutirlo con el MITRANS, de no ser aprobadas las fechas la Vicerrectoría de formación debe orientar unas nuevas para volver a hacer la solicitud. Una vez aprobado el plan se confecciona los listados con la distribución de los viajeros por (ómnibus, tren o barco) de acuerdo a su provincia y municipio.
Curso Normal de los eventos	
Acciones del Actor	Respuesta del proceso de negocio
1. Envía un documento con las fechas de salida/entrada del pase masivo al especialista de transporte con dos meses de antelación para que este lleve a cabo el proceso de solicitud de transporte.	2. Recibe el documento de la Vicerrectoría de formación.
	3. El especialista de transporte solicita a Akademos el listado de los estudiantes y a Recursos humanos el listado de todos los trabajadores internos.
	4. Akademos y Recursos humanos reciben la solicitud hecha por el especialista en transporte y entregan los mismos.
	5. El especialista de transporte elabora el plan de salida/entrada.

	6. El especialista de transporte va al MITRANS y presenta el plan de transportación para su aprobación en dependencia de las fechas y transportes disponibles.
	7. El MITRANS analiza las fechas y cantidad de viajeros por provincia.
	8. El MITRANS aprueba lo propuesto. Ver CA1.
	9. El MITRANS asigna los transportes que la UCI necesita (ómnibus, tren, barco).
	10. El especialista de transporte confecciona los listados con las distribuciones de los viajeros en los respectivos medios de transporte.
	11. El día planificado para la salida el especialista de transporte entrega por guagua el listado correspondiente de sus viajeros a un responsable.
	12. El especialista de transporte informa a la Vicerrectoría de formación que el pase masivo se llevó a cabo satisfactoriamente.
13. Logra que se cumpla el pase correctamente en las fechas establecidas.	
Curso Alternativo de los eventos	
CA1: De no aprobar el plan de salida/entrada el especialista de transporte se lo informa a la Vicerrectoría de formación, la cual cambia las fechas y vuelve a la actividad 9.	
Prioridad	
Mejoras	

Tabla 2.3 Descripción del CUN: “Reservar transporte”

2.2.8 Modelo de objetos del negocio

El modelo o diagrama de objetos del negocio, forma parte de los artefactos que propone la metodología RUP. Está formado por entidades y trabajadores del negocio y la relación entre ellos. Los trabajadores representan a personas o sistemas (software) dentro del negocio que son los que realizan las actividades, y las entidades del negocio representan un contenedor de información. La **Fig. 2.3** muestra el modelo de objetos del negocio.

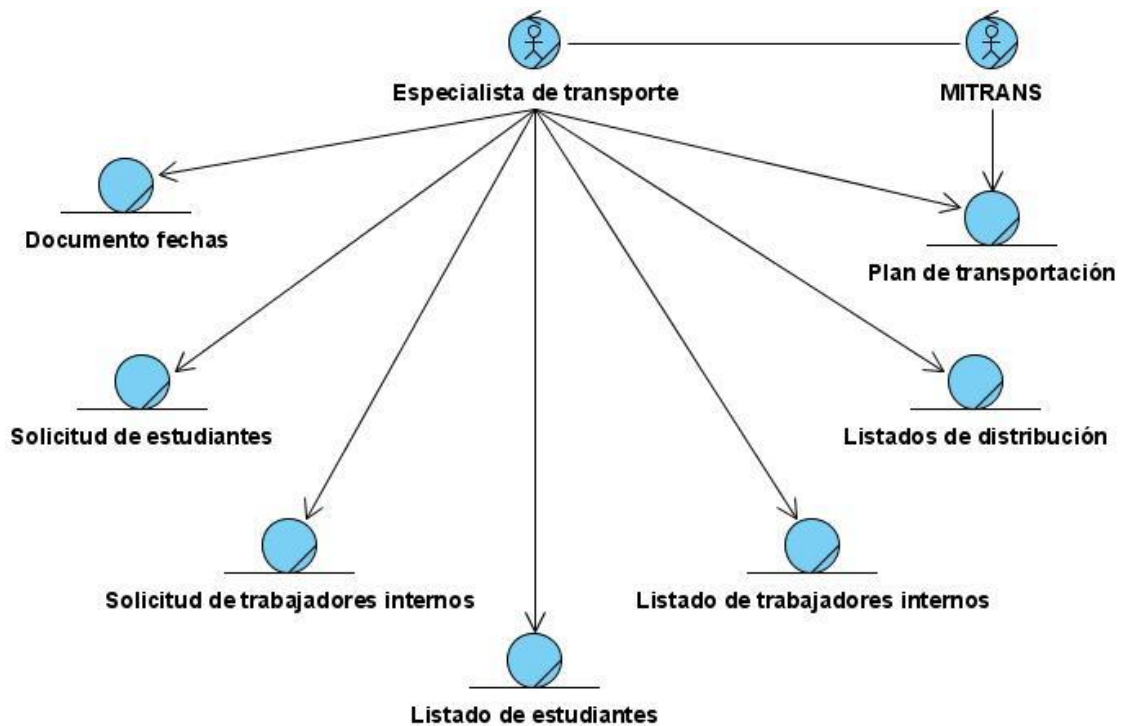


Figura 2.3 Modelo de Objeto del Negocio.

2.3 Especificación de los requisitos del software

1. La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

3. Una representación documentada de una condición o capacidad como en 1 o 2. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. (*Flujo de trabajo de requerimientos 2007*)

2.3.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Deben ser comprensibles por clientes, usuarios y desarrolladores. Deben tener una sola interpretación y estar definidos en forma medible y verificable. Se mantienen invariables sin importar con que propiedades o cualidades se relacionen. No alteran la funcionalidad del producto. Seguidamente se hará referencia a los requisitos funcionales del sistema:

1 Autenticar usuario

- 1.1 Pedir nombre de usuario y contraseña para entrar al sitio.
- 1.2 Introducir nombre de usuario y contraseña del dominio UCI.
 - 1.2.1 Validar datos introducidos usando el directorio activo UCI.
- 1.3 Mostrar al usuario las opciones a las que tiene acceso según el rol.

2 Gestionar usuario

- 2.1 Buscar usuario del dominio UCI.
- 2.2 Asignar un rol al usuario:
 - 2.2.1 **Administrador:** Acceso total al sistema. Permiso: control total.
 - 2.2.2 **Decanos:** Acceso a gestionar viajeros de su facultad y ver reportes. Permiso: lectura/escritura.
 - 2.2.3 **Residencia y Alimentos:** Acceso a los reportes. Permiso: lectura.
- 2.3 Insertar el usuario y el rol asignado en la BD.
- 2.4 Modificar usuario del sistema en cuanto al rol.

2.5 Cancelar el rol a un usuario del sistema.

2.6 Actualizar la BD cuando se ejecute alguna de estas acciones.

2.7 Mostrar sus datos en cada caso.

2.8 Registrar fecha, hora, IP y usuario cada vez que se realice algún cambio en el sistema por una persona autorizada.

3 Gestionar bloques de transportación

3.1 Crear bloques de transportación con los siguientes datos: fecha de creado, nombre del bloque y tipo de salida (estándar o personalizada).

3.2 Aplicar reglas al bloque (opcional).

3.3 Modificar bloques de transportación.

3.4 Eliminar bloques de transportación.

4 Gestionar viajes

4.1 Crear viaje con los siguientes datos: fecha, fecha y hora de inicio y fin del período de reservación y tipo de viaje (ida o regreso),

4.2 Aplicar reglas al viaje (opcional).

4.3 Modificar el viaje.

4.4 Eliminar el viaje.

5 Gestionar reglas

5.1 Crear reglas para restringir el acceso a la reservación por: provincia, municipio, año, tipo de persona, mundial, proyecto, tipo de viaje (ida o regreso) o facultad y establecer el orden en que se va a ejecutar la misma dentro del bloque o viaje al que pertenezca.

5.2 Eliminar reglas.

6 Bloquear usuario

6.1 Buscar usuario.

6.2 Mostrar de los usuarios (nombre y apellidos, carné de identidad (CI), facultad, grupo, provincia, municipio, localidad, solapín, foto).

6.3 Determinar un tiempo de bloqueo, fecha de inicio y fin del bloqueo.

6.4 Hacer observaciones de la sanción.

6.5 Bloquear usuario.

7 Gestionar reservación

7.1 Realizar reservación de viaje:

7.1.1 Reservar en el tipo de reservación disponible (estándar o personalizada) de acuerdo a las posibilidades que se brinden en cada caso:

7.1.1.1 Reservación estándar:

7.1.1.1.1 Si es estudiante confirma reservación y si desea puede entrar como dato la localidad.

7.1.1.1.2 Si es trabajador interno confirma la reservación con la posibilidad de escoger el destino (provincia y municipio) y si desea puede entrar como dato la localidad.

7.1.1.2 Reservación personalizada:

7.1.1.2.1 Ambos (estudiantes y trabajadores internos) reservan con la posibilidad de seleccionar el destino (provincia, municipio) y si lo desean pueden entrar como dato la localidad.

7.1.2 Registrar la fecha, IP y usuario para ambos tipos de reservación.

7.1.3 Adicionar los datos de la reservación a la BD de la aplicación.

7.1.4 Enviar correo de confirmación de la reservación realizada con la fecha, hora, IP, provincia, municipio y localidad de la reservación.

7.2 Modificar reservación de viaje

7.2.1 Mostrar al usuario los datos de la reservación hecha anteriormente con la fecha, hora y el IP de la última modificación que hizo en el sistema.

7.2.2 Verificar el plazo establecido para llevar a cabo la modificación, el cual debe estar dentro del período de reservación, de lo contrario solo podrá cancelar la reservación hasta el día de la salida.

7.2.3 Modificar los parámetros de interés:

7.2.3.1 Posibles parámetros a modificar: provincia, municipio, localidad y registra la fecha, hora y el IP de la última modificación.

7.2.4 Actualizar la BD.

7.2.5 Enviar por correo electrónico un mensaje de confirmación de la nueva reservación con los siguientes datos: provincia, municipio, localidad y la fecha, hora e IP de la máquina donde se hizo la modificación.

7.3 Cancelar reservación de viaje

7.3.1 Mostrar la(s) reservación(es) efectuada(s) por el usuario con sus datos: provincia, municipio, localidad y la fecha, hora y el IP de la última modificación que hizo en el sistema.

7.3.2 Verificar el tiempo que se estableció para la cancelación, este estará vigente hasta el día de la salida.

7.3.3 Eliminar la reservación de la BD.

7.3.4 Enviar por correo electrónico mensaje de confirmación con la provincia, municipio, localidad y la fecha, hora y el IP de la máquina en la que canceló.

7.4 Realizar reservación a un familiar

- 7.4.1 Mostrar los viajes a los que tiene acceso el usuario.
- 7.4.2 Brindar la posibilidad de realizarle la reservación a uno o varios familiares en el / los viaje(s) que haya(n) sido reservado(s) por el usuario.
- 7.4.3 Elegir el viaje, de los disponibles, en el que el familiar desea reservar.
- 7.4.4 Entrar sus datos y los de la reservación para realizarle la misma.
- 7.4.5 Mostrar mensaje de confirmación una vez realizada la reservación.
- 7.4.6 Registrar el usuario (administrador) que le realizó la reservación al familiar y la fecha, hora e IP del la máquina donde se hizo la acción.
- 7.4.7 Enviar un correo al usuario del dominio UCI con la notificación de que se le ha agregado el familiar, la fecha, hora, IP de la máquina y el usuario (administrador) que le hizo la reservación al familiar.

7.5 Cancelar reservación a un familiar

- 7.5.1 Mostrar los viajes a los que tiene acceso el usuario.
- 7.5.2 Brindar la posibilidad de acceder a su(s) familiar(es) para cancelarle la reservación.
- 7.5.3 Mostrar el / los familiar(es) y posibilitar la cancelación de su reservación.
- 7.5.4 Cancelar la reservación del familiar.
- 7.5.5 Mostrar mensaje de confirmación una vez realizada la cancelación.
- 7.5.6 Registrar el usuario (administrador) que le canceló la reservación al familiar y la fecha, hora e IP del la máquina donde se hizo la acción.
- 7.5.7 Enviar correo electrónico al usuario del dominio UCI con la notificación de que se le ha cancelado la reservación al familiar, la fecha, hora, IP de la máquina y el usuario (administrador) que le hizo la reservación al familiar.

8 Gestionar viajero

- 8.1** Realizar búsqueda de un usuario del dominio UCI.
- 8.2** Gestionar reservación del usuario buscado y la de sus familiares.
- 8.3** Posibilitar al decano el acceso al gestionar reservación de un usuario perteneciente a su facultad para cancelarle la reservación.
- 8.4** Gestionar reservaciones puntuales por parte del administrador.
- 8.5** Registrar en cada caso la fecha y hora, IP de la PC y usuario del administrador o decano que realizó la acción (reservar, modificar o cancelar según corresponda) sobre la persona buscada.
- 8.6** Enviar al usuario gestionado un correo con la notificación de la acción realizada, la fecha, hora, IP de la máquina y el usuario del administrador o decano que realizó dicha acción.

9 Gestionar reportes

9.1 Facultades

- 9.1.1** Hacer reportes de estudiantes por: facultad, año y mostrar el total en cada caso y general.
 - 9.1.1.1** Para el caso de los estudiantes cada reporte debe tener: Nombre, C.I, provincia, municipio, localidad, grupo.
- 9.1.2** Hacer reportes para trabajadores internos: área y mostrar el total en cada caso y general.
 - 9.1.2.1** Para el caso de los trabajadores cada reporte debe tener: nombre, C.I, provincia, municipio, localidad, cargo, edificio, apartamento, teléfono.
- 9.1.3** Posibilitar imprimir cualquier nivel de reporte.

9.2 Provincia

- 9.2.1** Hacer reporte para conocer la cantidad de viajeros por provincia y el total.
- 9.2.2** Hacer reporte para conocer la cantidad de viajeros por municipio y el total.
- 9.2.3** Mostrar de los viajeros CI, nombre, facultad / área, año, solapín, usuario, edificio, apartamento, teléfono.

9.2.4 Posibilitar imprimir cualquier nivel de reporte.

9.3 Fecha

9.3.1 Hacer reporte para conocer la cantidad de reservaciones por fecha.

9.3.1.1 De la fecha el total de personas que van a viajar.

9.3.1.2 Distribuidos por área (trabajadores) y por facultad (estudiantes).

9.3.2 Posibilitar imprimir cualquier nivel de reporte.

9.4 Usuarios bloqueados

9.4.1 Hacer reporte para conocer de los usuarios bloqueados: nombre, usuario, facultad, provincia, municipio, fecha inicio/fin del bloqueo y motivo.

9.4.2 Posibilitar imprimir reporte.

9.5 Familiares

9.5.1 Hacer reporte para conocer la cantidad de familiares que van a viajar por cada provincia.

9.5.2 Hacer reporte para conocer la cantidad de familiares que van a viajar por cada municipio.

9.5.3 Hacer reporte para conocer la cantidad de familiares que van a viajar con un destino diferente al de la persona UCI al que están adjuntos. El reporte debe mostrar los siguientes datos: CI, nombre, municipio y parentesco del familiar; nombre, provincia/municipio de la persona UCI.

9.5.4 Mostrar el total en cada caso.

9.5.5 Posibilitar imprimir cualquiera de estos reportes.

10 Gestionar provincias

10.1 Crear las provincias con los siguientes datos: nombre de la provincia, descripción, fecha de creada y tipo de transportación.

10.2 Modificar provincia.

10.3 Eliminar provincia.

11 Gestionar transportes

- 11.1 Crear transporte (ómnibus, barco o tren) con los siguientes datos: tipo, nombre, capacidad, fecha y hora de salida, fecha y hora de llegada, puntos intermedios, punto de salida, punto de llegada, profesor responsable, provincia, municipio, punto de concentración y asientos en el caso del ómnibus y el barco porque en el tren los asientos serían por coches.
- 11.2 Crear los coches que componen el tren con los siguientes datos: capacidad, profesor responsable y asientos. Con la posibilidad de modificarlos y eliminarlos.
- 11.3 Modificar datos del transporte.
- 11.4 Eliminar transporte.

12 Gestionar distribución

- 12.1 Distribuir por: provincia, municipio, orden de reservaciones, por año y facultad los estudiantes y por área los trabajadores internos.
- 12.2 Distribuir una o varias personas en un determinado transporte.
- 12.3 Mostrar las capacidades totales y las capacidades que van quedando después de la distribución por provincia y municipio.
- 12.4 Deshacer una distribución por ómnibus, provincia, municipio o usuario.

13 Consultar Información

- 13.1 Buscar datos del viajero de acuerdo a los patrones de búsqueda:
 - 13.1.1 Introducir datos para la búsqueda por: usuario, CI, solapín.
 - 13.1.2 Mostrar información de acuerdo al tipo de usuario:
 - 13.1.2.1 Usuario: Muestra los datos de la reservación en el medio de transporte de la ida y regreso (lugar, fecha y hora de salida, nombre, puntos intermedios, lugar, fecha y hora de llegada, salida, provincia / municipio).

13.1.2.2 Administrador: Muestra los datos de la reservación en el medio de transporte de la ida y regreso (lugar, fecha y hora de salida, nombre, puntos intermedios, lugar, fecha y hora de llegada, salida, provincia / municipio), así como la fecha, IP y hora de la última modificación hecha por el usuario.

14 Gestionar boletines

14.1 Generar documento con formato que permita imprimir los datos de la reservación hecha por los usuario (nombre, CI, facultad / área, grupo, provincia, municipio, localidad, fecha / hora de (salida y entrada), punto de concentración (salida y entrada), nombre del ómnibus).

14.2 Seleccionar la distribución que se desea imprimir (provincia, municipio, tipo de transporte, persona, facultad y grupo).

14.3 Imprimir selección.

2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, estas pueden ser características que hagan al producto atractivo, usable, rápido o confiable. Especifican propiedades del sistema como restricciones de ambiente y desarrollo, performance, dependencias de plataformas, mantenibilidad y confiabilidad. Están relacionados con los requerimientos funcionales pues una vez definido lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener y cuán rápido o grande debe ser. A continuación se mostrarán los requisitos no funcionales del sistema:

• Apariencia o interfaz externa

- El sistema tendrá una interfaz sencilla, amigable y con rápida respuesta del sistema.
- El empleo de imágenes y colores estarán relacionados con el negocio donde se implantará el sistema.
- Tendrá una interfaz de fácil navegación para los usuarios, donde los usuarios no necesiten adiestramiento para su utilización.

- **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, ya que los usuarios finales serán los estudiantes y trabajadores internos de la UCI

- **Portabilidad**

- El sistema deberá ser multiplataforma, es decir debe poder ejecutarse sobre los Sistemas Operativos Windows y Linux.

- **Rendimiento**

- Estará implementado sobre una tecnología web, para facilitar su uso en la red.
- Las páginas estarán poco cargadas de imágenes para garantizar repuesta rápida ante la solicitud de cualquier usuario.

- **Seguridad**

- Las acciones que se llevarán a cabo serán restringidas de acuerdo al rol que desempeñe cada usuario, que es el que define su nivel de acceso.
- Garantizar que la información sea solo manejada por quien tiene ese derecho.
- Asegurar las verificaciones sobre acciones definitivas (ejemplo: eliminar).

- **Confiabilidad**

- Validar la entrada de usuarios.
- El sistema debe ser lo más preciso posible evitando error en la información que se le suministra al usuario.
- La información que manejará el sistema debe estar protegida de cualquier acceso no autorizado.

- **Integridad**

- La información que va a manejar el sistema debe estar bajo estricta protección contra acciones inconscientes, por lo que deberán existir mecanismos de chequeo.

- **Disponibilidad**

- Estará disponible las 24 horas del día durante todo el año.
- Los usuarios autorizados tendrán acceso a la información siempre que lo deseen.

- **Funcionalidad**

- Sencillez a la hora de la navegación, minimizando la cantidad de páginas para ejecutar todas las funciones posibles y que las mismas estén preferentemente relacionadas.

- **Software**

- Para el cliente:
 - (1) Navegador Mozilla FireFox 2.0 ó Internet Explorer 5.5 ó superior.
 - (2) Sistema operativo Linux ó Windows 98 ó superior.
- Para el servidor:
 - (1) Sistema operativo Linux.
 - (2) Servidor web Apache y PHP 5.
 - (3) Servidor de Base de Datos PostgreSQL 8.2.

- **Hardware**

- Requerimientos mínimos para el cliente:
 - (1) Procesador Pentium III a 1GHz de velocidad de procesamiento y 128MB de memoria RAM.
 - (2) Tarjeta de red.
- Requerimientos mínimos para el servidor:

- (1) Procesador Intel Pentium Dual Core a 3.0HGz de velocidad de procesamiento y 2Gb de memoria RAM.
- (2) 50Gb de espacio libre en disco.
- (3) Tarjeta de red.

2.4 Descripción del sistema propuesto

El sistema que se propone diseñar cumplirá una serie de requisitos para satisfacer en su totalidad a los usuarios finales en cada una de las acciones que se llevarán a cabo en la aplicación. Primeramente está **autenticar usuario**, donde luego de introducir su usuario y contraseña del dominio UCI y ser validados por el sistema y comprobados en el directorio activo de la UCI; si es estudiante los datos a mostrar se extraen de Akademos y si es trabajador de Recursos humanos. Posteriormente podrá acceder a la aplicación mostrándole las opciones a las que tiene acceso según su rol. Una vez realizada la autenticación si el usuario es administrador verá la opción de **gestionar usuario** donde mediante una búsqueda de un usuario del dominio UCI se le asignará un rol como puede ser: UsuarioGR (administrador, decano, residencia, alimentos), UsuarioGV (administrador, decano) y administrador; también se podrá modificar o cancelar un rol anteriormente asignado; de cada acción realizada se actualizará la BD y se registrará la fecha, hora, IP y usuario del administrador que realizó dichas acciones. Seguidamente el administrador podrá **gestionar los bloques de transportación**, donde se crearán dichos bloques, a los que continuamente, si es necesario, se le añadirán reglas mediante la funcionalidad **gestionar reglas para el bloque** en caso que se desee restringir el acceso al mismo, dando la posibilidad de eliminarlas, así como tener la posibilidad de modificar y eliminar los bloques antes creados.

Dentro del bloque se brindará la posibilidad de **gestionar viajes** posibilitando crear, modificar y eliminar los mismos, una vez creado un viaje se le podrán añadir reglas mediante la funcionalidad **gestionar reglas para el viaje** en caso que se desee restringir el acceso a dicho viaje, estas podrán ser eliminadas.

Al administrador se le brindará la opción de **bloquear usuario**, donde primeramente se buscará el usuario deseado, se mostrarán sus datos y dará la oportunidad de aplicarle dicho bloqueo en un rango de tiempo escogido y de introducir el motivo por el cual fue bloqueado.

Posteriormente los usuarios podrán **gestionar su reservación**, esta funcionalidad incluye la gestión de los familiares por parte del administrador, en un viaje ya sea para reservar, modificar o cancelar la misma

de acuerdo a las posibilidades que se brinden en el tipo de reservación disponible (estándar o personalizada); se registrará fecha, IP y usuario de acceso al sistema en caso de llevarse a cabo algún tipo de gestión, estos datos serán adicionados a la BD y luego se le enviará a dicho usuario, por correo, la confirmación de la operación realizada. Las acciones de reservar y modificar reservación se realizan si están dentro del plazo o período de tiempo establecido para ese viaje, el administrador no tiene esta restricción.

El UsuarioGV (administrador, decano) si es administrador podrá **gestionar viajero**, donde tendrá la posibilidad de gestionarle la reservación a una persona que puede pertenecer o no al dominio UCI, en caso de pertenecer a dicho dominio, una vez realizada su reservación, podrá adjuntársele al usuario uno o varios familiares en el viaje que se desee. Si la persona no pertenece al dominio se le podrá gestionar una reservación puntual, siempre mostrando mensajes de confirmación de las acciones realizadas. El decano podrá cancelarle la reservación a un estudiante de su facultad. Registrará el usuario, fecha, IP y hora de la última acción llevada a cabo por el administrador o decano y enviará al usuario del dominio que se le haya realizado algún cambio en su reservación un correo de notificación.

Dentro de un viaje al UsuarioGR (administrador, decano, residencia, alimentos) se le van a ofrecer una serie de reportes a los cuales podrán acceder en su totalidad los administradores, residencia y alimentos; los decanos podrán ver el reporte por facultad. Los reportes se realizan a través de la funcionalidad **generar reportes**, los mismos se brindarán por: facultades, provincia, fecha, usuarios bloqueados y familiares. En todos los casos da la oportunidad de imprimir el reporte deseado.

Dentro de un viaje un administrador con la funcionalidad **gestionar provincias** podrá crear las provincias hacia las cuales existirán viajes con los siguientes datos: nombre de la provincia, descripción, y tipos de transportación; luego de creadas se permitirá modificarlas o eliminarlas. Dentro de una provincia se crean los transportes que viajarán a la misma con la funcionalidad **gestionar transporte** donde el administrador crea los transportes de acuerdo al medio (ómnibus, tren, barco), con la posibilidad de modificar y eliminar el transporte creado de ser necesario. En caso de crear un tren para gestionar los coches se utiliza la funcionalidad **gestionar coches** que permite crear, modificar y eliminar los mismos.

Dentro de un viaje la funcionalidad **gestionar distribución** es una función de vital importancia ejercida por el administrador donde se distribuirá a los viajeros ya sea por provincia, municipio, orden de reservaciones, por año y facultad los estudiantes y por área los trabajadores internos, de ser deseado se

podrá distribuir una o varias personas en un determinado transporte; se mostrarán las capacidades totales y las que van quedando luego de la distribución; en el momento que queden pocas capacidades libres, número determinado por el administrador, deberá mostrar un mensaje de aviso; brindará la opción de deshacer una distribución antes realizada.

Todos los usuarios del sistema tendrán la posibilidad de **consultar información** de su reservación y si lo desea la de otros viajeros mediante una búsqueda. En caso de que sea un administrador el que esté consultando información mediante una búsqueda se le mostrará además la fecha, IP y hora de la última modificación hecha por el usuario.

Dentro de cada viaje está **generar boletines** donde se generará un documento con formato que permita imprimir los datos de la reserva de un usuario, escogiendo la distribución que se deseada para su posterior impresión.

Cumpliendo con todas estas funcionalidades se logrará un producto de excelente calidad y sobre todo satisfaciendo las necesidades esenciales del cliente.

2.4.1 Descripción de los actores del sistema

Actores del sistema	Justificación
Usuario	Son los estudiantes y trabajadores internos de la UCI que utilizan el sistema para realizar, modificar o cancelar su reservación luego de autenticarse en el mismo, además consultan información de su reserva, así como los datos de la reserva de un usuario buscado.
UsuarioGR	Son los administradores, decanos, residencia y alimentos que tienen acceso a ver todos los reportes que brinda la aplicación. Los decanos solo tienen acceso a ver los reportes por facultad.
UsuarioGV	Son los administradores y decanos que tiene acceso a gestionar un viajero, este último solo con la posibilidad de cancelarle la reservación a un estudiante de su facultad.
Administrador	Gestiona toda la parte administrativa del sistema. Puede realizar cualquier acción.

Akaderos	Brinda los datos del estudiante autenticado.
Recursos humanos	Brinda los datos del trabajador autenticado.

Tabla 2.4 Actores del Sistema.

2.4.2 Diagrama de casos de uso del sistema

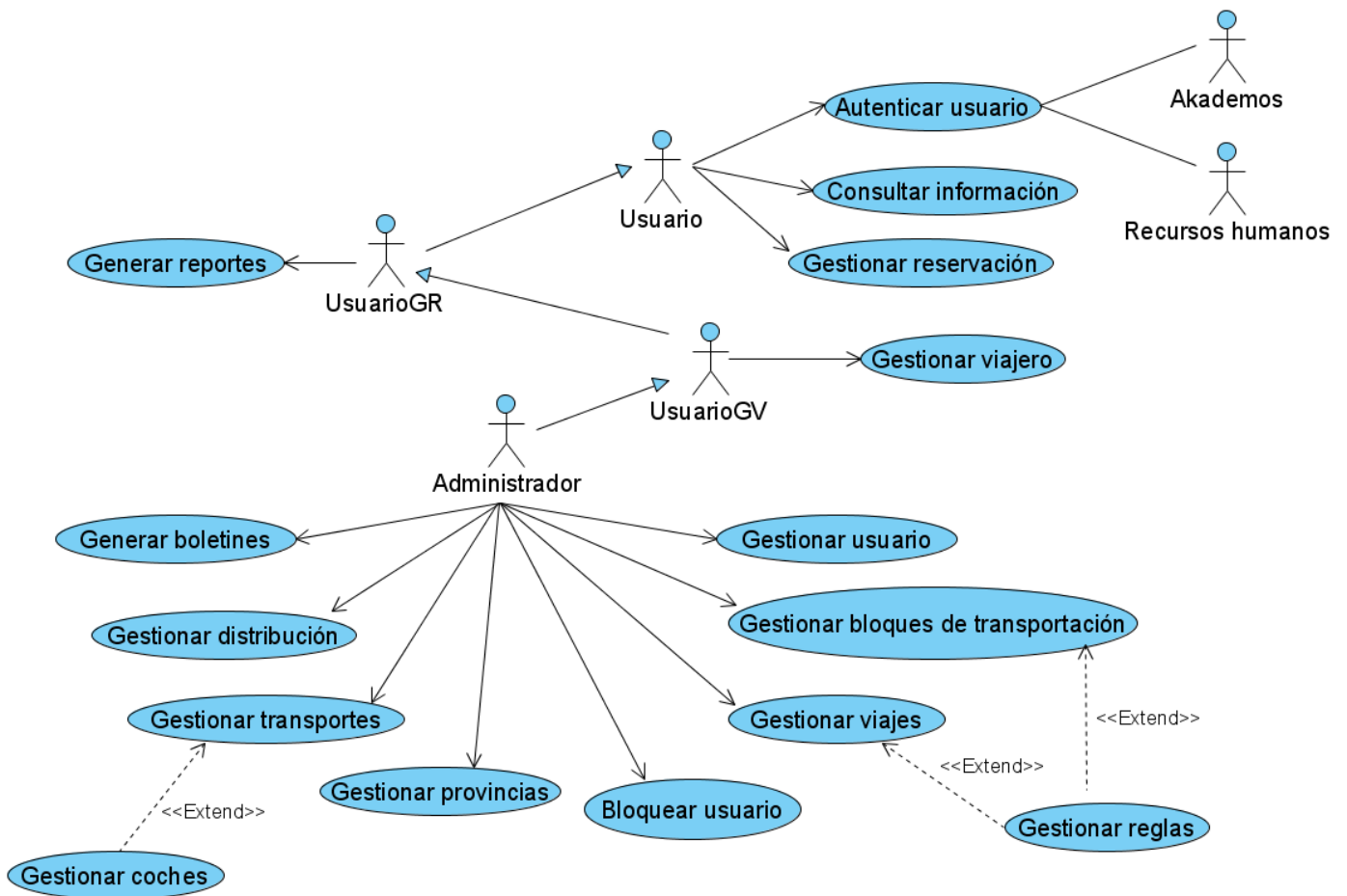


Figura 2.4 Diagrama de Casos de Uso del Sistema.

2.4.3 Descripción de los casos de uso del sistema

Para ver la descripción de los casos de usos del sistema ver **Anexo 1**.

2.5 Conclusiones

En es este capítulo se realizó la modelación del negocio, a fin de lograr una mejor comprensión de los procesos del mismo, logrando que los usuarios finales y desarrolladores tengan un entendimiento común. Se obtuvo el diagrama de casos de uso del negocio, así como la descripción de cada uno de ellos y se elaboró el diagrama de actividades. Se identificaron y describieron los actores y trabajadores del negocio, Además, se realizó el modelo de objetos del negocio. Se identificaron los requerimientos del sistema, obteniéndose así las funcionalidades que este debe tener, además se obtuvo un listado de requerimientos no funcionales a tener en cuenta para el desarrollo de la aplicación. Se realizó la descripción del sistema propuesto donde se explican de forma ordenada los requerimientos funcionales. Se identificaron y describieron los actores del sistema. Y por último se muestra el diagrama de casos de uso el sistema.

3
CAPÍTULO 3

CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

En este capítulo se comienza a construir la solución del problema descrito anteriormente a través del flujo de trabajo de análisis y diseño. Durante el análisis se analizan los requisitos descritos en el capítulo anterior, refinándolos y estructurándolos, para lograr una comprensión más precisa y así ayudar a estructurar el sistema; se dará una vista interna del producto para el mejor entendimiento de los desarrolladores. En el diseño se modela el sistema y se crea un punto de partida a la implementación. Se abordan aspectos relacionados con este flujo de trabajo como son las clases de análisis y diseño, diagramas de interacción, descripción de clases de diseño y diagrama de clases persistentes.

3.2 Modelo de análisis

3.2.1 Clases del análisis

Una clase de análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema. (IVAR JACOBSON 2000) Las clases de análisis definen atributos, participan en las relaciones y encajan en uno de tres estereotipos básicos como son: de interfaz, de control o de entidad; estos estereotipos se utilizan para ayudar a los desarrolladores a distinguir el ámbito de las diferentes clases.

A continuación se muestran algunas clases de análisis, para ver el resto de las clases, ver **Anexo 2**.



Figura 3.1 Diagrama de clases del análisis: CU “Autenticar usuario”

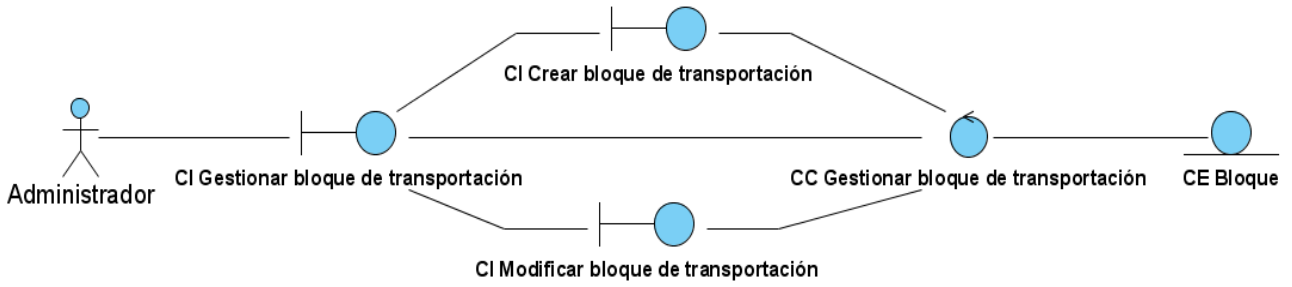


Figura 3.2 Diagrama de clases del análisis: CU "Gestionar bloques de transportación"

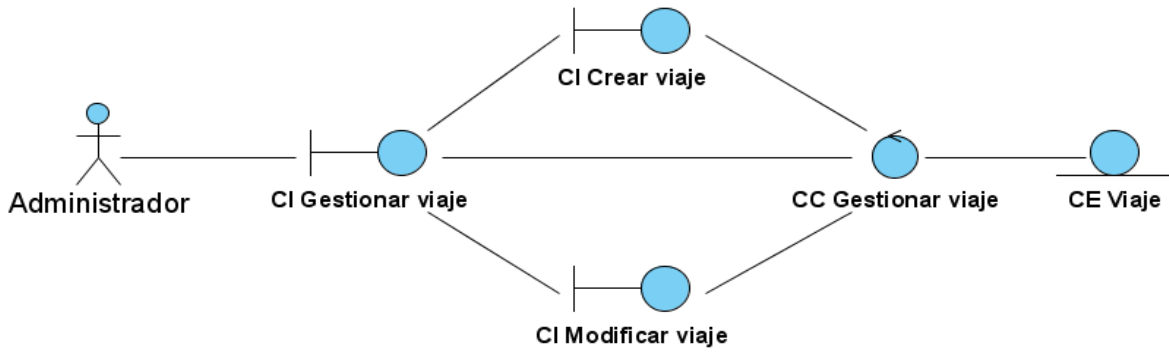


Figura 3.3 Diagrama de clases del análisis: CU "Gestionar viaje"

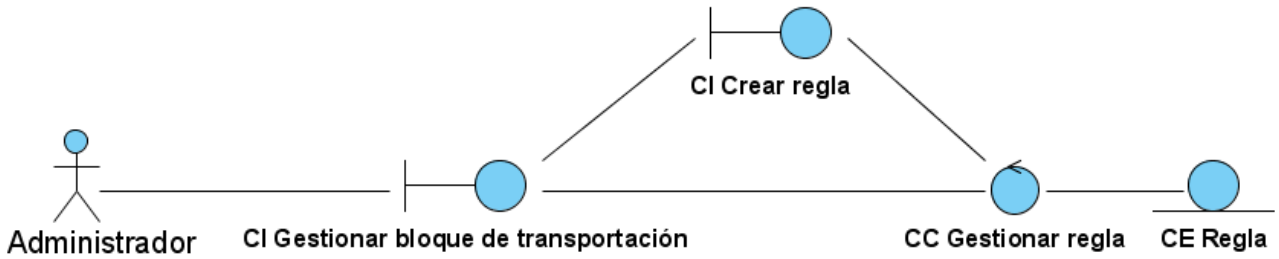


Figura 3.4 Diagrama de clases del análisis: CU "Gestionar reglas"

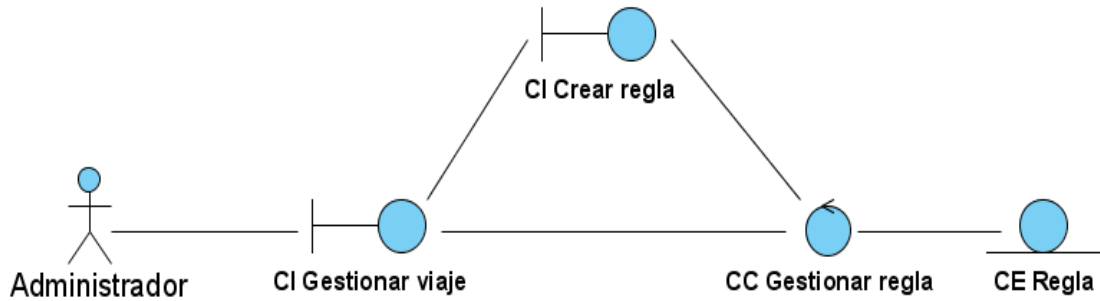


Figura 3.5 Diagrama de clases del análisis: CU “Gestionar reglas”

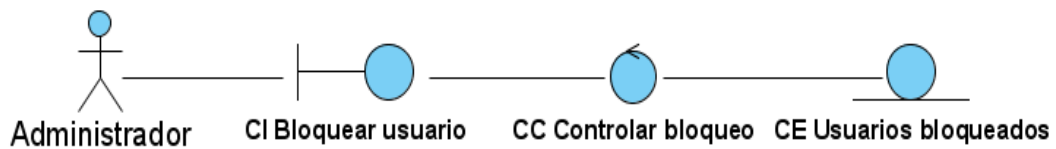


Figura 3.6 Diagrama de clases del análisis: CU “Bloquear usuario”

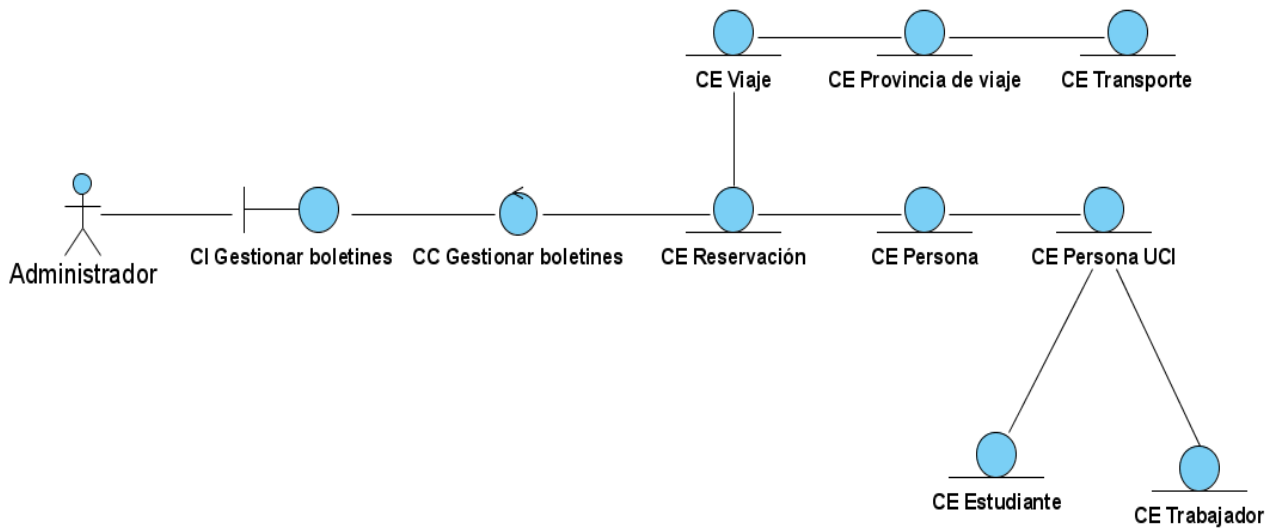


Figura 3.7 Diagrama de clases del análisis: CU “Generar boletines”

3.3 Modelo de diseño

3.3.1 Clases del diseño

Una clase del diseño es una abstracción sin costuras de una clase o construcción similar en la implementación del sistema. (IVAR JACOBSON 2000) El lenguaje que se utiliza en las clases de diseño es el mismo que el de programación; se especifican atributos y operaciones; puede realizar interfaces si tienen sentido para la programación; los métodos tienen correspondencia directa con los métodos en la implementación.

A continuación se muestran algunas clases del diseño, para ver el resto de las clases, ver **Anexo 3**.

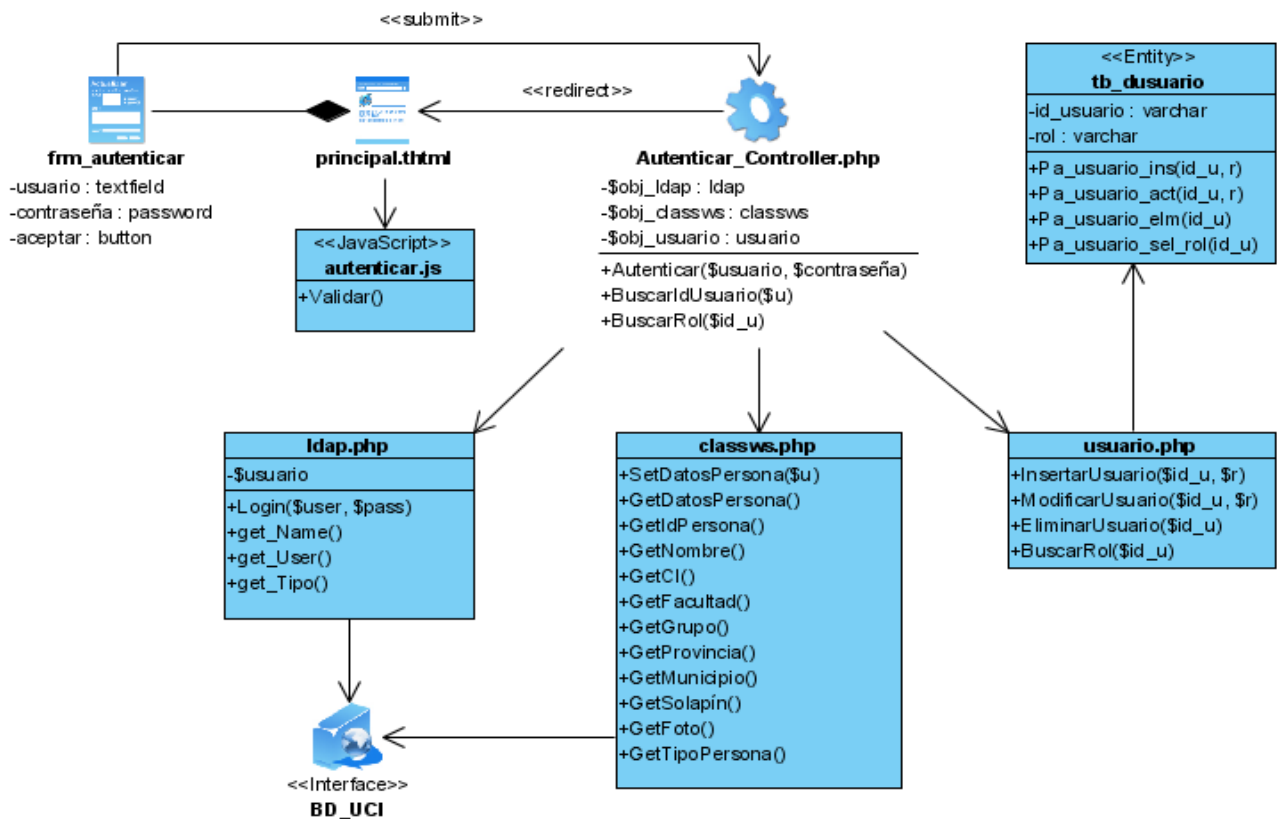


Figura 3.8 Diagrama de clases del diseño: CU “Autenticar usuario”

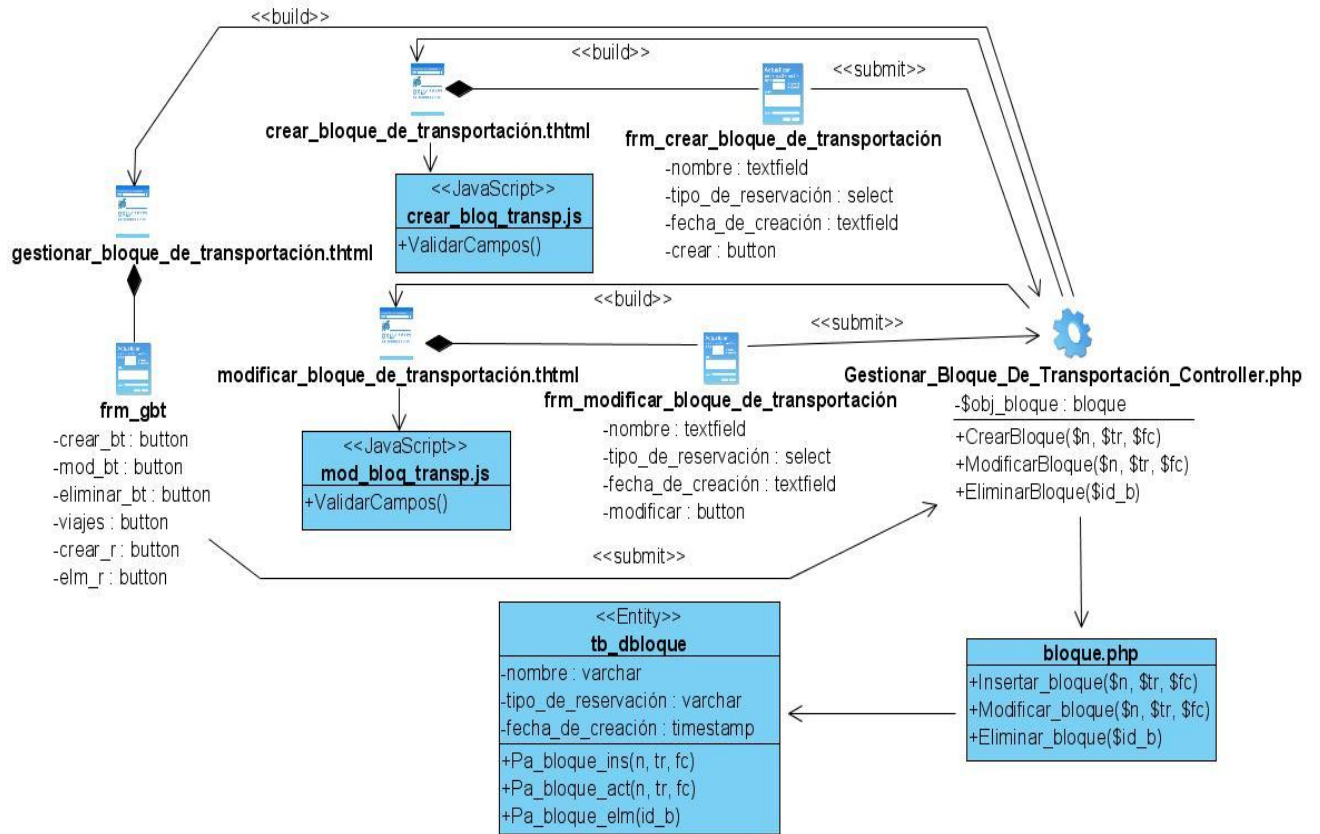


Figura 3.9 Diagrama de clases del diseño: CU “Gestionar bloques de transportación”

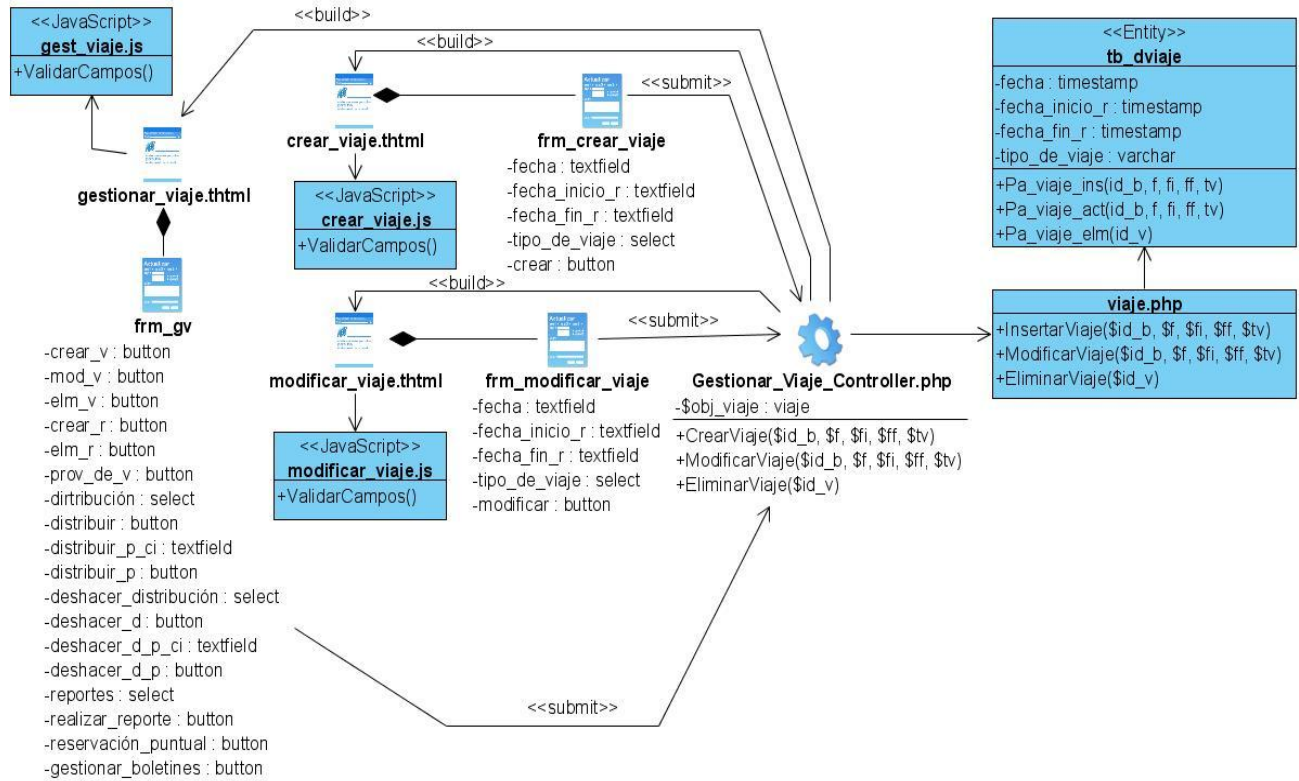


Figura 3.10 Diagrama de clases del diseño: CU "Gestionar viaje"

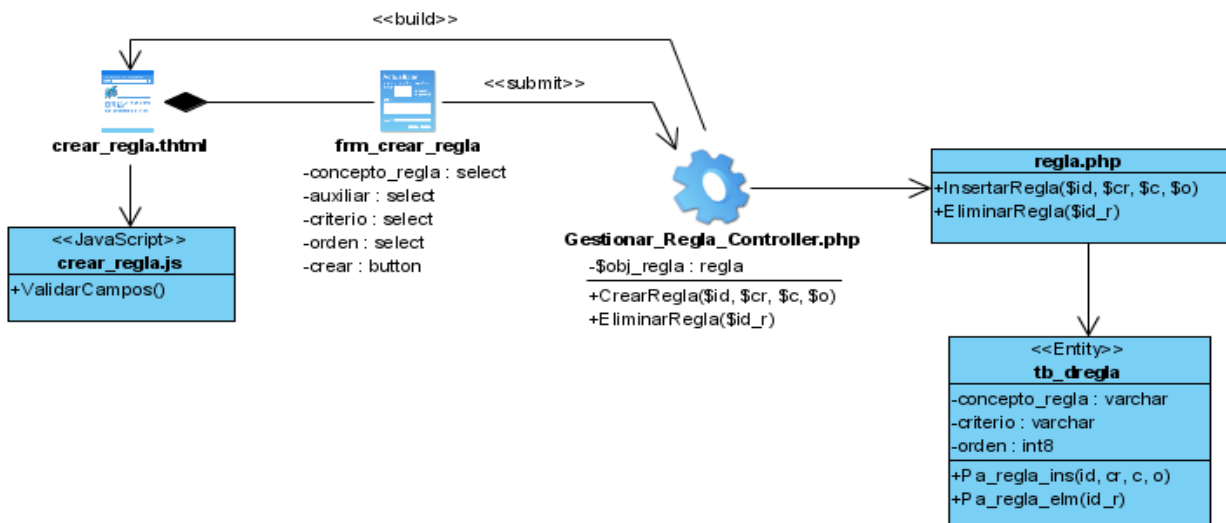


Figura 3.11 Diagrama de clases del diseño: CU "Gestionar reglas"

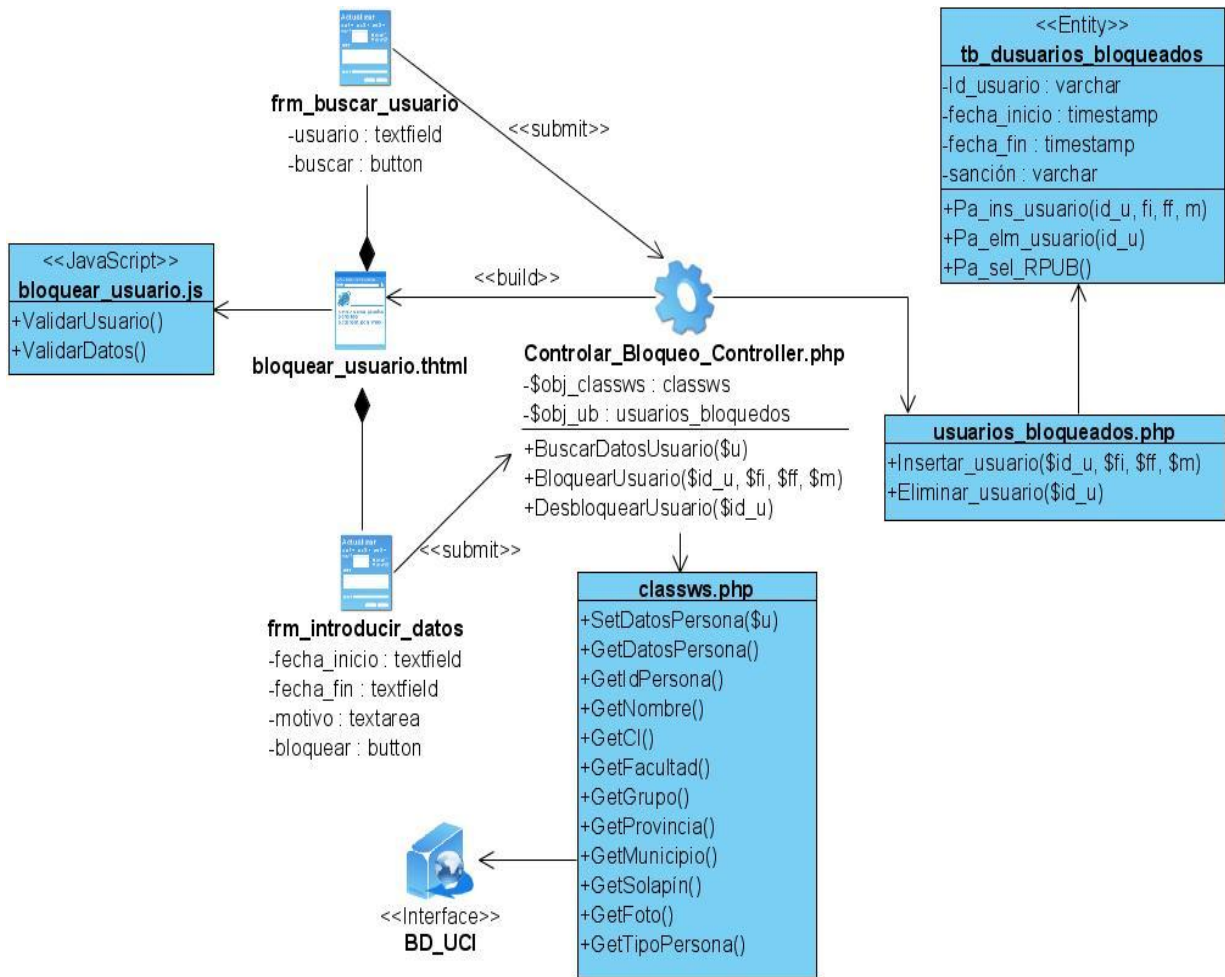
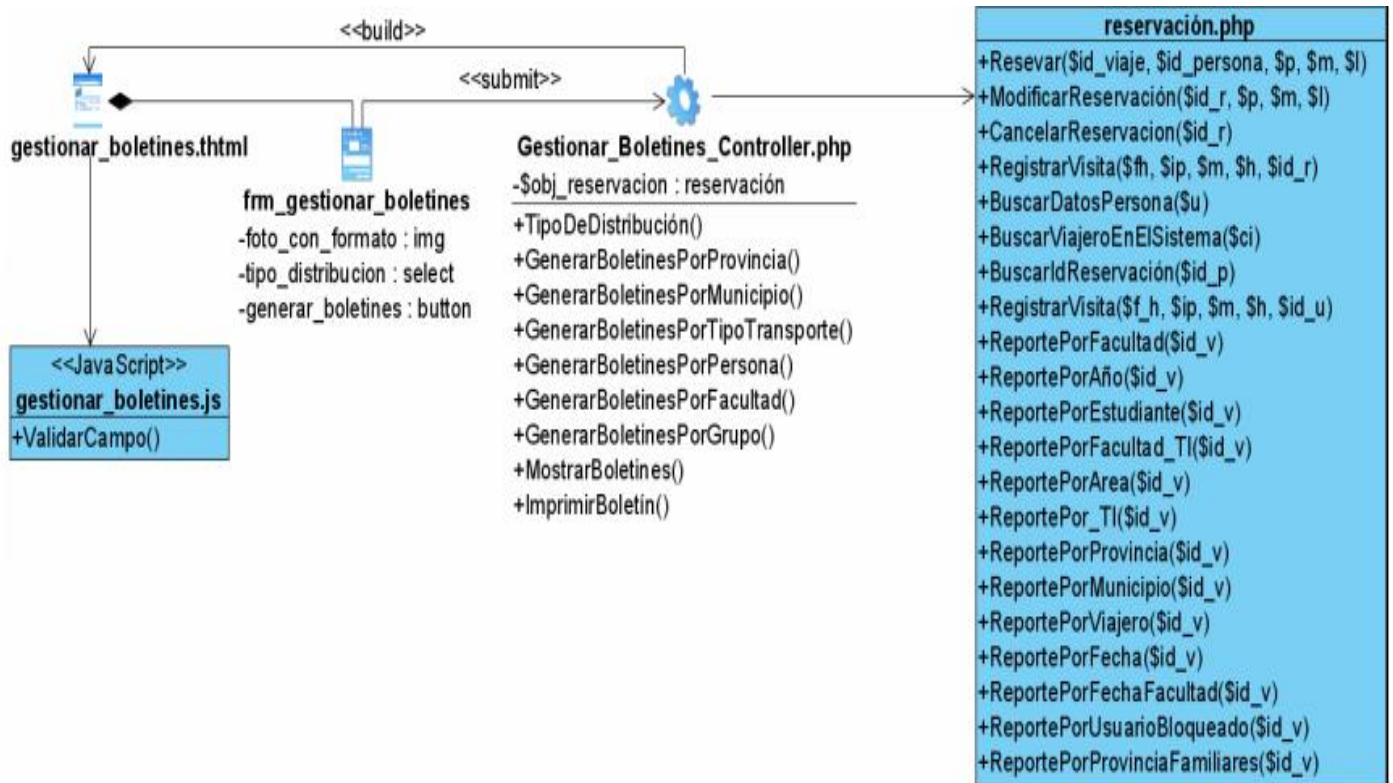


Figura 3.12 Diagrama de clases del diseño: CU "Bloquear usuario"



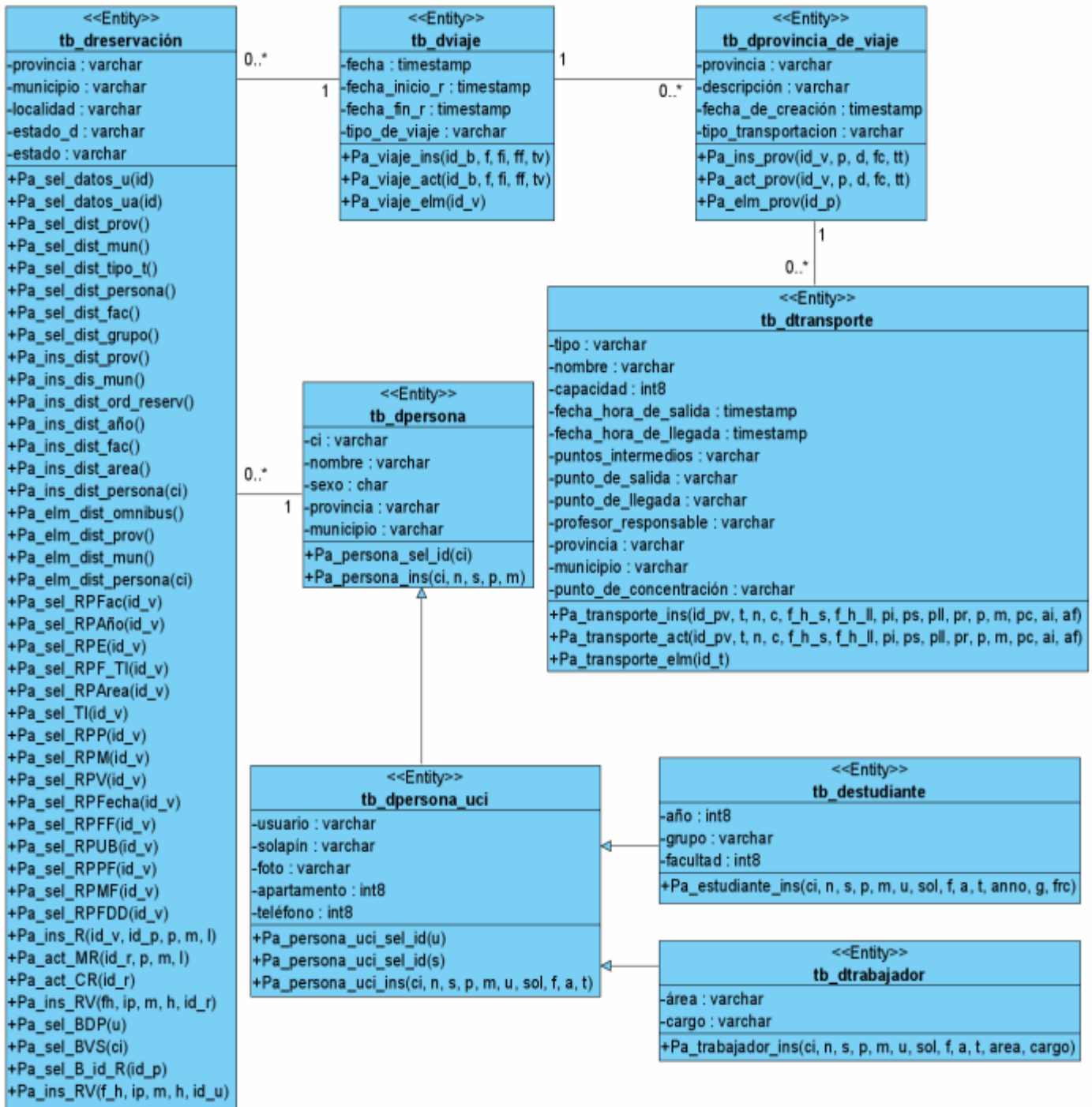


Figura 3.13 Diagrama de clases del diseño: CU “Generar boletines”

3.3.2 Descripción de las clases del diseño

Para ver las descripciones de las clases del diseño ver **Anexo 4**.

3.3.3 Diagramas de secuencia del diseño

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes, se utiliza para modelar los aspectos dinámicos de un sistema. Gráficamente, un diagrama de secuencia es una tabla que representa objetos, dispuestos a lo largo del eje X, y mensajes, ordenados según se suceden en el tiempo, a lo largo del eje Y.

A continuación se muestran algunos ejemplos de diagramas de secuencia, para ver el resto, ver **Anexo 5**.

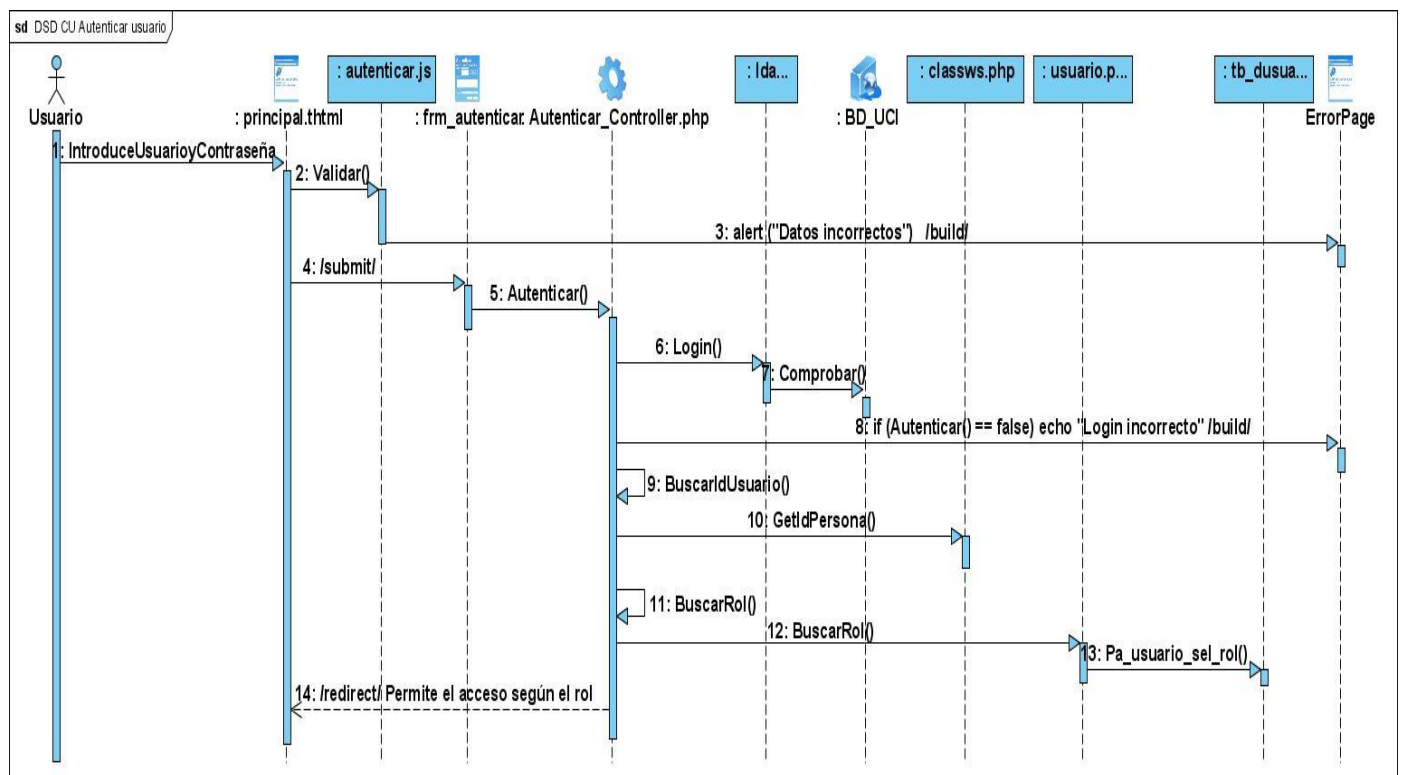


Figura 3.14 Diagrama de secuencia: CU "Autenticar usuario"

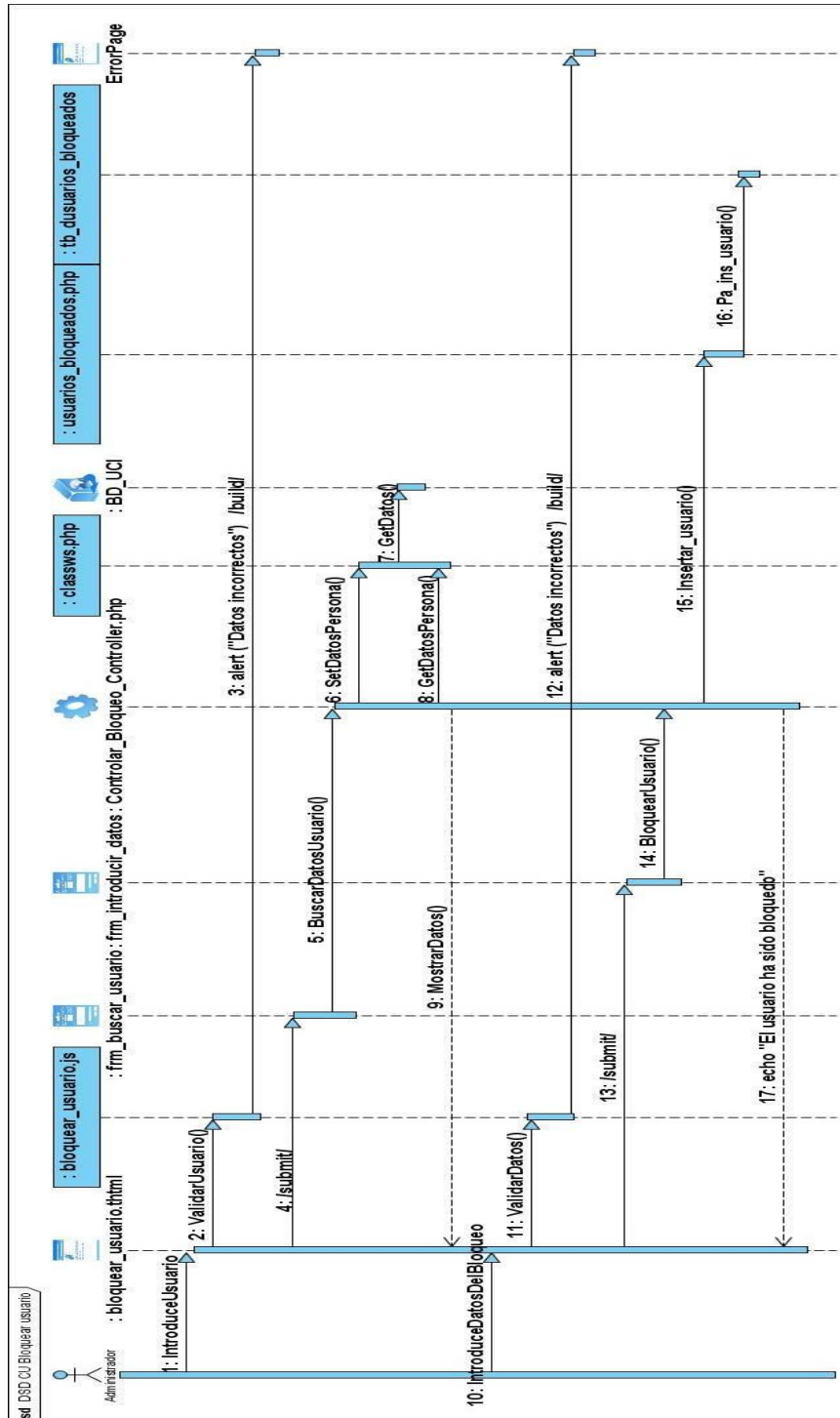


Figura 3.15 Diagrama de secuencia: CU “Bloquear usuario”

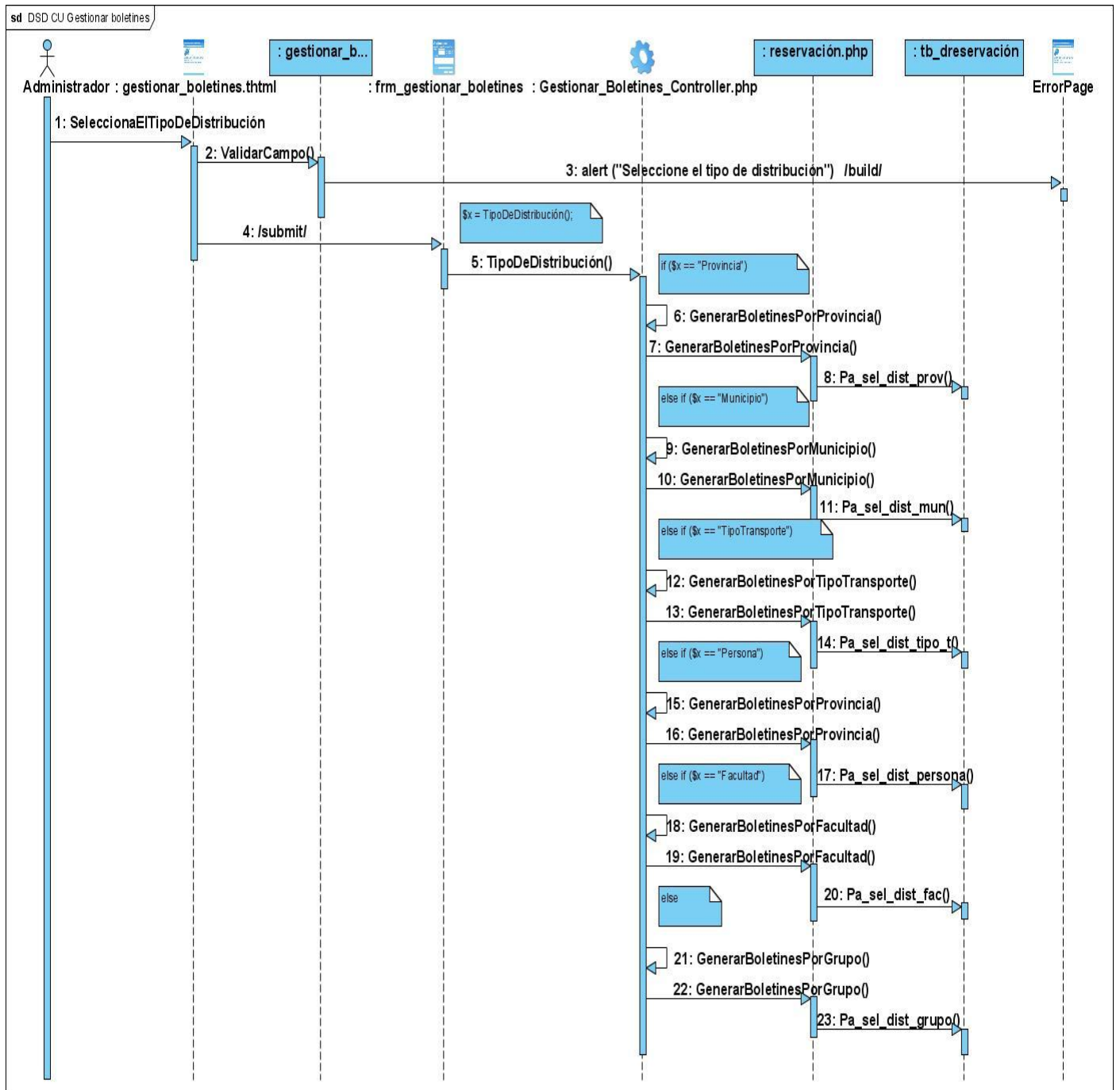


Figura 3.16 Diagrama de secuencia: CU "Generar boletines"

3.4 Principios de diseño

3.4.1 Estándares de la interfaz de aplicación

El sistema sigue la línea general de todo diseño web. Inicialmente se le presenta al usuario una página principal donde contiene en el banner una imagen que identifica el tema que se presenta. Se ha consultado con el diseñador con el fin de seleccionar colores y textos agradables a la vista del usuario. El color predominante es el azul, en caso de que se cometa algún error al llenar algún campo se utilizará el rojo para resaltar la falta cometida. A la izquierda de la página aparecerán los vínculos de acceso, el panel de usuario e información que se desee brindar. La fuente utilizada en los textos es negro, Arial 14 y 16. El fondo de la página se basa en una imagen de forma tal que haga contraste con el sitio y a la vez no llame la atención del usuario, para que el mismo se concentre en el centro de la página donde está la información que necesita. Debajo del banner existe un menú con vínculos a otras páginas de interés para el usuario. El idioma utilizado es el español.

3.5 Concepción general de la ayuda

La ayuda debe ser accesible para todos los usuarios del sistema, como es una aplicación sencilla y de poca complejidad en su uso, la misma se hará a través de mensajes de error, de confirmación y mostrando breves explicaciones que indicarán como usar el sistema. Cuando se desea eliminar o modificar se mostrará un mensaje para que confirme la acción que desea llevar a cabo. En caso de que se esté realizando una búsqueda y que se introduzca incorrectamente los datos se va a mostrar un mensaje que le indique la verificación del dato que ha introducido erróneamente.

3.6 Tratamiento de excepciones

El sistema trata de minimizar al máximo todos los posibles errores que pueden ocurrir, para ello se ha tenido en cuenta que los usuarios tengan que introducir la mínima cantidad de información, para ello se les presenta siempre de ser posible una lista donde escojan la opción deseada, logrando de esta forma que los datos sean válidos y evitando la ocurrencia de errores. En caso contrario, es decir, que el usuario tenga que teclear algún tipo de información y lo haga de forma incorrecta, se harán validaciones a través

del lenguaje JavaScript. Este lenguaje nos brinda la posibilidad de manipular dichas excepciones mostrando al usuario una información correcta sobre el posible error cometido; dichos mensajes se mostraran en color rojo para que resalten y llamen la atención del usuario.

3.7 Diagrama de clases persistentes

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. (*Flujo de trabajo Análisis y Diseño* 2007) El diagrama de clases persistentes representa los objetos que se deben almacenar en la BD y que son necesarios para que la información pueda persistir aún cuando el usuario abandona la sesión.

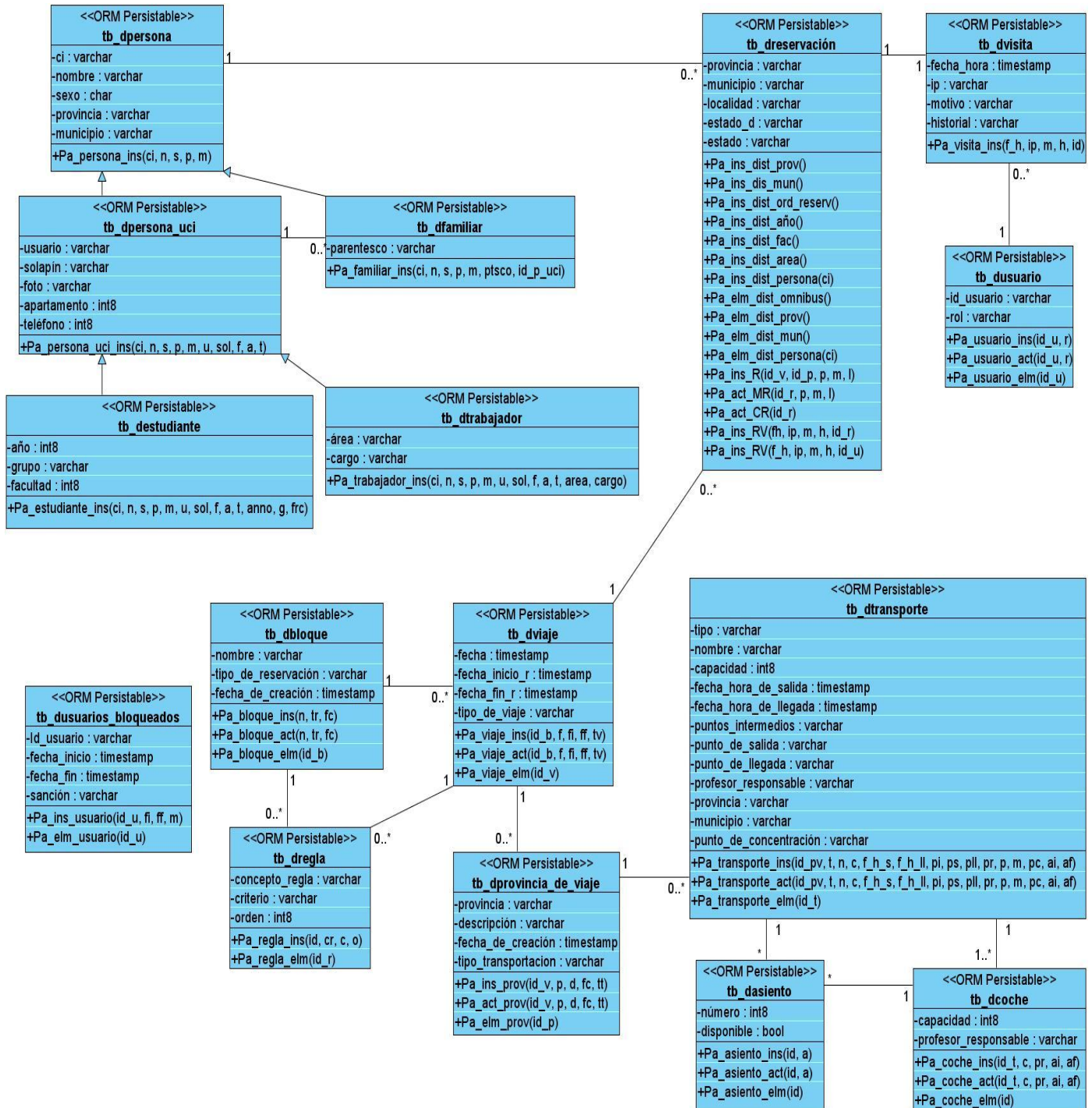


Figura 3.17 Diagrama de Clases Persistentes

3.8 Conclusiones

En este capítulo se han presentado los diagramas de clases del análisis y diseño de los casos de uso del sistema. Además se muestran diagramas de interacción en el diseño (secuencia). Se representa el diagrama de clases persistentes, así como otros temas de interés en el flujo de trabajo de análisis y diseño. De esta manera, se han logrado modelar todos los procesos que han sido objeto de estudio durante el transcurso del trabajo investigativo, proporcionando una idea completa de lo que realmente es el software. Los diagramas y especificaciones de diseño que se proponen constituyen una guía que puede ser fácilmente comprendida por los desarrolladores, los cuales tendrán la tarea de implementar la aplicación que se ha diseñado.



CAPÍTULO 4

ESTUDIO DE FACTIBILIDAD

4.1 Introducción

Para la realización de cualquier proyecto es necesario tener en cuenta el análisis del costo y los beneficios que reportará. Como resultado de estos análisis se obtiene el tiempo de desarrollo en meses, el costo de su producción y la cantidad de personas que se necesitan para desarrollarlo. La estimación es la base de todas las demás actividades de planificación del proyecto y sirve como guía para una buena ingeniería del software. En este capítulo se describen los análisis para la obtención de dichos resultados.

4.2 Planificación basada en casos de uso

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. (*Planificación y Estimación de Proyectos 2007*)

Paso 1. Cálculo de Punto de Casos de Usos sin ajustar

Esto se calcula a partir de la ecuación:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

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

- Cálculo del Factor Peso de los Actores sin ajustar (UAW).

Se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de Actor	Descripción	Factor de Peso	Actores	Total ($\sum \text{FactPeso} \cdot \text{Acts}$)
Simple	Sistema que interactúa con el sistema mediante una interfaz de programación.	1	0	0
Medio	Sistema que interactúa con el sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	4	12
Total ($\sum \text{Total}$)				12

Tabla 4.1 Factor Peso de los Actores sin ajustar (UAW).

- Cálculo del Factor Peso de los Casos de Usos sin ajustar (UUCW).

Se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos.

Tipo de Caso de Uso	Descripción	Factor de Peso	Cantidad de CU	Total ($\sum \text{FactPeso} \cdot \text{CantCU}$)
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	9	45
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	5	50
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	1	15
Total ($\sum \text{Total}$)				110

Tabla 4.2 Factor Peso de los Casos de Usos sin ajustar (UUCW).

Luego: **UUCP** = 12 + 110

UUCP = 122

Paso 2. Cálculo de Puntos de Casos de Uso ajustados

Luego que se tienen los Puntos de Casos de Uso sin ajustar, se procede a ajustar este valor mediante la ecuación:

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Puntos de Casos de Uso sin ajustar **UUCP** = 122

- Cálculo del Factor de complejidad técnica (**TCF**).

Se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

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

Significado de los valores:

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

Factor	Descripción	Peso	Valor	$\sum (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido.	2	2	4
T2	Objetivos de performance (funcionamiento) o tiempo de respuesta.	1	3	3
T3	Eficiencia del usuario final.	1	4	4
T4	Procesamiento interno complejo.	1	3	3
T5	El código debe ser reutilizable.	1	3	3
T6	Facilidad de instalación.	0.5	4	2
T7	Facilidad de uso.	0.5	5	2.5

T8	Portabilidad.	2	3	6
T9	Facilidad de cambio.	1	4	4
T10	Concurrencia.	1	4	4
T11	Incluye objetivos especiales de seguridad.	1	3	3
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	1	1
Total				39.5

Tabla 4.3 Clasificación del Factor de Complejidad Técnica del Sistema.

Finalmente calculamos el factor de complejidad técnica:

$$TCF = 0.6 + 0.01 * \sum (\text{Peso } i \times \text{Valor } i)$$

Según los datos de la tabla anterior tenemos que:

$$TCF = 0.6 + 0.01 * 39.5$$

$$TCF = 0.6 + 0.395$$

$$TCF = 0.995$$

- Cálculo del Factor Ambiente (EF).

Para determinar el factor ambiente hay que tener en cuenta las habilidades y experiencia del conjunto de personas que van a desarrollar el producto.

Estos factores ejercen una gran influencia a la hora de realizar estimaciones de tiempo, utilizando la ecuación:

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

En la siguiente tabla se muestra el valor y el peso de cada uno de estos factores:

Factor	Descripción	Peso	Valor	$\sum (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3	4.5
E2	Experiencia en la aplicación.	0.5	2	1
E3	Experiencia en orientación a objetos.	1	4	4

E4	Capacidad del analista líder.	0.5	3	1.5
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	3	6
E7	Personal part-time.	-1	3	-3
E8	Dificultad del lenguaje de programación.	-1	2	-2
Total				17

Tabla 4.4 Clasificación del Factor de Ambiente del Sistema.

Luego según los datos de la tabla anterior tenemos que:

$$EF = 1.4 - 0.03 * 17$$

$$EF = 1.4 - 0.51$$

$$EF = 0.89$$

Teniendo ya todos los valores que conforman la parte derecha de la ecuación (**UUCP**, **TCF**, **EF**) para calcular los Casos de Uso Ajustados (**UCP**), pasamos a la solución de la misma:

$$UCP = UUCP * TCF * EF$$

$$UCP = 122 * 0.995 * 0.89$$

$$UCP = 108.0371$$

Paso 3. Calcular esfuerzo a través de los Puntos de Casos de Uso

El esfuerzo en horas-hombres viene dado por:

$$E = UCP * CF$$

Donde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

- Cálculo del factor de conversión (**CF**)

Para obtener el factor de conversión (CF) se cuentan cuántos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8).

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

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

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

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

Como: $Total_{EF} = 1$

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

Luego:

$E = 108.0371 * 20$ horas-hombre

E = 2160.742 horas-hombres

Este método proporciona una estimación del esfuerzo en horas-hombres contemplando el desarrollo de la funcionalidad específica en los casos de uso.

Paso 4. Calcular esfuerzo de todo el proyecto

Este procedimiento aporta una estimación del esfuerzo en horas-hombre observando solo el desarrollo de la funcionalidad especificada en los casos de usos. Para lograr obtener una estimación más completa y exacta de la duración total del proyecto, se debe agregar a la estimación del esfuerzo lograda por los puntos de casos de uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para lograr esto podemos tener en cuenta el siguiente criterio que plantea de la distribución del esfuerzo entre las diferentes actividades de un proyecto:

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	540.1855 horas – hombre
Diseño	20%	1080.371 horas – hombre
Implementación	40%	2160.742 horas – hombre
Prueba	15%	810.27825 horas – hombre
Sobrecarga	15%	810.27825 horas – hombre
Total	100%	5401.855 horas – hombre

Tabla 4.5 Distribución del esfuerzo.

Si **ET = 5401.855 horas-hombre** y se estima que cada mes tiene 4 semanas y cada semana 5 días laborales, por tanto se trabajaría 20 días al mes y si se trabaja 7 horas diarias como promedio se trabajarían en un mes 140 horas laborables, eso daría un **ET = 38.58 mes-hombre**.

Si: $\text{Tiempo} = \text{ET} / \text{CH}$

$\text{Tiempo} = 38.58 / 2$

Tiempo = 19.29

Esto quiere decir que con 2 hombres trabajando en el proyecto el mismo se desarrolla en aproximadamente en 20 meses.

Paso 5. Costo del proyecto

Ahora se van a analizar los costes asociados al desarrollo del producto.

C = CHM * ET

Donde:

C: costo

CHM: cantidad de hombre-mes

ET: esfuerzo total del proyecto

- Cálculo de la cantidad de hombres-mes (**CHM**)

CHM = CH * SBM

Donde:

CH: cantidad de hombres trabajando en el proyecto

SBM: salario básico por hombre-mes

SBM = \$100

CH = 2

Por lo que: **CHM = 2 * 100**

CHM = 200 pesos/mes.

Entonces: **C = 200 * 19.29**

C = \$ 3858

Por tanto para el desarrollo del producto se obtiene un costo 3858 pesos.

4.3 Beneficios tangibles e intangibles

El desarrollo de esta aplicación está pensada para su uso en la UCI, o sea publicarla a disposición de toda la comunidad universitaria que reside en la misma, para que realicen reservaciones hacia sus respectivas provincias cuando la universidad oriente una salida masiva, siendo este el mayor beneficio que brinda dicha aplicación. Como beneficio intangible fundamental se tiene el producto en sí, que se podrá utilizar de manera sencilla, amigable y que va a permitir al usuario conocer información de interés relacionados con el tema. Como beneficios tangibles se tiene el ahorro de papel y material de oficina, así como el beneficio económico que trae arraigado debido a que se evita un gasto innecesario de recursos. Además traerá consigo una serie de beneficios intangibles como son:

- Disminución del tiempo y el esfuerzo que se invierte en el proceso de reservación de transporte.
- Un aumento de la calidad del proceso de reservación.
- Impresión de páginas e informes desde la aplicación.
- Integridad de los datos en la Base de Datos.
- Información legible.
- Búsquedas rápidas e eficientes.
- Rápida gestión de la información.
- Seguridad de la información.

4.4 Análisis de costo y beneficios

Desarrollar un producto informático trae consigo beneficios y costos. Los beneficios pueden ser económicos y sociales, la aplicación que se propone trae consigo estas dos ventajas, de forma que permite elevar y mejorar las condiciones de trabajo de la dirección de transporte a la hora de llevar a cabo una salida masiva de la universidad. Una vez que se implante el sistema va contribuir con la eficiencia de los servicios y con el ahorro del tiempo al llevar a cabo tan difícil tarea. La tecnología utilizada para el desarrollo de dicha aplicación es libre o se cuenta con la licencia, por lo que no se incurrirá en grandes gastos para su uso. Se desarrollará con pocos recursos y el tiempo de desarrollo depende de la cantidad de trabajadores. Analizando sus costes y los beneficios anteriormente se puede concluir que estos últimos

tienen mayor peso en el desarrollo del sistema, por lo que la implantación del mismo será completamente factible y de gran utilidad para la UCI.

4.5 Conclusiones

En este capítulo se ha hecho un estudio de la factibilidad del sistema propuesto mediante la utilización del método de planificación basada en casos de uso, se ha llevado a cabo un análisis de las ventajas y beneficios que traerá consigo la implantación del sistema, los cuales son superiores a los costos en los que se implican, ya que el sistema se pondría en marcha en poco tiempo y con escasos recursos. Todos estos aspectos son fundamentales tenerlos en cuenta para estimar los recursos que se van a utilizar y así obtener un software con calidad.

CONCLUSIONES GENERALES

Con el desarrollo de este trabajo se logró realizar el análisis y diseño de un software, para dar solución a los problemas existentes en la dirección de transporte de la UCI, favoreciendo la futura implementación del mismo. El modelado contribuirá a generar reportes automáticamente, teniendo de acuerdo a la información que solicite el usuario. La información manejada estará registrada en un BD, para su mayor seguridad y confiabilidad. Se pondrá en marcha un sistema completo, que cubre con todas las funcionalidades y necesidades del usuario, logrando una mayor eficiencia.

La propuesta realizada, permitirá a los clientes una mayor satisfacción y placer, a partir de la interfaz amigable y la ayuda que dispondrá. Brindará funcionalidades a la medida de las necesidades de los clientes. Permitirá llevar a cabo la reservación del pase online, logrando una mayor comodidad para los usuarios. Se logrará una mayor rapidez en la gestión y control de la información relacionada con transportación masiva de la universidad.

Fue confeccionado acorde con lo convenido por la metodología RUP cumpliendo con cada una de sus etapas, se utilizó como herramienta CASE Visual Paradigm, que utiliza como lenguaje de modelado UML, bajo el patrón de arquitectura MVC.

Por todo lo anterior expuesto se concluye que el objetivo fue cumplido satisfactoriamente, teniendo en cuenta las recomendaciones necesarias para una buena implementación y mejora del mismo.

RECOMENDACIONES

Se recomienda que:

- ✓ La propuesta de diseño expuesta en el trabajo sea implementada posteriormente por los desarrolladores para su futura utilización y puesta en marcha en la dirección de transporte de la UCI.
- ✓ Se realice la continuidad de la investigación sobre los sistemas de reservaciones existentes en el mundo, para tener en cuenta actualizaciones y mejoras a llevar a cabo sobre el sistema, para un mejor funcionamiento.
- ✓ Se amplíe la propuesta de diseño enriqueciendo su contenido y forma de presentar la información.
- ✓ Se expanda esta idea a otros procesos de reservaciones que impliquen gran cantidad de personas a viajar.

REFERENCIAS BIBLIOGRÁFICAS

- [1] ANDRÉS, M. M. M. *Herramientas CASE*, 2001. [Disponible en:
<http://www3.uji.es/~mmarques/f47/apun/node75.html>
- [2] *Arquitectura y Patrones de diseño. Conferencia_de_Arquitectura*. <http://teleformacion.uci.cu> 2008.
- [3] *Centro Nacional de Tecnologías de Información*. 2007. [Disponible en:
http://www.softwarelibre.gob.ve/index.php?option=com_content&task=view&id=634&Itemid=137
- [4] FAJARDO, L. B., 2008. [Disponible en: <http://www.colombiaseconecta.gov.co/spip.php?article104>
- [5] *Fase de Inicio. Modelo del Negocio. Conf_2_Modelacion_Negocio_.pdf*. <http://teleformacion.uci.cu>, 2007.
- [6] *Flujo de trabajo Análisis y Diseño. Culminación_FT_AyD._Modelo_de_datos.pdf*, 2007.
- [7] *Flujo de trabajo de requerimientos. Conferencia_3_estudiantes*. <http://teleformacion.uci.cu>, 2007.
- [8] GALLEGO, J. P. G. *Fundamentos de la metodología RUP*, 2007. [Disponible en:
http://www.scribd.com/word/full/297224?access_key=1jrpweegpyzi3
- [9] IVAR JACOBSON, G. B., JAMES RUMBAUGH. *El Proceso Unificado de Desarrollo de Software*. 2000. p. 84-7829-036-2
- [10] JUAN BERNARDO QUINTERO, R. A. D. P., JUAN CARLOS MARÍN, ALEX BILBAO LÓPEZ. *Un estudio comparativo de herramientas para el modelado con UML. Revista Universidad EAFIT*, 2005. 41.
- [11] LARMAN, C. *UML y Patrones*. 1999. p. 0-13-748880-7
- [12] *Metodologías de Desarrollo de Software*. 2007. [Disponible en:
<http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>
- [13] ORALLO, E. H. *El Lenguaje Unificado de Modelado* 2004. [Disponible en:
<http://www.infomanuales.com/Manuales/UML/UML.asp>
- [14] *Planificación y Estimación de Proyectos. Conferencia_Gestion_Proyectos_Estudiantes*. <http://teleformacion.uci.cu>, 2007.

- [15] *RENa (red escolar nacional)*. 2005. [2008]. Disponible en:
<http://www.rena.edu.ve/servicios/internet/queesinternet.html>
- [16] SANCHEZ, M. A. M. *Informatízate*, 2004. [Disponible en:
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
- [17] *SPL Sistemas de Información*. 2007. [Disponible en: http://www.spl-ssi.com/?sec=articulos&subsec=aplicaciones_web
- [18] VALDÉS, D. P. *Los diferentes lenguajes de programación para la web*, 2007. [Disponible en:
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>

BIBLIOGRAFÍAS

Ayuda Extendida de RUP.

Colectivo_Autores. 2008. Disciplina Análisis y Diseño. *Material_de_Apoyo_Conferencia_Diseño*. 2008.

Entorno virtual de Aprendizaje. [En línea] <http://teleformacion.uci.cu>

Ivar Jacobson, Grady Booch, James Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software*. 2000.

James Rumbaugh, Ivar Jacobson, Grady Booch. *El Lenguaje Unificado de Modelado. Manual de Referencia*.

larman, Craig. 1999. *UML y Patrones*. 1999.

Pressman., Roger. *Ingeniería del Software. Un enfoque práctico*.

Rodríguez, Luis Germán. 1999. Tecnologías de información y comunicación: la gestión de un impacto social positivo. [En línea] 1999. <http://bibliodoc.uci.cu/html/impacto.htm>.

Schmuller, Joseph. 2000. *Aprendiendo UML en 24 horas*. 2000.

Tecnología de la Información y la Comunicación. 2005.

Un estudio comparativo de las herramientas para el modelado con UML. Juan Bernardo Quintero, Juan Carlos Marin, Alex Bilbao. 2005

VISUAL-PARADIGM. Herramienta CASE para el modelado UML. [En línea] 2007. <http://www.visual-paradigm.com/product/vpum/>

GLOSARIO DE TÉRMINOS

Ada: es un lenguaje de programación estructurado y fuertemente tipado de forma estática, es un lenguaje multipropósito, orientado a objetos y concurrente.

AJAX: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

Akademios: es un grupo de desarrollo que contiene todos los datos de los estudiantes, es un sistema automatizado para la gestión académica.

Alimentos: es la dirección encargada de los alimentos en la universidad.

Banner: es un formato publicitario en Internet. Esta forma de publicidad online consiste en incluir una pieza publicitaria dentro de una página web.

Base de Datos: es un conjunto de datos persistentes utilizados por los sistemas de aplicación de alguna empresa, es además un conjunto de datos relacionados entre sí.

CORBA IDL: CORBA (Common Object Request Broker Architecture) IDL (Interface Definition Language) es un lenguaje utilizado para describir la interfaz de un componente de software. En él se describe una interfaz en forma de un lenguaje neutral, lo que permite la comunicación entre los componentes de software escritos en diferentes lenguajes.

CRUD: es el acrónimo de Crear, Obtener, Actualizar y Borrar (Create, Retrieve, Update y Delete en inglés).

C++: (pronunciado "ce más más" o "ce plus plus") es un lenguaje de programación orientado a objetos.

C#: es un lenguaje de programación orientado a objetos, evolución del lenguaje C++, desarrollado por Microsoft.

Delphi: lenguaje de programación.

Flash ActionScript: lenguaje de programación.

Formulario HTML: es una sección de un documento que contiene contenido normal, código, elementos especiales llamados controles (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, etc.), y rótulos (labels) en esos controles.

IP: Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora).

Java: es un lenguaje de programación orientado a objetos con el que podemos realizar cualquier tipo de programa.

Javascript: es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web.

MVC (Modelo Vista Controlador): es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

Objective-C: lenguaje de programación.

Perl: lenguaje de programación.

Python: es un lenguaje de programación que facilita la creación de programas compactos y legibles.

Recursos humanos: es la oficina que maneja los datos de los trabajadores.

Reservas on-line: son reservaciones que pueden hacer los usuarios a través de sitios web disponibles para este fin.

Residencia: es la dirección encargada de la parte de residencia en la universidad.

Ruby: lenguajes de programación.

Schema: es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML.

URL: las siglas URL en inglés quieren decir "Uniform Resource Locator," y se refiere al texto que identifica a una página web. Normalmente empieza por "http:/...".

VB.NET: Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET.

Vicerrectoría de formación: oficina encargada de la docencia.

XML: sigla en inglés de eXtensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

ANEXOS

Anexo 1. Descripción de los Casos de Uso del Sistema

Tabla 1. CUS Gestionar usuario

Caso de Uso	Gestionar usuario.	
Actor:	Administrador.	
Propósito:	Insertar usuarios al sistema con un rol determinado, posibilita modificar o cancelar su rol.	
Precondiciones:	Tiene que existir un administrador por defecto.	
Resumen:	Se realizan búsquedas de usuarios del dominio UCI para asignarle un rol dentro del sistema. Posibilita modificar y cancelar el rol de cualquier usuario dentro del mismo. En cada caso se registran datos de la persona y la máquina donde se realizó la acción.	
Referencias	RF2	
Flujo Normal de Eventos		
Sección “Asignar rol a usuario”		
Acción del Actor	Respuesta del Sistema	
2- Introduce el usuario para realizar la búsqueda.	1- El sistema da la posibilidad de realizar una búsqueda por usuario.	
4- Selecciona el rol que va a tener el usuario dentro de la aplicación.	3- Realiza la búsqueda del usuario y muestra sus datos y da la posibilidad de asignarle un rol (UsuarioGR, UsuarioGV o Administrador). En caso de no encontrar al usuario: Ver FA1 .	
	5- Registra en el sistema el id del usuario más el rol que se le asignó.	
	6- Registrar fecha, hora, IP y usuario de la persona que realizó la asignación del rol.	
	7- Muestra los datos del usuario más su rol una vez asignado el mismo.	
Flujos Alternos		
FA1: Muestra un mensaje informando que no se encontró al usuario buscado.		
Poscondiciones:	Se ha asignado un rol al usuario que fue buscado en el sistema.	
Sección “Modificar rol a usuario”		
Acción del Actor	Respuesta del Sistema	
2- Selecciona modificar el rol del usuario que desee.	1- Lista los usuarios que tienen un rol dentro del sistema (UsuarioGR, UsuarioGV o Administrador) con sus datos y el rol actual y da la posibilidad de modificarlo.	
4- Selecciona el nuevo rol.	3- Muestra una interfaz con la posibilidad de modificar el rol.	
	5- Actualiza el nuevo rol del usuario en la base de datos.	
	6- Registrar fecha, hora, IP y usuario de la persona que realizó la modificación del rol.	
	7- Muestra los datos del usuario más el nuevo rol.	
Poscondiciones:	Se le ha modificado el rol al usuario seleccionado.	

Sección "Cancelar rol a usuario"	
Acción del Actor	Respuesta del Sistema
2- Cancela el rol del usuario que desee	1- Lista los usuarios que tienen un rol dentro del sistema (UsuarioGR, UsuarioGV o Administrador) con sus datos y el rol actual y da la posibilidad de cancelarlo.
	3- Elimina al usuario que se le realizó la cancelación en la base de datos.
	4- Registrar fecha, hora, IP y usuario de la persona que realizó la cancelación del rol.
	5- Refresca la página para que se vean los cambios realizados.
Poscondiciones:	Se cancela el rol del usuario seleccionado.
Prioridad:	Secundario.

Tabla 2. CUS Gestionar reglas

Caso de Uso	Gestionar reglas.
Actor:	Administrador.
Propósito:	Crear y eliminar reglas.
Precondiciones:	Ser administrador del sistema.
Resumen:	Crear reglas dentro de los bloques y viajes de forma opcional para restringir el acceso a los mismos. Debe dar la posibilidad de eliminarlas.
Referencias	RF3, RF4, RF5
Flujo Normal de Eventos	
Sección "Crear regla"	
Acción del Actor	Respuesta del Sistema
6- Selecciona criterio, valor, orden y elige crear.	1- Muestra la interfaz crear regla por uno de estos criterios: provincia, municipio, año, tipo de persona, mundial, proyecto, tipo de viaje (ida o regreso) o facultad. Una vez seleccionado el criterio entonces se elige su valor correspondiente. Por último se establece el orden en que será ejecutada dentro del bloque o viaje.
	3- Crea la regla y se le aplica al bloque o viaje correspondiente. En caso de no seleccionar opciones para crear la regla: Ver FA1.
	4- Actualiza la base de datos.
	5- Muestra la regla creada dentro del bloque o viaje.
Flujos Alternos	
FA1: Muestra mensaje de error.	
Poscondiciones:	Se escogen las reglas para restringir el acceso a la aplicación.
Sección "Eliminar regla"	
Acción del Actor	Respuesta del Sistema
2- Selecciona la(s) regla(s) a eliminar.	1- Muestra las reglas existentes.

	3- Elimina las reglas seleccionadas.
	4- Actualiza la base de datos.
Poscondiciones:	Se eliminan las reglas seleccionadas.
Prioridad:	Secundario.

Tabla 3. Autenticar usuario

Caso de Uso:	Autenticar usuario.	
Actor:	Usuario, Akademos y Recursos humanos.	
Propósito:	Iniciar sesión en el sistema.	
Precondiciones:		
Resumen:	El usuario se autentica y muestra las opciones a las que tiene acceso según su rol.	
Referencias:	RF1	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
2- El usuario introduce usuario y contraseña.	1- Muestra una interfaz que le permite al usuario autenticarse con usuario y contraseña del dominio UCI.	
	3- Valida los datos introducidos por el usuario. En caso de ser incorrectos: Ver FA1.	
	4- Comprueba los datos en el directorio activo de la UCI. En caso de ser incorrectos: Ver FA2.	
6- Brinda los datos solicitados.	5- Solicita datos del usuario <ul style="list-style-type: none"> • Si es estudiante a Akademos. • Si es trabajador a Recursos humanos. 	
	7- Muestra al usuario las opciones a la que tiene acceso según el rol que tiene asignado.	
Flujos Alternos		
FA1: Mostrar mensaje de error.		
FA2: Mostrar mensaje de error.		
Poscondiciones:	Se autentica correctamente el usuario en el sistema.	
Prioridad:	Crítico.	

Tabla 4. Gestionar bloques de transportación.

Caso de Uso:	Gestionar bloques de transportación.	
Actor:	Administrador.	
Propósito:	Crear, modificar y eliminar bloques de transportación y agregarle reglas a los mismos.	
Precondiciones:	El usuario autenticado debe ser administrador.	
Resumen:	Crea, modifica y elimina bloques de transportación dando la posibilidad de agregarle reglas.	
Referencias:	RF3	
Flujo Normal de Eventos		
Sección “Crear bloques de transportación”		

Acción del Actor		Respuesta del Sistema
2-	Selecciona la opción de crear un bloque de transportación.	1- Muestra una interfaz que permita crear bloques de transportación.
4-	Introduce los datos del bloque.	3- Muestra la interfaz crear bloque de transportación con los siguientes campos: fecha de creado, nombre del bloque y tipo de reservación (estándar o personalizada).
		5- Crea el bloque de transportación. En caso de dejar campos vacíos ver FA1 .
7-	Selecciona crear reglas.	6- Muestra el bloque creado con la opción de crear las reglas que serán aplicadas al mismo. Estas reglas son opcionales.
		8- Ver sección <i>Crear regla</i> del caso de uso <i>Gestionar reglas</i> .
		9- Muestra el bloque creado con la(s) regla(s) aplicada(s) y da la posibilidad de eliminarlas. En caso de querer eliminar una regla: Ver FA2 .
Flujos Alternos		
FA1: Mostrar un mensaje de error.		
FA2: Ver sección <i>Eliminar regla</i> del caso de uso <i>Gestionar reglas</i> .		
Poscondiciones:	Se han creado los bloques de transportación.	
Sección "Modificar bloques de transportación"		
Acción del Actor		Respuesta del Sistema
2-	Selecciona el bloque a modificar.	1- Muestra los bloques existentes.
4-	Modifica datos de interés.	3- Muestra los datos del bloque seleccionado dando la posibilidad de modificarlos.
		5- Aplica la modificación.
		6- Actualiza la base de datos
		7- Muestra el bloque con las modificaciones realizadas.
Poscondiciones:	Se ha modificado un bloque de transportación.	
Sección "Eliminar bloques de transportación"		
Acción del Actor		Respuesta del Sistema
2-	Selecciona el bloque a eliminar.	1- Muestra los bloques existentes con la opción de eliminar.
		3- Elimina el bloque seleccionado.
		4- Actualiza la base de datos.
Poscondiciones:	Se elimina el bloque seleccionado.	
Prioridad:	Secundario.	

Tabla 5. Gestionar transportes

Caso de Uso:	Gestionar transportes.
Actor:	Administrador.
Propósito:	Crear transporte.
Precondiciones:	El usuario autenticado debe ser administrador y debe estar creada al menos una provincia

	de viaje.
Resumen:	Crea transporte de acuerdo al medio de transporte (ómnibus, tren, barco) dándole la posibilidad de modificarlo o eliminarlo.
Referencias:	RF11
Flujo Normal de Eventos	
Sección “Crear transporte”	
Acción del Actor	Respuesta del Sistema
2- Selecciona crear transporte.	1- Muestra una interfaz con la posibilidad de crear el medio de transporte (ómnibus, tren o barco) según este establecido en la provincia.
4- Introduce los datos del medio de transporte, en caso de que falte alguno: Ver FA1.	3- Muestra los campos para crear el medio de transporte: nombre, capacidad, fecha y hora de salida, fecha y hora de llegada, puntos intermedios, punto de salida, punto de llegada, profesor responsable, provincia, municipio, punto de concentración y asientos en caso de ser un ómnibus o un barco.
	5- Crea el transporte.
	6- Actualiza la base de datos
	7- Muestra el transporte creado con sus datos.
	8- Si el transporte creado es un tren, para gestionar sus coches ver: caso de uso <i>gestionar coches</i> .
Flujos Alternos	
FA1: Muestra mensaje de error.	
Poscondiciones:	Se ha creado un medio transporte.
Sección “Modificar transporte”	
Acción del Actor	Respuesta del Sistema
2- Selecciona el transporte a modificar.	1- Muestra los medios de transporte existentes con sus datos.
4- Modifica los datos de interés, en caso de que falte alguno de los datos siguientes de los medios de transporte Ver FA1.	3- Muestra una interfaz con la posibilidad de modificar sus datos.
	5- Aplica la modificación realizada por el usuario en la base de datos.
	6- Muestra el medio de transporte con las modificaciones realizadas.
Flujos Alternos	
FA1: Muestra mensaje de error.	
Poscondiciones:	Se ha modificado el medio de transporte seleccionado.
Sección “Eliminar transporte”	
Acción del Actor	Respuesta del Sistema
2- Selecciona el transporte a eliminar.	1- Muestra los medios de transporte existentes con la posibilidad de eliminarlos.
	3- Elimina el transporte seleccionado de la base de datos.
	4- Refresca la página.

Poscondiciones:	Se ha eliminado el medio de transporte seleccionado.
Prioridad:	Secundario.

Tabla 6. Gestionar provincias

Caso de Uso:	Gestionar provincias.	
Actor:	Administrador.	
Propósito:	Crear las provincias.	
Precondiciones:	El usuario autenticado debe ser administrador y debe existir al menos un viaje.	
Resumen:	Crea las provincias con sus datos, dándole la posibilidad de modificarla o eliminarla.	
Referencias:	RF10	
Flujo Normal de Eventos		
Sección "Crear provincia"		
Acción del Actor	Respuesta del Sistema	
2- Introduce los datos de la provincia.	1- Muestra una interfaz con la posibilidad de crear una provincia con los siguientes datos a llenar: nombre de la provincia, descripción, fecha de creada y tipo de transportación.	
	3- Crea la provincia de acuerdo a los datos introducidos por el usuario, en caso de dejar campos vacíos: Ver FA1.	
	4- Actualiza la base de datos del sistema.	
	5- Muestra la provincia creada.	
Flujos Alternos		
FA1: Mostrar mensaje de error.		
Poscondiciones:	Se ha creado la provincia.	
Sección "Modificar provincia"		
Acción del Actor	Respuesta del Sistema	
2- Selecciona la provincia a modificar.	1- Muestra una interfaz con las provincias existentes y con la posibilidad de modificarlas.	
4- Modifica los datos de interés.	3- Muestra los datos de la provincia seleccionada.	
	5- Aplica la modificación realizada, en caso de dejar al menos un campo vacío: Ver FA1.	
	6- Actualiza la base de datos del sistema.	
	7- Muestra la provincia con la modificación realizada.	
Flujos Alternos		
FA1: Mostrar mensaje de error.		
Poscondiciones:	Se ha modificado la provincia seleccionada.	
Sección "Eliminar provincia"		
Acción del Actor	Respuesta del Sistema	
2- Selecciona la provincia a eliminar.	1- Muestra una interfaz con las provincias existentes y con la posibilidad de eliminarlas.	
	3- Elimina la provincia seleccionada.	
	4- Actualiza la base de datos.	
Poscondiciones:	Se ha eliminado la provincia seleccionada.	

Prioridad:	Secundario.
-------------------	-------------

Tabla 7. Gestionar reservación

Caso de Uso:	Gestionar reservación.	
Actor:	Usuario, Administrador.	
Propósito:	Posibilitar al usuario realizar, modificar o cancelar su reservación de viaje. En caso que el usuario sea administrador permite adjuntarle al usuario buscado, en caso de tener una reservación realizada, uno o varios familiares con la posibilidad de cancelarle a los mismos su reservación.	
Precondiciones:	El usuario tiene que estar autenticado.	
Resumen:	Realiza la reservación (estándar o personalizada) del viaje de acuerdo a las posibilidades que se brinden en cada caso. Adiciona datos de la reservación a la BD y envía correo de confirmación de reservación o cancelación. Modificar reservación: muestra datos de la anterior reservación, verifica el plazo establecido para llevar a cabo la reservación, se hace la modificación, se actualiza la BD y se envía correo de confirmación de la nueva reservación. Cancelar reservación: muestra la última reservación efectuada por el usuario, verifica el tiempo establecido para la cancelación, elimina la reservación de la BD y envía por correo mensaje de confirmación. En caso que el usuario sea administrador permite adjuntarle al usuario buscado, si este tiene una reservación realizada, uno o varios familiares con la posibilidad de cancelarle a los mismos su reservación.	
Referencias:	RF7	
Flujo Normal de Eventos		
Sección “Realizar reservación de viaje”		
Acción del Actor	Respuesta del Sistema	
2- Realiza la reservación de acuerdo a su tipo de usuario (estudiante / trabajador interno) y a los permisos que tiene asignados.	1- Muestra una interfaz con el tipo de reservación publicada (estándar / personalizada) y los viajes a los que tiene acceso (ida o regreso) con los datos del mismo. <u>Estándar:</u> <ul style="list-style-type: none"> • Si es estudiante confirma reservación y si desea puede entrar como dato la localidad. • Si es trabajador interno confirma la reservación con la posibilidad de escoger el destino (provincia y municipio) y si desea puede entrar como dato la localidad. <u>Personalizada:</u> <ul style="list-style-type: none"> • Ambos (Estudiantes y Trabajadores internos) realizan la reservación con la posibilidad de seleccionar el destino (provincia, municipio) y si lo desean pueden entrar como dato la localidad. 	
	3- Registra la fecha, hora y el IP para ambos tipos de reservación, en caso de cancelar la reservación: Ver FA1.	
	4- Adiciona los datos de la reservación a la base de datos de la aplicación.	

	5- Envía al usuario un correo de confirmación de reservación con la fecha, hora, IP de la máquina donde se hizo la reservación y los siguientes datos de la reservación: provincia, municipio y localidad.
Flujos Alternos	
FA1: Envía al usuario un correo de confirmación de cancelación de la reservación con la fecha, hora, IP de la máquina donde se hizo la cancelación y los siguientes datos de la reservación: provincia, municipio y localidad.	
Poscondiciones:	El usuario ha hecho una reservación.
Sección “Modificar reservación de viaje”	
Acción del Actor	Respuesta del Sistema
	1- Muestra al usuario los datos de la reservación hecha anteriormente además de la fecha, hora y el IP de la última modificación que hizo en el sistema con la posibilidad de modificarla.
3- De los posibles parámetros a modificar (provincia, municipio, localidad) modifica el/los de su interés.	2- Verifica el plazo establecido para llevar a cabo la modificación de la reservación, el cual debe de estar en el período de reservación, de lo contrario: Ver FA1.
	4- Aplica la modificación de la reservación en la base de datos y registra la fecha, hora y el IP de la máquina donde se realizó la modificación.
	5- Envía un correo electrónico al usuario con la confirmación de la nueva reservación y los datos de la misma como: fecha, hora, IP, provincia, municipio y localidad de la reservación.
Flujos Alternos	
FA1: Muestra un mensaje de error.	
Poscondiciones:	Se ha modificado la reservación.
Sección “Cancelar reservación de viaje”	
Acción del Actor	Respuesta del Sistema
2- Selecciona la reservación que va a cancelar.	1- Muestra la(s) reservación(es) efectuada(s) por el usuario con sus datos: provincia, municipio, localidad y la fecha, hora y el IP de la última modificación que hizo en el sistema además de la posibilidad de cancelarla.
	3- Verifica el tiempo que se estableció para la cancelación, este estará vigente hasta el día de la salida. En caso de pasar el día de la salida Ver FA1.
	4- Cancela la reservación en la base de datos.
	5- Registra la fecha, hora y el IP de la máquina donde se realizó la cancelación.
	6- Envía un correo al usuario con la confirmación de la cancelación y los datos de la misma: provincia, municipio, localidad y la fecha, hora y el IP de la máquina en la que canceló.

Flujos Alternos	
FA1: Muestra un mensaje de error.	
Poscondiciones:	Se ha cancelado la reservación.
Sección "Realizar reservación a un familiar"	
Acción del Actor	Respuesta del Sistema
4- Selecciona realizar reservación a familiar.	1- Muestra los datos de la reservación realizada a / por un usuario y brinda la posibilidad de realizarle la reservación a sus familiares.
4- Introduce los datos de la reservación del familiar. En caso de introducir incorrectamente los datos: Ver FA1.	3- Muestra una interfaz que permita realizarle la reservación a un familiar con los siguientes campos: provincia, municipio y localidad de destino y además los datos del familiar: CI, nombre, sexo, provincia, municipio y parentesco.
	7- Agrega la reservación del familiar a la base de datos del sistema y se lo adjunta a la persona buscada. En caso de dejar algún campo vacío: Ver FA2.
	8- Muestra mensaje de confirmación de que el familiar fue agregado.
	9- Registra el usuario (administrador) que agregó al familiar y la fecha, hora e IP del la máquina donde se hizo la acción.
	10- Envía correo al usuario del dominio UCI con la notificación de que se le ha agregado el familiar, la fecha, hora, IP de la máquina y el usuario (administrador) que le hizo la reservación al familiar.
Flujos Alternos	
FA1: Muestra mensaje de error. FA2: Muestra mensaje de error.	
Poscondiciones:	Se le ha realizado una reservación a un familiar.
Sección "Cancelar reservación a un familiar"	
Acción del Actor	Respuesta del Sistema
2- Selecciona el cancelar reservación del familiar que desee,	1- Muestra una interfaz que permita cancelarle la reservación a el/los familiar (es).
	3- Cancela la reservación del familiar.
	4- Muestra mensaje de confirmación de que de reservación del familiar fue cancelada.
	5- Registra el usuario (administrador) que canceló la reservación del familiar y la fecha, hora e IP del la máquina donde se hizo la acción.
	6- Envía correo al usuario del dominio UCI con la notificación de que se le ha cancelado la reservación al familiar, la fecha, hora, IP de la máquina y el usuario (administrador) que le canceló la reservación al familiar.
Flujos Alternos	
Poscondiciones:	Se le ha cancelado la reservación a un familiar.
Prioridad:	Crítico.

Tabla 8. Gestionar viajero

Caso de Uso:	Gestionar viajero.	
Actor:	UsuarioGV.	
Propósito:	Buscar un viajero para gestionar su reservación y permite gestionar una reservación puntual.	
Precondiciones:	El usuario que esté autenticado tiene que ser administrador.	
Resumen:	Se busca al usuario del dominio UCI y se le brinda la posibilidad de gestionar su reservación, además de permitirle gestionar una reservación puntual llevándolo a la interfaz <i>gestionar bloques de transportación</i> para que elija el bloque y viaje en el que vaya a realizar dicha gestión.	
Referencias:	RF8	
Flujo Normal de Eventos		
Sección “Gestionar reservación a un usuario UCI”		
Acción del Actor	Respuesta del Sistema	
2- Introduce el usuario. En caso de introducir incorrectamente el usuario. Ver FA1.	1- Muestra una interfaz que permita realizar una búsqueda por usuario de la persona a la cual se le realizará la reservación.	
	3- Busca algunos datos de la persona, estos son: foto, nombre y apellidos, grupo / área y solapín.	
5- Selecciona gestionar reservación.	4- Da la posibilidad de gestionar su reservación.	
	6- Para gestionar la reservación ver caso de uso <i>Gestionar reservación</i> .	
Flujos Alternos		
FA1: Muestra mensaje de error.		
Poscondiciones:	Se le ha realizado una reservación a un usuario del dominio.	
Sección “Realizar reservación puntual”		
Acción del Actor	Respuesta del Sistema	
2- Selecciona el bloque, el viaje y la opción de gestionar la reservación puntual.	1- Muestra una interfaz que permita seleccionar el bloque y el viaje donde se va a gestionar la reservación puntual, este último brinda la posibilidad de gestionar dicha reservación.	
4- Selecciona realizar reservación.	3- Muestra una interfaz con la posibilidad de de seleccionar la acción que se va a realizar, en este caso realizar reservación.	
6- Introduce los datos de la persona y pulsa realizar reservación. En caso de dejar campos vacíos. Ver FA1.	5- Muestra los siguientes campos a llenar con los datos de la persona: CI, nombre, sexo, municipio, provincia y localidad. Debe dar la posibilidad de realizar la reservación.	
	7- Realiza la reservación.	
	8- Muestra un mensaje de confirmación.	
	9- Registra la fecha y hora, IP y usuario del administrador que realizó la reservación.	
Flujos Alternos		
FA1: Muestra un mensaje de error.		
Poscondiciones:	Se ha realizado una reservación puntual.	
Sección “Modificar reservación puntual”		
Acción del Actor	Respuesta del Sistema	

2- Selecciona el bloque, el viaje y la opción de gestionar la reservación puntual.	1- Muestra una interfaz que permita seleccionar el bloque y el viaje donde se va a gestionar la reservación puntual, este último brinda la posibilidad de gestionar dicha reservación.
4- Selecciona modificar reservación.	3- Muestra una interfaz con la posibilidad de de seleccionar la acción que se va a realizar, en este caso modificar reservación.
6- Introduce el CI. En caso de dejar el campo vacío. Ver FA1.	5- Brinda la posibilidad de buscar en la base de datos del sistema a una persona por el CI.
	7- Realiza la búsqueda y muestra los datos de la persona.
9- Introduce los datos de la persona y pulsa modificar reservación. En caso de dejar campos vacíos. Ver FA2.	8- Muestra los siguientes campos a llenar con los datos de la persona: CI, nombre, sexo, municipio, provincia y localidad. Debe dar la posibilidad de modificar la reservación.
	10- Modifica la reservación.
	11- Muestra un mensaje de confirmación.
	12- Registra la fecha y hora, ip y usuario del administrador que realizó la reservación.
Flujos Alternos	
FA1: Muestra un mensaje de error. FA2: Muestra un mensaje de error.	
Poscondiciones:	
Sección "Cancelar reservación puntual"	
Acción del Actor	Respuesta del Sistema
2- Selecciona el bloque, el viaje y la opción de gestionar la reservación puntual.	1- Muestra una interfaz que permita seleccionar el bloque y el viaje donde se va a gestionar la reservación puntual, este último brinda la posibilidad de gestionar dicha reservación.
4- Selecciona cancelar reservación.	3- Muestra una interfaz con la posibilidad de de seleccionar la acción que se va a realizar, en este caso cancelar reservación.
6- Introduce el CI. En caso de dejar el campo vacío. Ver FA1.	5- Brinda la posibilidad de buscar en la base de datos del sistema a una persona por el CI.
8- Pulsa cancelar reservación.	7- Realiza la búsqueda, muestra los datos de la persona y posibilita cancelar su reservación.
	9- Cancela la reservación.
	10- Muestra un mensaje de confirmación.
	11- Registra la fecha y hora, IP y usuario del administrador que realizó la reservación.
Flujos Alternos	
FA1: Muestra un mensaje de error.	
Poscondiciones:	Se ha cancelado la reservación.
Prioridad:	Crítico.

Tabla 9. Gestionar distribución

Caso de Uso:	Gestionar distribución.
Actor:	Administrador.
Propósito:	Distribuir viajeros en los medios transportes.
Precondiciones:	Existencia de reservaciones en la base de datos.
Resumen:	Inicia cuando se distribuye a los viajeros en sus determinados transportes, luego muestra las capacidades restantes, valida en caso de escasez reservaciones libres, además da la oportunidad de deshacer una distribución realizada anteriormente.
Referencias:	RF12
Flujo Normal de Eventos	
Sección “Realizar distribución”	
Acción del Actor	Respuesta del Sistema
2- El administrador selecciona el tipo de distribución que desea realizar.	1- Muestra una interfaz que permita al administrador realizar la distribución de los viajeros por: provincia, municipio, orden de reservaciones, por año y facultad los estudiantes y por área los trabajadores internos.
	3- Distribuye a los viajeros de acuerdo a la selección, que pueden ser las siguientes, por: <ul style="list-style-type: none"> • Provincia: Distribuye a los viajeros por provincia en el tipo y número del transporte que desee el administrador que estén disponibles para esta provincia. • Municipio: Distribuye a los viajeros por municipio en el tipo y número del transporte que desee el administrador que estén disponibles para este municipio. • Orden de reservaciones: Distribuye a los viajeros por provincia en el tipo y número del transporte que desee el administrador que estén disponibles en el orden que hayan reservado. • Año: Distribuye a los viajeros por provincia en el tipo y número del transporte que desee el administrador que estén disponibles de acuerdo al año al que pertenezca. • Facultad: Distribuye a los viajeros por provincia en el tipo y número del transporte que desee el administrador que estén disponibles de acuerdo a la facultad que pertenezca. • Área: Distribuye a los viajeros por provincia en el tipo y número del transporte que desee el administrador que estén disponibles de acuerdo al área que pertenezca. • Una o varias personas: Distribuye a una o varias en personas en un determinado transporte.
	4- Guarda la distribución en la base de datos.

	5- Muestra la distribución realizada, capacidades totales por provincia y municipio y las que van quedando.
Poscondiciones:	Se ha realizado una distribución.
Sección “Deshacer distribución”	
Acción del Actor	Respuesta del Sistema
2- Selecciona el tipo de distribución que desea deshacer.	1- Muestra la posibilidad de deshacer una distribución por: ómnibus, provincia, municipio o usuario.
	3- Deshace la distribución seleccionada.
	4- Actualiza la base de datos.
Poscondiciones:	Se ha deshecho una distribución.
Prioridad:	Crítico.

Tabla 10. Bloquear usuario

Caso de Uso:	Bloquear usuario.
Actor:	Administrador.
Propósito:	Bloquear usuarios deseados.
Precondiciones:	Introducir correctamente datos para la búsqueda de usuario.
Resumen:	Inicia con una búsqueda de usuario, luego se muestran sus datos y se bloquea el usuario, determinándole un tiempo de inicio/fin del bloqueo, y haciéndole una observación del motivo de la sanción.
Referencias	RF6
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
2- Introduce el usuario a buscar.	1- Muestra una interfaz con la posibilidad de hacer una búsqueda a los usuarios del sistema introduciendo el usuario.
4- Establece el tiempo que el usuario va a estar bloqueado, la sanción y realiza el bloqueo.	3- Si la búsqueda se realiza de forma correcta muestra del usuario los siguientes datos: nombre y apellidos, C.I, facultad, grupo, provincia, municipio, solapín, foto con la posibilidad de establecer la fecha de inicio y fin del bloqueo, hacer observaciones de la sanción y bloquearlo. En caso de contrario: Ver FA1.
	5- Aplica el bloqueo al usuario. En caso de no introducir un tiempo de bloqueo o la observación de la sanción: Ver FA2.
	6- Actualiza la base de datos.
	7- Muestra los datos del usuario con la confirmación del bloqueo realizado.
Flujos Alternos	
FA1: Muestra un mensaje de error.	
FA2: Muestra un mensaje de error.	
Poscondiciones:	Se ha bloqueado el usuario.

Prioridad:	Auxiliar.
-------------------	-----------

Tabla 11. Consultar información

Caso de Uso:	Consultar información.
Actor:	Usuario.
Propósito:	Obtener información del usuario deseado.
Precondiciones:	Estar autenticado por el dominio UCI.
Resumen:	El caso de uso inicia cuando se introducen los parámetros de búsqueda del viajero, se muestran los datos de la reservación realizada por el usuario buscado, si el usuario que esta autenticado es administrador se muestran además de los datos de la reservación del usuario buscado la fecha, IP y hora de la última modificación hecha por el mismo.
Referencias:	RF13

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
2- Introduce el parámetro de búsqueda (usuario, CI, solapín).	1- Muestra interfaz que permita realizar una búsqueda de un usuario.
	3- Si realiza la búsqueda correctamente muestra los datos de acuerdo al tipo de usuario. <ul style="list-style-type: none"> • Usuario: Muestra los datos de la reservación en el medio de transporte de la ida y regreso (lugar, fecha y hora de salida, nombre, puntos intermedios, lugar, fecha y hora de llegada, salida, provincia / municipio). • Administrador: Muestra los datos de la reservación en el medio de transporte de la ida y regreso (lugar, fecha y hora de salida, nombre, puntos intermedios, lugar, fecha y hora de llegada, salida, provincia / municipio), así como la fecha, IP y hora de la última modificación hecha por el usuario. En caso contrario: Ver FA1.

Flujos Alternos

FA1: Mostrar mensaje de error.	
Poscondiciones:	Se han mostrado los datos correspondientes al tipo de usuario.
Prioridad:	Auxiliar.

Tabla 12. Gestionar viajes

Caso de Uso	Gestionar viajes.
Actor:	Administrador.
Propósito:	Crear, modificar y eliminar un viaje.
Precondiciones:	Estar autenticado en el sistema y haber creado al menos un bloque de transportación.
Resumen:	El caso de uso inicia cuando se crea un viaje con todas sus características y se aplican las reglas (opcional), debe brindar la posibilidad de modificar o eliminar el viaje creado.

Referencias:	RF4
Flujo Normal de Eventos	
Sección “Crear viaje”	
Acción del Actor	Respuesta del Sistema
2- Introduce los datos del viaje.	1- Muestra una interfaz con los campos para crear un viaje: fecha, tipo de viaje (ida/regreso), fecha y hora de inicio y fin del período de reservación.
	3- Crea el viaje. En caso de dejar campos vacíos ver FA1.
5- Selecciona crear reglas.	4- Muestra el viaje creado con la opción de crear las reglas que serán aplicadas al mismo. Estas reglas son opcionales.
	6- Ver sección <i>Crear regla</i> del caso de uso <i>Gestionar reglas</i> .
	7- Muestra el viaje creado con la(s) regla(s) aplicada(s) y da la posibilidad de eliminarlas. En caso que se desee eliminar una regla: Ver FA2.
Flujos Alternos	
FA1: Mostrar mensaje de error.	
FA2: Ver sección <i>Eliminar regla</i> del caso de uso <i>Gestionar reglas</i> .	
Poscondiciones:	Se ha creado un viaje.
Sección “Modificar viaje”	
Acción del Actor	Respuesta del Sistema
3- Selecciona el viaje a modificar.	1- Muestra los viajes existentes con la opción de modificar.
4- Modifica los datos de interés.	3- Muestra los datos del viaje con la posibilidad de modificarlos.
	5- Aplica la modificación, en caso de dejar campos vacíos: Ver FA1.
	6- Actualiza la base de datos.
	7- Muestra el viaje con las modificaciones realizadas.
Flujos Alternos	
FA1: Mostrar mensaje de error.	
Poscondiciones:	Se ha modificado el viaje seleccionado.
Sección “Eliminar viaje”	
Acción del Actor	Respuesta del Sistema
2- Selecciona el viaje a eliminar.	1- Muestra los viajes existentes con la opción de eliminar.
	3- Elimina el viaje seleccionado.
	4- Actualiza la base de datos.
Poscondiciones:	Se ha eliminado el viaje seleccionado.
Prioridad:	Crítico.

Tabla 13. Generar reportes

Caso de Uso	Generar reportes.
Actor:	UsuarioGR.
Propósito:	Hacer reportes por facultades, provincia, fecha y de usuarios bloqueados.
Precondiciones:	El usuario autenticado debe tener algún tipo de permiso: administración, lectura/escritura o lectura.
Resumen:	En este caso de uso se brindan varios reportes ya sea por: facultad, provincia, fecha y por usuarios bloqueados. Muestra datos del viajero y brinda la posibilidad de imprimir reporte.
Referencias:	RF9
Flujo Normal de Eventos	
Sección “Brindar reporte por facultad”	
Acción del Actor	Respuesta del Sistema
2- Selecciona el tipo de reporte que desee.	1- Muestra una interfaz con todos los posibles reportes a realizar: facultad, provincia, fecha, usuarios bloqueados y familiares.
	<p>3- En caso de seleccionar reporte por:</p> <p>Facultad:</p> <ul style="list-style-type: none"> • Para estudiantes: muestra todas las facultades, el total de viajeros de cada una y un total general. Dentro de cada facultad muestra el total de viajeros por año y un total general. Dentro de cada año te muestra los datos de los estudiantes: nombre, C.I, provincia, municipio, localidad, facultad, grupo, año y total. • Para trabajadores internos: muestra todas las facultades, el total de viajeros de cada una y un total general. Dentro de cada facultad muestra las áreas, el total de viajeros de cada una y un total general. Dentro de cada área muestra los datos de los trabajadores: nombre, C.I, provincia, municipio, localidad, cargo, edificio, apartamento, teléfono. <p>Provincia:</p> <ul style="list-style-type: none"> • Muestra todas las provincias con el total de viajeros por cada una y un total general. Dentro de cada provincia muestra los municipios con el total de viajeros de cada uno y el total general. Dentro de cada municipio muestra los datos de los viajeros del mismo: CI, nombre, Facultad (estudiantes) / área (trabajadores), año (estudiantes), solapín, usuario, edificio, apartamento, teléfono. <p>Fecha:</p> <ul style="list-style-type: none"> • Muestra las fechas de salida, el total por cada fecha y un total general. Dentro de cada fecha muestra el total por facultades (estudiantes) y área (trabajadores) y un total general. <p>Usuarios bloqueados:</p>

	<ul style="list-style-type: none"> Muestra los datos de los usuarios bloqueados: nombre, usuario, facultad, provincia, municipio, fecha inicio/fin del bloqueo y sanción. <p>Familiares:</p> <ul style="list-style-type: none"> Muestra la cantidad de familiares que van a viajar por cada provincia y el total. Dentro de cada provincia muestra los municipios con el total de familiares que van a viajar y el total general. Muestra los familiares que van a viajar con un destino diferente al de la persona UCI al que están adjuntos. El reporte debe mostrar los siguientes datos: CI, nombre, municipio y parentesco del familiar; nombre, provincia/municipio de la persona UCI.
	4- Muestra el reporte seleccionado.
	5- Posibilita imprimir cualquier reporte.
Poscondiciones:	Se han obtenido reportes.
Prioridad:	Auxiliar.

Tabla 14. Generar boletines

Caso de Uso	Generar boletines.	
Actor:	Administrador.	
Propósito:	Crear e imprimir boletines con datos de la reservación.	
Precondiciones:	Tener al menos realizada una distribución.	
Resumen:	El caso de uso inicia cuando se genera un documento con formato que permita introducir datos de la reserva, luego se selecciona la distribución que se desea imprimir, para una posterior impresión de la misma.	
Referencias:	RF14	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
2- Selecciona la opción de generar boletín.	1- Muestra una interfaz con la opción de generar boletines de las reservaciones realizadas.	
4- Selecciona la distribución que desea imprimir.	3- Muestra documento con formato que permita imprimir los datos de la reservación hecha por los usuarios: nombre, ci, facultad / área, grupo, provincia, municipio, localidad, fecha y hora de salida y llegada, punto de concentración salida y llegada, nombre del ómnibus. Y la opción de escoger la distribución que desea imprimir: provincia, municipio, tipo de transporte, persona, facultad o grupo.	
	5- Imprime la distribución seleccionada.	
Poscondiciones:	Se ha imprimido el/los boletín(es) con la distribución deseada.	
Prioridad:	Auxiliar.	

Tabla 15. Gestionar coches

Caso de Uso:	Gestionar coches.	
Actor:	Administrador.	
Propósito:	Crear coche.	
Precondiciones:	Haber creado un tren.	
Resumen:	Crea un coche dándole la posibilidad de modificarlo o eliminarlo.	
Referencias:	RF11	
Flujo Normal de Eventos		
Sección "Crear coche"		
Acción del Actor	Respuesta del Sistema	
2- Introduce los datos. En caso de dejar campos vacíos: Ver FA1.	1- Muestra una interfaz con la posibilidad de crear un coche con los siguientes campos: capacidad, profesor responsable y asientos.	
	3- Crea el coche.	
	4- Actualiza la base de datos.	
	5- Muestra el coche creado con sus datos.	
Flujos Alternos		
FA1: Muestra mensaje de error.		
Poscondiciones:	Se ha creado un coche.	
Sección "Modificar coche"		
Acción del Actor	Respuesta del Sistema	
2- Selecciona el coche a modificar.	1- Muestra los coches del tren con sus datos y da la posibilidad de modificarlos.	
4- Modifica los datos de interés. En caso de que falte alguno de los datos Ver FA1.	3- Muestra una interfaz con la posibilidad de modificar sus datos.	
	5- Aplica la modificación realizada por el usuario en la base de datos.	
	6- Muestra el coche con las modificaciones realizadas.	
Flujos Alternos		
FA1: Muestra mensaje de error.		
Poscondiciones:	Se ha modificado el coche seleccionado.	
Sección "Eliminar coche"		
Acción del Actor	Respuesta del Sistema	
2- Selecciona el coche a eliminar.	1- Muestra los coches del tren con sus datos y da la posibilidad de eliminarlos.	
	3- Elimina el coche seleccionado de la base de datos.	
	4- Refresca la página.	
Poscondiciones:	Se ha eliminado el coche seleccionado.	
Prioridad:	Secundario.	