



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

SISTEMA DE RESERVACIONES DE EXTENSIÓN UNIVERSITARIA

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO



AUTORES

Andrés Ballester Marsal

Alejandro Notario Laborí

TUTOR

Ing. Isuel Méndez Roldán

Ciudad de La Habana. Marzo de 2006

COMUNES

Fabio, Matilde, Joel

Tutor

Diana, Susel y Arianne ¿Qué nos hubiéramos hecho sin ustedes...?

Profesores y compañeros

Trabajadores e Instructores de Extensión Universitaria

Adrián

DE ANDRÉS

DE ALEJANDRO

A Andrés, por permitirme compartir esta aventura y por las dudas...

A mis amigos en la UCI: Julio, Tamara, Pedro E., Pedro F., Adonis, Mailén, Alberto, Ariel, Jose, Ygraine, Ostwald, Raida; los que no están: Misael, Guille, Pavel, Manuel, Hernán, Prendes; y todos los demás que se me quedan por poner, que harían esta lista bien grande... gracias por un montón de cosas, especialmente por el cruce entre diferentes culturas...

A mis amigos de Pinar: Mario, Raúl, María, Josny, Marquito, Omar, Milaysi (la flaca), Mucky, René, Ernesto, Rocío, Eusebio, por los maravillosos momentos que hemos vivido juntos (y por los que nos faltan).

A Yesid (mi "hermana putativa" y "futura esposa"), por todo...

A mis profesores y compañeros de la UPR, especialmente a Beatriz, Rolando y Vázquez.

A Teresita, Pedro Mercedes, Frank, Marlene, Jorgito, Maribel, Yanara, Mariela P., y a toda la gente que me ha ayudado de una forma u otra a llegar hasta aquí y no cupo por falta de espacio...

DE ANDRÉS

DE ALEJANDRO

A mi papá, que no pudo verme graduado, pero que le hubiera gustado mucho.

A mi mamá por su amor y por ser mi apoyo en todo.

Y Mariela, Ive y Angelito por ser los mejores hermanos del mundo.

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN DEL TEMA	4
1.1 Introducción.....	4
1.2 La Vicerrectoría de Extensión Universitaria.	4
1.2.1 El Proceso de Reservaciones.	5
1.3 Situación Problemática.....	9
1.4 Propuesta de Solución.	10
1.5 Otros Sistemas Automatizados de Gestión de Reservaciones.....	10
1.6 Conclusiones.....	11
CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.....	12
2.1 Introducción.....	12
2.2 Internet y la Web.	12
2.2.1 World Wide Web.....	12
2.3 Las aplicaciones Web.	13
2.3.1 Modelo Cliente-Servidor.	13
2.3.2 Origen de las Aplicaciones Web.....	14
2.3.3 Concepto y características de las Aplicaciones Web.....	15
2.3.4 Arquitectura de una aplicación Web.....	16
2.4 Aplicaciones de Internet Ricas.	17
2.4.1 Características de las RIA.	18
2.4.2 Arquitectura de una RIA.	19
2.4.3 AJAX.....	20
2.5 Tecnologías Web del lado del cliente.....	22
2.5.2 HTML.....	22
2.5.3 XML.	22
2.5.4 Hojas de Estilo en Cascada.	24
2.5.5 Javascript.	24
2.6 Lenguajes de programación del lado del servidor.....	26
2.6.1 Perl (Practical Extracting and Reporting Language).	26
2.6.2 ASP (Active Server Pages).	26
2.6.3 PHP (Personal Home Page).....	27
2.6.4 Java y tecnologías asociadas.....	27
2.7 Sistemas Gestores de Bases de Datos.....	29
2.7.1 Microsoft SQL Server.	29
2.7.2 Oracle.	29
2.7.3 MySQL.....	30
2.7.4 PostgreSQL.....	30
2.8 Metodologías de Desarrollo de Software.	31
2.8.1 Algunas Metodologías de Desarrollo de Software.	31
2.8.2 El Proceso Unificado de Racional.	32
2.8.2.1 Características de RUP.	33
2.8.2.2 La vida de un producto de Software según RUP.	34
2.8.3 El Lenguaje Unificado de Modelado.....	35
2.9 Software Libre.	36
2.10 Justificación de la propuesta de solución.....	37
2.11 Conclusiones.....	37
CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	38
3.1 Introducción.....	38

3.2. Modelo del negocio propuesto.....	38
3.2.1 Actores / Trabajadores del negocio.....	38
3.2.2 Reglas del Negocio.....	40
3.2.3 Descripción de los procesos del negocio propuesto.....	40
3.3 Especificación de los requerimientos de software.....	44
3.3.1 Requerimientos funcionales.....	44
3.3.2 Requerimientos no funcionales.	49
3.4 Modelo del sistema.....	50
3.4.1 Actores del sistema.....	50
3.4.2 Modelo de casos de uso del Sistema.....	51
3.4.3 Descripción de los casos de uso del sistema.....	52
3.4.3.1 Reservasiones.....	53
3.4.3.2 Gestión de Reservasiones / Gestión de Salidas.....	56
3.4.3.3 Gestión de Reservasiones / Gestión de Avisos.....	61
3.4.3.4 Gestión de Reservasiones / Gestión de Usuarios.....	62
3.4.3.5 Gestión de Reservasiones / Gestión de Reportes.....	64
3.5 Conclusiones.....	65
CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	66
4.1 Introducción.....	66
4.2 Patrones.....	66
4.2.1 Patrones de Arquitectura.....	66
4.2.2 Patrones de Diseño.....	67
4.3 Modelo de Diseño.....	67
4.3.1 Clases Persistentes.....	67
4.3.3 Diagramas de Clases Web.....	67
4.4 Modelo de Datos.....	68
4.5 Diseño de la Interfaz de la Aplicación.....	68
4.5.1 Estándares de la interfaz.....	69
4.5.2 Concepción de la ayuda.....	69
4.5.3 Tratamiento de errores.....	69
4.6 Modelo de Despliegue.....	69
4.7 Modelo de Implementación.....	70
4.8 Conclusiones.....	70
CAPÍTULO 5. ESTUDIO DE FACTIBILIDAD.....	71
5.1 Introducción.....	71
5.2 El Modelo de Estimación de Costos COCOMO II.....	71
5.3 Estimación del Costo.....	72
5.4 Beneficios.....	75
5.5 Análisis de costo / beneficio.....	76
5.6 Conclusiones.....	76
CONCLUSIONES	78
RECOMENDACIONES	79
BIBLIOGRAFÍA	80
GLOSARIO DE TÉRMINOS	81
ANEXOS	82
Anexo 1. Modelo del Negocio Propuesto.....	82
Anexo 2. Modelo del Sistema.....	92
Anexo 3. Diagrama de Clases Persistentes.....	94
Anexo 4. Diagramas de Clases Web.....	95
Anexo 5. Modelo de Datos.....	96
Anexo 6. Interfaz de la Aplicación.....	97

Anexo 7. Modelo de Despliegue.	98
Anexo 8. Diagrama de Componentes.	99

ÍNDICE DE TABLAS

Capítulo 3.

Tabla 3. 1 Actores y trabajadores del negocio propuesto.....	40
Tabla 3. 2 Actores del sistema.....	51
Tabla 3. 3 Descripción del CUS Autenticar Usuario.....	53
Tabla 3. 4 Descripción del CUS Comenzar Reservaciones.....	53
Tabla 3. 5 Descripción del CUS Finalizar Cancelaciones.....	53
Tabla 3. 6 Descripción del CUS Explorar Salidas Disponibles.....	54
Tabla 3. 7 Descripción del CUS Reservar Salida.....	55
Tabla 3. 8 Descripción del CUS Cancelar Reservación.....	55
Tabla 3. 9 Descripción del CUS Ver Avisos.....	56
Tabla 3. 10 Descripción del CUS Ver Perfil.....	56
Tabla 3. 11 Descripción del CUS Explorar Salidas Publicadas.....	57
Tabla 3. 12 Descripción del CUS Publicar Salida.....	57
Tabla 3. 13 Descripción del CUS Eliminar Salida.....	58
Tabla 3. 14 Descripción del CUS Ver Datos de Una Salida.....	58
Tabla 3. 15 Descripción del CUS Modificar Datos.....	59
Tabla 3. 16 Descripción del CUS Ver Lista de Usuarios Reservados.....	59
Tabla 3. 17 Descripción del CUS Registrar Inasistencia.....	60
Tabla 3. 18 Descripción del CUS Confirmar Pago.....	60
Tabla 3. 19 Descripción del CUS Archivar Datos.....	61
Tabla 3. 20 Descripción del CUS Explorar Avisos Publicados.....	62
Tabla 3. 21 Descripción del CUS Publicar Aviso.....	62
Tabla 3. 22 Descripción del CUS Buscar Usuario.....	64
Tabla 3. 23 Descripción del CUS Elaborar Reporte.....	64

Capítulo 5.

Tabla 5. 1 Entradas Externas.....	72
Tabla 5. 2 Salidas Externas.....	72
Tabla 5. 3 Consultas Externas.....	73
Tabla 5. 4 Ficheros Lógicos Internos.....	73
Tabla 5. 5 Ficheros de Interfaz Externa.....	73
Tabla 5. 6 Puntos de Función Desajustados.....	74
Tabla 5. 7 Instrucciones Fuente.....	74
Tabla 5. 8 Factores de Escala.....	74
Tabla 5. 9 Factores Modificadores del Esfuerzo.....	75

ÍNDICE DE FIGURAS

Capítulo 1.

Fig. 1. 1 Instancias de la VEU que intervienen en la gestión del proceso de reservaciones.	7
Fig. 1. 2 Ejemplo de aviso sobre oferta de salida publicado en la intranet.	8

Capítulo 2.

Fig. 2. 1 Arquitectura de una aplicación Web.....	17
Fig. 2. 3 Diferencia entre el modelo clásico de una aplicación Web y el modelo que propone Ajax.	21
Fig. 2. 5 Fases y flujo de trabajo de RUP.....	35

Capítulo 3.

Fig. 3. 1 Actores / Trabajadores del Negocio	39
Fig. 3. 2 Diagrama de Casos de Uso del Negocio	41
Fig. 3. 3 Actores del Sistema	50
Fig. 3. 4 Diagrama de Casos de Uso del Sistema	52

Anexos.

Fig. 6. 1 Diagrama de Actividad. CUN Planificar Salidas.....	82
Fig. 6. 2 MON. CUN Planificar Salidas.....	82
Fig. 6. 3 Diagrama de Actividad. CUN Aprobar Plan de Salidas en Reunión de Coordinación. .	83
Fig. 6. 4 MON. CUN Aprobar Plan de Salidas en Reunión de Coordinación.....	83
Fig. 6. 5 Diagrama de Actividad. CUN Publicar Aviso.....	84
Fig. 6. 6 MON. CUN Publicar Aviso.	84
Fig. 6. 7 Diagrama de Actividad. CUN Reservar Salida.....	85
Fig. 6. 8 MON. CUN Reservar Salida.....	85
Fig. 6. 9 Diagrama de Actividad. CUN Pagar Reservación.....	86
Fig. 6. 10 MON. CUN Pagar Reservación.....	86
Fig. 6. 11 Diagrama de Actividad. CUN Recoger Entrada.	87
Fig. 6. 12 MON. CUN Recoger Entrada.....	87
Fig. 6. 13 Diagrama de Actividad. CUN Cancelar Reservación.....	88
Fig. 6. 14 MON. CUN Cancelar Reservación.....	88
Fig. 6. 15 Diagrama de Actividad. CUN Efectuar Salida.	89
Fig. 6. 16 MON. CUN Efectuar Salida.....	89
Fig. 6. 17 Diagrama de Actividad. CUN Archivar Información.	90
Fig. 6. 18 MON. CUN Archivar Información.	90
Fig. 6. 19 Diagrama de Actividad. CUN Pedir Reporte.	91
Fig. 6. 20 MON. CUN Pedir Reporte.....	91
Fig. 6. 21 Diagrama de CUS. Subsistema Reservaciones.....	92
Fig. 6. 22 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Salidas.	92
Fig. 6. 23 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Avisos.	93
Fig. 6. 24 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Usuarios....	93
Fig. 6. 25 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Reportes. ..	93
Fig. 6. 26 Diagrama de CUS. Subsistema Administración.....	93

Fig. 6. 27 Diagrama de Clases Persistentes	94
Fig. 6. 28 Diagrama de Despliegue.....	98

La Universidad de las Ciencias Informáticas (UCI) utiliza una fuerte infraestructura tecnológica, informática y telemática para la formación de profesionales altamente calificados y la producción de aplicaciones informáticas. El uso eficiente de esta infraestructura tecnológica constituye un reto para la Universidad y por esta razón una de las tareas fundamentales que se llevan a cabo es la de informatizar los procesos que forman parte de la vida universitaria.

La Vicerrectoría de Extensión Universitaria, brinda a la comunidad universitaria un servicio de reservaciones para salidas y visitas a centros culturales y recreativos. Actualmente el proceso de reservación para dichas actividades presenta dificultades, pues el mismo se realiza de manera manual, lo que trae como consecuencias:

- Largas colas para efectuar las reservaciones que provocan pérdida de tiempo y dificultades en el trabajo del personal, así como malestar entre los usuarios de este servicio.
- No existe la posibilidad de controlar la asistencia a las actividades, lo que provoca la falta de optimización de los recursos de transporte que se ponen a disposición de las salidas.
- Los avisos de ofertas de salidas disponibles se publican en la Intranet por falta de un espacio propio para este fin y en ocasiones los usuarios no llegan a tener conocimiento de dichas ofertas, en muchos casos por falta de costumbre de consultar esta sección de la intranet.
- Dificultades en la elaboración de estadísticas, lo que provoca que no se conozca con certeza cuál es la tendencia de los usuarios en cuanto a la preferencia por las distintas opciones, haciendo que el espectro de ofertas sea más limitado, no solo en cuanto a gustos, sino también por la incidencia que tiene dicha actividad en la formación integral.

De lo anteriormente dicho, se deriva la necesidad de un sistema que automatice el proceso de reservaciones de salidas y visitas a centros culturales y recreativos, en virtud de lo cual, el **problema científico** de la presente investigación queda formulado

como: *¿Qué sistema automatizado se requerirá para optimizar la gestión del proceso de reservaciones para salidas y visitas a centros culturales y recreativos que ofrece la Vicerrectoría de Extensión Universitaria de la UCI?*

En correspondencia con el problema, el **objeto de estudio** lo constituye: *El proceso de reservaciones para salidas y visitas a centros culturales y recreativos de la Vicerrectoría de Extensión Universitaria de la UCI.* El **campo de acción** incluye la *optimización de la gestión del proceso de reservaciones para salidas y visitas a centros culturales y recreativos.*

El **objetivo general** de esta investigación es: *Desarrollar un sistema automatizado que permita optimizar la gestión del proceso de reservaciones para salidas y visitas a centros culturales y recreativos de la Vicerrectoría de Extensión Universitaria de la UCI.*

De donde se derivan los siguientes **objetivos específicos**:

- Diseñar un módulo que permita a los usuarios la reservación para ofertas de salidas y visitas culturales y recreativas.
- Diseñar un módulo que facilite la gestión del proceso de reservaciones para salidas, avisos, usuarios y reportes estadísticos.
- Diseñar un módulo que brinde las funcionalidades administrativas que den soporte a los dos módulos anteriores.
- Implementar las funcionalidades priorizadas, sobre todo las relacionadas con las reservaciones y la gestión de salidas y avisos.

Tareas de la Investigación:

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

Estructuración del Contenido:

Capítulo 1. Fundamentación del tema: se describe la situación actual de la Vicerrectoría de Extensión Universitaria y del proceso de reservaciones, se exponen los principales problemas que motivan el desarrollo de esta investigación, se describe brevemente en qué consiste la propuesta y se hace referencia a otros sistemas automatizados de reservaciones.

Capítulo 2. Tendencias y tecnologías actuales a considerar: Se hace un estudio referente a las tendencias y tecnologías existentes en la actualidad que se deben considerar para hacer la selección de aquellas que se utilizarán en la confección del proyecto y se plantea dicha selección fundamentado en cada caso.

Capítulo 3. Descripción de la solución propuesta: Se describen los procesos actuales a través de un modelo de negocio, el cual sirve de base para determinar qué es lo que se va a desarrollar. Se determinan las funcionalidades del sistema propuesto y se describen en detalle.

Capítulo 4. Construcción de la solución propuesta: Aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de diseño, se plantea el modelo de datos, y se especifican los principios para el diseño gráfico y la implementación.

Capítulo 5. Estudio de Factibilidad: Contiene el estudio de factibilidad realizado para el sistema, se enumeran los beneficios y se analizan los costos que representa la elaboración de la propuesta de solución.

1.1 Introducción

En este capítulo se describe el flujo de trabajo actual en la gestión del proceso de reservaciones para salidas y visitas a centros culturales y recreativos que brinda la Vicerrectoría de Extensión Universitaria y se identifican los principales problemas que se detectaron. Se expone también la propuesta de solución y se analizan otros sistemas automatizados que existen actualmente para reservaciones.

1.2 La Vicerrectoría de Extensión Universitaria.

La Vicerrectoría de Extensión Universitaria (VREU) de la Universidad de las Ciencias Informáticas tiene como Misión: La formación cultural, de valores, actitudes y comportamientos de forma continua de la comunidad universitaria en un espacio abierto, cooperativo y flexible, donde se desarrollen proyectos, actividades y programas a través del protagonismo de cada uno y de todos, que permita su incidencia positiva y conciente en el proyecto social que construimos. Y la Visión es: Convertir la extensión universitaria en un espacio dinámico para el crecimiento, desarrollo, promoción y difusión de la cultura, la creatividad en todos y en todo a través del acceso a niveles de conocimientos y códigos culturales, que permitan a cada quien ser sujeto activo en el proceso de la transformación y elevación de la calidad de vida del entorno social en el cual interactuamos.

Sus objetivos fundamentales son:

1. Desarrollar una comunidad cultural que asuma en lo cognoscitivo y en los modos de actuación los valores esenciales de la cultura local, nacional y universal.
2. Consolidar la Extensión en la Universidad de las Ciencias Informáticas (UCI) como una herramienta de transformación social y educativa, que contribuya de forma organizada y participativa al desarrollo integral de la comunidad

universitaria, a través del crecimiento y desarrollo de opciones, proyectos y servicios acorde también a sus intereses.

3. Integrar los procesos universitarios a partir del carácter totalizador de la extensión en tanto transmisora entre la acepción más amplia de la palabra cultura y la transformación de la sociedad inmersa en el uso cada vez más amplio de los servicios informáticos.
4. Ejecutora del sistema de acciones integradas del ciclo reproductivo de la cultura a través de la creación, conservación, difusión y disfrute para el logro de niveles superiores en el desarrollo cultural de la comunidad universitaria y extrauniversitaria.
5. Ejecutar acciones distintivas desde el uso de las TIC para la superación cultural del claustro profesores en aras de la formación del profesional de las Ciencias Informáticas.
6. Promocionar los resultados de la ciencia y la innovación tecnológica a partir del aprovechamiento intensivo de las TIC, y el fortalecimiento las investigaciones en el campo de la extensión universitaria y de las ciencias que son afines a este proceso.
7. Ejecutar un trabajo sociocultural comunitario a través de la utilización efectiva de las TIC en función de contribuir a la informatización de la sociedad cubana desde la creación artística y literaria.

Como una vía para dar cumplimiento a estos objetivos la VREU brinda a través de de las Direcciones de Recreación y Extensión Universitaria de la VREU, brindan un servicio de reservaciones para salidas y visitas recreativas y de interés cultural, el cual se describe a continuación.

1.2.1 El Proceso de Reservaciones.

La DR y la DEU gestionan el acceso de los miembros de la comunidad universitaria a un grupo de opciones culturales y recreativas.

La DR oferta las siguientes opciones:

- **Salidas de Paseo.** Se ofertan los domingos y son generalmente al Parque de la Fraternidad y a Coppelia. Estas salidas se ofertan para que los interesados puedan pasear por la ciudad por su cuenta. Los ómnibus salen de la UCI a las 9:00 a.m., dejan a los reservados en estos puntos y los recogen a las 4:00 p.m.
- **Visitas Dirigidas.** Son salidas a centros recreativos (Ej.: Acuario Nacional, Parque Morro-Cabaña, Zoológico, etc.). Estas salidas son generalmente los fines de semana, aunque excepcionalmente pudieran ser ofertadas para otros días.

Por su parte, la DEU se encarga de gestionar:

- **Salidas a Centros y Eventos Culturales.** Son las salidas a centros culturales y eventos culturales en general (Ej: Conciertos, Exposiciones de Artes Plásticas, Vista a Museos, etc.). Algunas de estas opciones se coordinan con antelación con los teatros y demás centros culturales, para las actividades que incluyen pago. Se gestiona con la entidad un número de entradas y se ofertan a los interesados que para reservar deben abonar el pago correspondiente (Ej: Función del Ballet Nacional de Cuba, Galas Culturales, Conciertos, etc.).

En la decisión de la propuesta de salidas las dos direcciones siguen un proceso muy similar. En el caso de la DR, los encargados de coordinar y decidir qué opciones se ofertarán son los miembros del equipo de trabajo del Departamento Técnico. Por la parte de la DEU se encargan de esta labor la Jefa del Departamento de Promoción y la Especialista General de dicho departamento (Ver Figura 1.1). La mecánica de trabajo que se sigue consiste en hacer un sondeo de las opciones disponibles y si son necesarias, hacer las coordinaciones con los centros recreativos y culturales, además de efectuar el pago de las entradas en los teatros y demás instituciones para las actividades que así lo requieran después de efectuado el proceso de reservaciones. Cuando se han decidido las propuestas que se ofertarán, se elabora un plan con las mismas para que sea aprobado por los Directores de ambas direcciones y posteriormente se lleva a la Reunión de Coordinación.



Fig. 1. 1 Instancias de la VEU que intervienen en la gestión del proceso de reservaciones.

La Reunión de Coordinación se realiza todos los jueves y asisten las Direcciones (Transporte, Alimentación, Recreación, Extensión Universitaria, etc.) y Organizaciones de Masa (UJC, FEU, etc.) de la universidad. En esta reunión se puntualizan las actividades que quedan pendientes en la semana y se coordinan, discuten y aprueban las propuestas de salidas para la semana próxima.


Cuando se tienen definidas las salidas que se ofertarán, se publica una nota en la sección de Avisos de la Intranet informando sobre las mismas para el conocimiento de la comunidad universitaria. También en caso de que se suspenda una actividad, se les informa a los interesados por esta vía.

El texto de estos avisos generalmente incluye una descripción sobre la salida que se oferta, así como el día y la hora de la actividad, la hora de salida del transporte, la fecha, hora y lugar del inicio de las reservaciones y si la salida incluye pago, se informa además el precio de las entradas (Ver Figura 1.2).

AVISOS
Oferta de Teatro

Sección de Avisos de la Intranet

Oferta de Teatro
16/01/2006
Danza Contemporánea.



Obra: Compás.
Gran Teatro de La Habana, Sala García Lorca.

Días: Sábado 21, a las 8:30 pm, con salida de la UCI a las 7:00 pm.
Domingo 22, a las 5:00 pm, con salida de la UCI a las 3:45 pm.

Precio: 5.00 MN.

Todos los estudiantes y profesores internos interesados pueden realizar su reservación este martes 17, a las 7:00 pm, en el Centro Cultural.

Dir.de Extensión Universitaria

Texto del aviso

Fig. 1. 2 Ejemplo de aviso sobre oferta de salida publicado en la intranet.

Los usuarios a los que va dirigido el servicio de reservaciones son los estudiantes y profesores internos de la UCI, ya que estos no tienen la posibilidad de asistir a las actividades fuera de la Universidad. Aunque en contadas ocasiones, excepcionalmente se le brinda el servicio a profesores externos y trabajadores (Ej.: para la Feria del libro).

Para hacer las reservaciones los usuarios deben dirigirse al lugar señalado en el aviso, el día y la hora indicados, portando el solapín para ser identificados. Un trabajador de la entidad, encargado de este proceso irá anotando a los reservados en una lista hasta cubrir las capacidades disponibles. El usuario puede solicitar reservar para acompañantes, aunque esto depende de la salida y se define con anterioridad si se admitirá y en caso afirmativo, el número de acompañantes permitido. Si la salida requiere de pago se le solicita al usuario la cantidad de dinero necesaria y se le informa cuando debe regresar a recoger la invitación o entrada.

Cuando llega el momento de abordar el transporte para efectuar la salida, un trabajador de la entidad lleva el listado de reservados y comienza a llamar por el orden y a medida que los usuarios van subiendo al transporte, se les hace una marca de confirmación en

la lista. Para algunos tipos de salidas (Por ejemplo las Salidas de Paseo) se les entrega a los usuarios un ticket que le permite usar la transportación de regreso a la UCI.

1.3 Situación Problémica.

Al realizar un análisis del proceso actual, se han detectado problemas que frenan el óptimo funcionamiento de tan importante servicio.

El proceso se lleva a cabo de forma manual, lo que trae consigo pérdidas de tiempo y dificultades en el trabajo, de la misma manera, al realizarse las reservaciones los usuarios forman largas colas y junto a la pérdida de tiempo, se ocasionan molestias a los mismos y a los que brindan el servicio.

Como no existe un control preciso, no es posible detectar la inasistencia de los usuarios a las actividades luego de haber reservado. Esto provoca que no se puedan optimizar los recursos de transporte que se ponen a disposición de las salidas.

Los avisos de ofertas de salidas disponibles se publican en la Intranet por falta de un espacio propio para este fin y en ocasiones los usuarios no llegan a tener conocimiento de dichas ofertas, en muchos casos por falta de costumbre de consultar esta sección de la Intranet.

Las estadísticas sobre el proceso se guardan en “copia dura”, lo que dificulta la tarea de elaborar reportes sobre varias cuestiones, por ejemplo sobre los usuarios que usan el servicio y a los encargados de realizar la planificación de ofertas de salidas les es difícil conocer las preferencias, lo que provoca que se limite el espectro de opciones, que en ocasiones no se tengan en cuenta los gustos y se dificulte el trabajo de promoción hacia las actividades que son menos elegidas y que pudieran contribuir a la formación integral.

1.4 Propuesta de Solución.

Teniendo en cuenta la situación problemática planteada, se propone la elaboración de un sistema informático que permita automatizar el proceso de reservaciones para salidas y visitas a centros culturales y recreativos de la Vicerrectoría de Extensión Universitaria.

Dicho sistema deberá ser capaz de posibilitar a los trabajadores de la VREU gestionar el proceso de reservaciones para salidas y visitas a centros culturales y recreativos de una forma más óptima, ya que suprimirá todo el trabajo manual en lo referente a las reservaciones, la publicación de avisos y la elaboración de reportes estadísticos y les brindará nuevas funcionalidades que no existían antes de la automatización, como por ejemplo, el envío de confirmaciones vía correo electrónico, la búsqueda de usuarios, las penalizaciones, entre otras más. De la misma forma, los usuarios de dicho servicio se beneficiarán al suprimirse las colas para efectuar las reservaciones.

1.5 Otros Sistemas Automatizados de Gestión de Reservaciones.

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

Un ejemplo de esto es Ticketmaster (<http://www.ticketmaster.com>), que es la mayor compañía en Internet dedicada a la distribución de entradas y reservaciones para conciertos, museos, teatros y múltiples categorías de eventos en general en varios países. La empresa vendió 98 millones de tickets valorados en 5 billones de dólares durante el año 2004.

En Cuba, también se ha comenzado a explotar este medio, sobre todo las empresas de la rama del turismo, y algunas ya tienen en sus sitios de Internet sistemas para la reservación de autos, habitaciones de hotel, billetes de avión, etcétera. Buenos ejemplos son la empresa Cubanacán, que brinda un servicio de renta de autos online desde su sitio www.cubanacan.cu/FRANCES/CUBACAR/reserv.asp y la empresa CITMATEL, con su portal CUBA (www.cuba.cu) que brinda un servicio de reservación de habitaciones de hotel y vuelos a Cuba.

Es importante destacar la gestión del Teatro Tomás Terry, de Cienfuegos, que brinda a sus usuarios un servicio de reservaciones de entradas online en el sitio http://www.azurina.cult.cu/out_sites/A_ESCENI/Terry/Online/Reservar_Entrada.asp y es la primera iniciativa relacionada con las reservaciones culturales y recreativas en Cuba.

En la UCI se cuenta también con un sistema de reservaciones del pase de fin de semana (<http://intranet.uci.cu/pase/default.pl>) y aunque este no se relaciona con el sistema que se propone desarrollar en esta investigación, constituye un ejemplo del uso de la infraestructura tecnológica, informática y telemática de la universidad con el fin de informatizar los procesos que forman parte de la vida universitaria.

1.6 Conclusiones.

Se han descrito las características de la Vicerrectoría de Extensión Universitaria y el flujo actual del proceso de reservaciones para salidas y visitas a centros culturales y recreativos que brinda la misma, determinándose los problemas que presenta. Se ha planteado una propuesta de solución y se han expuesto algunos sistemas automatizados de gestión de reservaciones existentes en la actualidad.

2.1 Introducción.

En este capítulo se exponen los principales fundamentos teóricos en los que se basa la presente investigación y se hace un análisis de las tecnologías y tendencias actuales, lenguajes de programación, sistemas gestores de bases de datos y la metodología de desarrollo de que pueden ser adecuadas para la construcción del sistema que se pretende desarrollar.

2.2 Internet y la Web.

Internet, la red de redes, nació a mediados de la década de los setenta, bajo los auspicios de DARPA, la Agencia de Proyectos Avanzados para la Defensa de Estados Unidos. DARPA inició un programa de investigación de técnicas y tecnologías para unir diversas redes de conmutación de paquetes, permitiendo así a los ordenadores conectados a estas redes comunicarse entre sí de forma fácil y transparente [6]. Es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP.

En estos momentos disponer de una dirección de correo electrónico, de acceso a la Web, etc., ha dejado de ser una novedad para convertirse en algo normal en muchos países del mundo. Por eso las empresas, instituciones, administraciones y demás están migrando rápidamente todos sus servicios, aplicaciones, tiendas, etc., a un entorno Web que permita a sus clientes y usuarios acceder a todo ello por Internet.

2.2.1 World Wide Web.

La WWW (World Wide Web) o, de forma más coloquial, la Web, es un sistema hipertexto interactivo desarrollado sobre Internet. Para ver la información se utiliza una aplicación llamada navegador Web para extraer elementos de información (llamados

páginas Web), de los servidores Web (o sitios) y mostrarlos en la pantalla del usuario. El usuario puede entonces seguir los enlaces que hay en la página a otros documentos o incluso enviar información al servidor para interactuar con él. A la acción de seguir enlaces se le suele llamar navegar por la Web o explorar la Web.

En 1994 se fundó el WWW Consortium, que se convertiría en el motor de desarrollo de los estándares predominantes en la Web (<http://www.w3c.org>). A partir de ese momento, el crecimiento ya fue constante, convirtiéndose hacia finales de los noventa en el servicio insignia de Internet y dando lugar al crecimiento imparable de los servicios en línea experimentado actualmente.

El éxito espectacular de la Web se basa en dos puntales fundamentales: el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones que nos permite enviar cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones y reduzcan los costes de despliegue. El otro nos proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple [6].

2.3 Las aplicaciones Web.

Con la introducción de Internet y de la Web en concreto, se han abierto infinitas posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web.

2.3.1 Modelo Cliente-Servidor.

La utilización de las diferentes aplicaciones o servicios de Internet se lleva a cabo respondiendo al llamado modelo cliente-servidor. Cuando se utiliza un servicio en

Internet, como consultar una base de datos, transferir un fichero o participar en un foro de discusión, se establece un proceso en el que entran en juego dos partes. Por un lado, el usuario, quien ejecuta una aplicación en el ordenador local: el denominado programa cliente. Este programa cliente se encarga de ponerse en contacto con el ordenador remoto para solicitar el servicio deseado. El ordenador remoto por su parte responderá a lo solicitado por el programa cliente mediante otro programa, denominado programa servidor. Los términos cliente y servidor se utilizan tanto para referirse a los programas que cumplen estas funciones, como a los ordenadores donde son ejecutados esos programas.

El programa o los programas cliente que el usuario utiliza para acceder a los servicios de Internet realizan dos funciones distintas. Por una parte, se encargan de gestionar la comunicación con el ordenador servidor, de solicitar un servicio concreto y de recibir los datos enviados por éste; y por otra, es la herramienta que presenta al usuario los datos en pantalla y que le ofrece los comandos necesarios para utilizar las prestaciones que ofrece el servidor.

2.3.2 Origen de las Aplicaciones Web.

En los comienzos de Internet, la Web era simplemente una gran colección de páginas estáticas que ofrecía documentos, imágenes, etc., que podían consultarse y/o descargarse. En el transcurso del tiempo, la Web ha ido evolucionando y con ella los contenidos y servicios que se ofrecen.

El siguiente paso en su evolución fue la de incluir un método para la confección de páginas Web con contenidos dinámicos que permitieran que lo mostrado variara en función de las peticiones enviadas al servidor Web. Dicho método fue denominado CGI (common gateway interface). Los CGI ofrecían una pasarela de intercambio de información (gateway) entre un servidor Web y programas externos.

El esquema de funcionamiento de los CGI tenía y sigue teniendo un punto débil bastante importante: cada vez que se recibe una petición, el servidor Web lanza un proceso hijo que ejecuta el programa correspondiente. Esto provoca dos tipos de situaciones que pueden perjudicar seriamente al servidor:

- Si la Web tuviera muchos accesos CGI, el servidor podría llegar a saturarse debido a la gran cantidad de programas ejecutados simultáneamente.
- Los CGI, al proporcionar una gran variedad de lenguajes de programación para ser desarrollados, pueden estar escritos en algún lenguaje interpretado (Perl, Python, C, etc.) o en algún lenguaje que requiera de un runtime environment (VB, Java, etc.), y esto implicaría una gran carga para el servidor Web.

Por estos motivos, se empezaron a desarrollar tecnologías alternativas a los CGI que solucionaran este grave problema de rendimiento. Las soluciones vinieron principalmente por dos vías diferentes. Por un lado se diseñan sistemas de ejecución de módulos integrados en el servidor que evitan que éste tenga que instanciar y ejecutar multitud de programas. La otra vía consiste en dotar al servidor Web de un intérprete de algún lenguaje de programación (PHP, ASP, VBScript, etc.) que nos permita incluir las páginas Web en el código de manera que el servidor Web sea quien lo ejecute, reduciendo así el tiempo de respuesta.

A partir de este momento, se vive una explosión del número de arquitecturas y lenguajes de programación que nos permiten desarrollar aplicaciones Web. Todas ellas siguen alguna de las dos vías ya mencionadas. De ellas, las más útiles y las que más se utilizan son las que permiten mezclar los dos sistemas, es decir, un lenguaje de programación integrado que permita al servidor interpretar comandos que “incrustemos” en las páginas HTML y un sistema de ejecución de programas más enlazado con el servidor que no presente los problemas de rendimiento de los CGI.

2.3.3 Concepto y características de las Aplicaciones Web.

Se considera una aplicación Web a “un sitio Web donde la navegación a través de él y la entrada de datos por parte de un usuario, afectan el estado de la lógica del negocio. En esencia, una aplicación Web usa un sitio Web como entrada a una aplicación típica... Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web.” [2].

Las características básicas de una aplicación Web son las siguientes:

- Está alojada en un servidor Web y sigue una arquitectura cliente/servidor
- Es accesible a través de una red telemática y mediante la utilización de un navegador Web
- La lógica de la aplicación Web se ejecuta íntegramente en el servidor Web y el navegador del cliente sólo representa los datos
- La navegación a través de ella y la entrada de datos por parte de un usuario afectan el estado de la lógica de la aplicación
- Se ejecuta íntegramente en el servidor y puede interactuar con otros procesos y aplicaciones Web
- El acceso a la aplicación puede ser público o restringido, dependiendo hacia quien vaya dirigida
- Las actualizaciones y el mantenimiento de la aplicación es transparente para los usuarios, ya que éstos no deberán instalar ni actualizar nada
- Las aplicaciones son multiplataforma ya que pueden ejecutarse en cualquier Sistema Operativo que posea un navegador Web.

2.3.4 Arquitectura de una aplicación Web.

Teniendo en cuenta las múltiples opciones que tenemos al desarrollar una aplicación Web, por lo general éstas suelen estar estructuradas siguiendo un modelo basado en tres niveles (Figura 2.1).

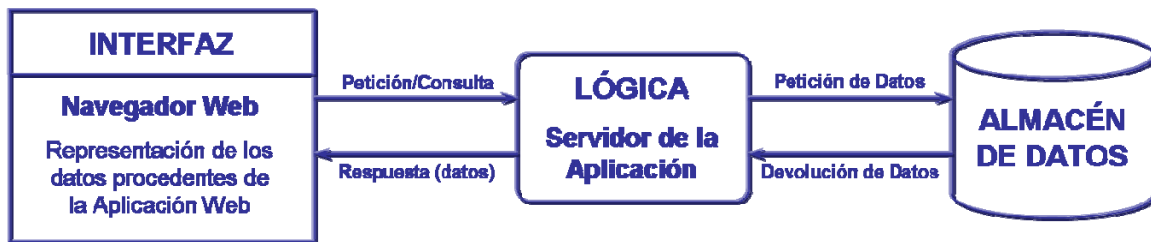


Fig. 2. 1 Arquitectura de una aplicación Web.

El primer nivel es la interfaz. Las funciones que se realizan en este nivel se limitan únicamente a enviar las peticiones/consultas que genera el usuario, obtener los datos generados por la aplicación Web (servidor) y finalmente representar los datos en el navegador.

En el segundo nivel se sitúa la lógica. En este nivel se encuentra el núcleo principal de la aplicación y es el encargado de dotar a la aplicación Web del contenido dinámico.

Las funciones que pueden ser realizadas en este nivel son las siguientes:

- interpretación de las consultas generadas por el cliente
- ejecución de la lógica necesaria para esas consultas
- obtención/almacenaje de datos
- interacción con otras aplicaciones Web
- envío de los datos generados al cliente

En el tercer nivel estarían los datos. Éste podría estar formado por una base de datos, por ficheros XML o por cualquier otro tipo de sistema de almacenaje de datos dependiendo de las necesidades de la aplicación.

2.4 Aplicaciones de Internet Ricas.

Una Aplicación de Internet Rica o RIA (Rich Internet Application) es un nuevo tipo de aplicación Web cuyo objetivo es el de incrementar y mejorar las opciones y capacidades de las aplicaciones Web tradicionales.

Las limitaciones en la capa de presentación de los actuales navegadores Web y del lenguaje HTML ha sido lo que ha impulsado a los desarrolladores a utilizar este nuevo tipo de aplicaciones que permiten, entre otras cosas, mejorar la experiencia entre el usuario y la aplicación, la ejecución de contenido multimedia y la carga de aplicaciones online/offline, dependiendo de la tecnología RIA que se utilice.

2.4.1 Características de las RIA.

Las RIA cumplen, en su mayoría, con una serie de características elementales que son las siguientes:

- Mejora de la experiencia visual gracias a la aportación de nuevos componentes más avanzados.
- Permite crear aplicaciones más atractivas mediante la utilización de audio, video y gráficos.
- La mayoría de tecnologías RIA se basan en el lenguaje XML, tanto para las interfaces gráficas como para el intercambio de datos.
- Los servidores donde se ejecutan las aplicaciones son bastante variados, pero los que más predominan son los servidores basados en Java.
- Disminuye el ancho de banda utilizado por la aplicación ya que permite almacenar más información en el cliente de manera que se reduce el uso de transacciones HTTP. Además, con esta medida, se minimiza la memoria utilizada en el servidor Web.
- El modelo petición/respuesta no es necesario para cada acción realizada en la interfaz de usuario. Con las RIA, el usuario interactúa con la interfaz de usuario y sólo se realizan comunicaciones con el servidor cuando son necesarias.
- Requiere la utilización de un rich client en el lado del usuario, que será el motor de la tecnología RIA utilizada (habitualmente será un plugin).
- Visualización y ejecución en múltiples plataformas y dispositivos heterogéneos.
- Desvinculación de la presentación de la lógica.
- Conectividad. La mayoría de las RIA permiten ejecutar las aplicaciones online/offline.

- Se posibilita la detección de eventos en la mayoría de componentes y la actualización de los mismos sin la necesidad de refrescos de pantalla.
- Permiten realizar funciones como drag&drop (arrastrar y soltar), resize (cambiar el tamaño) de objetos, etc.
- La serialización del estado de la aplicación se realiza mediante objetos
- La programación utilizada en las diferentes RIAs es compatible para todos los usuarios (navegadores) y evita tener que realizar diferentes implementaciones del mismo código.

2.4.2 Arquitectura de una RIA.

Primero se tiene el navegador Web, que contiene el plugin (rich client) necesario para interpretar los datos que le llegan de la aplicación, ya sea ficheros de video, textos, audio, etc. En esta parte de la arquitectura, a diferencia que la de las aplicaciones Web tradicionales, se almacena la parte gráfica de la aplicación (esquema XML). Con esto se evita tener que renderizar una nueva página Web cada vez que se quieren mostrar nuevos datos en pantalla.

En segundo lugar el controlador de la aplicación y el gateway. El controlador de la aplicación sería la parte de la aplicación que se encarga de interactuar con el cliente y viceversa. Luego esta parte de la aplicación utilizaría otras partes para realizar/ofrecer los servicios solicitados. La pasarela de información o gateway se encargaría de transformar los datos que maneja la aplicación a un formato de datos (en este caso XML), que pudiera entender el navegador del cliente.

El siguiente elemento de la arquitectura de una RIA es el servidor donde se alojaría la aplicación. Este servidor de aplicaciones compilará la aplicación Web en caso necesario y permitiría la comunicación de nuestra aplicación RIA con otras aplicaciones o servlets. Esta parte de la arquitectura se encargaría también de obtener los datos necesarios para la aplicación, ya sea de una base de datos o de ficheros.

Por último, en lo referido al almacenaje de los datos, que habitualmente será una base de datos donde se almacenarán aquellos relacionados con el servicio que está ofreciendo la aplicación.

2.4.3 AJAX.

AJAX, acrónimo inglés de Asynchronous JavaScript and XML (JavaScript y XML asíncrono) es una técnica, no tecnología, de desarrollo Web para crear aplicaciones ricas.

AJAX propone el desarrollo de aplicaciones Web de la siguiente forma:

- Presentación basada en estándares: XHTML y CSS
- Cambios dinámicos en la visualización del contenido mostrado y control de eventos del usuario a través del DOM (JavaScript)
- Intercambio de datos y manipulación usando XML y XSLT
- Obtención de datos de forma asíncrona a través de XMLHttpRequest
- JavaScript para unirlo todo

AJAX no constituye una tecnología en sí (por eso carece de un entorno de desarrollo propio), pero es un término que engloba a un grupo de éstas que trabajan conjuntamente.

El hecho de que el intercambio de datos se realice de forma asíncrona sirve para hacer que las aplicaciones Web funcionen de una manera casi transparente al usuario en términos de comunicación con el servidor.

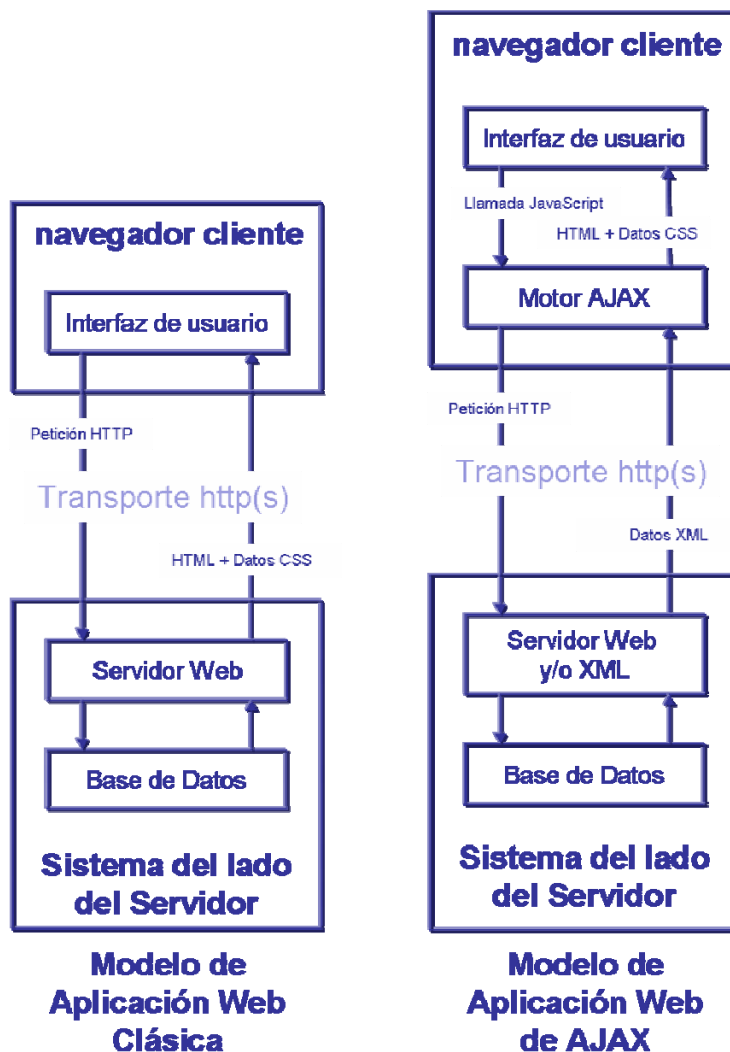


Fig. 2. 2 Diferencia entre el modelo clásico de una aplicación Web y el modelo que propone Ajax.

En el modelo clásico, cada vez que se quiere cargar una nueva página Web con nuevos datos, se envía una petición al servidor Web, y éste devuelve la página entera, que incluye tanto los datos a mostrar como la presentación de la misma.

Sin embargo utilizando el modelo AJAX, cuando se quiere cargar datos nuevos, se envía una petición HTTP al servidor Web que devuelve únicamente los datos a mostrar (en formato XML), no la presentación de la página. Con este sistema se consigue reducir el volumen de tráfico entre cliente y servidor, y también que no se tengan que cargar páginas HTML enteras cada vez que se quieren representar nuevos datos.

2.5 Tecnologías Web del lado del cliente.

A continuación se relacionan y explican las tecnologías del lado del cliente más utilizadas actualmente en el desarrollo de aplicaciones Web.

2.5.2 HTML.

El HTML, acrónimo inglés de Hypertext Markup Language (lenguaje de formato de documentos de hipertexto), es un “lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto” [7], que es el formato estándar de las páginas Web. El HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. HTML es hijo de SGML, aunque hay unas versiones de XHTML que son descendientes de XML y exigen que se escriba mucho más para facilitar la vida a los navegadores, que son aquellos programas que nos muestran información en pantalla. HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos en pantalla.

2.5.3 XML.

XML son las siglas de eXtensible Markup Language, lenguaje extensible de marcas. Se trata de un estándar del Word Wide Web Consortium, cuyo objetivo original consistía en permitir afrontar los retos de la publicación electrónica de documentos a gran escala. Actualmente, XML está empezando a desempeñar un papel muy importante en el intercambio de una gran variedad de información en la Web y en otros contextos.

XML deriva del lenguaje de marcas SGML. Concretamente, es un subconjunto de SGML que pretende que éste pueda ser servido, recibido y procesado en la Web de la misma forma que el HTML. XML ha sido diseñado buscando la simplicidad de

implementación y la interoperabilidad con SGML y HTML y tratando, por otra parte, de que sea usado para diseñar aplicaciones centradas en los datos.

XML surge en 1996 cuando se ponen de manifiesto algunas de las más destacadas carencias de HTML:

- HTML estaba optimizado para ser fácil de aprender, no para ser fácil de procesar:
 - Un solo conjunto de marcas (independientemente de las aplicaciones).
 - Una semántica predefinida para cada marca.
 - Estructuras de datos predefinidas.
- HTML sacrifica la potencia para conseguir facilidad de uso.
- HTML resulta adecuado para aplicaciones simples, pero es poco adecuado para aplicaciones complejas:
 - Conjuntos de datos complejos.
 - Datos que deben ser manipulados de formas diversas.
 - Datos para controlar programas.
 - Sin capacidad de validación formal.

Ante todo esto, el W3C desarrolló este nuevo lenguaje (XML), que proporciona:

- Extensibilidad: se pueden definir nuevas marcas y atributos según sea necesario.
- Estructura: se puede modelar cualquier tipo de datos que esté organizado jerárquicamente.
- Validez: se pueden validar automáticamente los datos (de forma estructural).
- Independencia del medio: se puede publicar el mismo contenido en multitud de medios.

En resumen, XML es una versión simplificada de SGML muy fácil de implementar, que no constituye un lenguaje, sino un meta-lenguaje diseñado para permitir la definición de un número ilimitado de lenguajes para propósitos específicos, pero que pueden ser procesados empleando las mismas herramientas independientemente del propósito para el que se han construido.

2.5.4 Hojas de Estilo en Cascada.

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML.

Las principales ventajas de utilizar CSS son, entre otras:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web remoto, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

2.5.5 Javascript.

Un lenguaje de scripts es un lenguaje usado para manipular, personalizar o automatizar las facilidades de un sistema existente. En el caso de JavaScript, dicho sistema es típicamente un Navegador Web y sus tecnologías asociadas (HTML, CSS y XML).

JavaScript es el principal lenguaje de scripts que se usa hoy en la Web del lado del cliente. Es ampliamente usado en tareas que van desde la validación de formularios de datos a la creación de complejas interfaces de usuario. Aunque el lenguaje ofrece capacidades que muchos de sus usuarios aun no han descubierto, JavaScript puede ser utilizado para manipular las marcas en el documento en el cual está contenido. Mientras mas desarrolladores descubren su verdadera fortaleza, JavaScript se está convirtiendo en una tecnología Web del lado del cliente de primera clase, junto con XHTML, CSS y XML.

El nombre JavaScript puede traer confusiones si no se conoce su historia, ya que a pesar de la similitud, este no está relacionado con Java. Netscape introdujo el lenguaje bajo el nombre LiveScript en una versión beta del Navigator 2.0 en 1995, y el lenguaje inicialmente se enfocaba a la validación de formularios. Luego fue renombrado a Javascript dada la fascinación que existía en la industria hacia todo lo relacionado con Java así como por el potencial que ofrecían ambos lenguajes para ser integrados en la construcción de aplicaciones Web.

Mientras que el nombre ha llevado a algunas confusiones por parte de los usuarios, ha sido ampliamente adoptado por los proveedores de navegadores. Actualmente los fabricantes han desarrollado sus propias extensiones del lenguaje y la mayoría de los navegadores soportan variadas versiones de JavaScript o del JScript de Microsoft.

JavaScript como casi todo lenguaje, es mejor para algunos tipos de aplicaciones. Algunos de sus usos más comunes son:

- Validación de formularios
- Efectos especiales
- Sistemas de navegación
- Generación dinámica de documentos
- Manipulación de documentos estructurados

Aun así, JavaScript tiene sus limitantes. No da soporte a las características de control de errores robustas, especificación precisa de tipos de datos, o las facilidades útiles para construir aplicaciones de gran escala. A pesar de estas carencias y muchos de los malentendidos que rodean al lenguaje, ha tenido un amplio éxito y es uno de los más populares y ampliamente usados.

2.6 Lenguajes de programación del lado del servidor.

En cualquier aplicación Web, es necesario programar el servidor, es decir, realizar una serie de programas que respondan a las peticiones del usuario, y generen páginas Web dinámicas. Como ejemplos más sobresalientes de lenguajes que se ejecutan en el servidor (en inglés, server side languages), tenemos PERL, ASP, PHP, los módulos CGI, Java, entre otros. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la información, etc.

2.6.1 Perl (Practical Extracting and Reporting Language).

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para la Web, y es considerado el lenguaje perfecto para este fin dadas sus facilidades en cuanto a la manipulación de texto. Es un lenguaje de libre uso. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP.

2.6.2 ASP (Active Server Pages).

Es una tecnología propietaria de Microsoft. Se utiliza exclusivamente en los servidores Web de Microsoft (Internet Information Server y Personal Web Server), lo cual constituye su principal desventaja. Los scripts ASP se ejecutan, por lo tanto, en el servidor y puede utilizarse conjuntamente con HTML y JavaScript para realizar tareas

interactivas y en tiempo real con el cliente. Con ASP se pueden realizar fácilmente páginas de consulta de bases de datos, funciones sencillas como obtener la fecha y la hora actual del sistema servidor, cálculos matemáticos simples, etc. Actualmente se ha presentado ya la segunda versión de ASP: el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona, y que presenta algunas diferencias en cuanto a sintaxis, de modo que se ha de tratar de distinta manera uno de otro. Para implementar ASP.NET es necesario instalar en el Servidor la Plataforma .NET.

2.6.3 PHP (Personal Home Page).

Es el acrónimo de Hypertext Preprocessor. Es un lenguaje de programación del lado del servidor gratuito, de código abierto, e independiente de plataforma, muy rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML. Su sintaxis es muy parecida a la del lenguaje C, por lo que para cualquier programador que esté familiarizado con dicha sintaxis será fácil aprender a programar en PHP. Actualmente se encuentra en su versión 5. Es evidente que se ha convertido en la gran tendencia en el mundo de Internet. Las estadísticas arrojan que cada mes su uso crece en un 15% y cuenta con una de las comunidades más grandes de Internet, lo cual facilita encontrar ayuda, documentación, y otros recursos relacionados.

2.6.4 Java y tecnologías asociadas.

Java es una plataforma virtual de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales.

J2EE son las siglas de Java 2 Enterprise Edition que es la edición empresarial del paquete Java, comprende un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales. J2EE aprovecha muchas de las

características de la plataforma Java, como la portabilidad "Write Once, Run Anywhere", el Application Program Interface (API) JDBC para el acceso a bases de datos, la tecnología CORBA para la interacción con los recursos existentes de la empresa y un modelo de seguridad que protege los datos incluso en las aplicaciones para Internet. Sobre esta base, Java 2 Enterprise Edition añade el soporte completo para componentes Enterprise Java Beans, el API Java Servlets y la tecnología JavaServer Pages.

Los **Servlets** son módulos que extienden los servidores orientados a petición-respuesta, como los servidores Web compatibles con Java. Por ejemplo, un servlet podría ser responsable de tomar los datos de un formulario de entrada de pedidos en HTML y aplicarle la lógica de negocios utilizada para actualizar la base de datos de pedidos de la compañía. Los Servlets son un reemplazo efectivo para los scripts CGI. Proporcionan una forma de generar documentos dinámicos que son fáciles de escribir y rápidos en ejecutarse. Los Servlets también solucionan el problema de hacer la programación del lado del servidor con APIs específicos de la plataforma: están desarrollados con el API Java Servlet, una extensión estándar de Java.

La tecnología **Java Server Pages** (JSP) permite a los diseñadores y desarrolladores de sitios Web crear rápidamente y mantener de manera sencilla las páginas Web dinámicas, ricas en información sobre las que se basan los sistemas de negocio. Como parte de la familia Java, la tecnología JSP posibilita el desarrollo de aplicaciones basadas en Web que son independientes de la plataforma usada. La tecnología JSP separa la interfaz de usuario de la generación de contenidos, permitiendo a los diseñadores cambiar el formato de la página sin alterar el contenido dinámico subyacente.

Hibernate es un mapeador objeto-relacional u ORM (por las siglas en inglés de object/relational mapping). Trabajar con software orientado a objetos y bases de datos relacionales puede incrementar el tiempo de desarrollo debido a la diferencia entre el modelo relacional y el paradigma orientado a objetos. Un ORM se encarga de realizar la

transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa. Hibernate no solo realiza esta transformación sino que proporciona potentes capacidades para la obtención y almacenamiento de datos de la base de datos. Hibernate es el mapeador objeto-relacional de código abierto más maduro y más completo que existe actualmente. Se utiliza muy ampliamente, se desarrolla activamente y soporta una extensa comunidad de Open Source, además tiene la ventaja de que es totalmente transparente el uso de la base de datos pudiendo cambiar de base de datos simplemente cambiando los ficheros de configuración, sin necesidad de cambiar una línea de código de la aplicación.

2.7 Sistemas Gestores de Bases de Datos.

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. En la actualidad existe una gran variedad de SGBD, tanto de tipo comercial como libre. A continuación se exponen los más conocidos.

2.7.1 Microsoft SQL Server.

Es un sistema de gestión de bases de datos relacionales desarrollado por Microsoft. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET. El SQL Server permite lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones, pero presenta el inconveniente de no ser multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft [14].

2.7.2 Oracle.

Es considerado el SGBD más completo que existe. Sus características más destacadas son el soporte de transacciones, su gran estabilidad y seguridad, su escalabilidad, así

como que es un sistema multiplataforma, entre otras ventajas. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de otros SGBD. Su mayor defecto es su enorme precio y también ha sido criticado por algunos especialistas por la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, que incrementan el nivel de exposición de los usuarios [11].

2.7.3 MySQL.

Es uno de los Sistemas Gestores de bases de Datos más populares desarrollados bajo la filosofía de código abierto. MySQL tiene como una de sus principales ventajas la velocidad en la lectura de datos, pero a costa de eliminar un conjunto de facilidades que presentan otros SGBD: integridad referencial, bloqueo de registros, procedimientos almacenados, entre otros. En recientes versiones de MySQL se incluyen algunas de estas características, pero indudablemente esto va en detrimento de la velocidad [10].

2.7.4 PostgreSQL.

Está considerado el SGBD de código abierto más avanzado del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales de alto calibre tales como Oracle.

Es un SGBD objeto-relacional, ya que aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas, el control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y valores no atómicos (atributos basados en vectores y conjuntos).

Es altamente extensible: soporta operadores y tipos de datos definidos por el usuario. Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92. Cuenta con una API (del inglés Application Program Interface)

flexible lo cual ha permitido dar soporte para el desarrollo con PostgreSQL en diversos lenguajes de programación entre los que se incluyen: Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike. Tiene soporte para lenguajes procedurales internos, incluido un lenguaje nativo denominado PL/pgSQL, el cual es comparable con el lenguaje procedural de Oracle PL/SQL. Presenta como ventaja frente a MySQL (que tiene sus restricciones en las licencias), que es totalmente libre [12].

2.8 Metodologías de Desarrollo de Software.

El desarrollo vertiginoso que desde hace algunos años ha venido experimentado la industria de la informática, hace que los sistemas informáticos sean cada vez más complejos y que los usuarios exijan por la calidad de los mismos. Por otro lado, el alto nivel de competitividad existente en este mercado hace que pequeños retrasos en la entrega de un sistema pudiera implicar la pérdida de gran cantidad de usuarios finales e importantes clientes. “El problema del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las múltiples cadenas de trabajo de un gran proyecto de software” y “a menos que renovemos nuestros métodos, no podremos cumplir con el objetivo de desarrollar el software complejo que se necesita actualmente” [3].

2.8.1 Algunas Metodologías de Desarrollo de Software.

Durante los últimos años se han desarrollado dos corrientes en lo referente a las metodologías de desarrollo de software, las llamadas “pesadas” y las llamadas “ligeras o ágiles”. Las primeras se basan en la idea de conseguir el objetivo común por medio de orden y documentación, mientras que las segundas tratan de lograrlo por medio de la comunicación directa e inmediata entre aquellos que intervienen en el proceso.

La **Programación Extrema** o eXtreme Programming (XP) es una aproximación a la ingeniería de software formulada por Kent Beck. Se trata de un proceso ágil de desarrollo de software. La base para el desarrollo del software que usa esta

metodología son las llamadas User Stories, historias escritas por el cliente en las que describe escenarios sobre el funcionamiento del sistema y que no sólo están limitados a la interfaz de usuario, sino que también pueden describir modelos, dominio, etc. Estas User Stories junto a la arquitectura que se persigue, sirve de base para crear un plan de “entregas de software” entre el equipo de desarrollo y el cliente, para cada una las cuales se definen objetivos y las iteraciones (generalmente cortas) necesarias para cumplirlos. Las User Stories y los casos de pruebas son la base sobre la que se asienta el trabajo del desarrollador.

Desarrollo Guiado por la Funcionalidad (FDD por las en inglés de Feature Driven Development) es más bien una metodología ligera. Está pensada para proyectos con un tiempo de desarrollo relativamente corto (menos de un año). Se basa en un proceso iterativo con iteraciones cortas de aproximadamente dos semanas que producen un software funcional, el cual puede ser examinado por el cliente y la dirección de la empresa. Cada iteración se define en término de funcionalidades (de ahí su nombre) que son pequeñas partes del sistema con significado para el cliente.

RUP son las siglas en inglés de Rational Unified Process. Esta es una de las metodologías más generales de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto, no sólo de software. Es el producto final de treinta años de desarrollo y uso práctico. Se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal. En el siguiente epígrafe se realiza un estudio más detallado de esta metodología.

2.8.2 El Proceso Unificado de Racional.

El Proceso Unificado de Rational o RUP es una metodología que apareció en 1998, creada por James Rumbaugh, Grady Booch e Ivar Jacobson para la Rational Corporation. Según sus autores: “Un proceso de desarrollo de software es el conjunto

de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas” y “utiliza el *Lenguaje Unificado de Modelado* (Unified Modeling Language, UML) para preparar todos esquemas de un sistema software, de hecho, UML es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos” [4].

2.8.2.1 Características de RUP.

Los aspectos definitorios de RUP que lo convierten en único se resumen en tres frases claves: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

- Dirigido por los casos de uso:

Teniendo en cuenta que la razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado usuario y los utiliza tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígase diseño, implementación y prueba.

- Estar centrado en la arquitectura:

La arquitectura es una vista del diseño completo con las características más importantes, dejando a un lado los detalles. Esta no solo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red; con los que debe coexistir el sistema. En otras palabras, la arquitectura representa la forma del sistema,

la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

- Ser iterativo e incremental:

La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. A cada uno de estos mini-proyecto se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

2.8.2.2 La vida de un producto de Software según RUP.

El Proceso Unificado establece que desde el nacimiento de un sistema de software hasta su muerte, se suceden en el tiempo un conjunto de ciclos, cada uno de los cuales concluyen con la entrega de una versión del sistema. Cada ciclo se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada una de las fases, que según la complejidad del sistema se divide en iteraciones, termina con un hito, en el cual los directivos, basados en el conjunto de artefactos obtenidos, toman una serie de decisiones para la continuación o no en la siguiente fase (ver Figura 2.5).

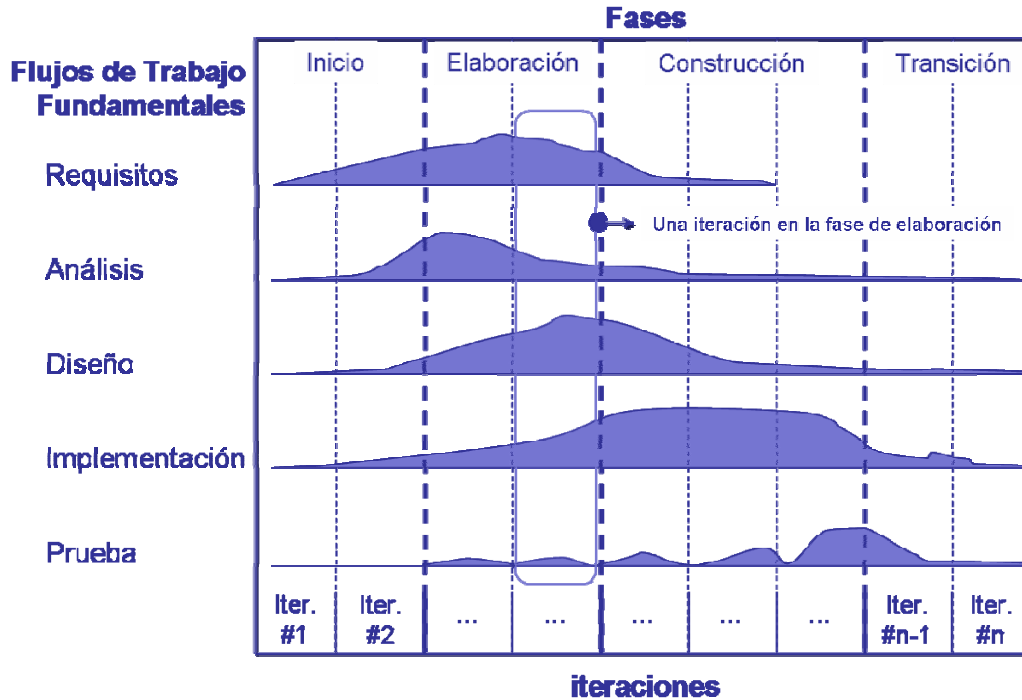


Fig. 2. 3 Fases y flujo de trabajo de RUP.

2.8.3 El Lenguaje Unificado de Modelado.

El Lenguaje Unificado de Modelado (UML) es un lenguaje estándar de modelado para software. Es un lenguaje para la visualización, especificación, construcción y documentación de los artefactos de sistemas en los que el software juega un papel importante. Básicamente, UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados.

UML (por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. EL UML ofrece un estándar para escribir un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. El UML cuenta

con varios tipos de modelos, los cuales muestran diferentes aspectos de las entidades representadas y se ha convertido rápidamente en el estándar de facto para construir software orientado a objetos [15].

2.9 Software Libre.

Se entiende por software libre todo aquel software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

El Software Libre es un tipo particular de software que le permite al usuario el ejercicio de cuatro libertades básicas:

- Ejecutarlo con cualquier propósito
- Estudiar como funciona y adaptarlo a sus necesidades
- Distribuir copias
- Mejorarlos, y liberar esas mejoras al público.

Con la única restricción del copyleft (o sea, cualquiera que redistribuya el software, con o sin cambios, debe dar las mismas libertades que antes), y con el requisito de permitir el acceso al código fuente (imprescindible para ejercer las libertades 1 y 3).

En la actualidad existe una creciente tendencia hacia la utilización de software libre. Cada vez se promociona más la migración desde los sistemas con licencia comercial que están en poder de unos pocos monopolios de la rama de la informática que se reservan el derecho de vender sus productos a quienes consideren pertinente, hacia aquellos que se denominan “libres”.

En la Universidad de las Ciencias Informáticas, y en nuestro país en general, se ha comenzado a seguir una política de migración hacia el software libre y partiendo de esta

premisa, la presente investigación propone hacer uso de herramientas y tecnologías libres para la elaboración del sistema propuesto.

2.10 Justificación de la propuesta de solución.

Por todo lo anteriormente expuesto en el capítulo, para la elaboración de la propuesta de solución que plantea esta investigación se desarrollará una aplicación Web, lo que permite que el sistema pueda ser utilizado desde distintos lugares y sin más requerimientos que una computadora con navegador Web y conexión a la red. Del lado del servidor la programación se realizará en Java, específicamente J2EE, usando JSP y Servlets. El Sistema Gestor de Bases de Datos a emplear será Postgresql. La decisión del lenguaje de programación del servidor y el SGBD está en concordancia con las políticas de migración hacia software libre de la UCI, por las ventajas y prestaciones que brindan ambos para la construcción de la propuesta y además por requerimiento del usuario. Se usará Javascript del lado del cliente para lograr la interactividad con el usuario en el navegador y específicamente la técnica Ajax, ya que la misma permite actualizar parte de una página en cualquier momento, dándole a los usuarios una respuesta instantánea a sus ingresos y consultas. Como metodología de desarrollo de software se utilizará RUP.

2.11 Conclusiones.

En este capítulo se hizo un estudio de las tecnologías a utilizar en el desarrollo de la propuesta de solución así como algunos conceptos y tendencias que se deben tener en cuenta. Se fundamentó la elección del lenguaje de programación, el sistema gestor de bases de datos, y la metodología de desarrollo de software; así como el uso de otras técnicas y tecnologías.

3.1 Introducción.

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

3.2. Modelo del negocio propuesto.

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

3.2.1 Actores / Trabajadores del negocio

Los actores del negocio son aquellas personas o sistemas que obtienen un resultado de valor de uno o varios procesos del negocio y los trabajadores del negocio son aquellas personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos, pero no obtienen ningún resultado de valor. La Figura 3.1 muestra la representación UML.

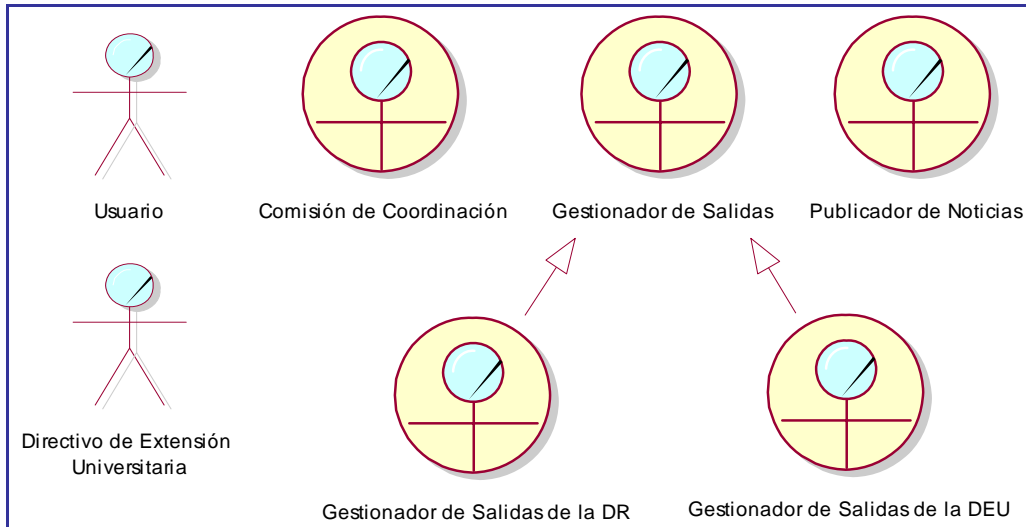


Fig. 3. 1 Actores / Trabajadores del Negocio

En la siguiente tabla se definen los actores y trabajadores del negocio propuesto:

Actor	Justificación
Usuario	Estudiantes y Profesores internos interesados en utilizar el servicio de reservaciones.
Directivo de Extensión Universitaria	Son los directivos de la VREU (Vicerrector, Directores, Jefes de Departamento, etc.), interesados en que el proceso de reservaciones se desarrolle con calidad. Rigen y supervisan dicho proceso.
Trabajador	Justificación
Gestor de Reservaciones	Trabajador de una de las direcciones de la VREU que interviene en el proceso de reservaciones. Es el encargado de llevar a cabo casi la totalidad de los procesos del negocio
Gestor de Reservaciones de la Dirección de Recreación	Se encarga del Proceso de Reservaciones en la DR.
Gestor de Reservaciones de la Dirección de Extensión Universitaria	Se encarga del Proceso de Reservaciones en la DEU.
Comisión de Coordinación	Son los representantes de las Direcciones y Organizaciones de Masas de la universidad, que participan en las Reuniones

	de Coordinación convocadas por la Dirección de Transporte donde se discute y aprueba el Plan de Salidas semanal.
Publicador de Noticias	Tiene la tarea de publicar los avisos sobre las ofertas de salidas para conocimiento de los usuarios.

Tabla 3. 1 Actores y trabajadores del negocio propuesto.

3.2.2 Reglas del Negocio

- El servicio de reservaciones está dirigido a los estudiantes y profesores internos.
- Solo en casos excepcionales y definido con antelación se les permite a otros miembros de la comunidad universitaria usar el servicio de reservaciones.
- Antes de hacer pública una oferta de salida esta debe ser aprobada en primera instancia por un directivo de la Vicerrectoría y posteriormente en la Reunión de Coordinación.
- Los trabajadores solo pueden intervenir en el proceso de reservaciones relacionado con el área a la que pertenecen.

3.2.3 Descripción de los procesos del negocio propuesto.

La Figura 3.2 muestra el Diagrama de Casos de Uso del Negocio Propuesto.

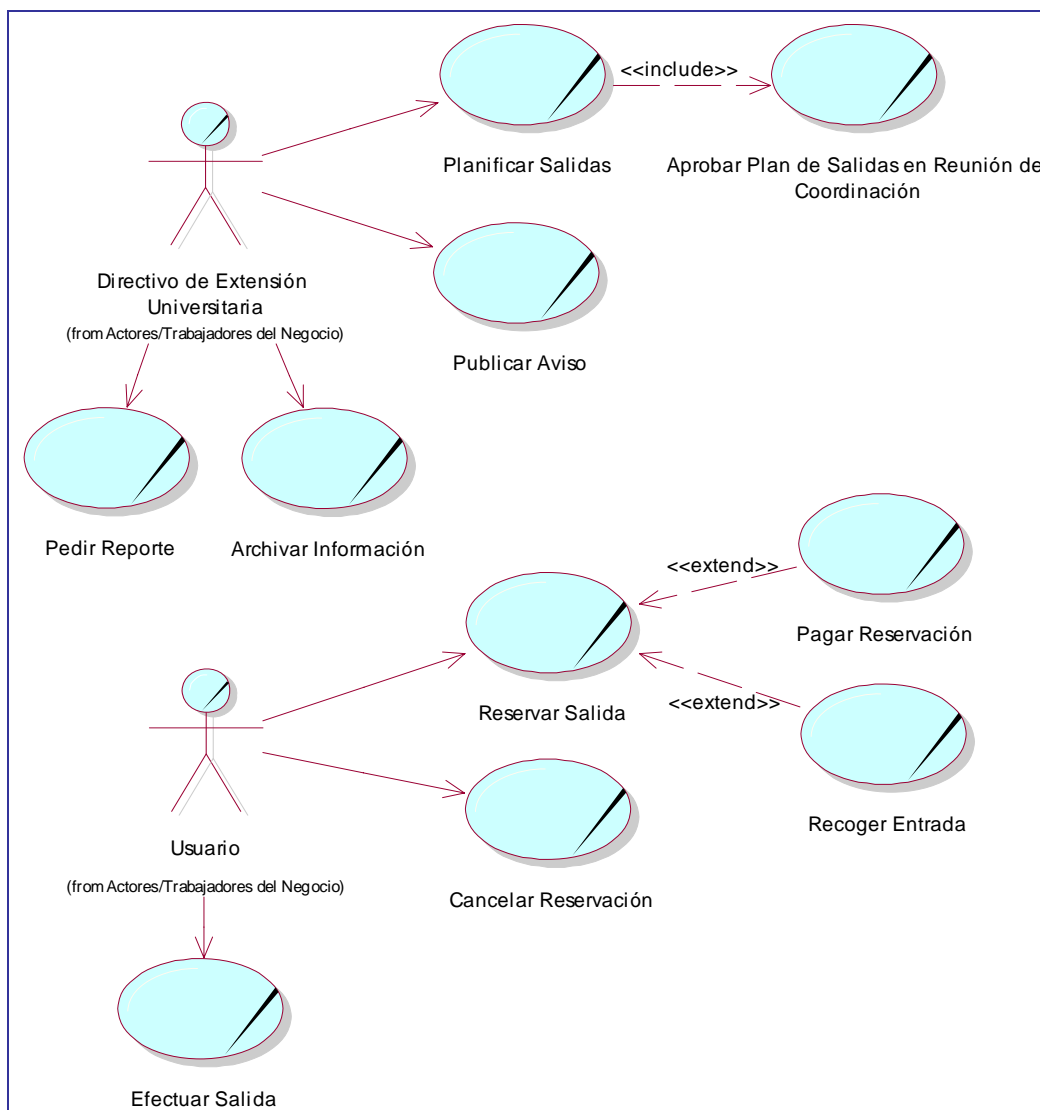


Fig. 3. 2 Diagrama de Casos de Uso del Negocio

A continuación se describen los procesos del negocio propuesto, en el Anexo 1. Modelo del Negocio Propuesto. pueden consultarse los diagramas de Actividad y modelos de Objetos del Negocio correspondientes.

CUN Planificar Salidas: El caso de uso inicia y es infinito. El Directivo de Extensión Universitaria encarga al Gestor de Reservas el Plan de Salidas de la semana, y a partir de ahí este se encarga semanalmente de elaborar una propuesta, la pone a consideración del Directivo de Extensión Universitaria, el cual la evalúa y aprueba, en caso de rechazarla, el Gestor de Reservas debe modificar la

propuesta hasta que sea aprobada por el Directivo de Extensión Universitaria. Este proceso incluye el CU Aprobar Plan de Salidas en Reunión de Coordinación (Fig. 6. 1 y Fig. 6. 2).

CUN Aprobar Plan de Salidas en Reunión de Coordinación: Inicia cuando el Gestor de Reservas presenta la propuesta del Plan de Salidas en la Reunión de Coordinación. En esta reunión se discute y analiza la propuesta presentada hasta que se aprueba el Plan de Salidas de la semana y se le entrega al Gestor de Reservas el cual crea la Lista de Usuarios Reservados para cada oferta que contiene el plan (Fig. 6. 3 y Fig. 6. 4).

CUN Publicar Aviso: El proceso es iniciado cuando el Directivo de Extensión Universitaria le encarga al Gestor de Reservas la redacción de un Aviso para poner en conocimiento de los Usuarios las ofertas de salidas de la semana. El Gestor de Reservas envía el Aviso al Publicador de Noticias. Este publica el Aviso y le remite la confirmación de Aviso publicado al Gestor de Reservas (Fig. 6. 5 y Fig. 6. 6).

CUN Reservar Salida: Se inicia cuando el Usuario verifica la publicación de los avisos de salidas y decide reservar para una de las ofertas existentes. Se presenta ante el Gestor de Reservas para solicitar su reservación, el cual verifica que queden capacidades para dicha salida; en caso negativo le informa al Usuario que no es posible reservarle y el caso de uso termina. De existir capacidades, lo adiciona en la Lista de Usuarios Reservados y verifica si la salida requiere pago. Si la salida es libre de pago le informa al Usuario de su reservación y el caso de uso termina. En caso contrario va al caso de uso extendido Pagar Reservación (Fig. 6. 7 y Fig. 6. 8).

CUN Pagar Reservación: El proceso inicia cuando el Gestor de Reservas pide al Usuario abonar el pago por una reservación que así lo requiere. El usuario abona la cantidad necesaria y el Gestor de Reservas anota en la Lista de

Usuarios Reservados la constancia del pago, la cual firma el Usuario y el caso de uso termina (Fig. 6. 9 y Fig. 6. 10).

CUN Recoger Entrada: Inicia cuando un Usuario pide recoger la entrada para una reservación. El Gestor de Reservas verifica que el Usuario está reservado y que ha pagado la entrada si la salida incluye pago (de no ser así, deniega la solicitud), el Usuario recibe su entrada y el caso de uso termina (Fig. 6. 11 y Fig. 6. 12).

CUN Cancelar Reservación: El caso de uso se inicia cuando el Usuario se presenta ante el trabajador Gestor de Reservas con el propósito de cancelar una reservación. El Gestor de Reservas busca al usuario, previamente identificado, en la Lista de Usuarios Reservados y lo elimina terminando el caso de uso. Asimismo si el Usuario no aparece como reservado, el Gestor de Reservas le informa y termina el caso de uso (Fig. 6. 13 y Fig. 6. 14).

CUN Efectuar Salida: Es iniciado cuando el Usuario va hacia el lugar de transportación y solicita al trabajador Gestor de Reservas efectuar la salida. Este ultimo busca al Usuario, previamente identificado, en la Lista de Usuarios Reservados. En caso de que se encuentre, le informa al usuario que puede utilizar el servicio, confirma en la lista de usuarios reservados la asistencia y el caso de uso termina. (Fig. 6. 15 y Fig. 6. 16).

CUN Archivar Información: El proceso es iniciado cuando el Directivo de Extensión Universitaria orienta archivar la documentación generada durante el proceso de reservaciones. El Gestor de Reservas comprueba si no quedan ofertas de salida en la semana (esto ocurre hasta que se cumpla dicha condición) y de ser así archiva el Plan de Salidas cumplido con sus correspondientes Listas de Usuarios Reservados. Este proceso se realiza cada semana. (Fig. 6. 17 y Fig. 6. 18).

CUN Pedir Reporte: Un Directivo de Extensión Universitaria inicia el caso de uso cuando pide al Gestor de Reservas un reporte sobre el proceso de

reservaciones. Dicho Reporte puede contener información estadística relacionada con los Usuarios o sobre el cumplimiento del Plan de Salidas. El Gestor de Reservaciones elabora el Reporte, se lo entrega al Directivo de Extensión Universitaria para que este lo analice y termina el caso de uso (Fig. 6. 19 y Fig. 6. 20).

3.3 Especificación de los requerimientos de software.

En este epígrafe se relaciona la lista de requerimientos funcionales y no funcionales del sistema a desarrollar.

3.3.1 Requerimientos funcionales.

R1. Autenticar Usuario

- 1.1. Introducir nombre de usuario y contraseña de dominio UCI.
- 1.2. Validar datos introducidos.
- 1.3. Mostrar al usuario las opciones a las que tiene acceso según el rol o permisos que posee.

R2. Mostrar las ofertas de salidas existentes.

- 2.1. Agrupar las salidas por tipo (Salidas de Paseo, Visitas Dirigidas y Salidas Culturales).
- 2.2. Mostrar de cada salida: Nombre, Capacidades, Fecha, Hora, Precio (si incluye pago), Características y Detalles específicos.
- 2.3. Mostrar el estado actual de las reservaciones (en espera del inicio de las reservaciones, capacidades disponibles, capacidades agotadas).
 - 2.3.1. Mostrar la opción de reservar si existen capacidades disponibles y sólo en caso de que el usuario cumpla con las características de la salida.
 - 2.3.2. Mostrar la opción de cancelar una reservación en caso de que el usuario haya reservado y si está dentro del tiempo establecido para la cancelación de las reservaciones.

R3. Reservar para una salida.

- 3.1. Mostrar todos los datos de la salida en cuestión.

3.2. Seleccionar la opción de reservar acompañantes (opcional y siempre que esté permitido para la salida en cuestión).

3.2.1. Adicionar un acompañante introduciendo su nombre de usuario del dominio UCI.

3.2.2. Eliminar un acompañante.

3.3. Mostrar (si la salida incluye pago) la cantidad total a pagar (sumando la de los acompañantes si se han incluido) y la fecha, hora y lugar en que se efectuará el pago.

3.4. Efectuar la reservación.

3.4.1. Mostrar mensaje de confirmación de la reservación.

3.4.2. Enviar un mensaje de correo electrónico informando a los acompañantes (si se han incluido acompañantes en la reservación).

R4. Cancelar una reservación.

4.1. Mostrar los datos de la salida que se va a cancelar.

4.2. Efectuar la cancelación.

4.2.1. Mostrar un mensaje de confirmación de la cancelación.

4.2.2. Enviar un mensaje de correo electrónico informando a los acompañantes (si se han incluido acompañantes en la reservación).

R5. Mostrar avisos publicados.

5.1. Mostrar una lista con los títulos de avisos sobre cancelación, modificaciones en las salidas o reservaciones u otras informaciones importantes relacionadas con el servicio de reservaciones.

5.1.1. Mostrar el texto completo de un aviso seleccionado.

R6. Mostrar Perfil.

6.1. Mostrar las reservaciones que ha efectuado el usuario.

6.2. Mostrar las penalizaciones impuestas y la fecha de fin de las mismas.

R7. Gestionar Salidas

7.1. Mostrar las salidas publicadas (permitiendo selección).

7.1.1. Mostrar datos de una salida

7.1.1.1. Modificar los datos de la salida.

7.1.1.1.1. Mostrar un mensaje de confirmación.

7.1.1.1.2. Mostrar la opción de publicar un aviso.

7.1.1.1.3. Enviar correo electrónico a los reservados.

7.1.1.2. Mostrar la lista de usuarios reservados.

7.1.1.2.1. Imprimir la lista de usuarios reservados

7.1.1.3. Registrar la Inasistencia de los usuarios a una salida

7.1.1.4. Confirmar el pago de los usuarios.

7.1.1.5. Archivar los datos de la salida.

7.1.1.5.1. Verificar si se efectuó la salida (comprobando la confirmación de asistencia de los usuarios)

7.1.1.5.1.1. En caso negativo informar y no realizar ninguna acción.

7.1.2. Eliminar una salida.

7.1.2.1. Mostrar un mensaje de confirmación.

7.1.2.2. Mostrar la opción de publicar un aviso.

7.1.2.3. Enviar correo electrónico a los reservados.

7.2. Publicar una nueva salida

7.2.1. Introducir todos los datos de la salida (Tipo, Nombre, Capacidades, Fecha, Hora, Precio (si incluye pago), Características, Detalles específicos, etc.).

R8. Gestionar Avisos

8.1. Mostrar los avisos publicados (permitiendo selección)

8.1.1. Modificar texto un aviso.

8.1.2. Eliminar un aviso.

8.2. Publicar un nuevo aviso.

8.2.1. Introducir datos del aviso (título y texto del aviso).

R9. Gestionar Usuarios

9.1. Buscar un usuario del dominio UCI.

9.1.1. Mostrar datos del usuario.

9.1.2. Mostrar las reservaciones (si ha efectuado alguna)

9.1.2.1. Cancelar una reservación hecha por el usuario (para reservaciones relacionadas con el área del Gestor de Reservaciones).

9.1.2.2. Enviar correo electrónico informando al usuario.

9.1.3. Mostrar las penalizaciones (si sufre alguna).

- 9.1.3.1. Despenalizar al usuario (si la penalización está relacionada con el área a la cual pertenece el Gestor de Reservas).
- 9.1.3.2. Enviar correo electrónico informando al usuario.
- 9.1.4. Penalizar al usuario.
 - 9.1.4.1. Seleccionar el tipo de penalización a aplicar.
 - 9.1.4.2. Introducir la causa de la penalización.
 - 9.1.4.3. Enviar correo electrónico informando al usuario.
- 9.1.5. Reservar al usuario para una salida.
 - 9.1.5.1. Introducir el nombre de usuario del dominio UCI.
 - 9.1.5.2. Seleccionar de una lista la salida a reservar (si existen capacidades y solo para las ofertas de salidas publicadas de la misma área a la que pertenece el Gestor de Reservas).
 - 9.1.5.3. Enviar correo electrónico informando al usuario.

R10. Gestionar Reportes

- 10.1. Mostrar una lista con los tipos de reportes disponibles (permitiendo selección).
- 10.2. Seleccionar un tipo de reporte.
 - 10.2.1. Marcar las opciones para la elaboración del reporte (rango de tiempo, tipos de salida a tener en cuenta, tipo de usuarios, entre otras más).
 - 10.2.2. Mostrar el reporte solicitado
 - 10.2.3. Imprimir el reporte.

R11. Administrar áreas.

- 11.1. Mostrar una lista de las áreas registradas en el sistema (permitiendo selección).
- 11.2. Añadir un área.
 - 11.2.1. Introducir el nombre de la nueva área.
- 11.3. Eliminar un área.
 - 11.3.1. Verificar si hay personas que pertenezcan a dicha área.
 - 11.3.2. En caso afirmativo informar al administrador y no realizar ninguna acción.
- 11.4. Modificar un área (nombre)

R12. Administrar personas.

12.1. Mostrar una lista de las personas registradas en el sistema (permitiendo selección).

12.2. Añadir una persona.

12.2.1. Introducir el nombre de usuario (de dominio UCI) de la persona que se desea registrar.

12.2.2. Seleccionar de una lista el área a la que pertenece.

12.2.3. Seleccionar de una lista el rol que juega en el sistema.

12.3. Eliminar una persona.

12.4. Modificar una persona (área a la que pertenece, rol que juega).

R13. Administrar permisos.

13.1. Mostrar una lista de los permisos con que cuenta el sistema (permitiendo selección).

13.2. Añadir un permiso.

13.2.1. Introducir un nombre para el nuevo permiso.

13.2.2. Seleccionar de una lista las acciones que comprende el nuevo permiso.

13.3. Eliminar un permiso.

13.3.1. Verificar que ningún rol tenga asignado ese permiso.

13.3.2. En caso afirmativo informar al administrador y no realizar ninguna acción.

13.4. Modificar un permiso (nombre, acciones que comprende)

R14. Administrar roles.

14.1. Mostrar una lista de los roles del sistema (permitiendo selección).

14.2. Crear un rol.

14.2.1. Introducir un nombre para el nuevo rol.

14.2.2. Seleccionar de una lista los permisos que posee.

14.3. Eliminar un rol.

14.3.1. Verificar que ninguna persona tenga asignado ese rol.

14.3.2. En caso positivo informar al administrador y no realizar ninguna acción.

14.4. Modificar un rol (nombre, permisos que posee).

R15. Administrar penalizaciones.

15.1. Mostrar una lista de las penalizaciones con que cuenta el sistema (permitiendo selección).

15.2. Añadir nueva penalización.

15.2.1. Introducir un nombre para la nueva penalización.

15.2.2. Elegir el criterio de penalización en caso de que esta sea automática.

15.2.3. Introducir tiempo de duración de la penalización.

15.3. Eliminar penalización.

15.4. Modificar penalización (nombre, criterio y tiempo de duración).

3.3.2 Requerimientos no funcionales.

Apariencia o interfaz externa:

- Diseño sencillo, permitiendo la utilización del sistema sin mucho entrenamiento.
- Diseñado para la resolución 800x600, pero preparado para verse en otras resoluciones.

Soporte:

- Garantía de instalación y prueba del sistema, además de un breve entrenamiento a los futuros usuarios.

Implementación:

- Usar J2EE como plataforma de desarrollo.
- Utilizar PostgreSQL como Sistema Gestor de Bases de Datos.

Portabilidad:

- Independencia de la plataforma.

Seguridad:

- Existencia de distintos roles que establezcan las acciones que pueden realizar los usuarios.
- Verificación sobre acciones irreversibles (por ejemplo las eliminaciones).

Confiabilidad:

- Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

3.4 Modelo del sistema

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

3.4.1 Actores del sistema.

Los actores de un sistema son agentes externos, es decir, aquellas personas o sistemas que interactúan con él. En la Figura 3.3 se puede observar la representación UML de los actores del sistema propuesto.

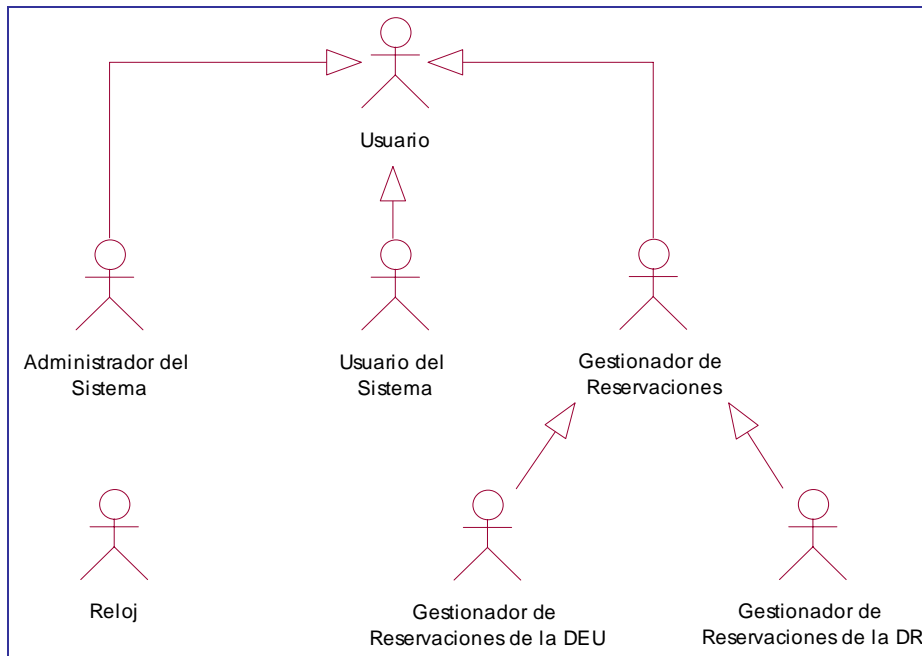


Fig. 3. 3 Actores del Sistema

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

Actor	Justificación
-------	---------------

Usuario	Generaliza a todos los usuarios del sistema. Realiza las operaciones comunes a ellos.
Administrador del Sistema	Es el encargado de definir las configuraciones con las que trabaja el sistema.
Usuario del Sistema	Son los estudiantes y profesores internos interesados en usar el sistema de reservaciones.
Gestionador de Reservaciones	Trabajador de la VREU que se encarga de realizar las actividades de gestión que componen el proceso de reservaciones.
Gestionador de Reservaciones de la DEU	Se ocupa del proceso de reservaciones en la DEU.
Gestionador de Reservaciones de la DR	Se ocupa del proceso de reservaciones en la DR.
Reloj	Actor ficticio para indicar un temporizador, el cual ejecuta ciertas tareas programadas del sistema.

Tabla 3. 2 Actores del sistema.

3.4.2 Modelo de casos de uso del Sistema.

Los casos de uso son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación. Los casos de uso del sistema se han organizado en paquetes según su área de aplicación (Ver Figura 3.4). En el Anexo 2. Modelo del Sistema. se muestra el diagrama de casos de uso de cada uno de los tres subsistemas, los cuales son:

- *Reservaciones* (Fig. 6. 21 Diagrama de CUS. Subsistema Reservaciones.).
- *Gestión de Reservaciones*, que se divide a su vez en *Gestión de Salidas* (Fig. 6. 22 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Salidas.), *Gestión de Avisos* (Fig. 6. 23 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Avisos.), *Gestión de Usuarios* (Fig. 6. 24 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Usuarios.) y *Gestión de Reportes* (Fig. 6. 25 Diagrama de CUS. Subsistema Gestión de Reservaciones / Gestión de Reportes.).
- *Administración* (Fig. 6. 26 Diagrama de CUS. Subsistema Administración.).

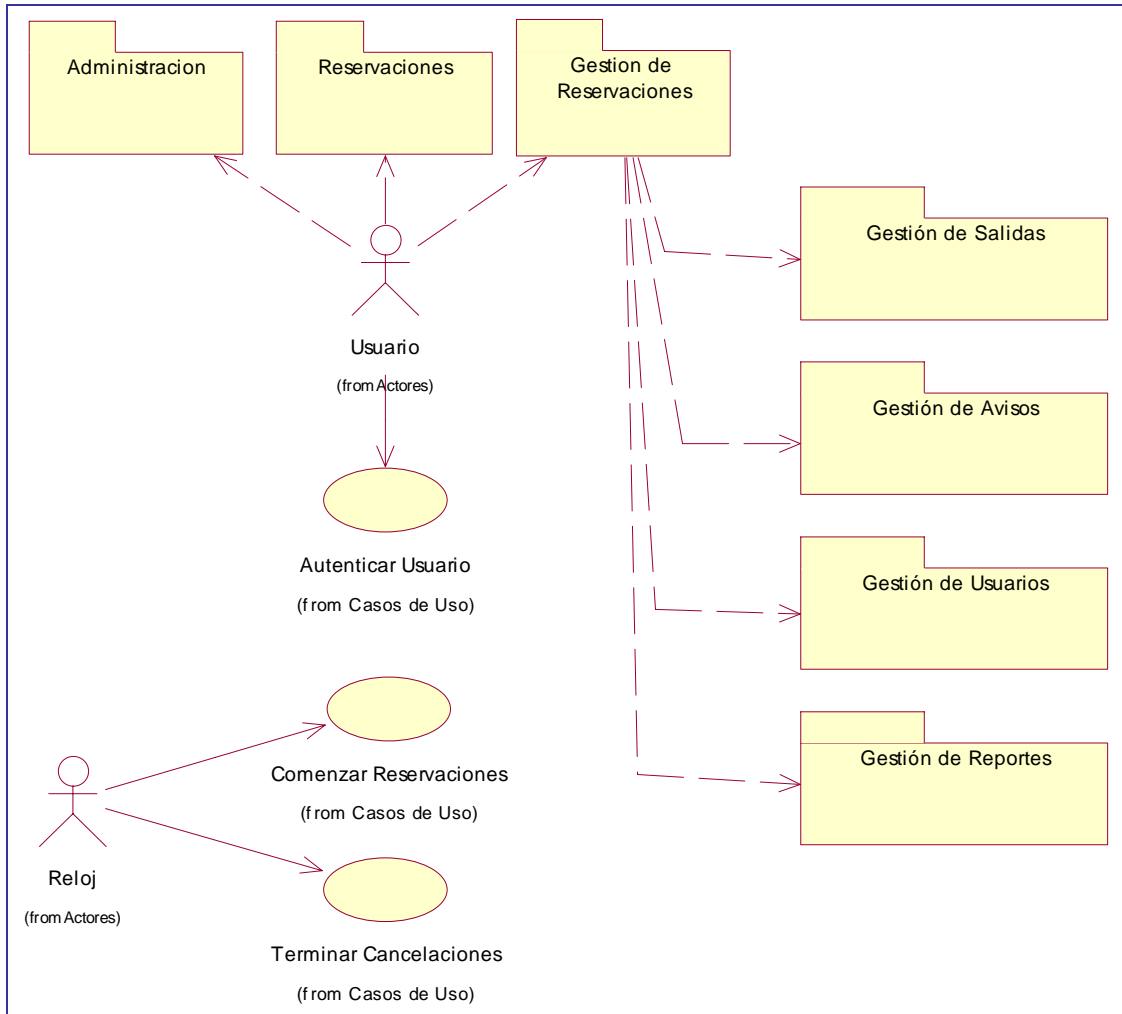


Fig. 3. 4 Diagrama de Casos de Uso del Sistema

3.4.3 Descripción de los casos de uso del sistema.

A continuación se describen en detalle los casos de uso del sistema más importantes.

CUS1	Autenticar Usuario	
Propósito	Conocer en todo momento quién accede al sistema.	
Actores	Usuario	
Resumen	Inicia cuando el usuario ingresa sus datos en el sistema (Nombre de usuario y contraseña del dominio UCI). Se le informa si los datos introducidos son correctos o no y termina el caso de uso.	
Referencias	R1	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1. Muestra los controles que permiten al usuario introducir su nombre de usuario y contraseña de	

	dominio UCI.
2. Introduce los datos y acepta la operación.	3. Valida la identidad del usuario y le informa que está autenticado. En caso de que los datos no sean correctos, ver FA1.
Flujos alternativos	
FA1. Los datos introducidos son incorrectos.	El sistema muestra un mensaje de error y regresa al paso 1.

Tabla 3. 3 Descripción del CUS Autenticar Usuario.

CUS2	Comenzar Reservas	
Propósito	Dar la posibilidad a los usuarios del sistema de reservar para una salida	
Actores	Reloj	
Resumen	El caso de uso inicia cuando llega la fecha y hora de comenzar las reservas para una oferta de salida. Se muestra la opción de reservar a los usuarios y termina el caso de uso.	
Referencias	R2.3.1	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1. Chequea si ha llegado a la fecha y hora de comenzar las reservas y habilita el botón Reservar para la salida.	

Tabla 3. 4 Descripción del CUS Comenzar Reservas.

CUS3	Terminar Cancelaciones	
Propósito	Poner un límite de tiempo para que los usuarios cancelen la reserva para una salida.	
Actores	Reloj	
Resumen	El caso de uso inicia cuando llega la fecha y hora de tope para realizar cancelaciones para una reserva. Se inhabilita la opción de cancelar reserva a los usuarios que han reservado y termina el caso de uso.	
Referencias	R2.3.2	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1. Chequea si ha llegado a la fecha y hora tope para cancelar la reserva e inhabilita el botón Cancelar Reserva para la salida reservada.	

Tabla 3. 5 Descripción del CUS Finalizar Cancelaciones.

3.4.3.1 Reservas.

CUS4	Explorar Salidas	
Propósito	Mostrar las ofertas de salidas disponibles.	
Actores	Usuario del Sistema	
Resumen	El caso de uso se inicia cuando el usuario entra al sistema y se muestra una lista con las salidas disponibles, sus características y la opción de reservar o cancelar la reserva en dependencia del estado de las reservas para la salida, los permisos que tenga el usuario y las restricciones de cada salida. El caso de uso no termina hasta que el usuario no lo especifique explícitamente.	

Referencias	R2
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	1. Muestra las ofertas de salidas disponibles organizadas por categorías. Para cada salida debe mostrarse el estado de la misma y la opción de reservar (Ver CUS5) o de cancelar la reservación (Ver CUS6) siempre y cuando el usuario autenticado pueda realizar estas acciones teniendo en cuenta el estado de las reservaciones y las restricciones de la salida. El usuario también tiene la opción de ver detalles de una salida (ver sección: "Ver Detalles de una Salida").
Sección "Ver Detalles de una Salida"	
Acción del Actor	Respuesta del sistema
1. Selecciona la opción de ver los detalles de una salida.	2. Muestra los detalles de la salida seleccionada y se da la opción de ocultar los detalles.
3. Selecciona ocultar los detalles.	4. Oculta los detalles de la salida.
Casos de Uso asociados	
CUS5 Reservar Salida (extend).	
CUS6 Cancelar Reservación (extend).	

Tabla 3. 6 Descripción del CUS Explorar Salidas Disponibles.

CUS5	Reservar Salida
Propósito	Reservar para una salida.
Actores	Usuario del Sistema
Resumen	El caso de uso se inicia cuando un usuario decide hacer una reservación para una salida. Selecciona las opciones disponibles para la salida, escoge la opción Reservar, se le confirma del éxito de la reservación y termina el caso de uso.
Referencias	R3, CUS4
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Elige la opción de reservar para una salida.	2. Muestra todos los datos disponibles sobre la salida. Si la salida lo permite, muestra la opción de Incluir Acompañantes (Ver Sección "Incluir Acompañantes") y si requiere de pago, ver FA4.
3 Elige la opción de confirmar la reservación.	4.1 Chequea si se reservó para acompañantes, en caso afirmativo, ver FA5. 4.2 Registra la reservación y muestra la confirmación de la operación, terminando el caso de uso.
Sección "Incluir Acompañantes"	
Acción del Actor	Respuesta del sistema
1. Selecciona la opción de Incluir Acompañantes.	2. Muestra los controles necesarios para incluir los acompañantes
3. Introduce el nombre de usuario del dominio UCI del acompañante.	4. Chequea si es válido y si cumple las restricciones de la salida y adiciona el nombre del usuario introducido a la lista de acompañantes, en caso contrario ver FA2. Si es posible adicionar

	otro acompañante, regresa al paso 2. El usuario tiene la opción de eliminar un acompañante, ver FA3
Flujos alternativos	
FA1. El usuario canceló la operación de efectuar la reservación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.
FA2. El nombre de usuario no es válido o no cumple las restricciones de la salida.	Se muestra un mensaje de error y regresa al paso 2 (Sección Incluir Acompañantes).
FA3. El usuario elige la opción eliminar un acompañante.	El sistema lo elimina de la lista de acompañantes.
FA4. La salida incluye pago	El sistema muestra la información de la fecha, hora y lugar donde debe efectuarse el pago y la cantidad requerida (incluyendo la de los acompañantes si los hay).
FA5. Se incluyeron acompañantes en la reservación.	Se envía automáticamente por correo electrónico a los acompañantes la información de la reservación (CUS Enviar Correo Electrónico).
Casos de Uso asociados	
CUS Enviar Correo Electrónico (extend)	

Tabla 3. 7 Descripción del CUS Reservar Salida.

CUS6	Cancelar Reservación	
Propósito	Cancelar la reservación para una salida.	
Actores	Usuario del Sistema	
Resumen	El caso de uso se inicia cuando un usuario decide cancelar su reservación para una salida. Se le muestra información sobre la salida y cuando el usuario acepta se cancela su reservación y se le informa, terminando el caso de uso.	
Referencias	R4, CUS4	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Elige la opción de cancelar una reservación.	2. Muestra los datos disponibles sobre la salida así como la información sobre la cancelación.	
3. Elige la opción de confirmar cancelar la reservación.	4.1 Chequea si reservó para acompañantes, en caso afirmativo, ver FA2. 4.2 Cancela la reservación y confirma la operación, terminando el caso de uso.	
Flujos alternativos		
FA1. El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.	
FA2. Se incluyeron acompañantes en la reservación.	Se envía automáticamente por correo electrónico a los acompañantes la información de que el usuario que les reservó ha cancelado la reservación (CUS Enviar Correo Electrónico).	
Casos de Uso asociados		
CUS Enviar Correo Electrónico (extend)		

Tabla 3. 8 Descripción del CUS Cancelar Reservación.

CUS7	Explorar Avisos
Propósito	Mostrar los avisos disponibles.

Actores	Usuario del Sistema	
Resumen	El caso de uso se inicia cuando el usuario entra al sistema y se muestra una lista con los avisos disponibles. El usuario puede elegir ver detalles de un aviso. Este caso de uso no termina hasta que el usuario no lo especifique explícitamente.	
Referencias	R5	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1. Muestra el título de los avisos disponibles organizados por fecha y permite la opción de ver detalles de un aviso (ver sección: "Ver Detalles de un Aviso").	
Sección "Ver Detalles de un Aviso"		
Acción del Actor	Respuesta del sistema	
1. Selecciona la opción de ver los detalles de un aviso.	2. Se muestran los detalles del aviso seleccionado y se da la opción de ocultar los detalles.	
3. Elige ocultar los detalles	4. Oculta los detalles del aviso.	

Tabla 3. 9 Descripción del CUS Ver Avisos.

CUS8	Ver Perfil	
Propósito	Mostrar su perfil al usuario.	
Actores	Usuario del Sistema	
Resumen	El caso de uso se inicia cuando el usuario accede a su perfil seleccionando la opción Ver Perfil. Se le informa de su estado (Salidas que ha reservado y/o penalizaciones). El caso de uso termina cuando el usuario elige cerrar su perfil.	
Referencias	R6	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Elige la opción Ver Perfil.	2. Muestra los datos del usuario, las salidas que ha reservado y/o las penalizaciones, así como advertencias o informaciones de interés.	
3. Elige la opción Cerrar Perfil	4. Cierra el perfil y termina el caso de uso.	

Tabla 3. 10 Descripción del CUS Ver Perfil.

3.4.3.2 Gestión de Reservaciones / Gestión de Salidas.

CUS9	Explorar Salidas Publicadas	
Propósito	Mostrar las salidas publicadas por un Gestor de Reservaciones.	
Actores	Gestor de Reservaciones	
Resumen	El caso de uso se inicia mostrándose una lista con las salidas publicadas, permitiéndose la selección para ver los datos y la opción de eliminar para cada salida. El caso de uso no termina hasta que no se especifique explícitamente.	
Referencias	R7.1	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Gestionar Salidas.	2. Muestra las diferentes salidas que ha publicado en el sistema el Gestor de Reservaciones organizadas por fecha de publicación. Para cada	

	salida se brinda la opción de ver sus datos (Ver CUS12) y de eliminarla (Ver CUS11), además de la opción de publicar una nueva salida (CUS10).
Casos de Uso asociados	
CUS10 Publicar Salida (extend)	
CUS11 Eliminar Salida (extend)	
CUS12 Ver Datos de una Salida (extend)	

Tabla 3. 11 Descripción del CUS Explorar Salidas Publicadas.

CUS10 Publicar Salida	
Propósito	Publicar una oferta de salida en el sistema
Actores	Gestionador de Reservasiones
Resumen	El caso de uso es iniciado cuando el Gestionador de Reservasiones elige la opción de publicar una nueva salida en el sistema, introduce los datos necesarios y publica la salida, terminando el caso de uso.
Referencias	R7.2
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Elige la opción Publicar Salida.	2. Muestra los controles necesarios para introducir los datos de la salida a publicar.
3.1 Inserta todos los datos de la salida que se dispone a publicar. 3.2 Acepta la operación.	4.1 Chequea que los datos introducidos son válidos y si no quedan campos en blanco, de no ser así ver FA2. 4.2 Inserta la nueva salida en el sistema y regresa a la página anterior, terminando el caso de uso.
Flujos alternativos	
FA1. El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.
FA2. Los datos introducidos no son válidos o existen campos sin llenar.	Se muestra un mensaje de error y regresa al paso 2, manteniendo los datos introducidos correctamente.

Tabla 3. 12 Descripción del CUS Publicar Salida.

CUS11 Eliminar Salida	
Propósito	Eliminar Una Salida Publicada por el Gestionador de Reservasiones.
Actores	Gestionador de Reservasiones
Resumen	Se inicia cuando el Gestionador de Reservasiones elige la opción de eliminar una salida publicada. Se muestran los datos de la salida y se pide confirmación. El caso de uso termina cuando se elimina la salida del sistema.
Referencias	R7.1.2, CUS9
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Elige la opción de Eliminar una Salida.	2. Muestra los datos de la salida que se desea eliminar y pide confirmación de la acción.
3. Confirma la eliminación de la oferta de salida.	4.1 Chequea si la salida tenía usuarios reservados, en caso afirmativo ver FA2. 4.2 Elimina la salida del sistema, registra la acción y muestra la opción de publicar un aviso (CUS19), terminando el caso de uso.

Flujos alternativos	
FA1. El usuario canceló la operación de eliminar la salida antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.
FA2. Existían usuarios reservados para la salida.	Se envía automáticamente por correo electrónico a los usuarios reservados la información de la eliminación de la salida (CUS Enviar Correo Electrónico).
Casos de Uso asociados	
CUS19 Publicar Aviso (extend) CUS Enviar Correo Electrónico (include).	

Tabla 3. 13 Descripción del CUS Eliminar Salida.

CUS12	Ver Datos de una Salida	
Propósito	Ver los datos de una salida publicada por el Gestor de Reservas.	
Actores	Gestor de Reservas	
Resumen	El caso de uso se inicia cuando el Gestor de Reservas elige una de las salidas que ha publicado, se muestran sus datos y las acciones que puede realizar relacionadas con dicha salida. El caso de uso no termina hasta que el usuario no lo especifique explícitamente.	
Referencias	R7.1.1, CUS9	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Ver Datos de una Salida.	2. Muestra todos los datos de la salida seleccionada y las acciones que se pueden realizar (Ver CUS13, CUS14, CUS15, CUS16 y CUS17)	
Casos de Uso asociados		
CUS13 Modificar Datos (extend) CUS14 Imprimir Lista de Usuarios Reservados (extend) CUS15 Registrar Inasistencia (extend) CUS16 Confirmar Pago (extend) CUS17 Archivar Datos (extend)		

Tabla 3. 14 Descripción del CUS Ver Datos de Una Salida.

CUS13	Modificar Datos	
Propósito	Modificar los datos de una salida publicada por el Gestor de Reservas	
Actores	Gestor de Reservas	
Resumen	El caso de uso es iniciado cuando el Gestor de Reservas elige la opción de modificar los datos de una salida que ha publicado anteriormente, se le muestran los datos y se le permite hacer los cambios necesarios. El usuario escoge guardar los cambios y el sistema realiza la acción, terminando el caso de uso.	
Referencias	R7.1.1.1, CUS12	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Elige la opción de Modificar los Datos de una salida publicada.	2. Muestra los datos de la salida permitiendo introducir cambios.	
3.1 Inserta los cambios deseados en los datos de la salida.	4.1 Chequea si la salida tenía usuarios reservados, en caso afirmativo ver FA2.	

3.2 Elige la opción Guardar Cambios.	4.2 Guarda los cambios realizados, muestra la opción de publicar un aviso (CUS19) y termina el caso de uso.
Flujos alternativos	
FA1. El usuario canceló la operación de modificar datos de la salida antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.
FA2. Existían usuarios reservados para la salida.	Se envía automáticamente por correo electrónico a los usuarios reservados la información de los cambios introducidos en los datos de la salida (CUS Enviar Correo Electrónico).
Casos de Uso asociados	
CUS19 Publicar Aviso (extend) CUS Enviar Correo Electrónico (include)	

Tabla 3. 15 Descripción del CUS Modificar Datos.

CUS14	Ver Lista de Usuarios Reservados	
Propósito	Ver la lista de usuarios reservados e imprimirla para controlar el uso del transporte asignado a una salida.	
Actores	Gestionador de Reservaciones	
Resumen	El caso de uso es iniciado cuando el Gestionador de Reservaciones elige la opción de Ver la Lista de Usuarios Reservados para la salida. Se le muestra dicha lista y se le da la opción de imprimirla.	
Referencias	R7.1.1.2, CUS12	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Ver la Lista de Usuarios Reservados	2. Muestra la Lista de Usuarios reservados para la salida con las opciones de Imprimir (ver sección "Imprimir Lista de Usuarios Reservados") y Volver Atrás.	
3. Elige la opción Volver Atrás	4. Regresa a la página anterior y termina el caso de uso.	
Sección "Imprimir Lista de Usuarios Reservados"		
Acción del Actor	Respuesta del sistema	
1. Selecciona la Opción Imprimir	2. Muestra el diálogo de impresión.	
3. Escoge las opciones de impresión deseadas y elige aceptar, en caso contrario ver FA1.	4. Imprime la Lista de Usuarios Reservados y va al paso 2 del Flujo normal de eventos.	
Flujos alternativos		
FA1. Cancela la opción de imprimir.	Regresa al paso 2 del Flujo normal de eventos.	

Tabla 3. 16 Descripción del CUS Ver Lista de Usuarios Reservados.

CUS15	Registrar Inasistencia	
Propósito	Guardar registro de la inasistencia de usuarios a una salida efectuada.	
Actores	Gestionador de Reservaciones	
Resumen	Inicia cuando el Gestionador de Reservaciones elige la opción de Registrar Inasistencia de Usuarios a una Salida, se muestra la lista de usuarios reservados y marca los usuarios que no se presentaron a la salida. Elige la opción de salvar y el sistema almacena la información, terminando el caso de uso.	

Referencias	R7.1.1.3, CUS12	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de Registrar Inasistencia.	2. Muestra la lista de usuarios reservados (CUS14) con la opción de marcarlos para identificar a los que inasistieron.	
3.1 Señala los usuarios que no se presentaron a la salida. 3.2 Elige la opción de salvar.	4 Guarda los datos introducidos y termina el caso de uso.	
Flujos alternativos		
FA1. El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.	
Casos de Uso asociados		
CUS14 Ver lista de Usuarios Reservados (include)		

Tabla 3. 17 Descripción del CUS Registrar Inasistencia.

CUS16	Confirmar Pago	
Propósito	Guardar la confirmación del pago de los usuarios para una reservación.	
Actores	Gestionador de Reservaciones	
Resumen	Inicia cuando el Gestionador de Reservaciones elige la opción de Confirmar Pago de Usuarios, se muestra la lista de usuarios reservados y marca los usuarios que han pagado su reservación. Elige la opción de salvar y el sistema almacena la información, terminando el caso de uso.	
Referencias	R7.1.1.4, CUS12	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Elige la opción de Confirmar Pago.	2. Muestra la lista de usuarios reservados (CUS14) con la opción de marcarlos para identificar a los que han pagado la reservación.	
3.1 Señala los usuarios que han efectuado el pago de la reservación. 3.2 Elige la opción de salvar.	4. Guarda los datos introducidos y regresa a la página anterior, terminando el caso de uso.	
Flujos alternativos		
FA1. El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.	
Casos de Uso asociados		
CUS14 Ver lista de Usuarios Reservados (include)		

Tabla 3. 18 Descripción del CUS Confirmar Pago.

CUS17	Archivar Datos	
Propósito	Archivar los datos de una salida.	
Actores	Gestionador de Reservaciones	
Resumen	Inicia el caso de uso cuando el Gestionador de Reservaciones elige la opción de Archivar Datos de una Salida. Se chequea si han concluido las reservaciones y no hay datos sin introducir sobre la salida, en caso afirmativo se archivan los datos y termina el caso de uso.	
Referencias	R7.1.1.5, CUS10 (extend)	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Elige la opción de Archivar Datos de la Salida.	2.1 Chequea si ha culminado el proceso de reservaciones y todos los datos sobre la salida	

	han sido introducidos (ej. confirmación de pago si la salida lo incluye). 2.2 Muestra los datos que se van a archivar (Datos de la Salida, Lista de Usuarios Reservados, Inasistencias, confirmación de pago, etc.) y brinda la opción de Archivar Datos. En caso contrario, ver FA2
3. Acepta la operación, seleccionando Archivar Datos.	4.1 Guarda los datos de la salida, eliminando la oferta del sistema. 4.2 Regresa a la página anterior, terminando el caso de uso.
Flujos alternativos	
FA1. El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.
FA2. No han concluido las reservaciones para la salida o existen datos sin introducir.	Se muestra un mensaje de error y regresa a la página anterior.

Tabla 3. 19 Descripción del CUS Archivar Datos.

3.4.3.3 Gestión de Reservaciones / Gestión de Avisos.

CUS18	Gestionar Avisos	
Propósito	Realizar acciones (modificar, eliminar y publicar) sobre los avisos.	
Actores	Gestionador de Reservaciones	
Resumen	El caso de uso se inicia mostrándose una lista con los avisos publicados, permitiéndose la selección para ver el texto completo y las opciones de modificar y eliminar para cada aviso. El caso de uso no termina hasta que no se especifique explícitamente.	
Referencias	R8.1	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de gestionar avisos.	1. Muestra los diferentes avisos que ha publicado en el sistema el Gestionador de Reservaciones organizados por fecha de publicación. Para cada aviso se brinda la opción de modificar sus datos (Ver Sección “Modificar Datos del Aviso”) y de eliminarlo (Ver Sección “Eliminar Aviso”). Además se brinda la opción de publicar un nuevo aviso (CUS20)	
Sección “Modificar Datos del Aviso”		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción de modificar los datos del aviso.	2. Muestra los datos actuales y los controles necesarios para modificar el aviso.	
3.1 Inserta los nuevos datos. 3.2 Acepta la operación.	4.1 Chequea si existen campos vacíos, en ese caso ver FA2. 4.2 Guarda los cambios y termina el caso de uso regresando a la página anterior.	
Sección “Modificar Datos del Aviso”		
Acción del actor	Respuesta del sistema	
1. Elige la opción de eliminar un aviso.	2. Muestra el aviso a eliminar y pide confirmación.	

3. Confirma la eliminación del aviso	4. Elimina el aviso y regresa a la página anterior
Flujos alternativos	
FA1. El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.
FA2. Existen campos sin llenar.	Se muestra un mensaje de error y regresa al paso 2 de la sección “Modificar Datos del Aviso”.
Casos de Uso asociados	
CUS20 Publicar Aviso (extend)	

Tabla 3. 20 Descripción del CUS Explorar Avisos Publicados.

CUS19	Publicar Aviso
Propósito	Publicar un aviso en el sistema
Actores	Gestionador de Reservasiones
Resumen	El caso de uso es iniciado cuando el Gestionador de Reservasiones elige la opción de publicar un nuevo aviso en el sistema, se le muestran los controles necesarios, introduce los datos y acepta, publicándose el aviso y terminado el caso de uso.
Referencias	R8.2, CUS11, CUS13 y CUS18
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de publicar un nuevo aviso.	2. Muestra los controles necesarios para introducir los datos del aviso a publicar.
3.1 Inserta todos los datos del aviso (Título y texto). 3.2 Acepta la operación.	4.1 Chequea la existencia de campos en blanco, de ser así ver FA2. 4.2 Inserta el nuevo aviso en el sistema y regresa a la página anterior, terminando el caso de uso.
Flujos alternativos	
FA1. El usuario canceló la operación antes de concluir el caso de uso.	Se retorna a la página anterior sin guardar los cambios.
FA2. Existen campos sin llenar.	Se muestra un mensaje de error y regresa al paso 2, manteniendo los datos introducidos.

Tabla 3. 21 Descripción del CUS Publicar Aviso.

3.4.3.4 Gestión de Reservasiones / Gestión de Usuarios.

CUS20	Gestionar Usuarios
Propósito	Manejar información y realizar acciones sobre los usuarios del sistema.
Actores	Gestionador de Reservasiones
Resumen	
Referencias	R9.1
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Elige la opción de gestionar usuarios.	2. Muestra los controles necesarios para realizar una búsqueda.
3. Introduce el nombre de usuario el dominio UCI que desea buscar.	4.1 Realiza la búsqueda y en caso de que no encuentre al usuario, ver FA1. 4.2 Muestra los datos del usuario, las reservasiones que ha efectuado y las penalizaciones, con las opciones de cancelar

	reservación (ver sección “Cancelar una Reservación del Usuario”) y despenalizar (ver sección “Despenalizar Usuario”), ambas en el caso de que sea posible, junto a las opciones de penalizar (ver sección “Penalizar Usuario”) y reservarle para una salida (ver sección “Reservar Usuario”).
Sección “Cancelar una Reservación del Usuario”	
Acción del actor	Respuesta del sistema
1. Elige la opción de cancelar la reservación del usuario para una salida.	2. Pide confirmación de la cancelación de la reservación del usuario.
3. Acepta la operación.	4.1 Cancela la reservación del usuario 4.2 Envía notificación por correo electrónico al usuario (CUS Enviar Correo Electrónico) y regresa a la página anterior.
Sección “Despenalizar Usuario”	
Acción del actor	Respuesta del sistema
1. Elige la opción de despenalizar al usuario	2. Pide confirmación de la despenalización del usuario.
3. Acepta la operación.	4.1 Despenaliza al usuario. 4.2 Envía notificación por correo electrónico al usuario (CUS Enviar Correo Electrónico) y regresa a la página anterior.
Sección “Penalizar Usuario”	
Acción del actor	Respuesta del sistema
1. Elige la opción de penalizar al usuario.	2. Muestra los controles necesarios para introducir los datos de la penalización.
3.1 Señala el tipo de penalización y el motivo. 3.2 Acepta la operación.	4.1 Chequea si no quedaron campos vacíos y aplica la penalización al usuario, en caso contrario ver FA2. 4.2 Envía notificación por correo electrónico al usuario (CUS Enviar Correo Electrónico) y regresa a la página anterior.
Sección “Reservar Usuario”	
Acción del actor	Respuesta del sistema
1. Elige la opción de reservar al usuario	2. Muestra una lista con las reservaciones disponibles (relacionadas al área del Gestionador de Reservaciones y que posean capacidades) y la opción de reservar al usuario. Si no existen salidas que cumplan esas características, ver FA3.
3.1 Selecciona la oferta de salida en la que va a reservar al usuario. 3.2 Acepta la operación.	4.1 Adiciona al usuario a la Lista de Usuarios Reservados de la salida 4.2 Envía notificación por correo electrónico al usuario (CUS Enviar Correo Electrónico) y regresa a la página anterior.
Flujos alternativos	
FA1. La búsqueda no arrojó un resultado.	Informa al usuario y regresa al paso 2.

FA2. Existen campos sin llenar.	Se muestra un mensaje de error y regresa al paso 2 de la sección "Penalizar Usuario", manteniendo los datos introducidos correctamente.
FA3. No existen ofertas de salida del área a la que pertenece el Gestor de Reservas con capacidades disponibles	Muestra la información y regresa a la página anterior.
Casos de Uso asociados	
CUS Enviar Correo Electrónico (extend)	

Tabla 3. 22 Descripción del CUS Buscar Usuario.

3.4.3.5 Gestión de Reservas / Gestión de Reportes.

CUS21		Elaborar Reporte	
Propósito	Elaborar un reporte estadístico sobre el proceso de reservas		
Actores	Gestor de Reservas		
Resumen	El caso de uso se inicia cuando el Gestor de Reservas elige la opción de elaborar un reporte, seleccionándolo de una lista. Especifica las opciones necesarias y acepta la operación, se muestra el resultado permitiendo la impresión y termina el caso de uso.		
Referencias	R10		
Flujo normal de eventos			
Acción del actor		Respuesta del sistema	
1. Elige la opción de gestionar reportes.		2. Muestra una lista con los reportes disponibles, permitiendo selección.	
3. Selecciona el tipo de reporte que desea elaborar.		4. Muestra los controles necesarios para ingresar las opciones del reporte.	
5.1 Introduce los datos necesarios. 5.2 Elige la opción de Mostrar el Reporte.		6.1 Valida los datos introducidos (ver FA2) y elabora el reporte. 6.2 Muestra los resultados, dando la opción de imprimirlos (ver sección "Imprimir Reporte") o de regresar a la página anterior para terminar el caso de uso.	
Sección "Imprimir Reporte"			
Acción del Actor		Respuesta del sistema	
1. Selecciona la opción de Imprimir el Reporte.		2. Muestra el diálogo de impresión.	
3. Escoge las opciones de impresión deseadas y elige aceptar, en caso contrario ver FA1.		4. Imprime el reporte y va al paso 6.2 del Flujo normal de eventos.	
Flujos alternativos			
FA1. El usuario canceló la operación antes de concluir el caso de uso.		Se retorna a la página anterior sin guardar los cambios.	
FA2. Los datos introducidos no son correctos o son insuficientes.		Muestra un mensaje de error y regresa al paso 6.1 del Flujo normal de eventos.	

Tabla 3. 23 Descripción del CUS Elaborar Reporte

3.5 Conclusiones.

En este capítulo se realizó una modelación del negocio propuesto, identificándose los actores, trabajadores y los casos de uso correspondientes; se plantearon los requerimientos funcionales y no funcionales de la aplicación a desarrollar y se modeló la misma en términos de casos de uso de sistema.

4.1 Introducción.

En el presente capítulo se diseña la propuesta de solución, la base de datos, se especifican los principios de diseño gráfico, y el diseño de la interfaz de aplicación. Finalmente, se utiliza el modelo de despliegue y el modelo de implementación para una mejor descripción de la solución propuesta.

4.2 Patrones.

“En la terminología de objetos, el patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas. (...) Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software. Todo lo contrario: intentan codificar el conocimiento, las expresiones y los principios ya existentes: cuanto más trillados y generalizados, tanto mejor.”[4]

4.2.1 Patrones de Arquitectura.

Se empleó una Arquitectura Multicapas que consiste en hacer una división de subsistemas en los que se ubican clases con responsabilidades propias. Se hacen abstracciones de las distintas funcionalidades o responsabilidades del sistema agrupándolas en capas y obteniéndose clases que serán independientes del resto, con lo que se podrán reutilizar fácilmente (Ver Figura X). Además se usó el patrón DAO (Data Access Object) cuyo propósito es crear un contrato para manejar la persistencia de una información ocultando la implementación de esta persistencia. Este patrón es útil para abstraer y encapsular los accesos, gestionar las conexiones a la base de datos

y obtener los datos almacenados. Para ello se ha utilizado el poderoso framework de java, Hibernate.

4.2.2 Patrones de Diseño.

En el diseño de la aplicación se tuvieron en cuenta principalmente los patrones Experto, y Creador. El primero plantea que siempre se debe asignar una responsabilidad al experto en información, o sea, a la clase con toda la información necesaria para llevarla a cabo. El segundo expresa que la responsabilidad de crear una instancia de una determinada clase debe asignarse a otra clase, siempre que esta agregue, contenga, registre o utilice específicamente los objetos de aquella. Además se tuvieron en cuenta los patrones Bajo Acoplamiento y Alta Cohesión. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

4.3 Modelo de Diseño.

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto los funcionales como los no funcionales, creándose así una entrada apropiada para las actividades de implementación.

4.3.1 Clases Persistentes.

El diagrama de clases persistentes es usado para modelar la estructura lógica de la base de datos con clases representando tablas, y atributos de clase representando columnas. (Ver el Anexo 3. Diagrama de Clases Persistentes.).

4.3.3 Diagramas de Clases Web.

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

Se elaboró un diagrama de clases Web para cada caso de uso del sistema de forma tal que se facilite la comprensión de cómo se relacionan los distintos componentes en la realización de cada uno de ellos. En el Anexo 4. Diagramas de Clases Web. se incluyen los diagramas correspondientes a los casos de uso principales.

4.4 Modelo de Datos.

El diagrama del modelo de datos se corresponde con la representación física de la base de datos. (Ver el Anexo 5. Modelo de Datos.).

4.5 Diseño de la Interfaz de la Aplicación.

El diseño Web no se limita a la apariencia estética, a la combinación de colores, o a un logo más o menos acertado. De él depende que la información sea útil, que los servicios se puedan usar, es decir, el diseño convierte a una aplicación Web en algo atractivo para el usuario, por su estética y su utilidad.

Una aplicación con una interfaz bien diseñada debe tener, además de un buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos.

Persiguiendo este objetivo se han seguido, para el diseño de la interfaz de usuario del Sistema de Reservaciones de Extensión Universitaria, los siguientes principios:

- Requerir un mínimo proceso de aprendizaje y permitir su utilización desde el primer momento, por cualquier persona que tenga un mínimo dominio de la computación.
- Evitar el uso inadecuado o excesivo de las metáforas, que puede dificultar enormemente el aprendizaje del usuario.

- Garantizar la legibilidad, el color de los textos debe contrastar con el del fondo, y el tamaño de fuente debe ser suficientemente grande.
- Evitar elementos invisibles de navegación que han de ser inferidos por los usuarios, menús desplegados, indicaciones ocultas, etc.
- Requerir de los usuarios un mínimo esfuerzo para alcanzar sus objetivos.
- Evitar las caídas inesperadas de la aplicación y los enlaces rotos.
- Mostrar al usuario solamente aquellas opciones a las que, dado su rol en el negocio, tiene derecho a acceder.
- Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse de que es correcta la opción seleccionada.
- Mostrar la mayor cantidad de información acerca de las opciones brindadas en un momento dado, de modo que el usuario siempre sepa cuáles son las operaciones a las que puede acceder y en qué consiste exactamente cada una.

4.5.1 Estándares de la interfaz.

Ver el Anexo 6. Interfaz de la Aplicación.

4.5.2 Concepción de la ayuda.

4.5.3 Tratamiento de errores.

4.6 Modelo de Despliegue.

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite

comprender la correspondencia entre la arquitectura software y la arquitectura hardware (ver el Anexo 7. Diagrama de Despliegue.).

4.7 Modelo de Implementación.

El modelo de implementación describe cómo se implementan los elementos del modelo de diseño. En el Anexo 8. Diagrama de Componentes. se puede observar una vista general del diagrama de componentes, así como una vista detallada de cada uno de los paquetes en que se han dividido los mismos, con vistas a lograr una mayor claridad y comprensión del modelo.

4.8 Conclusiones.

En este capítulo se modeló la construcción de la solución propuesta, a través del modelo de diseño, de implementación y de despliegue, así como también el modelo de datos y se analizaron los patrones de diseño y los principios de diseño de la interfaz de la aplicación.

5.1 Introducción.

En el presente capítulo se realiza una estimación de los costos que generará el desarrollo de la aplicación usando el modelo COCOMO II y se analizan los beneficios para determinar la factibilidad de la implantación del Sistema de Reservas de Extensión Universitaria.

5.2 El Modelo de Estimación de Costos COCOMO II.

En la actualidad existen diversos modelos para estimar los costos de los proyectos de software. Uno de los más aceptados es el modelo COCOMO II (del inglés Constructive Cost Model). El modelo COCOMO fue originalmente publicado en software Engineering Economics por Barry Boehm en 1981 y la versión COCOMO II del 2000 actualiza y extiende este estándar de estimación.

Este modelo expresa el esfuerzo de desarrollo en términos de Personas Mes (cantidad de tiempo que una persona dedica a trabajar sobre el proyecto de desarrollo durante un mes). Finalmente se hace una estimación del costo monetario del proyecto teniendo en cuenta el tiempo estimado de desarrollo, la cantidad de personas involucradas y el salario de estas personas.

En el caso del Sistema de Reservas de Extensión Universitaria, al ser un sistema a desarrollar en la Universidad de las Ciencias Informáticas, no se hace necesario calcular el costo monetario, pues la elaboración del sistema no supone gastos adicionales por concepto de salario para la institución, ya que cuenta con el capital humano necesario y no necesita contratar otro personal. Por otra parte la universidad cuenta con la infraestructura tecnológica necesaria para llevar a cabo el proyecto.

5.3 Estimación del Costo.

Obtención de los puntos de función (UFP).

ENTRADAS EXTERNAS			
Nombre de la Entrada Externa	Cantidad de Ficheros	Cantidad de Elementos de Datos	Clasificación
Reservar Salida	1	3	Simple
Cancelar Salida	1	5	Simple
Publicar Salida	5	20	Complejo
Eliminar Salida	6	25	Complejo
Registrar Inasistencia	2	3	Simple
Confirmar Pago	1	1	Simple
Modificar Datos	5	20	Complejo
Archivar Datos	7	28	Complejo
Gestionar Avisos	1	4	Simple
Publicar Aviso	1	4	Simple
Gestionar Usuarios	3	11	Medio
Administrar Áreas	2	4	Simple
Administrar Permisos	2	6	Medio
Administrar Roles	2	4	Simple
Administrar Personas	3	8	Medio
Administrar Penalizaciones	1	2	Simple

Tabla 5. 1 Entradas Externas

SALIDAS EXTERNAS			
Nombre de la Salida Externa	Cantidad de Ficheros	Cantidad de Elementos de Datos	Clasificación
Explorar Salidas	5	24	Complejo
Reservar Salida	2	6	Medio
Explorar Avisos	1	3	Simple
Elaborar Reporte	5	20	Complejo

Tabla 5. 2 Salidas Externas

CONSULTAS EXTERNAS			
Nombre de la Consulta Externa	Cantidad de Ficheros	Cantidad de Elementos de Datos	Clasificación
Autenticar Usuario	2	4	Simple
Comenzar Reservaciones	2	3	Simple
Terminar Cancelaciones	1	1	Simple
Enviar Correo Electrónico	1	5	Simple
Reservar Salida	2	17	Medio
Ver Perfil	4	11	Complejo
Explorar Salidas Publicadas	6	11	Complejo
Ver Datos de una Salida	5	20	Complejo

Ver Lista de Usuarios Reservados	2	4	Simple
Gestionar Avisos	1	4	Simple
Gestionar Usuarios	3	11	Medio
Administrar Áreas	1	3	Simple
Administrar Permisos	1	2	Simple
Administrar Roles	2	4	Simple
Administrar Personas	3	7	Medio
Administrar Penalizaciones	1	2	Simple

Tabla 5. 3 Consultas Externas

FICHEROS LÓGICOS INTERNOS			
Nombre del Fichero Lógico Interno	Cantidad de Records	Cantidad de Elementos de Datos	Clasificación
Área	1	3	Simple
Persona	1	3	Simple
Rol	1	2	Simple
Rol _ Permiso	1	2	Simple
Permiso	1	2	Simple
Usuario _ Penalizado	1	4	Simple
Penalización	1	2	Simple
Salida	1	9	Simple
Salida _ Pago	1	4	Simple
Salida _ Restricción	1	2	Simple
Restricción	1	2	Simple
Usuario _ Reservado	1	5	Simple
Aviso	1	4	Simple
Archivo	1	3	Simple
Usuario _ Ausente	1	2	Simple

Tabla 5. 4 Ficheros Lógicos Internos

FICHEROS DE INTERFAZ EXTERNA			
Nombre del Fichero de Interfaz Externa	Cantidad de Records	Cantidad de Elementos de Datos	Clasificación
Buscar Estructura	3	8	Simple
Buscar Datos de la Persona	3	9	Simple
Buscar Tipo de Persona	2	4	Simple

Tabla 5. 5 Ficheros de Interfaz Externa

PUNTOS DE FUNCIÓN DESAJUSTADOS							
Elementos	Simple	X Peso	Medios	X Peso	Complejos	X Peso	Subtotal
Ficheros Lógicos Internos	15	7	0	10	0	15	105
Ficheros de Interfaz Externa	3	5	0	7	0	10	15
Entradas Externas	9	3	3	4	4	6	63

Salidas Externas	9	4	1	5	2	7	55
Peticiones	10	3	3	4	3	6	60
Total 298							

Tabla 5. 6 Puntos de Función Desajustados

Estimación de la cantidad de instrucciones fuente.

INSTRUCCIONES FUENTE		
Características		Valor
Puntos de función desajustados		298
Lenguaje	Java	60%
	Javascript	35%
	SQL	5%
Instrucciones fuentes por puntos de función	Java	62
	Javascript	58
	SQL	39
Instrucciones Fuente		17716

Tabla 5. 7 Instrucciones Fuente

Aplicación de las Fórmulas de Bohem.

FACTORES DE ESCALA			
Factores	Complejidad	Valor	Justificación
PREC	Nominal	3.72	Existe Precedencia
FLEX	Alto	2.03	La flexibilidad debe ser alta.
TEAM	Muy Alta	1.10	El equipo está muy cohesionado.
RESL	Alto	2.83	Hay buena fortaleza en la arquitectura.
PMAT	Nominal	4.68	La madurez es nominal.
$\sum_{j=1}^5 SF_j = 14.36$			

Tabla 5. 8 Factores de Escala

FACTORES MODIFICADORES DEL ESFUERZO				
Parámetro	Atributo	Clasificación	Valor	Justificación
Producto	RELY	Nominal	1.00	
	DATA	Nominal	1.00	
	DOCU	Nominal	1.00	
	CPLX	Nominal	1.00	
	RUSE	Nominal	1.00	
Plataforma	TIME	Alto	1.11	
	STOR	Nominal	1.00	
	PVOL	Bajo	0.87	

Personal	ACAP	Alto	0.85	
	AEXP	Bajo	1.10	
	PCAP	Alto	0.88	
	PEXP	Bajo	1.09	
	LTEX	Bajo	1.09	
	PCON	Bajo	1.12	
Proyecto	TOOL	Alto	0.90	
	SITE	Nominal	1.00	
	SCED	Nominal	1.00	
$\prod_{i=1}^n EM_i = 0.95159037887$				

Tabla 5. 9 Factores Modificadores del Esfuerzo

Esfuerzo:

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

$$E = 0.91 + 0.01 \times 14.36 = 1.0536$$

$$PM = 2.94 \times (17.716^{1.0536}) \times 0.95159037887 \approx 58 \text{ personas / mes}$$

5.4 Beneficios.

La implantación del Sistema de Reservas de Extensión Universitaria disminuye considerablemente los gastos por concepto de papel e imprenta para la elaboración de las listas de usuarios reservados y los reportes estadísticos.

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

Se logra una disminución considerable del tiempo que se emplea en realizar las actividades, por ejemplo, el tiempo que emplean los trabajadores de la DR y la DEU en elaborar las listas de usuarios reservados para cada salida y el de los usuarios del

servicio en hacer la cola para obtener una reservación, evitándose también que después de la espera no queden capacidades disponibles.

Se facilita el trabajo con estadísticas y la obtención de reportes diversos, lo que elimina contenido de trabajo de los trabajadores y se hacen más confiables los resultados al eliminarse el error humano.

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

5.5 Análisis de costo / beneficio.

El Sistema de Reservaciones de Extensión Universitaria no requiere inversión en software pues todas las herramientas que se han empleado en su desarrollo son libres y de código abierto.

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

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

Por todo ello se considera que es factible el desarrollo de la aplicación y que el esfuerzo de desarrollo (58 Personas Mes) está plenamente justificado.

5.6 Conclusiones.

Del estudio realizado en este capítulo, a partir del análisis de los costos y beneficios, se concluye que es factible el desarrollo de esta aplicación, por todos los aportes económicos y sociales que genera.

BIBLIOGRAFÍA

- [1] Autor. Introducción a las aplicaciones Web. Año. Disponible en:
<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>
- [2] Connalen, Jim. "Building Web Applications with UML". Addison Wesley. 1999.
- [3] Jacobson, Ivar; Booch, Grady; Rumbaugh, James. "El Proceso Unificado de Desarrollo de Software". 2 Tomos. Addison Wesley. 2000.
- [4] Larman, Craig. "UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos". 2 Tomos. Addison Wesley. Prentice Hall. 1999. 507pp
- [5] Mariaca, Cinxgler. Título. Año. Disponible en:
http://glud.udistrital.edu.co/glud/areas/doc/articulos/1_articulo_ws/
- [6] Mateu, Carles. "Desarrollo de aplicaciones Web". Eureka Media, SL. 2004. 377pp.
- [7] Musciano, Chuck y Hill Knenedy. "HTML, La guía completa". 2da Edición. McGraw-Hill, 1999. 531pp.
- [8] Wikipedia. La Enciclopedia Libre. J2EE. Disponible en:
- [9] Wikipedia. La Enciclopedia Libre. Java. Disponible en:
- [10] Wikipedia. La Enciclopedia Libre. MySQL. Disponible en:
- [11] Wikipedia. La Enciclopedia Libre. Oracle. Disponible en:
- [12] Wikipedia. La Enciclopedia Libre. PostgreSQL. Disponible en:
- [13] Wikipedia. La Enciclopedia Libre. SGBD. Disponible en:
- [14] Wikipedia. La Enciclopedia Libre. SQL Server. Disponible en:
- [15] Wikipedia. La Enciclopedia Libre. UML. Disponible en:
http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado
- [16] Niederst, Jennifer. Web Design in a Nutshell.

GLOSARIO DE TÉRMINOS

DEU: Dirección de Extensión Universitaria.

DR: Dirección de Recreación.

UCI: Universidad de las Ciencias Informáticas.

VREU: Vicerrectoría de Extensión Universitaria.

Anexo 1. Modelo del Negocio Propuesto.

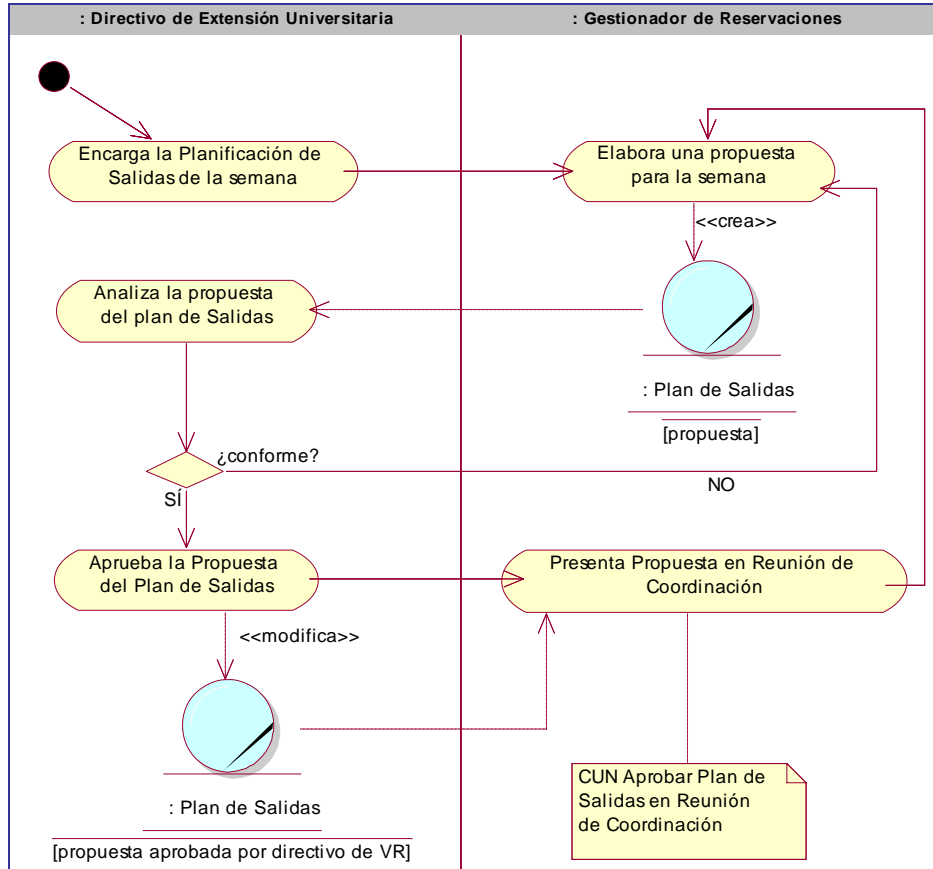


Fig. 6. 1 Diagrama de Actividad. CUN Planificar Salidas.

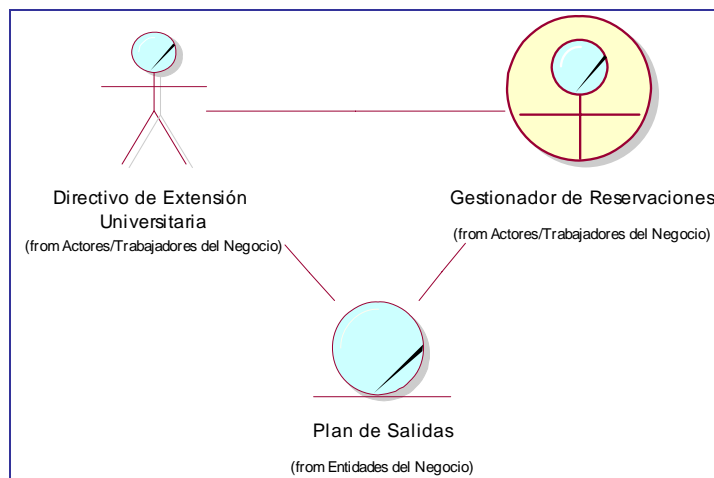


Fig. 6. 2 MON. CUN Planificar Salidas.

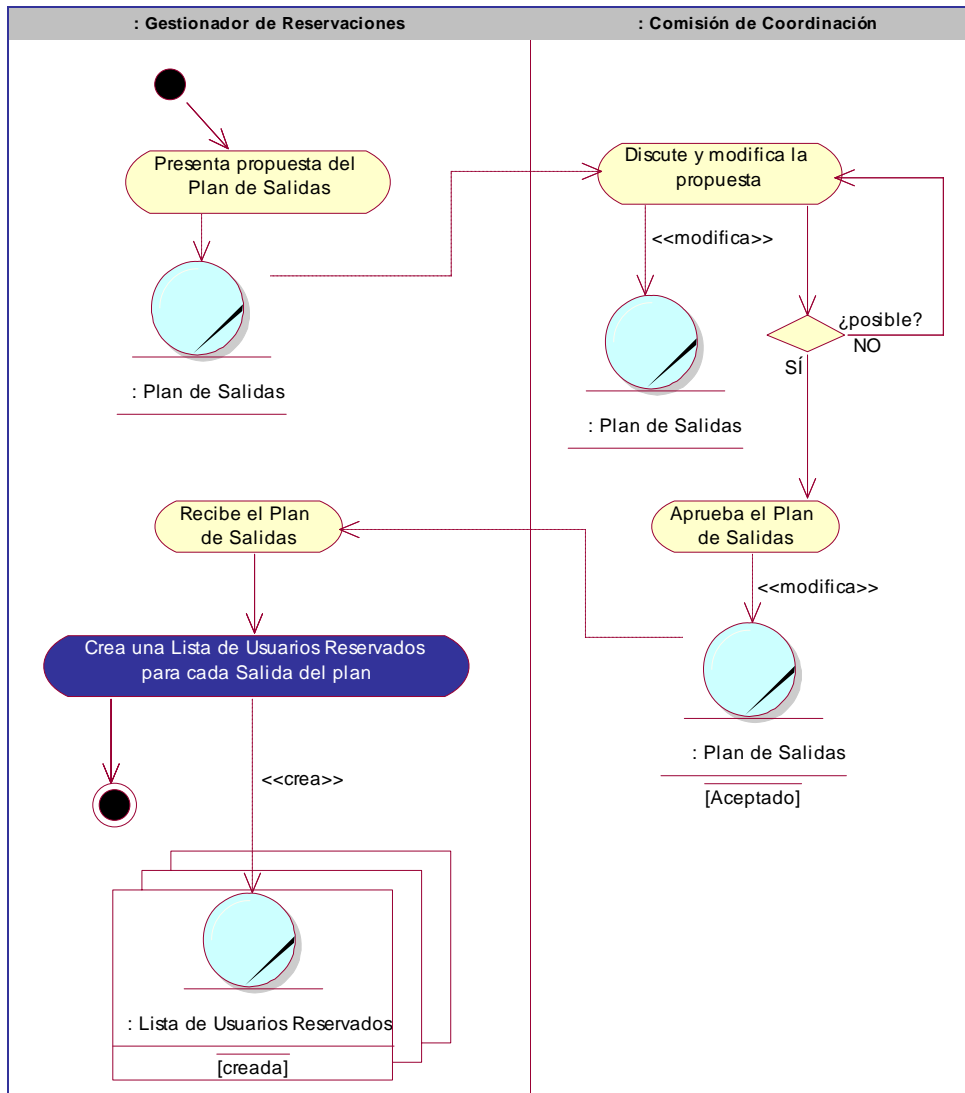


Fig. 6. 3 Diagrama de Actividad. CUN Aprobar Plan de Salidas en Reunión de Coordinación.

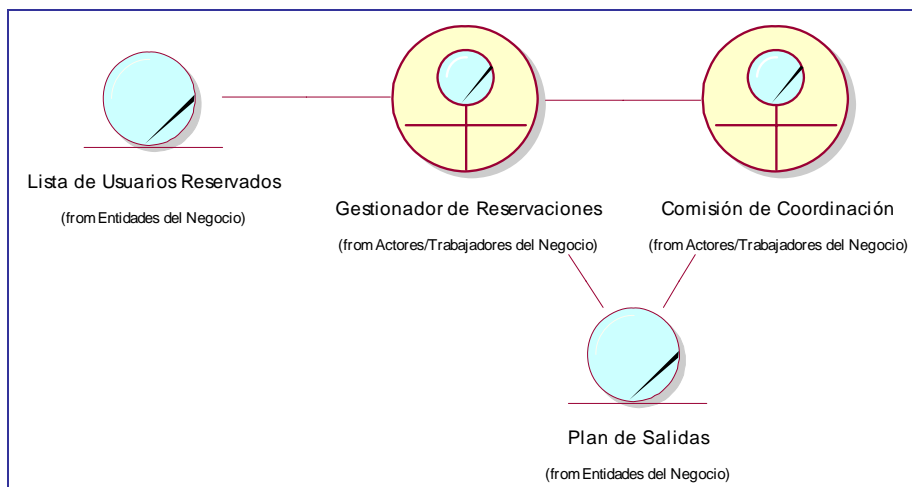


Fig. 6. 4 MON. CUN Aprobar Plan de Salidas en Reunión de Coordinación.

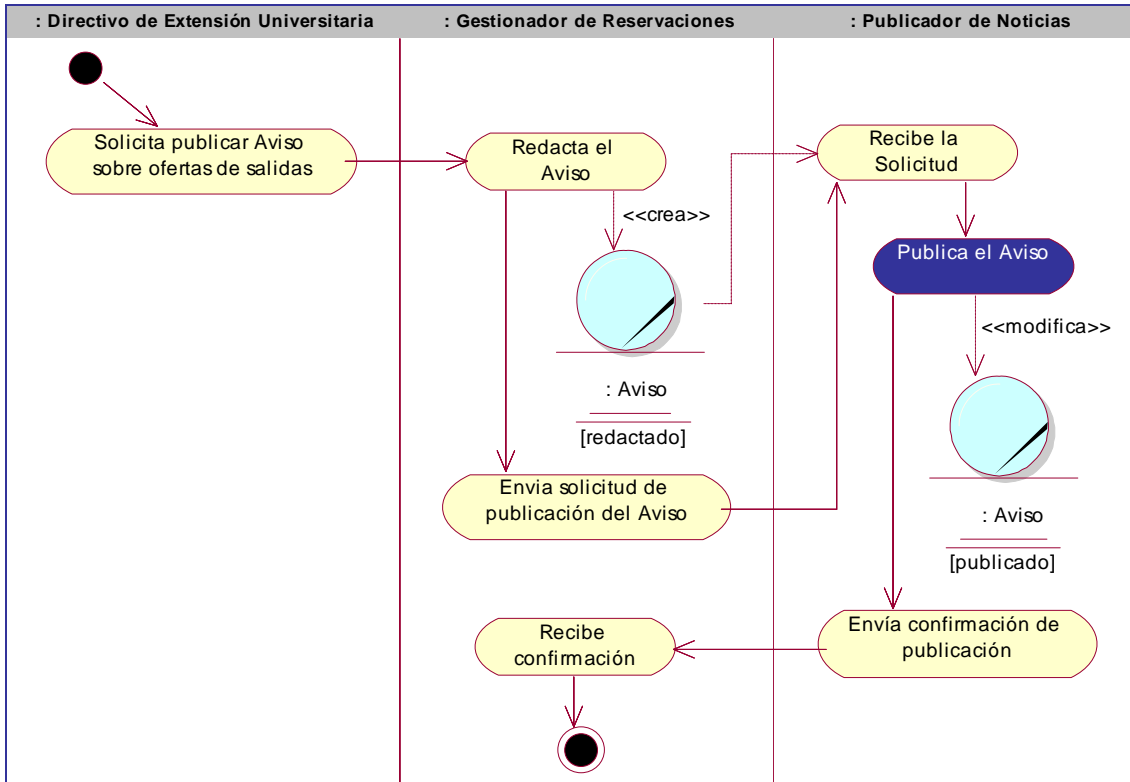


Fig. 6. 5 Diagrama de Actividad. CUN Publicar Aviso.

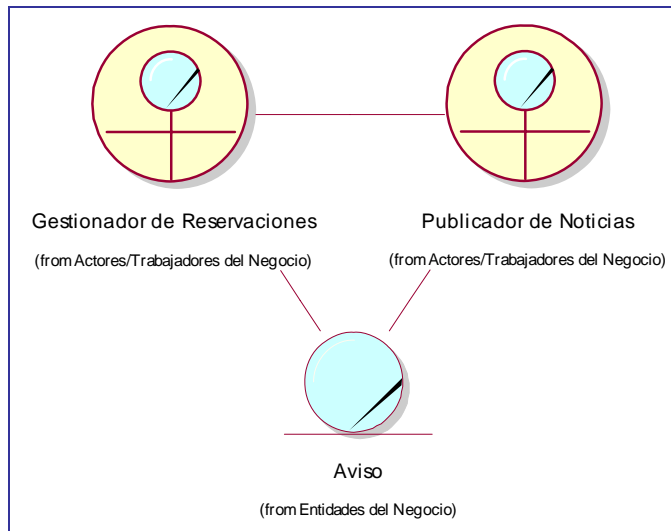


Fig. 6. 6 MON. CUN Publicar Aviso.

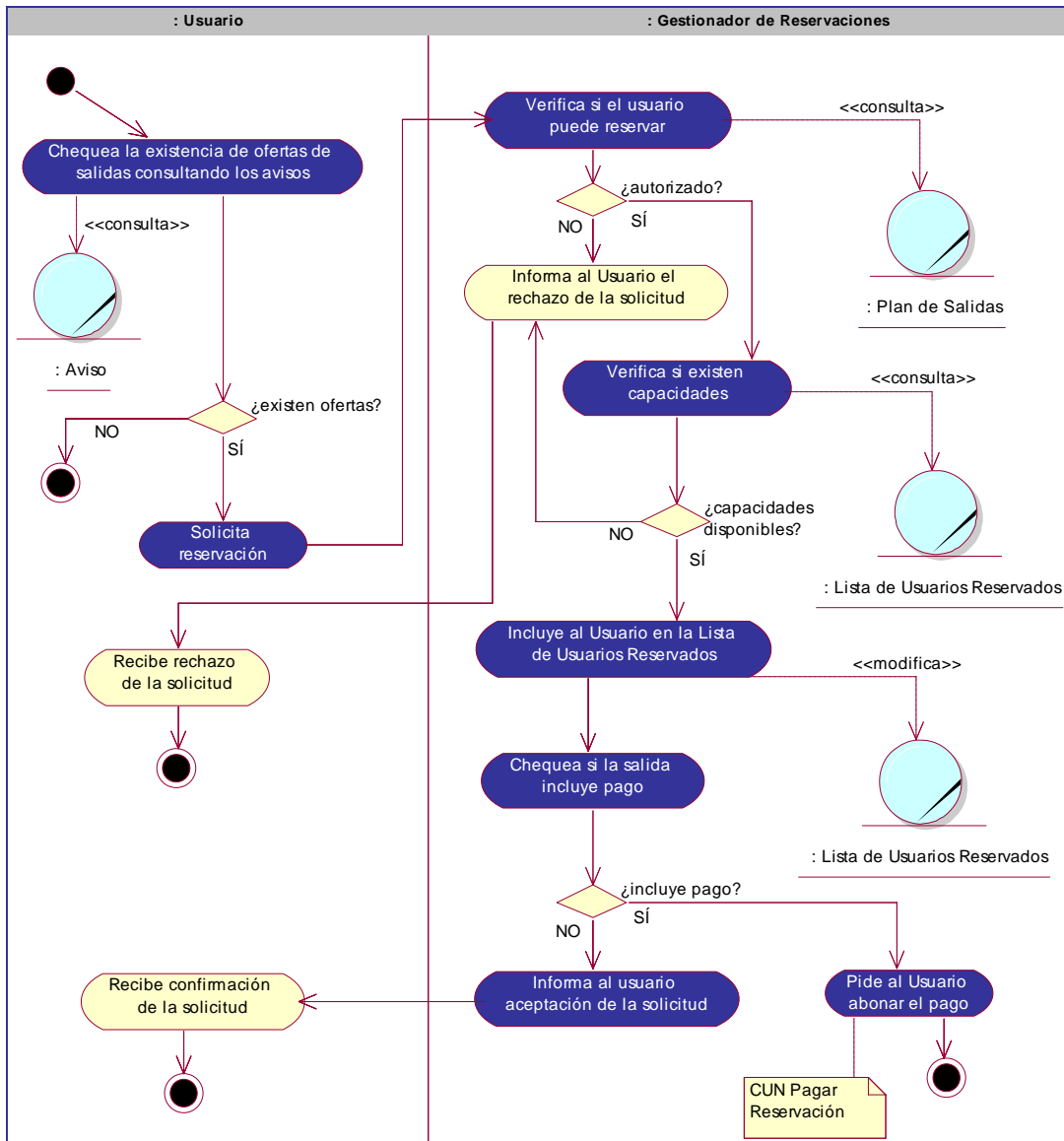


Fig. 6. 7 Diagrama de Actividad. CUN Reservar Salida.

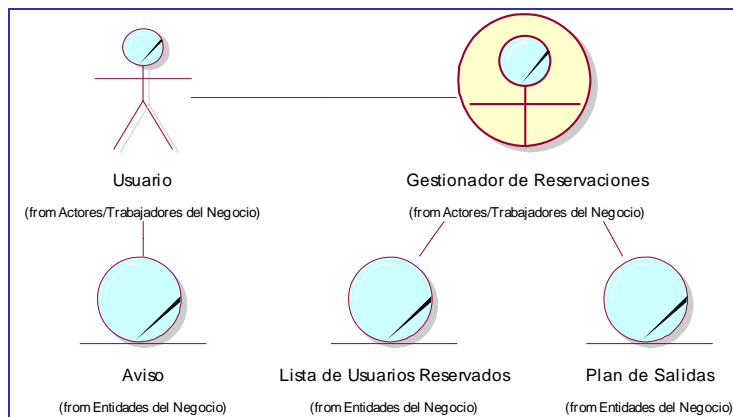


Fig. 6. 8 MON. CUN Reservar Salida.

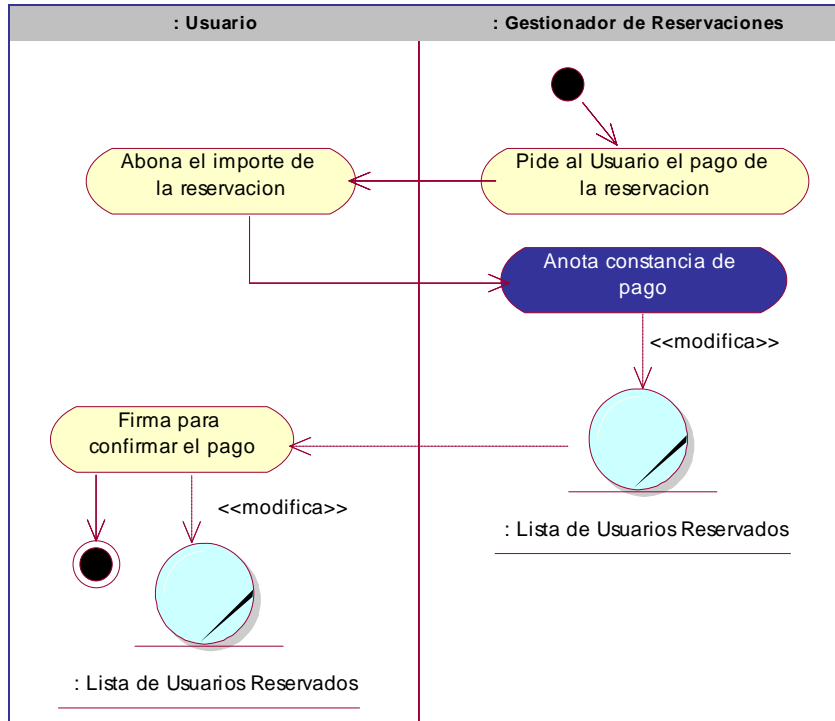


Fig. 6. 9 Diagrama de Actividad. CUN Pagar Reservación.

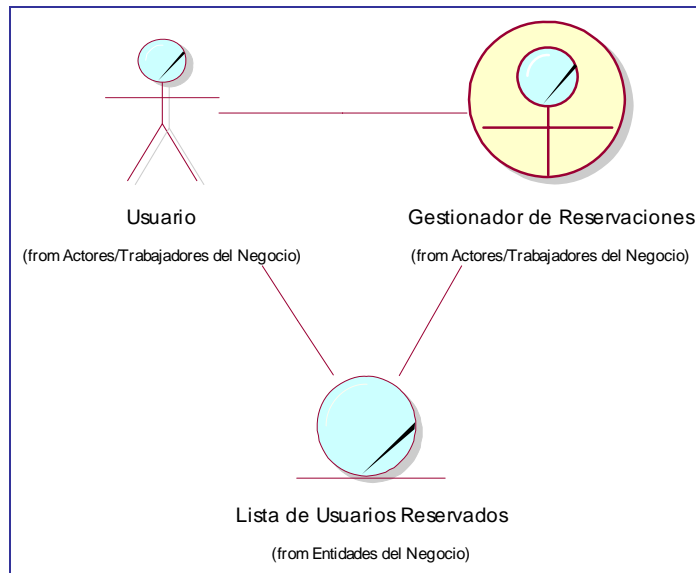


Fig. 6. 10 MON. CUN Pagar Reservación.

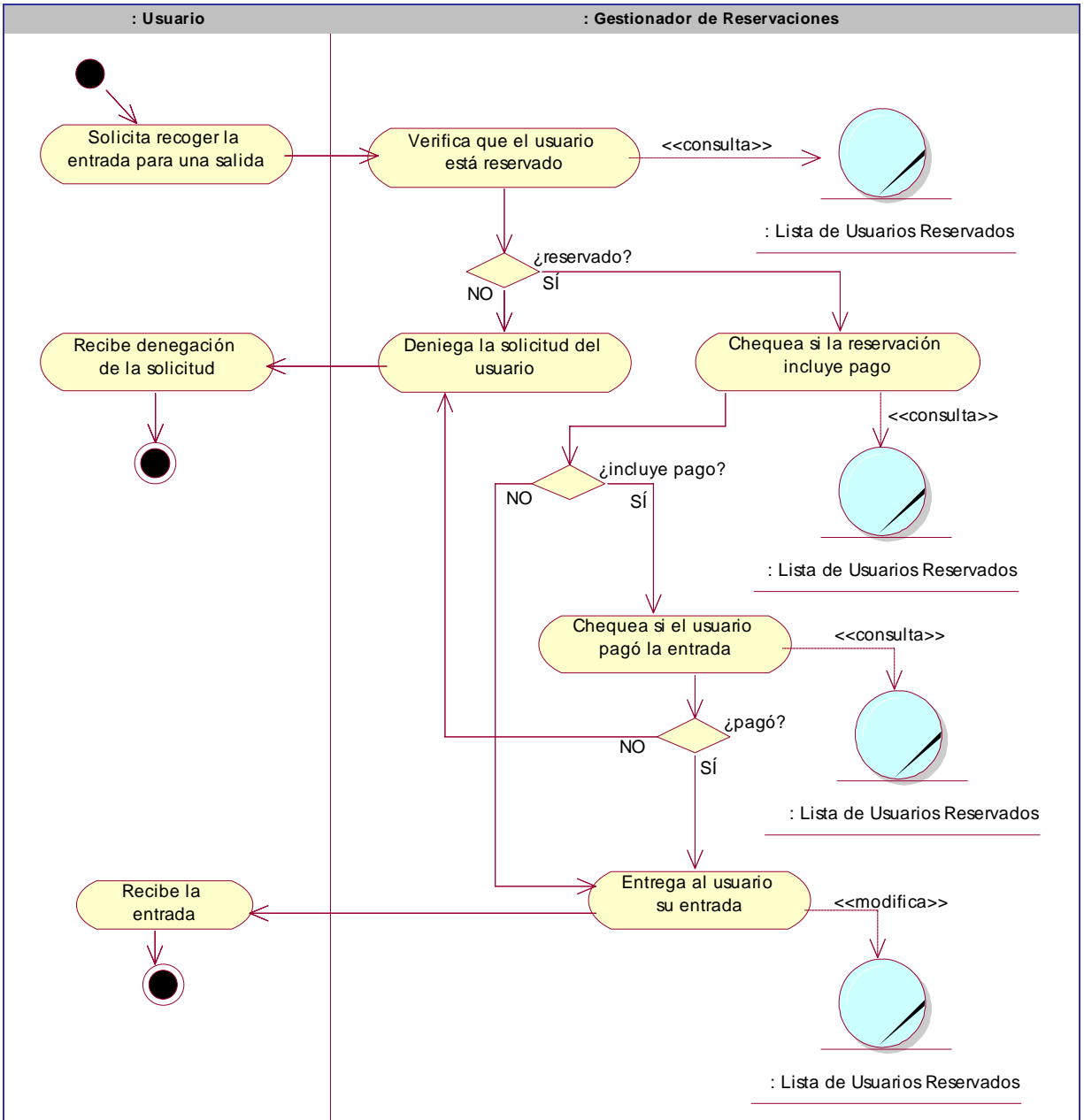


Fig. 6. 11 Diagrama de Actividad. CUN Recoger Entrada.

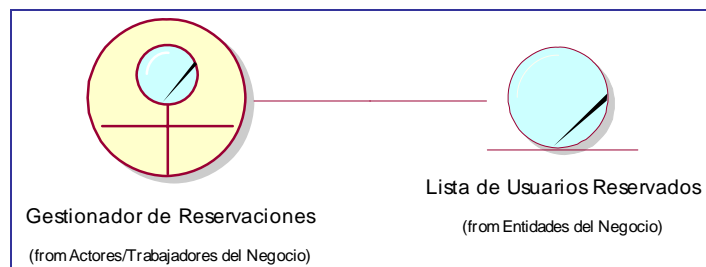


Fig. 6. 12 MON. CUN Recoger Entrada.

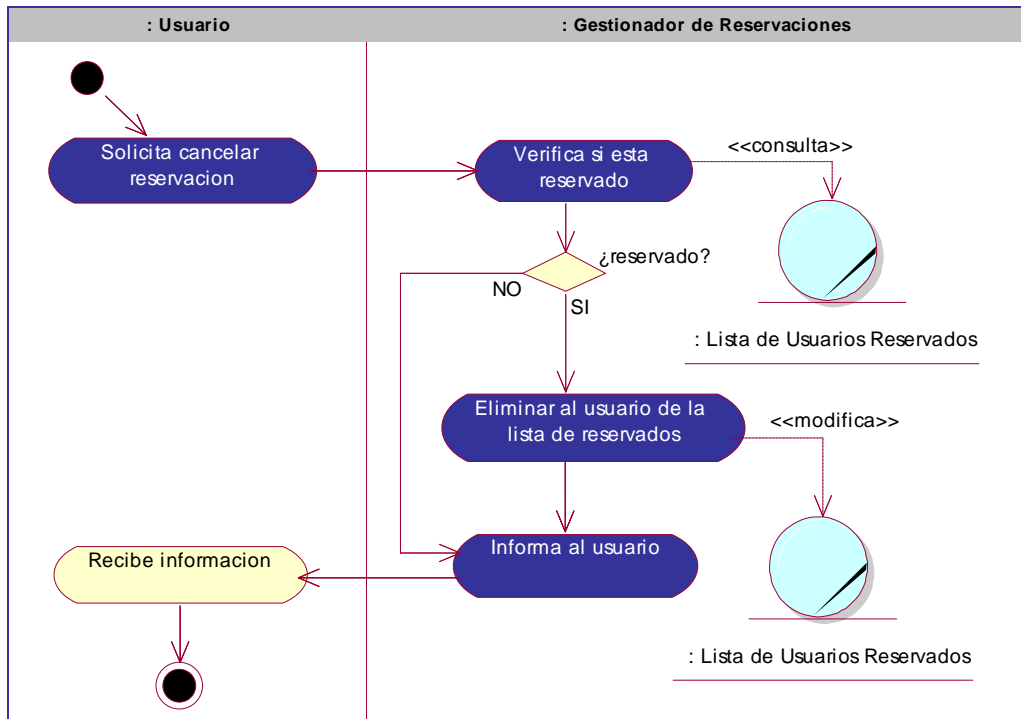


Fig. 6. 13 Diagrama de Actividad. CUN Cancelar Reservación.

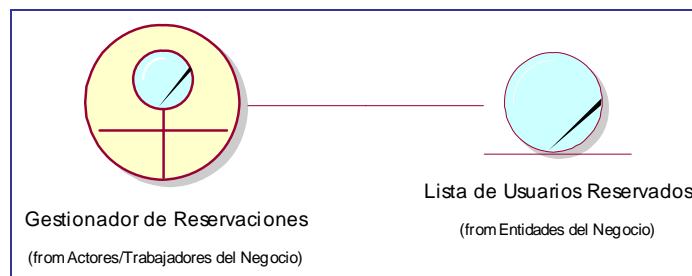


Fig. 6. 14 MON. CUN Cancelar Reservación.

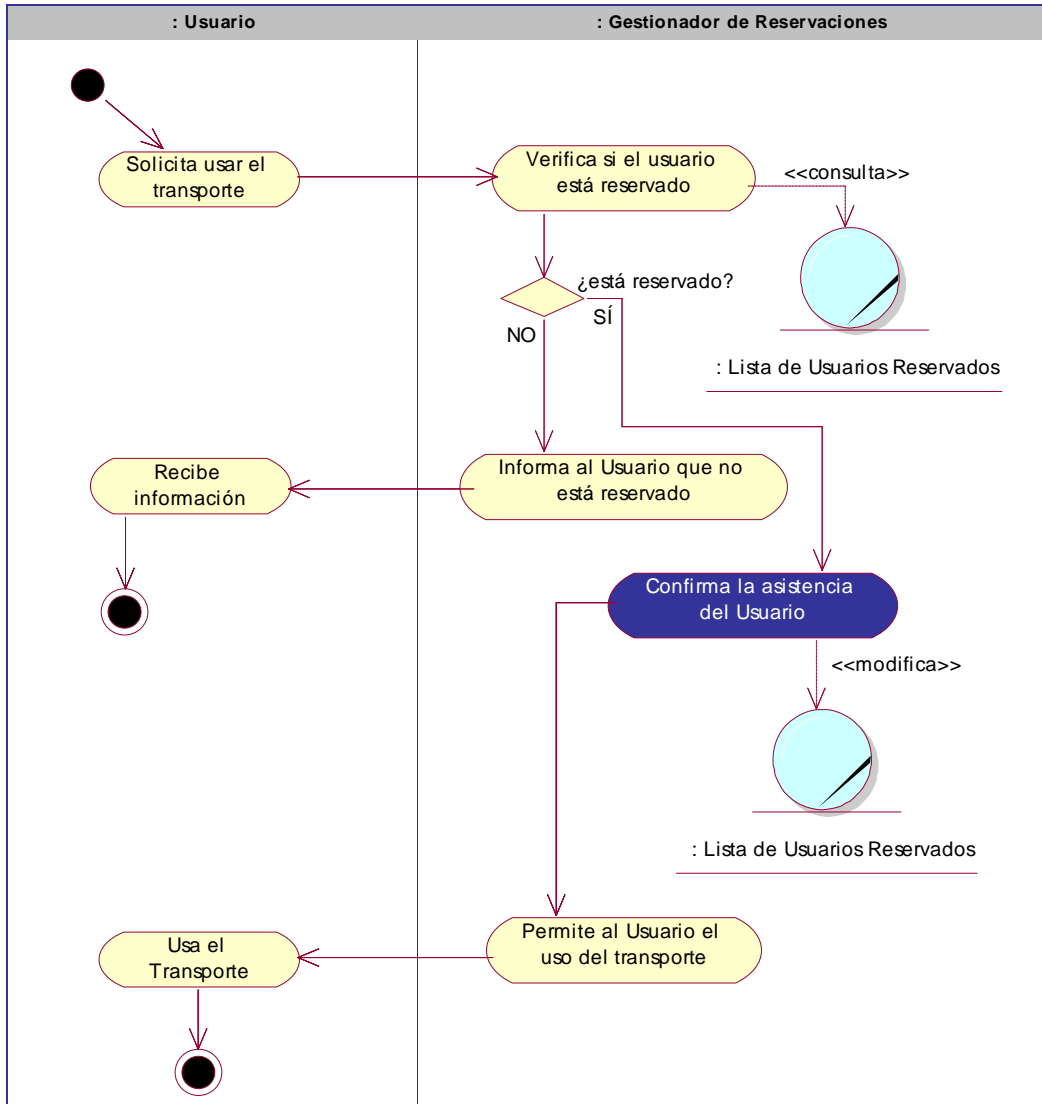


Fig. 6. 15 Diagrama de Actividad. CUN Efectuar Salida.

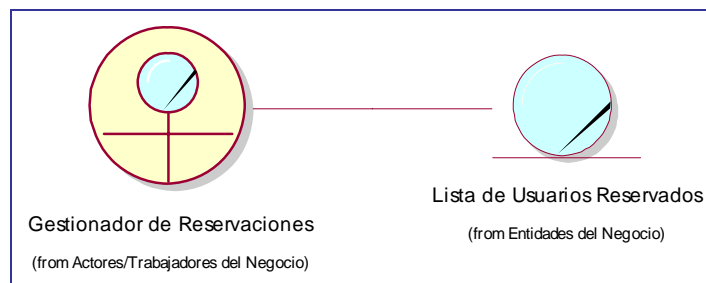


Fig. 6. 16 MON. CUN Efectuar Salida.

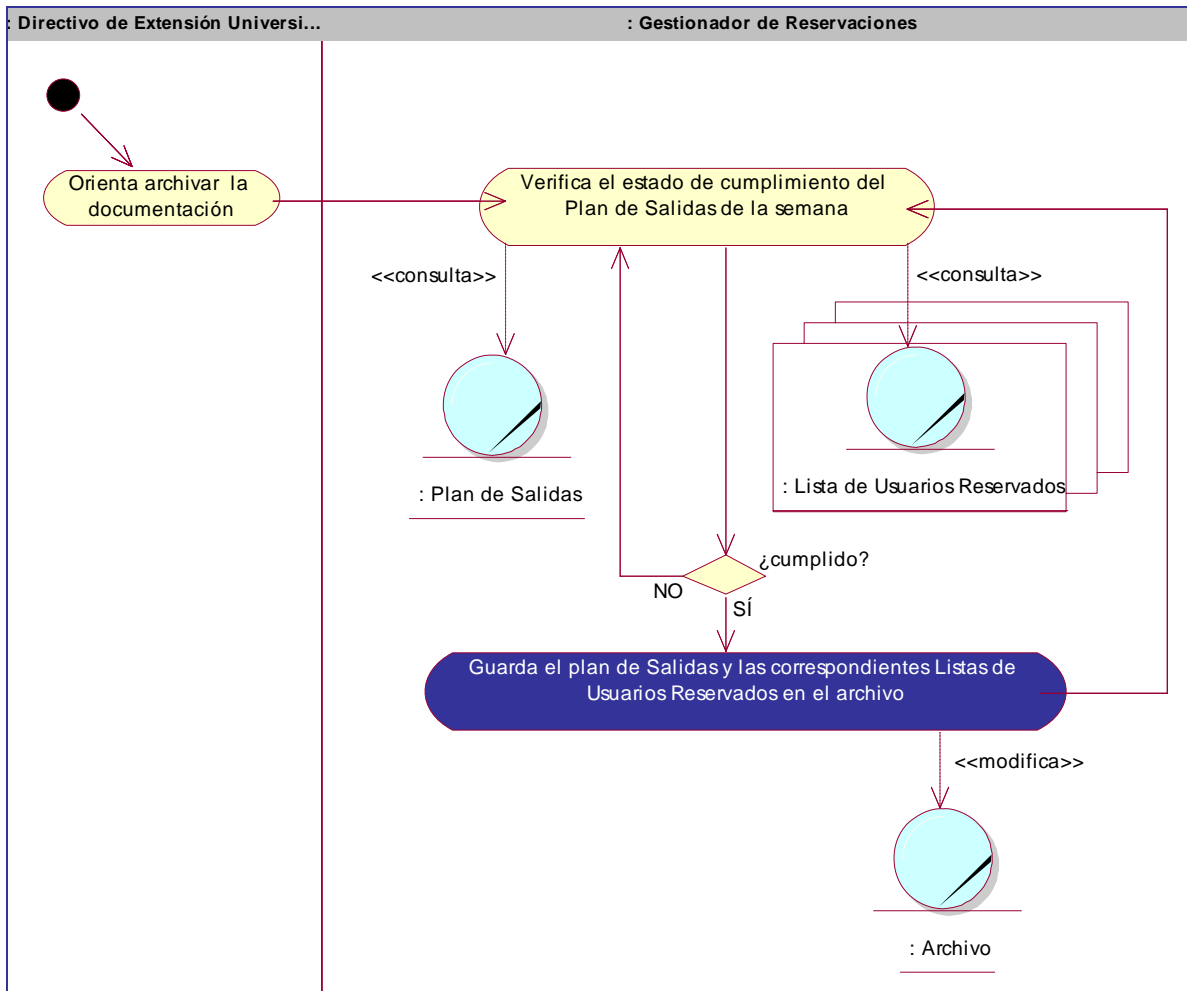


Fig. 6. 17 Diagrama de Actividad. CUN Archivar Información.

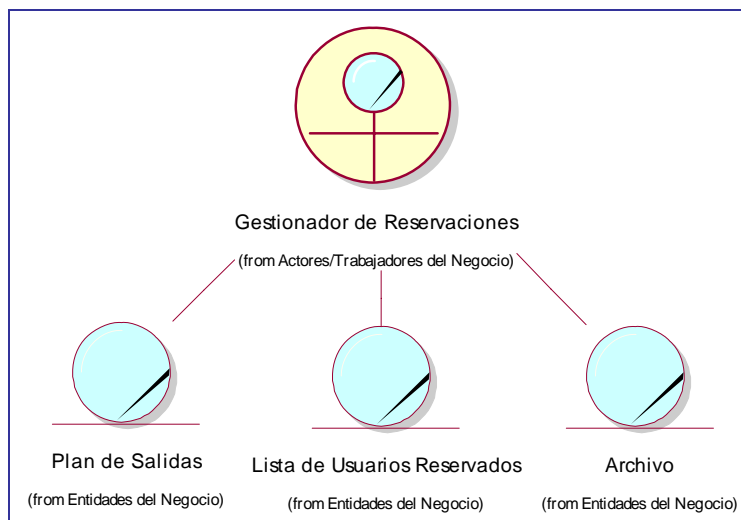


Fig. 6. 18 MON. CUN Archivar Información.

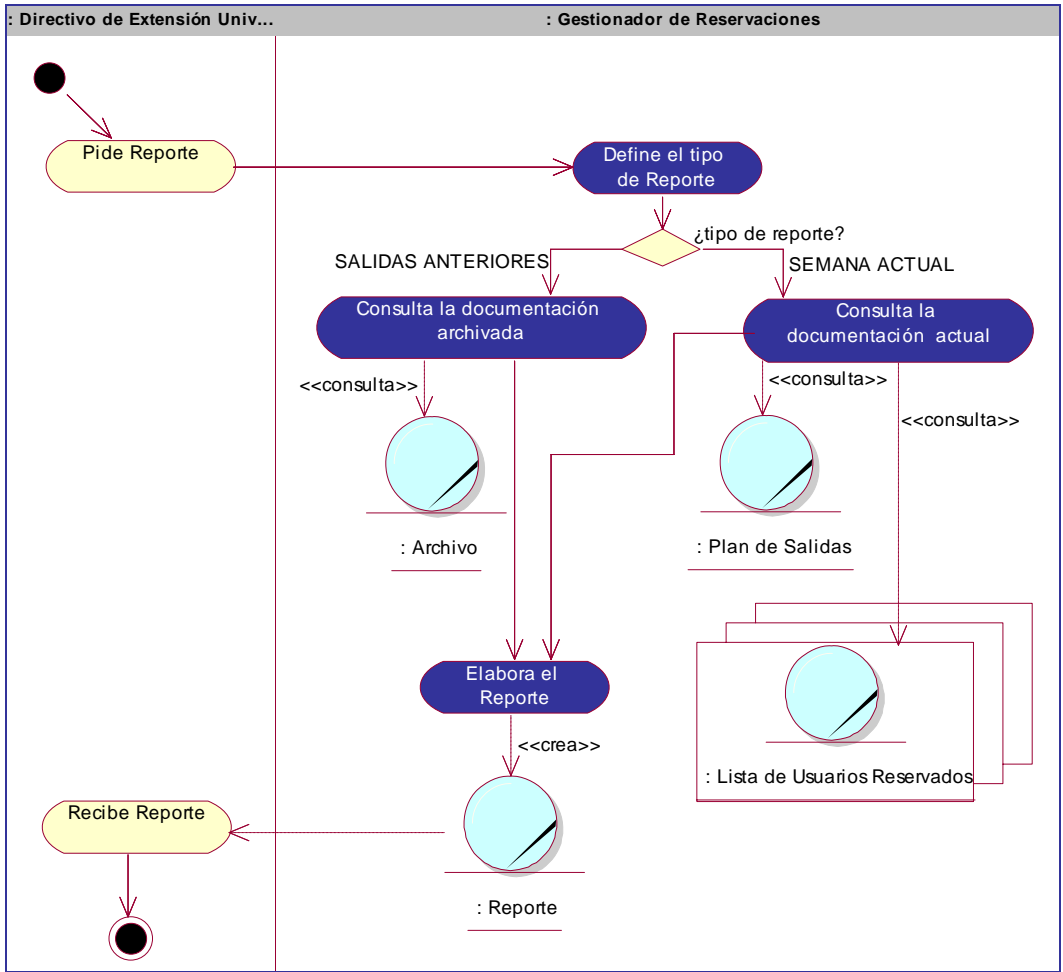


Fig. 6. 19 Diagrama de Actividad. CUN Pedir Reporte.

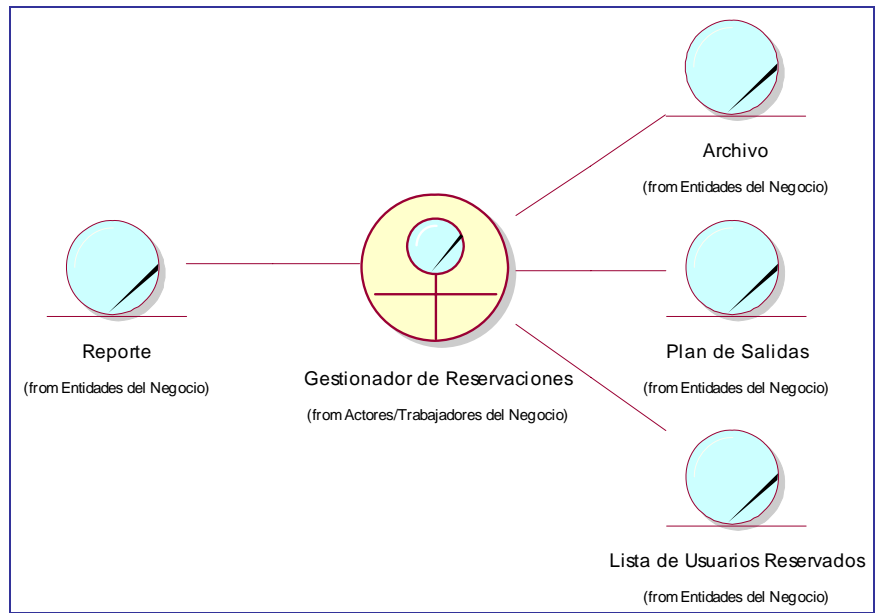


Fig. 6. 20 MON. CUN Pedir Reporte.

Anexo 2. Modelo del Sistema.

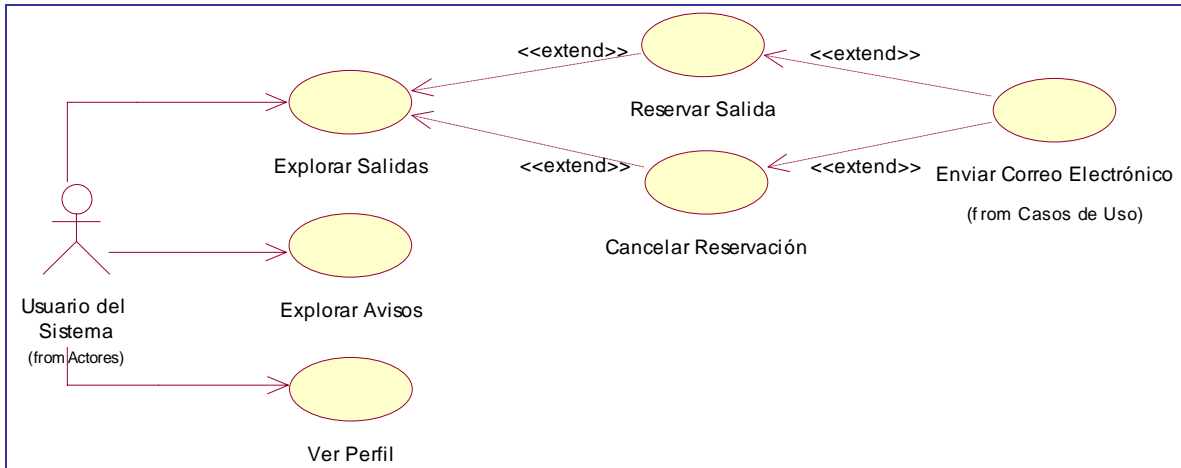


Fig. 6. 21 Diagrama de CUS. Subsistema Reservasiones.

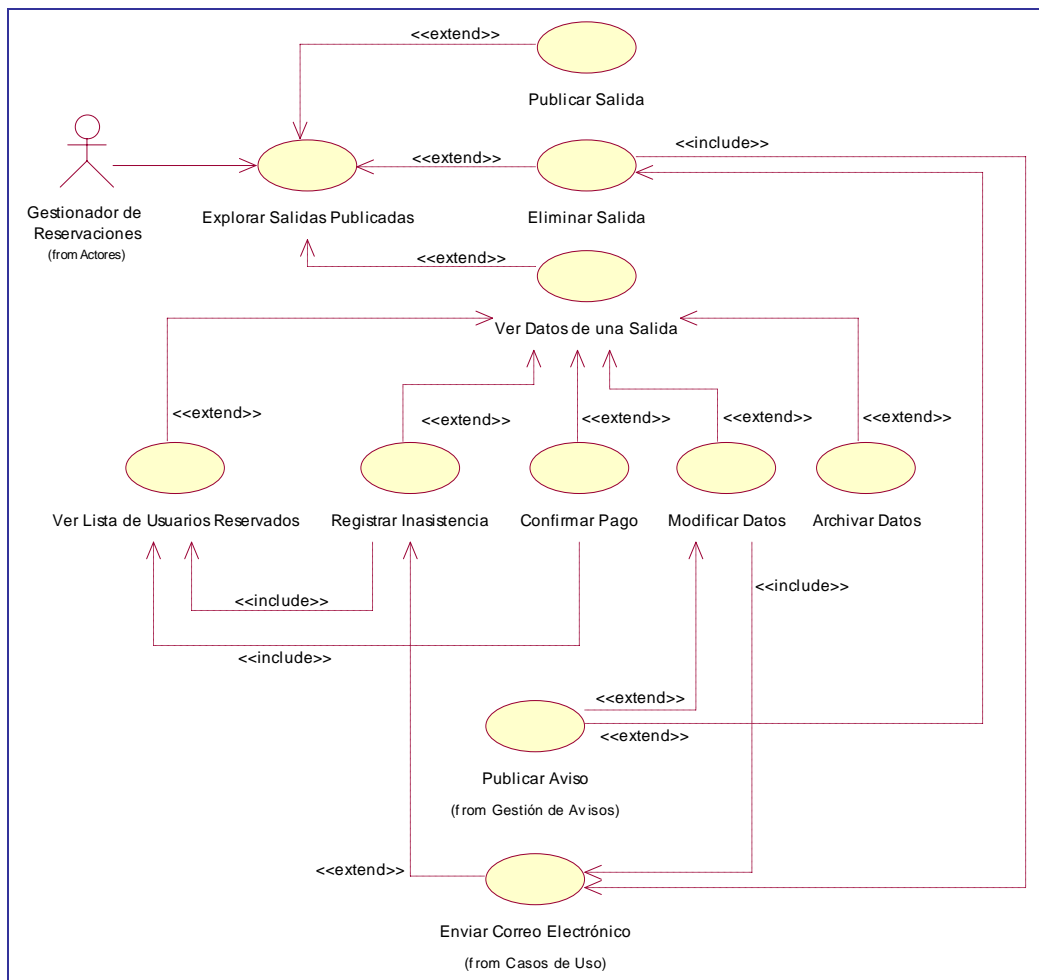


Fig. 6. 22 Diagrama de CUS. Subsistema Gestión de Reservasiones / Gestión de Salidas.

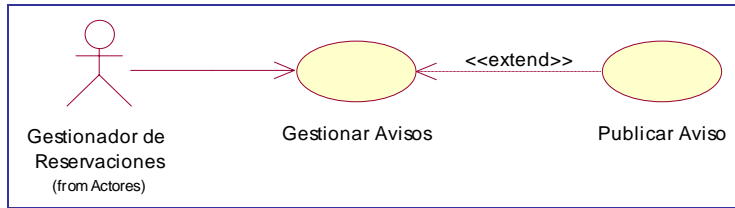


Fig. 6. 23 Diagrama de CUS. Subsistema Gestión de Reservasiones / Gestión de Avisos.

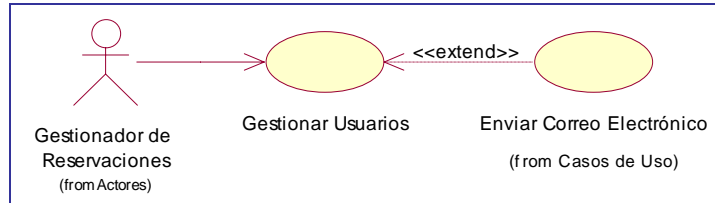


Fig. 6. 24 Diagrama de CUS. Subsistema Gestión de Reservasiones / Gestión de Usuarios.

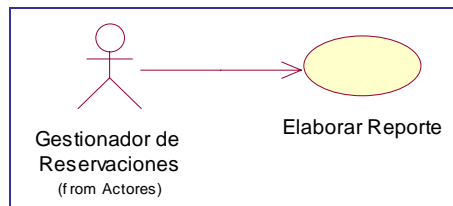


Fig. 6. 25 Diagrama de CUS. Subsistema Gestión de Reservasiones / Gestión de Reportes.

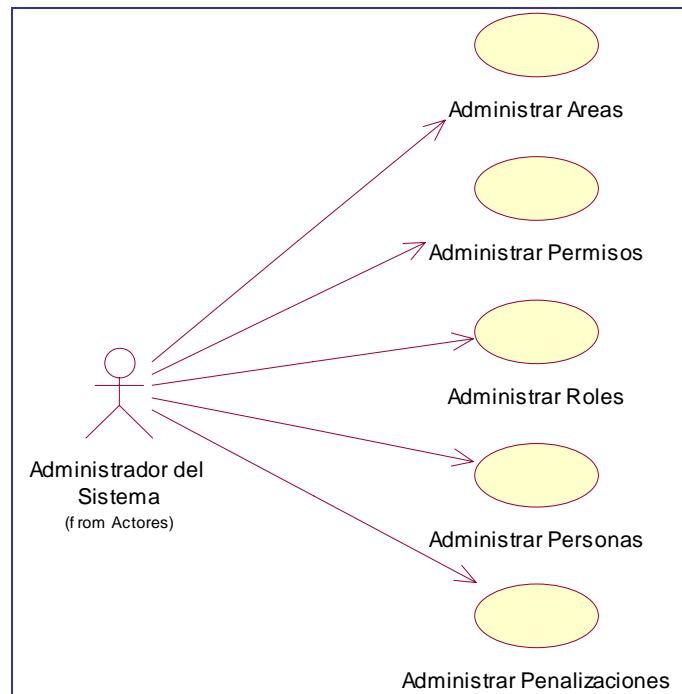


Fig. 6. 26 Diagrama de CUS. Subsistema Administración.

Anexo 3. Diagrama de Clases Persistentes.

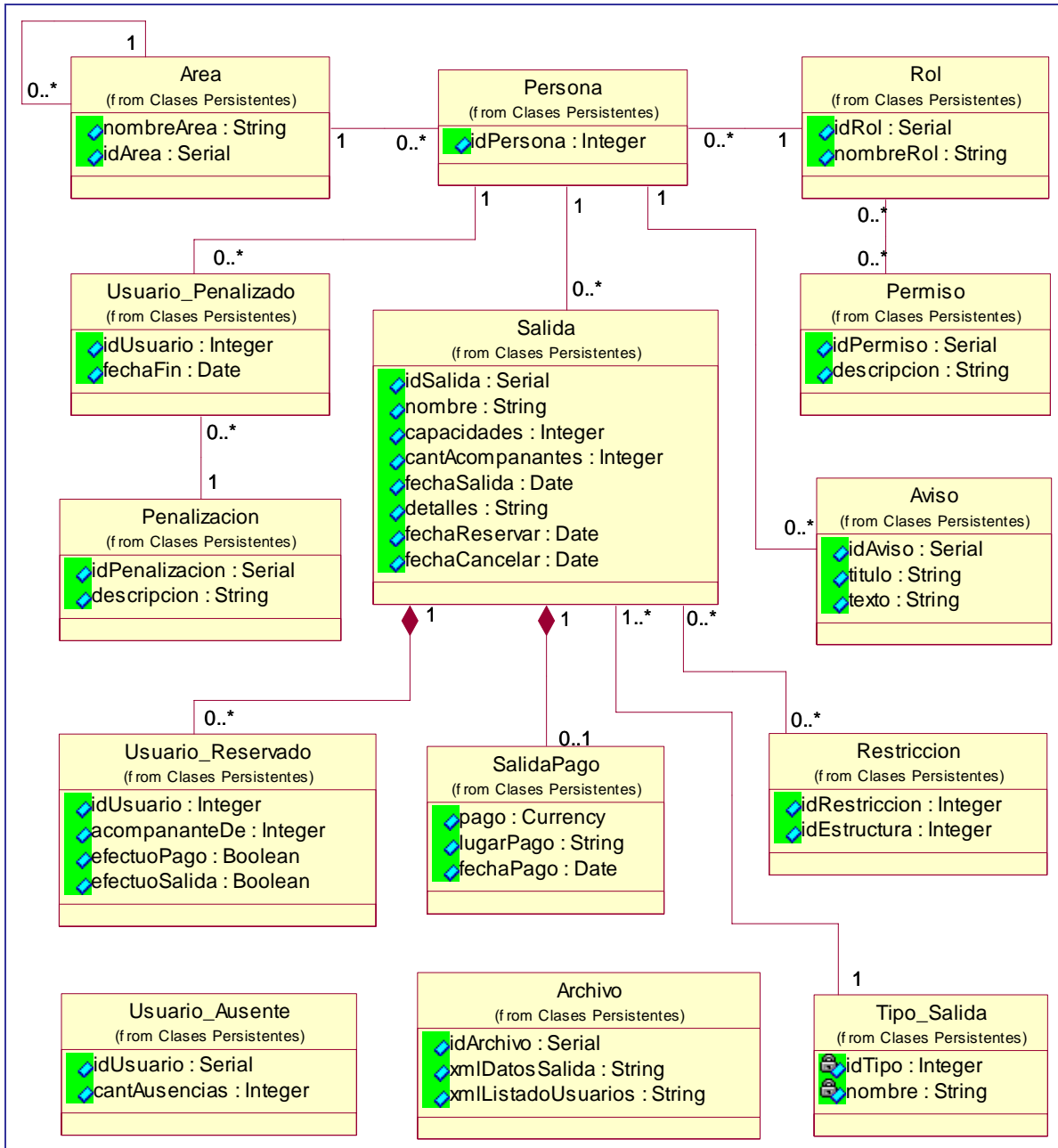


Fig. 6. 27 Diagrama de Clases Persistentes

Anexo 4. Diagramas de Clases Web.

Anexo 5. Modelo de Datos.

Anexo 6. Interfaz de la Aplicación.

Anexo 7. Diagrama de Despliegue.

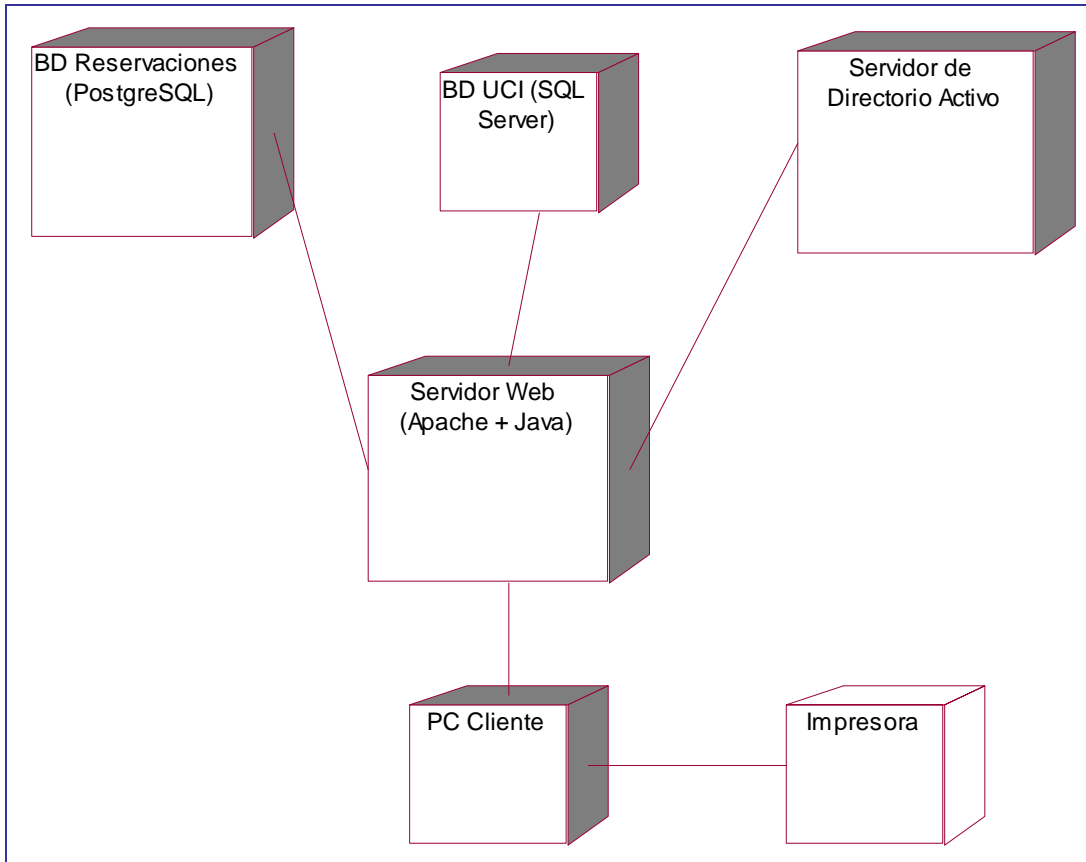


Fig. 6. 28 Diagrama de Despliegue

Anexo 8. Diagrama de Componentes.