

**PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA**

**Sistema de Reservas de Productos Turísticos mediante
Web Services y Dispositivos Móviles**

ERWIN HÉCTOR GUTIÉRREZ RUIZ

**INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA.**

Diciembre 2009

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**Sistema de Reservas de Productos Turísticos mediante
Web Services y Dispositivos Móviles**

ERWIN HÉCTOR GUTIÉRREZ RUIZ

Profesor Guía: **Claudio Cubillos Figueroa**

Profesor Correferente: **Rodrigo Alfaro Arancibia**

Carrera: **Ingeniería de Ejecución en Informática**

**INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA.**

Diciembre 2009

Resumen

La industria del Turismo cada vez va tomando una importancia significativa dentro de la economía, por lo que la demanda de productos turísticos va aumentando de forma paralela; por esto, a la hora de planificar un viaje se hace complicado si éste no se ha hecho con antelación. Por otro lado, los dispositivos móviles han tenido una fuerte evolución tecnológica, y de accesibilidad a las personas. El presente proyecto consiste en un sistema de reservas de productos turísticos que apoye al turista antes y durante su viaje, para la generación de itinerarios a través de Servicios Web y dispositivos móviles.

Palabras claves

Servicios Web, Itinerarios, Dispositivos Móviles, Turismo

Abstract

The tourism industry along time has taken a more relevant significance inside of the economy. Because of that, the demand and cost of tourism products is being increased in a parallel way; because of this, when making a travel plan, the task becomes complicated if it is not made with the proper advance. On the other way, mobile devices have had a strong technological revolution, and the accessibility to persons is one of these. The present project consists in a reservation system of touristic products that support the tourist before and during its travel, for the generation of travel itinerary through web services and mobile devices technology.

Key Words

Web Services, Itineraries, Mobile Devices, Tourism

|

**A mis padres Isaiás y Margarita
por su apoyo incondicional,
como Rosita y Moisés.**

ÍNDICE

1. Introducción	1
2. Objetivos	3
2.1. Objetivo General.....	3
2.2. Objetivos específicos	3
3. Marco Teórico Web Services	4
3.1. Arquitectura de Web Services	4
3.1.1. Roles	4
3.1.2. Pila de Protocolo.....	5
3.2. Servicio de Transporte	6
3.2.1. HTTP	6
3.2.2. FTP	6
3.2.3. SMTP.....	6
3.2.4. BEEP.....	6
3.2.5. JMS	6
3.3. Mensajería XML.....	7
3.3.1. XML	7
3.3.2. REST.....	8
3.3.3. XML-RPC.....	8
3.3.4. SOAP.....	10
3.4. WSDL	13
3.4.1. Especificación de WSDL	13
3.5. UDDI	14
3.5.1. Categorías.....	14
3.5.2. Modelo de datos UDDI.....	15
3.6. Integración de Servicios Web.....	15
3.6.1. Coreografía.....	15
3.6.2. Orquestación.....	15
3.6.3. Lenguajes de composición de servicios	16
3.7. Arquitectura Orientada a Servicios.....	17
3.7.1. Terminología	17

3.8. Uso de Tecnologías en Servicios Web	18
3.8.1. Tecnología .NET	18
3.8.2. JAVA	19
4. Área Turismo	21
4.1. Elementos del Turismo	21
4.2. Clasificación	22
4.3. El turista	22
4.4. La Empresa Turística	23
4.4.1. Tipos de Empresa	23
4.5. El producto turístico.....	23
4.5.1. Los recursos turísticos.....	24
4.5.2. Los atractivos turísticos	24
4.5.3. Planta turística	24
4.5.4. Servicios complementarios	25
4.5.5. Los medios de transporte	25
4.5.6. La infraestructura básica	25
4.6. Efectos del Turismo	26
4.7. Las Tecnologías de Información en el Turismo	27
4.8. Trabajos relacionados	28
4.8.1. Asired e-turismo	28
4.8.2. F.E.T.I.S.H	28
4.8.3. CINNTA.....	28
4.8.4. Eureka Tourism	28
4.8.5. Prototipo multiagente sobre dispositivos móviles para la planificación de viajes turísticos	29
5. Metodología	31
6. Estudio de factibilidad	33
6.1. Factibilidad Técnica.....	33
6.2. Factibilidad Económica.....	33
5.3. Factibilidad legal	34
7. Análisis de Riesgos	35
7.1. Planes de Contingencia	36

7.2. Planes de Mitigación.....	36
8. Solución Propuesta.....	37
8.1. Descripción de la solución propuesta.....	37
8.2. Análisis.....	38
8.3. Descripción de diseño.....	44
8.2.1. Arquitectura propuesta.....	58
8.3. Elección de Implementación.....	59
8.3.1. Elección del IDE.....	59
8.3.2. Parser XML.....	60
9. Plan de pruebas.....	61
9.1. Planificación de las pruebas.....	61
10. Detalles de la solución.....	65
10.1. Sistema Coordinador.....	65
10.1.1. Invocación Dinámica de Servicios.....	66
10.1.2. Invocación Asíncrona de Servicios.....	68
10.2. Aplicación cliente.....	70
10.3. Registro de Servicios.....	74
10.4. Modelo de datos.....	76
10.5. Descripción de Servicios,.....	78
11. Pruebas de Software.....	81
12. Conclusiones.....	84
13. Referencias.....	85
Anexo A.....	86
A.1 Sistema Coordinador.....	86
A.2 Servicio Web Hotel.....	94
A.3 Servicio Web Rent a Car.....	95
A.4 Servicio City Tour.....	97

ÍNDICE DE TABLAS

6.1. Costos de Software	33
6.2. Costos de Hardware.....	34
7.1. Análisis de Riesgos	35
7.2. Planes de Contingencia.....	36
7.3. Planes de Mitigación	36
9.1. Análisis de Límites.....	63

ÍNDICE DE ILUSTRACIONES

3.1. Modelo básico del funcionamiento de un Servicio Web	5
3.2. Elementos principales de un mensaje SOAP	11
3.3. Elementos de WSDL	14
3.4. Terminología SOA	17
5.1. Fases e interacciones de metodología RUP	31
8.1. Sistema basado en Web Services	37
8.2. Diagrama de casos de uso alto nivel	39
8.3. Diagrama de casos de gestionar perfil	43
8.4. Diagrama de casos de uso gestionar itinerario	43
8.5. Diagrama de casos de uso gestionar servicios	44
8.6. Diagrama de diseño de clases Sistema Planificador.....	45
8.7. Diagrama de diseño de clases Sistema Planificador(B).....	46
8.8. Clase Web Service sistema planificador	47
8.9. Diagrama de diseño de clases Sistema Hotel.....	48
8.10. Diagrama de diseño de clases Sistema Rent a Car	49
8.11. Diagrama de secuencias crear itinerario(A).....	50
8.12. Diagrama de secuencias crear itinerario(B).....	52
8.13. Diagrama de secuencias eliminar reserva	53
8.14. Diagrama de secuencias crear perfil.....	54
8.15. Diagrama de secuencias registrar servicio	55
8.16. Diagrama de actividad crear itinerario.....	56
8.17. Diagrama de actividad cancelar reserva	56
8.18. Diagrama de actividad ver modificaciones.....	57
8.19. Diagrama de actividad crear perfil	57
8.20. Diagrama de despliegue.....	59
9.1. Tester de Servicio Web planificador.....	65
9.2. Interfaz de usuario aplicación móvil menú principal	70
9.3. Interfaz de usuario aplicación móvil formulario búsqueda de servicios.....	70
9.4. Interfaz de usuario aplicación móvil listado producto.....	71
9.4. Interfaz de usuario aplicación móvil itinerarios	72
9.4. Interfaz de usuario aplicación móvil reservas	72

9.4. Interfaz de usuario aplicación móvil modificaciones	73
9.5. Interfaz de página web.....	74
9.5. Interfaz de página web para descarga de aplicación móvil.....	74
9.7. Modelo de datos sistema planificador	75
9.8. Modelo de datos servicio web hotel	76
9.9. Modelo de datos servicio web rent a car.....	76
9.10. Modelo de datos servicio web city tour	77

1. Introducción

La industria del Turismo es una de las más importantes del mundo, por la cantidad de elementos que la componen, además de involucrar distintos sectores económicos. La inclusión de las tecnologías de la información en la actividad turística ha hecho que la industria turística tome un papel fundamental en la globalización, además de alcanzar un desarrollo como industria; donde existe una gran variedad de sistemas informáticos para el apoyo a los turistas como a los proveedores de servicios, además de la expansión a nuevos mercados. De éstas tecnologías de la información, Internet es la más importante y utilizada por la industria Turística, donde se puede acceder a una cantidad de información relevante las 24 horas, 365 días y alrededor de todo el mundo. Sin embargo, la mayoría de esta información no representa el estado de productos y servicios turísticos en tiempo real. Una nueva forma de ver el mercado turístico es a través de la incorporación de tecnologías móviles, donde los dispositivos móviles están tomando un rol importante debido a la creciente evolución tecnológica, y la accesibilidad a los usuarios.

El presente proyecto consiste en desarrollar un Sistema basado Servicios Web que apoye al turista a crear un itinerario de viaje a su medida, donde el turista podrá consultar precio, disponibilidad, y reservar los servicios ofrecidos por los proveedores de productos turísticos para crear su propio paquete turístico de acuerdo a sus preferencias, además de poder modificar su itinerario. Por el lado de los proveedores de servicios turísticos podrán cambiar la información de los servicios, tales que estas modificaciones sean notificadas al turista. Para ello, se implementará una lógica del negocio que orqueste los servicios ofrecidos por los proveedores de productos turísticos, es decir, para que refleje como si se tratara de un solo servicio, donde el turista ante el sistema podrá ver la disponibilidad, precios y reservar los servicios de acuerdo a su presupuesto, localización y el tiempo que durará su viaje, además de acceder a través de un PC y/o dispositivos móviles, como teléfono celular o una PDA.

Este informe presenta al lector los temas que involucran tanto el problema como la solución, donde se presentará el marco teórico de los Servicios Web y sector del Turismo, además de presentar un plan de trabajo esperado para el desarrollo del proyecto, análisis y diseño. Además de dar una descripción detallada de la solución final desarrollada.

Se estructura el informe de la siguiente forma:

- En el Capítulo 2, se presentan los objetivos generales y específicos del proyecto.
- En el Capítulo 3, en este capítulo se abordará el marco teórico correspondiente a los servicios web, donde se describe su arquitectura, y los protocolos involucrados como mensajería, transporte, descripción y descubrimiento. Además se presentan el uso de tecnologías con servicios web, como el uso de Java y .Net.
- En el Capítulo 4, en este capítulo se presenta el marco teórico correspondiente al área del turismo. Se describe el producto turístico, los tipos de turista, las empresas turísticas y la relación de la industria turística con el use de tecnologías de información.
- En el Capítulo 5 cubre la metodología a utilizar en el proyecto.
- En el Capítulo 6, se presenta el estudio de factibilidad realizado, tanto la factibilidad económica, técnica y legal.
- En el Capítulo 7 se muestra el análisis de riesgos realizados de acuerdo al proyecto.
- En el Capítulo 8 se presenta una descripción de la solución a implementar, además de la presentación del análisis y diseño del sistema. Además contempla la elección de herramientas de desarrollo para implementación del sistema.
- En el Capítulo 9 se muestra el plan de pruebas.
- En el Capítulo 10 se presenta los detalles de la aplicación desarrollada con las interfaces de la aplicación cliente, y la comunicación con web services.
- En el Capítulo 11 se presenta algunos de los test de software realizado.
- Capítulo 12 corresponde a las conclusiones del informe.
- En el Anexo A se detallan lo servicios web correspondientes a los desarrollados para el funcionamiento del sistema. Estos detalles contemplan las descripciones de cada servicio, además de los formatos de datos de entrada.

2. Objetivos

2.1. Objetivo General

El objetivo de este proyecto es desarrollar un sistema de planificación de viajes turísticos a través de Web Services que ayude al turista a generar un itinerario de viaje a su medida.

2.2. Objetivos específicos

- Estudiar los protocolos que conforman a un Servicio Web.
- Estudiar la forma en que la lógica del negocio usará los Servicios Web disponibles para la solución del problema.
- Diseñar los Servicios Web que sean acorde a la solución del problema.
- Validar el sistema diseñado verificando que su especificación cumpla con los requerimientos de la metodología empleada y el prototipo a obtener durante el desarrollo del sistema.

3. Marco Teórico Web Services

Los Servicios Web son mecanismos de comunicación distribuida que permiten que las aplicaciones compartan información y puedan invocar funciones de otras aplicaciones, independientemente de cómo éstas hayan sido creadas, qué dispositivos son necesarios para acceder a ellas, y cuál sea su sistema operativo o la plataforma en las que se ejecutan. La W3C¹ define a los Servicios Web como “un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Este tiene una interfaz descrita en un formato procesable por una máquina (WSDL). Otros sistemas interactúan con el Servicio Web en una manera prescrita por su descripción usando mensajes SOAP, típicamente enviados usando HTTP con una socialización XML en relación con otros estándares relacionados con la Web”.

3.1. Arquitectura de Web Services

Hay dos formas de ver una arquitectura de Web Services. La primera es ver desde el punto de los roles individuales de cada actor de los Web Services; y la segunda es examinar la pila de protocolos de Web Services [1].

3.1.1. Roles

Hay tres roles importantes en la arquitectura de los Servicios Web.

- **Proveedor de Servicio:** El proveedor de servicio implementa el servicio y lo hace disponible en Internet.
- **Solicitador de Servicio:** Es cualquier consumidor de un Servicio Web. Un solicitante utiliza un Servicio Web existente para abrir una conexión de red y enviar una solicitud XML.
- **Registro de Servicio:** El registro de servicio proporciona un lugar central donde los desarrolladores pueden publicar nuevos servicios o encontrar servicios existentes. Por lo tanto, sirve como centro de un servicio centralizado para las empresas y sus servicios.

¹ <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>

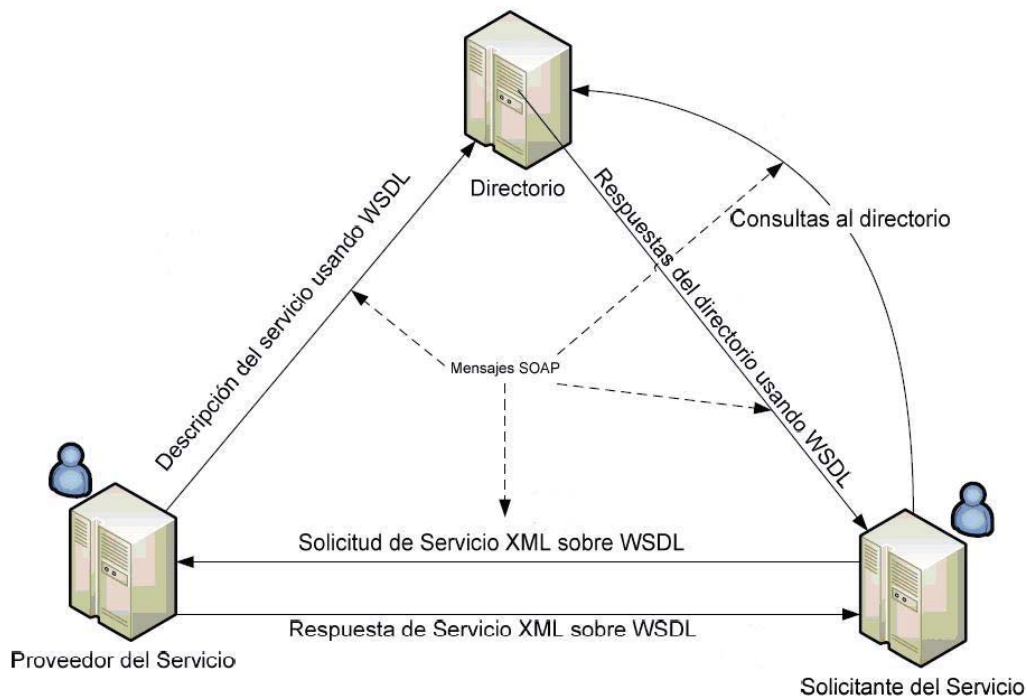


Figura 3.1. Modelo básico del funcionamiento de los Servicios Web

3.1.2. Pila de Protocolo

Una segunda forma de ver la arquitectura de los Web Services es ver la pila de protocolo; de ésta, cuatro capas son las más importantes. A continuación se entrega una breve descripción de ellas.

- **Servicio de Transporte:** El servicio de transporte está encargado de transportar mensajes entre aplicaciones sobre la red. El cual incluye varios protocolos a nivel de aplicación.
- **Mensajería XML:** Esta capa es la encargada de codificar mensajes en un formato XML común, tal que estos mensajes puedan ser entendidos entre los nodos de la red.
- **Descripción de Servicio:** La descripción del servicio es la capa responsable de describir la interfaz pública para especificar los Servicios Web. Actualmente la descripción de servicios es manejada vía WSDL.
- **Descubrimiento del servicio:** Esta capa es la encargada de centralizar los servicios en un registro común, además de ofrecer una funcionalidad de encontrar y publicar servicios fácilmente.

3.2. Servicio de Transporte

A continuación se nombrarán los protocolos de transporte más utilizados en los Servicios Web.

3.2.1. HTTP (HyperText Transfer Protocol)

Es el protocolo más utilizado en Internet. En cuanto a los Servicios Web es utilizado para la transferencia de las transacciones XML a través de la red, utilizando los mismos principios de HTML.

3.2.2. FTP (File Transfer Protocol)

Este protocolo en el ámbito de Servicios Web permite realizar modificaciones en equipos remotos evitando el uso de permisos sobre los archivos en la maquina cliente.

3.2.3. SMTP (Simple Mail Transfer Protocol)

El protocolo SMTP es un estándar de la capa de aplicación utilizado ampliamente para el envío de mensajes de correo electrónico a través de Internet.

3.2.4. BEEP (Block Extensible Exchange protocol)

Este protocolo es también conocido como BXXP, el que está diseñado para la interacción asíncrona punto a punto sobre una red TCP/IP. BEEP fue estandarizado por el IETF (Internet Engineering Task Force) y provee un marco para la administración punto a punto, transporte de mensajes, autenticación y manejo de errores.

3.2.5. JMS (Java Message Service)

Es una API de java para el envío de mensajes entre 2 o más clientes. JMS soporta 2 modelos, el modelo punto a punto y el modelo de publicación y suscripción. Una aplicación JMS está compuesta de las siguientes partes²:

- Clientes JMS que son componentes escritos en JAVA que producen y consumen los mensajes.
- Proveedor JMS, el cual implementa las interfaces que proveen las características de administración y el control.
- Los Mensajes que son los objetos dato entre clientes JMS.
- Objetos Administradores que son configurados previamente por un administrador del sistema para el uso de los clientes.
- Clientes Nativos que son programas que usan los mensajes de la API.

² http://java.sun.com/products/jms/tutorial/1_3_1-fcs/doc/basics.html

3.3. Mensajería XML

3.3.1. XML

XML (Extensible Markup Language) es un metalenguaje extensible de etiquetas que fue diseñado básicamente para estructurar, almacenar y para intercambiar datos entre diferentes aplicaciones. Se convirtió en un estándar, ya que es extensible y puede ser utilizado por cualquier aplicación independientemente de la plataforma. Además de permitir definir la gramática de lenguajes específicos, por ejemplo, XHTML, SVG, MathML. [1].

3.3.1.1. Partes de un documento XML

1. **Prólogo:** El prólogo es un elemento opcional, los documentos XML pueden empezar con unas líneas que describen la versión XML, el tipo de documento, entre otras cosas. El prólogo contiene:
 - Una declaración XML. Es la sentencia que declara al documento como un documento XML.
 - Una declaración de tipo de documento. Enlaza el documento con su DTD (definición de tipo de documento), o el DTD puede estar incluido en la propia declaración o ambas cosas al mismo tiempo.
 - instrucciones de procesamiento.
2. **Cuerpo:** A diferencia del prólogo, el cuerpo no es opcional en un documento XML, el cual debe contener un y solo un elemento raíz.
3. **Elementos:** Los elementos XML pueden tener contenido o bien ser elementos vacíos. Los contenidos de los elementos XML pueden ser caracteres, elementos XML, o ambos.
4. **Atributos:** Los elementos pueden tener atributos, que son una manera de incorporar características o propiedades a los elementos de un documento. Deben ir entre comillas.

3.3.1.2. DTD (Document type definition)

La DTD define los tipos de elementos, atributos y entidades permitidas, y puede expresar algunas limitaciones para combinarlos. Los documentos XML que se ajustan a su DTD son denominados válidos.

3.3.1.3. XSD (XML Schema Definition)

Un Schema es algo similar a un DTD. Define qué elementos puede contener un documento XML, cómo están organizados y qué atributos y de qué tipo pueden tener sus elementos.

Ventajas de los Schemas frente a los DTDs:

- Usan sintaxis de XML, al contrario que los DTDs.
- Permiten especificar los tipos de datos.
- Son extensibles.

3.3.2. REST (Representational State Transfer)

REST es una agrupación de principios para el diseño de redes, el cual es usado para definir una interfaz de transmisión sobre HTTP similar como lo hace el protocolo SOAP; aunque REST en sí no es un estándar, posee un conjunto de estándares como URL, HTML, GIF, JPG, XML y MIME. El conjunto de principios que conforman REST son los siguientes:

- Generalidad de Interfaces.
- La escalabilidad en la interoperabilidad con los componentes.
- Puesta en funcionamiento independiente.
- Compatibilidad con componentes intermedios.

3.3.3. XML-RPC

El protocolo XML-RPC permite a los programas hacer funciones o llamadas a procedimientos a través de la red. XML-RPC utiliza el protocolo de transporte HTTP para pasar la información de un computa cliente a un computador servidor, describiendo la naturaleza de las peticiones y respuestas en un pequeño vocabulario XML. Los clientes especifican los nombres de los procedimientos y parámetros en una petición XML (XML request), y los servidores devuelven tanto un fallo o la respuesta en una respuesta XML (XML response).

3.3.3.1. Descripción general Técnica.

XML-RPC consiste en tres partes:

1. **Modelo de datos XML-RPC:** Es un conjunto de tipos para el uso de paso de parámetros, retorno de valores y mensajes de error.
2. **Estructuras de peticiones XML-RPC:** Una petición HTTP POST, que contiene el método y los parámetros de información.
3. **Estructuras de respuesta XML-RPC:** Una respuesta HTTP que contiene los valores de retorno y la información de fallo.

La combinación de estas tres partes define una llamada de procedimiento remoto.

- **Modelo de datos XML-RPC**

El modelo de datos define 6 tipos de datos básicos y 2 tipos de datos compuestos que representan la combinación de tipos.

1. **Datos Básicos:** Los tipos de datos básicos se dividen en int o i4, doublé, boolean, string, Base64, DateTime.iso8601. Estos datos de tipo básico siempre están encerrados en etiquetas (elementos XML) de nombre "value". Solo los string (y solo ellos) pueden omitir la etiqueta "string" dentro de las etiquetas "value".
2. **Datos Compuestos:** Los tipos básicos pueden ser combinados en 2 tipos complejos, en array y estructuras. Los array son indicados por las etiquetas XML "array", dentro de estas, se encuentran las etiquetas "data", las que contiene la lista de valores que forman el array. También los array, como los tipos básicos pueden estar encerrados dentro de las etiquetas "value". Las estructuras tienen su contenido desordenado, identificado por el nombre. Cada elemento "struct" contiene una de elementos (etiquetas XML) "member". Cada elemento "member", tiene un elemento "name" y uno "value".

▪ Estructura de petición XML-RPC

Las peticiones en este protocolo son una combinación de contenido XML y cabeceras HTTP. El contenido XML usa la estructura de datos a escribir para pasar parámetros y contenidos adicionales, identificando los procedimientos que serán invocados. Mientras las cabeceras HTTP proveen una envoltura para pasar la petición en la red.

Cada petición es un documento XML, donde el elemento raíz del documento es la etiqueta "methodCall". Cada elemento "methodCall" contiene el nombre de la función (etiqueta "methodName") y los parámetros del método (etiqueta "params").

Las cabeceras HTTP para estas peticiones tiene la siguiente estructura:

```
POST /target HTTP 1.0
User-Agent: Identificador
Host: host.making.request
Content-Type: text/xml
Content-Length: largo de la petición en bytes
```

Para el ejemplo de la petición del método areaCirculo, la estructura puede darse de la siguiente forma:

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
<?xml version="1.0"?>
<methodCall>
<methodName> areaCirculo </methodName>
<params>
<param>
<value><double>2.41</double></value>
</param>
</params>
</methodCall>
```

▪ Estructura de respuesta XML-RPC

Las respuestas son similares a las peticiones. Si la respuesta es satisfactoria, es decir, si la el procedimiento fue encontrado, ejecutado correctamente y retornó los valores, la respuesta será parecida a la de petición, excepto que las etiquetas `methodCall` son reemplazadas por las etiquetas `methodResponse`, y estas no tiene las etiquetas `methodName`.

Las respuestas en este protocolo, solo pueden tener un solo parámetro, el cual está encerrado en la etiqueta `params`. Este parámetro, puede ser un array o una estructura, si el método devolviese varios valores.

Si llegase a ocurrir alguna falla en el procesamiento del método, el elemento `methodResponse` tendrá una etiqueta XML de nombre `fault`. El elemento `fault` tanto como el elemento `params` (si el procesamiento del método es correcto) debe tener solo un valor. Una respuesta fallida podría darse de la siguiente forma:

```
<?xml version="1.0"?>
<methodResponse>
<fault>
<value><string> No existe el método! </string></value>
</fault>
</methodResponse>
```

3.3.4. SOAP

SOAP es un protocolo basado en XML para el intercambio de información entre computadores. Aunque SOAP puede ser usado para una variedad de sistema de envío y ser emitido por una variedad de protocolos, el foco inicial de SOAP es la llamada a procedimientos de forma remota vía HTTP [1]. Por lo tanto SOAP permite aplicaciones clientes para una fácil conexión a los servicios remotos y la invocación de métodos remotos.

Ventajas de SOAP

- SOAP no está ligado a ningún lenguaje de programación, ya que no especifica una API. Por lo que los desarrolladores pueden elegir el lenguaje de programación como JAVA o .NET por ejemplo.
- La especificación de SOAP no describe como se deberían asociar los mensajes con HTTP. Como un mensaje SOAP no es más que un documento XML, este puede ser transportado por cualquier protocolo de transporte que se capaz de enviar texto.
- Aprovecha los estándares existentes. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema. Por otro lado, como se mencionó anteriormente, es independiente al protocolo de transporte, por lo que un mensaje SOAP puede ser enviado tanto por el protocolo HTTP o SMTP por ejemplo.
- Las aplicaciones que se ejecuten con en plataformas con los estándares compatibles con SOAP pueden comunicarse con mensajes SOAP con otras aplicaciones que se ejecuten en otras plataformas, es decir, permite la interoperabilidad entre múltiples entornos.

3.3.4.1. El mensaje SOAP

Un mensaje unidireccional como una petición de un cliente a un servidor, o una respuesta de un servidor a un cliente es considerado un mensaje SOAP. Todos los mensajes SOAP tiene dos elementos (etiquetas XML) obligatorios, como los son los elementos *envelope* y *body*; además de contar con un elemento opcional llamado *header*. Estos elementos están asociados a un conjunto de reglas.

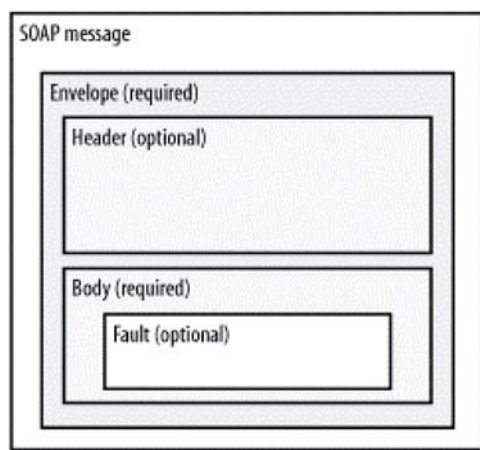


Figura 3.2. Elementos principales de un mensaje SOAP

- **Envoltura:** Todo mensaje SOAP tiene un elemento raíz llamado *envelope* (envoltura), el que contiene los elementos *header*, *body* y *foult*. En contraste con otras especificaciones, tales como HTTP y XML, SOAP no define un modelo tradicional de versiones sobre la base (por ejemplo, HTTP 1.0 versus HTTP 1.1). Por el contrario, SOAP utiliza espacios de nombre XML (namespaces) para diferenciar las versiones. La versión debe ser referenciada en el elemento *envelope*. Por ejemplo:

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

- **Header** (cabecera): El elemento opcional *header* ofrece un marco de trabajo flexible para una especificación adicional de requerimientos en el nivel de aplicación. Por ejemplo este elemento puede ser usado para especificar una firma digital para servicios de contraseña protegida. Además de proveer administración de transacciones y autorización de pago, por ejemplo, para especificar un número de cuenta para un servicio de pago.

- **Body** (cuerpo): El elemento Body es obligatorio para todos los mensajes SOAP. Este elemento contiene el mensaje propiamente tal, almacena el documento XML que el receptor final procesará.
- **Fault**: En caso de error, el elemento body incluirá al elemento fault, el que contiene la información de la naturaleza del error.
- **Petición SOAP** : Una petición de cliente debe contener el nombre del método y los parámetros necesarios para su funcionamiento. A continuación se presenta un ejemplo de una petición.

```

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<ns1:getTemp
xmlns:ns1="urn:xmethods-Temperature"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<zipcode xsi:type="xsd:string">10016</zipcode>
</ns1:getTemp>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

3.3.4.2. Diferencias entre SOAP y REST

La principal diferencia entre SOAP (Simple Object Access Protocol) y REST (Representational State Transfer) es que este último es un estilo arquitectónico y SOAP un protocolo.

En lo referente a peticiones y respuestas, estas en SOAP son más complejas que el envío exclusivo de los datos (generalmente XML) sobre HTTP, por su parte REST depende directamente del protocolo HTTP (Hypertext Transfer Protocol) en un sentido de petición-respuesta (request-response) utilizando un conjunto simple y pequeño de operaciones bien definidas: POST, GET, PUT y DELETE.

Las APIs de programación de servicios web en Java y .NET ocultan esta complejidad de envío que tiene SOAP, sin embargo, en lenguajes de tipo script (como PHP) resulta más cómodo implementar e invocar servicios con enfoque REST que servicio SOAP. Por lo que las APIs de programación de servicios web en J2EE y .NET son sencillas de usar, pero complican la invocación de los servicios a los lenguajes de tipo script. Este punto a llevado a que los servicios REST ganen relevancia en determinados escenarios, un ejemplo de aquello es Amazon este ofrece una gran cantidad de servicios web para interactuar con su tienda electrónica para la búsqueda de libros, típicamente desde otro sitio Web para ello no se necesita agrupamiento de operaciones en

transacciones o aspectos avanzados de seguridad. Amazon posee ambos estilos de uso de sus servicios Web.

Por otra parte SOAP lleva siempre asociado la definición de servicio web. Y como tal, su definición a través de un WSDL. REST por el contrario no ofrece esta posibilidad. REST es mucho más sencillo, es lo más óptimo para simplificar una interfaz a utilizar por consumidores externos (el ejemplo más claro lo tenemos con Google, que expone sus servicios web vía REST y da como respuesta un xml con formato de sindicación RSS o ATOM) o cachear resultados a nivel de recursos. Pero si se quieren implementar cosas más complejas es necesario SOAP: Por ejemplo, REST lleva asociado HTTP como único protocolo de transporte. Si tenemos necesidad de utilizar otro tipo de transporte como JMS, FTP, SMTP por ejemplo, esto si es posible con el protocolo SOAP, pero no así el caso de REST.

3.4. WSDL

WSDL (Web Services Description Language) es una especificación que define cómo describir servicios Web usando una gramática XML, describiendo en forma abstracta operaciones y mensajes. WSDL consta de 4 piezas importantes que indican qué hace el servicio, qué datos utiliza en sus mensajes, cómo se utiliza y dónde se encuentra dicho servicio.

Utilizando WSDL, un cliente puede localizar un servicio web e invocar cualquiera de sus funciones disponibles y que estén publicadas. Se puede decir que WSDL representa una piedra angular de los servicios web, porque provee un lenguaje común para describir los servicios, y una plataforma para integrar estos servicios.

3.4.1. Especificación de WSDL

Como se menciono anteriormente WSDL es una gramática XML para describir servicios web. Su especificación se divide en 6 partes importantes.

1. **Definitions:** El elemento definitions debe ser el elemento raíz para todo documento WSDL. Este elemento define el nombre del servicio web, declara múltiples espacios de nombres utilizados en todo el resto del documento.
2. **Types:** El elemento type describe todos los tipos de datos utilizados entre el cliente y el servidor. WSDL utiliza el XML Schema como opción predeterminada. Si un servicio usa solo XML Schema con tipos de datos simples, como strings o enteros, el elemento type ya no sería necesario.
3. **Message:** Este elemento describe un mensaje unidireccional, el que puede ser un mensaje de petición o respuesta. El elemento message define el nombre del mensaje, además de contener 0 o más subelementos llamados part, los cuales pueden referirse a parámetros o valores de retorno.

4. **PortType**: El elemento portType combina múltiples elementos message para formar operaciones unidireccionales o en ambos sentidos. Un elemento portType puede utilizar múltiples operaciones que definen el intercambio ordenado de los mensajes.
5. **Binding**: Describe las características concretas de cómo el servicio será implementado en la red.
6. **Service**: Este elemento define la dirección para la invocación del servicio requerido.

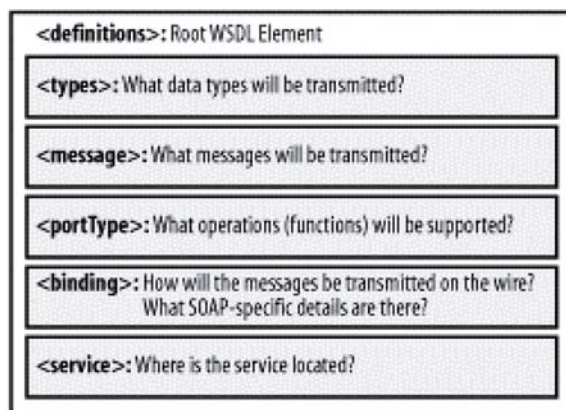


Figura 3.3. Elementos de WSDL

En adición a los 6 elementos principales, también se podrían agregar los siguientes elementos.

- **Documentation**: El elemento documentation es usado para proveer una documentación legible, la cual puede ser incluida en un documento WSDL.
- **Import**: Este elemento es utilizado para importar otros documentos WSDL o esquemas XML.

3.5. UDDI

UDDI (Universal Description, Discovery and Integration) es una descripción técnica para encontrar, describir e integrar servicios web. UDDI consiste en 2 partes:

1. Un directorio con los metadatos de todos los servicios web, incluyendo un puntero a la descripción de cada servicio (descripción WSDL).
2. Las definiciones de “port types” WSDL para manipular y buscar en ese directorio.

3.5.1. Categorías

La información capturada con UDDI puede ser dividida en 3 categorías principales.

1. **Páginas Blancas**: Estas incluyen información general acerca de una compañía específica, por ejemplo, el nombre del negocio, descripción del negocio, información de contacto, dirección y números telefónicos.

2. **Páginas Amarillas:** Estas páginas permiten buscar servicios por clasificación o categorías. Por ejemplo, la información de una página amarilla puede contener producto, la industria, la localización geográfica, esto para UDDI V1, que para UDDI V2 se extienden.

3. **Páginas Verdes:** Esta categoría contiene información técnica acerca de los servicios web. Generalmente, ésta incluye un puntero a una especificación externa y a una dirección para invocar el servicio web requerido.

3.5.2. Modelo de datos UDDI

El modelo de datos UDDI incluye un XML Schema que describe 4 elementos principales.

1. **businessEntity:** Incluye información acerca del negocio actual. Esta información contempla el nombre del negocio, descripción, dirección e información de contacto.
2. **businessService:** Este elemento incluye información acerca de un servicio o un grupo de servicios web relacionados. La información corresponde al nombre, descripción y una lista opcional de elementos
3. **bindingTemplates:** El elemento bindingTemplates contiene información acerca de cómo y dónde acceder a los servicios web requeridos.
4. **tModel:** Representa las especificaciones técnicas, metadatos sobre las especificaciones del documento, el nombre puntero URL, es presentado en forma de un documento WSDL.

3.6. Integración de Servicios Web

3.6.1. Coreografía

En la coreografía de Servicios Web se pueden definir colaboraciones entre cualquier tipo de aplicaciones, independiente de la plataforma en que se ejecuten o el lenguaje de programación que hayan sido creadas. En un proceso de coreografía, ésta no es controlada por un solo participante de la interacción.

3.6.2. Orquestación

En la orquestación de servicios, el proceso es controlado por una única entidad. Esta entidad define completamente las interacciones con los servicios componentes, además de la lógica necesaria para conducir correctamente éstas interacciones. De un punto de vista práctico, las compañías pueden exponer sus aplicaciones de software y recursos como servicios web, donde otros puedan encontrar y utilizar estos servicios para sus procesos de negocio. El objetivo de la orquestación de servicios web es conectar los servicios entre sí para así poder crear procesos de negocios de alto nivel.

3.6.3. Lenguajes de composición de servicios

3.6.3.1. BPML (Business Process Management Language)

Es un lenguaje basado en XML para la descripción de procesos de negocios. En principio BPML se diseñó para soportar procesos que un sistema de gestión de proceso de negocios pudiese ejecutar.

BPML incluye las características necesarias para dar soporte a la persistencia, y para realizar composiciones recursivas para así poder construir procesos de negocio que a su vez consten de subprocesos de negocio. Además de permitir planificar las tareas para que se ejecuten en instantes específicos.

3.6.3.2. XLANG

XLANG utiliza WSDL para la descripción de los Servicios Web, se centra en la creación de procesos de negocio y en los patrones de intercambio de mensajes entre los Servicios Web. Proporciona tanto el modelo de una orquestación de servicios, así como los contratos de colaboración entre orquestaciones. Además ofrece opciones para la captura de excepciones y soporte para transacciones de larga duración (Microsoft).

3.6.3.3. WSFL (Web Service Flow Language)

Este lenguaje describe los flujos de trabajo públicos y privados. Pueden definirse tanto los intercambios de datos, como la secuencia de ejecución y la descripción de cada uno de los pasos a realizar dentro del flujo para operaciones específicas. La interfaz de WSFL es similar a la de WSDL. Además permite capturar excepciones aunque no soporta directamente transacciones (IBM).

3.6.3.4. WSCI (Web Services Choreography Interface)

WSCI define una extensión para WSDL para la colaboración, y la coreografía global o el intercambio de mensajes entre los Servicios Web. Además soporta correlación, reglas de secuenciación, manejo de excepciones, transacciones y colaboración dinámica.

3.6.3.5. BPEL4WS (Business Process Execution Language for Web Services)

BPEL o BPEL4WS es una especificación para la orquestación de servicios web, es decir, tiene como objetivo definir un nuevo servicio a partir de otros servicios web existentes. BPEL utiliza la sintaxis XML para describir la lógica de control necesaria para la coordinación de los servicios web, donde BPEL define como secuenciar las operaciones definidas por WSDL.

3.7. Arquitectura Orientada a Servicios

Una arquitectura orientada a servicios (SOA) es descrita como un conjunto de servicios que apuntan a los negocios que son combinados (orquestración y composición) para cumplir con los objetivos de negocio [2].

3.7.1. Terminología

- Servicios (Services): son las entidades lógicas, contratos definidos por uno o más interfaces públicas.
- Proveedor de servicio (Service provider): es la entidad de software que implementa una especificación de servicio.
- Consumidor de servicio (Service consumer): la entidad de software que llama a un proveedor de servicios. Un consumidor de servicio puede ser una aplicación de usuario final o un mismo servicio.
- Localizador de servicio (Service locator): es una clase específica de servicio, el cual se comporta como un registro y permite la búsqueda de interfaces de los proveedores de servicio y localización de servicios.
- Corredor de servicios (Service broker): tipo especial de servicio que puede pasar ante una solicitud de servicio a uno o más proveedores de servicios.

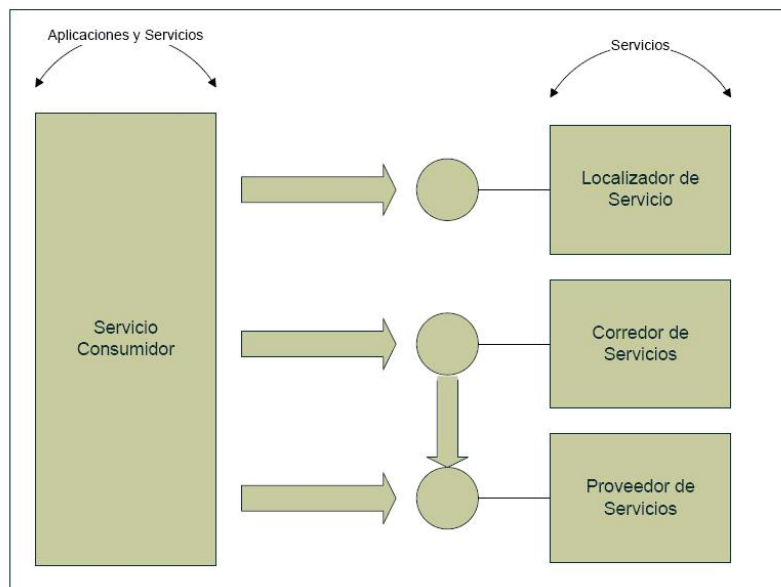


Figura 3.4 Terminología SOA

3.7.2. Propiedades de SOA

SOA es una arquitectura de sistemas distribuidos que está caracterizada por las siguientes propiedades.

- **Orientación al mensaje:** el servicio es definido en términos de los mensajes intercambiados entre el agente solicitante y el agente proveedor.
- **Vista Lógica:** es la vista que proporciona la imagen de los componentes que conforman al sistema, como el proceso de negocio, bases de datos, programas, etc., y explicando qué hace cada uno de los componentes.
- **Orientación a la descripción:** un Servicio Web es descrito por metadatos procesables por máquina. Esta descripción debe soportar la naturaleza pública de la arquitectura.
- **Orientación a la Red:** los Servicios Web deben ser creados para ser usados sobre la red, sin embargo, esto no es obligatorio.
- **Granularidad:** los servicios deben tener la tendencia de realizar pequeños números de operaciones con una gran cantidad de mensajes.
- **Plataforma Neutral:** los mensajes deben ser creados para una plataforma neutral, utilizando un lenguaje estándar, para ello la mejor opción sería XML.

3.8. Uso de Tecnologías en Servicios Web.

Para el desarrollo de Servicios Web existen varios lenguajes de programación que soportan el desarrollo y el consumo de estos, entre los cuales se encuentran Pearl, PHP, Java y .NET entre otros. Aunque en el mercado de los Servicios Web actualmente son 2 lenguajes que son los principales y más utilizados, estos son Java y .NET. En esta sección se abordará la relación entre estos 2 lenguajes y los Servicios Web.

3.8.1. Tecnología .NET

Microsoft .Net es una plataforma para generar, ejecutar aplicaciones distribuidas, la cual consta de los siguientes componentes:

Herramientas de desarrollo.

Son las herramientas necesarias para construir aplicaciones y Servicios Web. Estas herramientas son Visual Studio.NET y .NET Framework.

Servidores.

Contenedores sobre los cuales construir, proveer y desplegar las aplicaciones y servicios.

Cliente.

Software que provee dispositivos inteligentes, permitiendo a los usuarios interactuar y experimentar la plataforma .NET.

3.8.2. JAVA

Hay varias posibilidades de implementación de Servicios Web en Java. Una de las más conocidas es JAX-RPC o APIs de Java para llamadas a procedimientos remotos basadas en XML, en español. A continuación se muestra una breve descripción de JAX-RPC.

3.8.2.1.JAX-RPC

Esta API (interfaz de programación de aplicaciones, en español) sirve tanto para registrar y utilizar Servicios Web en Java. Una de las principales características de esta API, el cual siendo consecuente con las definiciones de Servicios Web en cuanto a la interoperabilidad es que, un cliente desarrollado en JAX-RPC puede consumir un Servicio Web residente en un servidor remoto a través de Internet, aunque este servicios esté desarrollado en otro lenguajes, o se ejecute en una plataforma diferente. Para esto, JAX-RPC requiere el soporte de SOAP en HTTP.

Modo de operación.

El modo de operación de JAX-RPC para la ejecución de servicios Web está dado de la siguiente forma.

Por el lado del cliente, las llamadas del método son transformadas en mensajes SOAP, luego este mensaje es enviados como una petición HTTP.

En lado del servidor, el sistema de ejecución JAX-RPC recibe la petición, traduce el mensaje SOAP generado en el cliente, a una llamada al método y finalmente lo invoca.

Finalmente, una vez procesado la petición el sistema de ejecución JAX-RPC, realiza unos pasos similares a los anteriores para enviar la respuesta al cliente, es decir, transformando la respuesta en mensajes SOAP.

Características de JAX-RPC.

Un cliente de un servicio web simplemente realiza las llamadas a los métodos de Java. Los procesos de serialización, deserialización y los detalles de transmisión se llevan a cabo automáticamente.

En JAX-RPC existe una herramienta que se utiliza para generar stubs utilizando documentos WSDL, el nombre de dicha herramienta es wscompile.

También existe la herramienta wsdeploy que se utiliza para desplegar los servicios. Esta herramienta también puede ser utilizada para crear los documentos WSDL.

3.8.2.2.JAX-WS

Actualmente existe otra API estandarizada para Java EE para el desarrollo de Servicios Web en Java. Esta API es JAX-WS, la cual sustituye a JAX-RPC, anteriormente mencionada. El principal cambio entre estas formas de implementación, es moverse a hacia servicios Web orientados a documentos. Una de las principales ventajas de JAX-WS es que es un modelo de programación que simplifica el desarrollo de aplicaciones mediante el soporte de un modelo basado

en anotaciones estándar para desarrollar clientes y aplicaciones de servicios Web. Estas anotaciones son:

`@WebService()` indica que la clase será un Web Service

`@Webmethod()` crea una operación para el Web Service, en base al método anotado

`@WebParam()` indica el nombre del parámetro en el web service.

Otras características a destacar de JAX-WS son el soporte de documentos SOAP 1.2, además de constar con nuevas herramientas como `wsgen` y `wsimport`. En cuanto al soporte de SOAP, JAX-WS soporta documentos SOAP 1.1 y 1.2, donde en SOAP 1.2, se pueden enviar documentos adjuntos binarios, como imágenes y archivos.

3.8.2.2.1. Modelo de programación de Cliente JAX-WS

Cliente proxy dinámico

El modelo de programación de cliente estático de JAX-WS se denomina el Cliente proxy dinámico. El Cliente proxy dinámico invoca un servicio Web basado en una interfaz de punto final de servicio (SEI) que debe proporcionarse. El Cliente proxy dinámico es parecido al cliente stub en el modelo de programación JAX-RPC (Java API for XML-based RPC). Aunque el Cliente proxy dinámico JAX-WS y el cliente stub JAX-RPC se basan en la Interfaz de punto final de servicio (SEI) que se genera a partir de un archivo WSDL, existe una diferencia importante entre ambos. El Cliente proxy dinámico se genera dinámicamente durante la ejecución utilizando la funcionalidad de proxy dinámico de Java 5, mientras que el cliente stub basado en JAX-RPC es un archivo Java no portátil que se genera utilizando las herramientas. A diferencia de los clientes stub JAX-RPC, el Cliente proxy dinámico no necesita que regenere un stub antes de ejecutar el cliente en un servidor de aplicaciones de otro proveedor, ya que la interfaz generada no requiere la información específica del proveedor.

Cliente de asignación

Los servicios Web basados en XML utilizan mensajes XML en las comunicaciones entre los servicios Web y los clientes de servicios Web. Las API JAX-WS proporcionan métodos de alto nivel para simplificar y ocultar los detalles de conversión entre las invocaciones de método Java y sus mensajes XML asociados. No obstante, en algunos casos, puede que desee trabajar a nivel de mensaje XML. La API de Cliente de asignación proporciona soporte para invocar servicios a nivel de mensaje XML. La API de Cliente de asignación, `javax.xml.ws.Dispatch`, es una interfaz de programación de clientes JAX-WS dinámica. Para grabar un Cliente de asignación, debe tener experiencia con las API de Cliente de asignación y los tipos de objetos soportados, así como conocimientos sobre las representaciones de mensajes del archivo WSDL (Web Services Description Language) asociado. El Cliente de asignación puede enviar datos en modalidad MESSAGE o PAYLOAD. Cuando se utiliza la modalidad `javax.xml.ws.Service.Mode.MESSAGE`, el Cliente de asignación sólo es responsable de proporcionar el sobre SOAP completo, incluidos los elementos `<soap:Envelope>`, `<soap:Header>` y `<soap:Body>`. Cuando se utiliza la modalidad `javax.xml.ws.Service.Mode.PAYLOAD`, el Cliente de asignación sólo es responsable de proporcionar el contenido de `<soap:Body>` y JAX-WS incluye la carga en un elemento `<soap:Envelope>`.

4. Área Turismo

El turismo es uno de los sectores económicos más importantes y dinámico a nivel mundial. Conceptualmente el turismo es un sector complejo, heterogéneo y multisectorial, donde se puede definir y clasificar de acuerdo a las actividades que abarca, los destinatarios del producto turístico, y el consumo asociados a estos productos. Es decir, desde el consumo de productos netamente turísticos (hoteles, hospedajes, etc.), hasta productos de uso común (restaurantes, carreteras). La definición que propone la Organización Mundial del Turismo (OMT) es “El turismo comprende las actividades que realizan las personas durante sus viajes y estancias en lugares distintos a su entorno habitual, por un periodo de tiempo consecutivo inferior a un año con fines de ocio, por negocios”.

4.1. Elementos del Turismo

Se pueden distinguir 4 elementos fundamentales en el área turística.

1. La demanda: es el elemento formado por el conjunto de consumidores (reales o potenciales) de bienes y servicios turísticos.
2. La oferta: compuesta por el conjunto de servicios, productos y organizaciones involucradas en la experiencia Turística.
3. Espacio Geográfico:
 - Espacio Turístico: es el lugar geográfico determinado donde se asienta la oferta turística y hacia el que fluye la demanda.
 - Destino Turístico: constituye el objetivo del turista. El destino turístico es una agrupación de instalaciones y servicios diseñados para satisfacer las necesidades de los turistas.
4. Los operadores del mercado: son los agentes participantes en la actividad turística, generalmente, como intermediarios entre el consumidor final y el producto turístico. Entre estos podemos encontrar las agencias de viajes, empresas transportadoras, centrales de reservas, etc.

4.2. Clasificación

El turismo se puede clasificar de las siguientes maneras:

1. Turismo Interino: es el tipo de turismo donde los residentes de un país dando viajan únicamente dentro del mismo país.
2. Turismo Receptor: es el tipo donde, los no residentes viajan dentro de un país dado.
3. Turismo Emisor: el de los residentes de un país dado viaje a otro país.
4. Turismo Internacional: es el que comprende el tipo de turismo emisor y receptor.

La temporada Turística es el conjunto de condiciones homogéneas que presentan las zonas, caracterizadas por la situación de acceso de la demanda y la oferta, dando lugar así a modificaciones constantes de precios. La temporada turística se divide en 3 tipos:

1. Temporada alta: se caracteriza porque la oferta y los atractivos turísticos se encuentran en la mejor situación posible. Esto coincide generalmente con un acceso masivo de los turistas a la zona, lo que provoca un incremento considerable del precio de los productos y servicios, derivado de la limitación temporal de las personas para la determinación de los periodos vacacionales.
2. Temporada media: ésta temporada se caracteriza por la pérdida de las condiciones y características de los atractivos turísticos de la zona, donde la demanda turística reduce su acceso que tiene como consecuencia una reducción significativa del fenómeno turístico. La temporada media se utiliza como temporada de promoción cuando el atractivo turístico tienen buenas condiciones pero los turistas tienen limitaciones temporales.
3. Temporada baja: es el periodo de tiempo donde los atractivos turísticos han perdido sus características y condiciones esenciales, en donde la demanda turística está limitada por que son los periodos establecidos para el trabajo, lo que provoca una reducción progresiva de la demanda solo al alcance del turismo de elite, del turismo social o para personas ajenas al ámbito laboral. Por lo tanto el precio de los productos turísticos alcanza sus niveles más baratos y se provoca la baja rentabilidad de las empresas, donde algunas o muchas de ellas tendrán que cerrar hasta las otras temporadas.

4.3. El turista

Todos los tipos de viajeros que realizan turismo son descritos como visitantes, éste término puede ser dividido de la siguiente manera [11].

1. Visitante internacional: es toda aquella persona que viaja por un periodo inferior a los 12 meses a un país distinto de aquel en el que reside habitualmente, y cuyo principal motivo es no ejercer actividades que remunere en el país visitado.

2. Visitante interino: es la persona que reside y viaja por un periodo inferior a los 12 meses, en un lugar dentro del país, pero diferente al de su entorno habitual. Cuyo objetivo del viaje es no ejercer una actividad que remunere en el lugar visitado.
3. Visitante que pernocta: es el visitante que permanece en un medio de alojamiento privado o colectivo en un lugar visitado por lo menos una noche.
4. Visitante del día: es el tipo de visitante que no pernocta en ningún medio de alojamiento privado o colectivo en el lugar visitado. En este grupo están comprendidos los propietarios o pasajeros de embarcaciones de placer, los que están alojados en un tren, y los pasajeros que participan en un viaje de grupo.

4.4. La Empresa Turística

La calificación de turística que se le atribuye a una empresa corresponde al consumidor que va dirigida. Es decir, si dichos consumidores son turistas que viajan movidos por necesidades de disfrutar del tiempo libre o de la cultura [10].

4.4.1. Tipos de Empresa

1. Empresas de Alojamiento: hoteles, moteles, casas rurales, hospedajes, balnearios (que ofrezcan servicio de hospedaje además del servicio termal).
2. Empresas de Restauración: restaurantes, cafeterías, salas de fiesta, etc.
3. Empresas Distribuidoras: agencias de viajes (tanto mayoristas, minoristas, mixtas o centrales de reservas), tours- operadores.
4. Empresas de Transporte y Auxiliares: compañías aéreas, compañías de autobuses, alquiler de coches, etc.
5. Servicios Financieros o Seguros: empresas que venden cheques o seguros de viajes.

4.5. El producto turístico

El producto turístico es el conjunto de bienes y servicios puestos a disposición del usuario (turista) en un destino determinado. El producto Turístico está compuesto por:

- Los recursos turísticos.
- Los atractivos turísticos.
- La planta turística.
- Los servicios complementarios.
- Los medios de transporte.
- La infraestructura básica.

4.5.1. Los recursos turísticos

Son los elementos principales de la oferta turística. Los recursos turísticos son aquellos elementos naturales, culturales y humanos que pueden motivar el desplazamiento de los turistas, es decir, generar demanda turística. Estos recursos están categorizados en:

- Sitios naturales.
- Folklore.
- Manifestaciones culturales
- Acontecimientos programados

4.5.2. Los atractivos turísticos

Son todos los recursos turísticos que tiene las condiciones necesarias para ser disfrutados y visitados por el turista, es decir, que cuentan con planta turística, medios de transporte, servicios complementarios e infraestructura básica, descritos a continuación.

4.5.3. Planta turística

Es el conjunto de instalaciones, equipos, empresas y personas que prestan servicio al turismo y fueron creados para este fin. Estos se clasifican en:

1. Alojamiento.

Según la Organización Mundial del Turismo (OMT), éste se divide en hoteleros y extrahoteleros. Los primeros están conformados por:

- Hoteles que ofrecen alojamiento con o sin servicios complementarios como alimentación, congresos y eventos. Estos dependen de la categoría de cada establecimiento que se puede clasificar de 5 a 1 estrellas, según la reglamentación vigente en nuestro país, siendo el de 5 estrellas el que brinda mayores y mejores servicios complementarios.
- Hoteles-apartamento que pueden ofrecer todos los servicios de los hoteles, pero que cuentan adicionalmente con instalaciones y equipamiento para la conservación, preparación y consumo de alimentos fríos y calientes. Van de 5 a 3 estrellas.
- Moteles, se encuentran ubicados en las carreteras o autopistas, poseen entrada independiente desde el exterior al alojamiento y pueden tener garaje individual o parqueo colectivo. En el Perú aún no se cuenta con esta clasificación.
- Hostal o pensión: Cumplen las funciones de un hotel, pero no alcanzan las condiciones mínimas indispensables para considerárseles como tales. Puede ser de 3 a 1 estrellas.

Los extrahoteleros agrupan a:

- Campamentos que son terrenos de uso privado en los que se instalan sistemas de alojamiento, tales como tiendas de campañas o caravanas móviles. Cuentan con instalaciones comunes (sanitarios, lugares para el aseo, el lavado y el tendido de ropa, zonas recreativas).
- Casa de alojamiento, normalmente particulares, que prestan servicios en época de alta demanda turística y que contribuyen a ampliar la oferta de un determinado destino.
- Casas rurales, comprende las habitaciones de las comunidades campesinas que han sido adaptadas especialmente para recibir a turistas.
- Apartamentos que han sido adaptados para permanencias largas.

2. Restaurantes.

Son aquellos establecimientos que expenden comidas y bebidas preparadas al público en el mismo local, prestando el servicio en las condiciones señaladas en el reglamento de restaurantes de cada país, y de acuerdo a las normas sanitarias correspondientes.

4.5.4. Servicios complementarios

Servicios complementarios son los servicios requeridos o empleados por los turistas que no dependen del sector turismo, tales como:

- Sistema bancario.
- Servicios de transporte diversos.
- Servicios de salud.
- cibern-cafe.
- Comercio en general.

4.5.5. Los medios de transporte

Son los medios que permiten al viajero acceder a los destinos elegidos. Los medios de transporte se clasifican en:

- Transporte terrestre.
- Transporte acuático.
- Transporte aéreo.

4.5.6. La infraestructura básica

Es el conjunto de obras y servicios que permiten el desarrollo socioeconómico de un país, y que el sector turístico utiliza para impulsar su actividad.

- Rutas de acceso

Pueden ser terrestre, aérea, lacustre, marítima, redes viales, aeropuertos, terminales, etc.

- Comunicaciones.

Abarca los servicios de teléfono, correo, telégrafo, Internet, correo electrónico, etc.

- Equipamiento urbano

Comprende los servicios agua, desagüe, alcantarillado y energía eléctrica.

4.6. Efectos del Turismo

La industria turística es muy compleja de acuerdo a la variedad de elementos que la componen, por lo que el impacto en el medio también es diverso. Los efectos más importantes se clasifican de la siguiente manera.

1. Efecto Económico

La industria turística en la calidad de producto de “exportación” trae consigo una serie de beneficios y rentas al país receptor. Estos beneficios se reflejan en la creación de empleos, la entrada de divisas, el aumento de los ingresos públicos, y el fomento de la actividad empresarial. La actividad turística en los países receptores, presenta uno de los medios más importantes en la generación del Producto Interno Bruto (PIB). El año 2007 la industria del turismo alcanzó el 3% del PIB nacional según el Servicio Nacional de Turismo (SERNATUR).

2. Efecto Cultural

El turismo pone en contacto a diferentes culturas, como la receptora (local) con la emisora (visitante), lo que supone un intercambio de pautas de conducta, formas de vida, hábitos de diversa índole como gastronómicos, lingüísticos, estéticos, etc. Además el Turismo, suele estimular el interés por los residentes por la propia cultura, sus tradiciones, costumbres y patrimonio histórico. También puede ser un factor que acelere cambios sociales positivos en una comunidad, como la fomentación de la tolerancia, el bien común, la búsqueda del mejoramiento de la calidad de vida, entre otras.

3. Efecto Sociopolítico

El turismo es una actividad importante dentro de las naciones receptoras, por lo que es una actividad económica mas dentro de ellas, donde debe existir un control publico de fiscalización, seguridad, sanidad entre otras. Por lo que el desarrollo de la actividad turística en gran escala requiere la intervención del gobierno para mejorar la competitividad a nivel internacional para popularidad turística de la nación, además de la creación de leyes que regulen la actividad turística, y la adopción medidas de protección del medio ambiente, el cual es la materia prima del turismo.

4.7. Las Tecnologías de Información en el Turismo

En la industria del turismo las tecnologías de información (TI) han desempeñado un rol fundamental para la eficiencia de las organizaciones y la entrega de los servicios [8]. Donde las industrias hoteleras, servicios de viajes, restaurantes han sido modificadas por la inserción de las TI.

Las primeras aplicaciones se remontan a los años 60, donde las aerolíneas crearon los primeros sistemas de reserva de boletos de avión, adoptando las agencias de viajes estos sistemas una década después. En cuanto a la industria hotelera, la relación con las TI, se remontaba a la reservación de un computador central. Las operaciones de reservación y venta de pasajes son realizadas por Sistemas Globales de Distribución o GDS, donde, tanto aerolíneas como la industria hotelera, ofrecen información actualizada a través de estos sistemas para promover sus productos. Esto implica que las agencias de viajes se benefician al contar con una sola base de datos toda la información actualizada para el uso de los productos ofrecidos.

A nivel mundial los principales GDS son:

- Galileo
- Sabre
- Amadeus
- Worldspan
- System One
- Book Hotel

Actualmente la tecnología de información que impacta como ninguna otra en el sector turístico es Internet, el cual ha cambiado los esquemas de comercio y de competencia mundiales de los servicios turísticos. A través del uso de Internet se está cambiando a una economía que funciona las 24 horas y los 365 días del año en cualquier parte del mundo. Así, Internet permite a la gente encontrar información de forma rápida, y exacta de cualquier destino o esparcimiento que sea de su gusto.

A través de la ayuda de las tecnologías de información, la industria del turismo ha logrado grandes beneficios como:

- Mejorar las operaciones de la empresa.
- Conocer mejor las necesidades de los clientes.
- Ofrecer mejor la entrega de los servicios
- Expansión de información
- Llegar un número mayor de clientes.
- Abrirse a nuevos mercados.
- Optimizar recursos, logrando así aumentar la eficiencia.

4.8. Trabajos relacionados

4.8.1. Asired e-turismo

Sistema basado en servicios web xml, para la distribución de servicios turísticos³. Asired principalmente trabaja con la integración de servicios de alojamiento para promover, comercializar, vender sus productos y servicios turísticos, donde estos servicios de alojamiento corresponden a alberges, hostales, hoteles, campings. Entrega soporte tanto a tours operadores, agencias mayoristas y minoristas y centrales de reserva [15].

4.8.2. F.E.T.I.S.H

F.E.T.I.S.H: Sigla de Federated European Tourism Infrastructure System Harmonization (Federación Europea Sistema de Harmonización de Infraestructura Turística), definido como un Entorno Inteligente para Servicios y Sistemas de Inter operable Valor Añadido. El valor de la proposición de FETISH, consiste en conectar sistemas de información (bus de servicios basado en la net) y servicios de valor añadido (VAS) en una comunidad donde una masa critica de recursos y datos europeas puedan ser compartidos sobre una red a través de una amplia geografía, así como una red de servicios de referencia capaz de crear una ventaja competitiva dentro del mercado de proveedores de servicios turísticos. Su visión es la de convertirse en un líder en la entrega de soluciones de servicios distribuidos, pudiendo proveer información turística e integración de servicios, usando una arquitectura orientada a servicios, FETISH da a los proveedores de VAS un modelo de arquitectura de alto nivel que soporte la producción de confiables servicios independientes de la plataforma [13].

4.8.3. CINNTA

CINNTA⁴ (Consejería de Turismo, Comercio y Deporte de la Junta de Andalucía y el Patronato de Turismo de Córdoba): Impulsada por la Consejería de Turismo, Comercio y Deporte, se constituye como fundación y participan en ella 26 entidades del sector turístico, universidades, agentes económicos y sociales y centros tecnológicos. La incorporación de las innovaciones en la actividad turística requiere del seguimiento de los avances que puedan beneficiar al sector y de la difusión de la cultura de la innovación, para ello se plantea la creación de una Plataforma de Servicios on-line para la innovación turística en Andalucía, la organización de jornadas profesionales y la realización de acciones formativas dirigidas a mejorar la capacitación de los recursos humanos para la gestión de la innovación [14].

4.8.4. Eureka Tourism

1ª iniciativa internacional dedicada al I+D en Turismo. El principal objetivo de Eureka TOURISM es la definición de un marco donde las empresas, las asociaciones, los institutos de investigación y las administraciones públicas (a niveles local, regional, nacional, de la UNIÓN EUROPEA y EUREKA) trabajan juntas para: –Promueve el desarrollo sostenible del turismo europeo y la industria del ocio a través de la aplicación y nuevos servicios del entorno tecnológico; –Define, desde el conocimiento del sector, las líneas de trabajo fundamentales en el ámbito de la innovación y el desarrollo tecnológico, con la participación de todos lo actores implicados:

³ <http://www.asired.es>.

⁴ <http://www.cinnta.com>.

proveedores, productores, intermediarios, usuarios y organismos públicos, con especial enfoque en las PYMES; Intercambia información y puntos de vista para lanzar proyectos de colaboración en el marco de EUREKA y otras iniciativas de I+D+I(Investigación + Desarrollo + Innovación tecnológica); Establece contactos internacionales y busca socios, definir el marco de colaboración internacional para la aplicación de nuevas tecnologías; –Promueve el futuro soporte de iniciativas de los distintos organismos de la administración [12].

4.8.5. Prototipo multiagente sobre dispositivos móviles para la planificación de viajes turísticos

La solución presentada en el trabajo de título frente al problema de la planificación de viajes turísticos en dispositivos móviles, corresponde al desarrollo de un prototipo basado en un sistema multiagente sobre dispositivos móviles. Las características de la solución propuesta, son la ayuda al turista antes y durante su viaje, donde este podrá “contratar” los productos turísticos disponibles a través de dispositivos móviles tales como PDA, smartphones y otros. Además de ayudar a los proveedores de servicios, para expandirse a nuevos mercados, como también llegar a un número mayor de clientes. A continuación se presentan las características principales de la solución propuesta [3].

1. Generación de un itinerario

La generación de itinerarios está basada en la obtención de un perfil de usuario, preferencias de viaje, y los servicios que seleccione durante el proceso. Estos servicios seleccionados serán “contratados” y posiblemente reservados de acuerdo a las políticas del proveedor de servicios correspondiente.

2. Reservación de servicios

La reservación de servicios consiste en que el turista “contrata” un servicio deseado, esta operación es realizada automáticamente por el sistema. Si es necesario reservar el producto turístico en espacio físico donde se realizará el servicio, el turista debe confirmar la reservación con los datos que le entregue el servicio. La reservación se hará efectiva si el proveedor de servicios cuenta con esta funcionalidad, de no ser así, ésta se puede hacer efectivo a través de otro medio, como una llamada telefónica.

3. Publicación / Registro de Servicios

Esta característica corresponde a la publicación de un servicio turístico, para que se incorpore al sistema y poder ser visualizada y consumida por el usuario, la cual estará encargada de un agente específico.

4. Notificación de cambios durante el viaje

Cuando haya eventos externos que cambien un servicio, el proveedor de servicios podrá modificar la información de su servicio y estos serán notificados al turista en su dispositivo móvil.

El sistema está compuesto por 5 agentes principales, los cuales están encargados de llevar a cabo las funciones especificadas anteriormente. A continuación se presenta los agentes principales que conforman el sistema.

1. Personal Agent (Agente Personal)

Es el responsable de recordar el perfil del turista, seguir sus instrucciones, administrar el itinerario, buscar y notificar cambios en los servicios. Este agente depende del dispositivo en el que el turista está manipulando, es decir, como una PDA o un laptop.

2. Planner Agent (Agente Planificador)

Es el encargado de generar el itinerario, reservar servicios, evitar que el itinerario presente conflictos entre las fechas y los destinos de los servicios turísticos seleccionados. El planner agent revisa los requerimientos del turista, buscar los servicios a través del agente Broker Agent, filtrarlos y reservar los servicios.

3. Broker Agent (Agente Intermediario)

Recibe los requerimientos del usuario a través del agente planificador, para luego proponerle una lista de servicios que puedan satisfacerlos.

4. Service Agent (Agente de Servicio)

Este agente es responsable de publicar e informar sobre el servicio que publica un proveedor de servicios.

5. Factory Agent (Agente de Fábrica)

Cuando un Personal Agent necesita a un Planner Agent, o un Planner Agent necesita un Broker Agent, solicita al agente de fábrica la asignación del agente requerido.

5. Metodología

La metodología de desarrollo del presente proyecto será la del proceso unificado (UP). Esta metodología cuenta con una serie de fases bien definidas para obtener de alguna forma una guía. También contempla el desarrollo de prototipos ejecutables en tempranas fases del desarrollo del sistema, lo que favorece a entender y descubrir la magnitud y dificultad del sistema. Esto favorece una mitigación temprana de posibles riesgos altos, además de favorecer la curva de aprendizaje. Por otro lado, esta metodología soporta la incorporación de nuevos requerimientos en el sistema, por su carácter iterativo, lo que facilita enfocarse a los requerimientos de mayor relevancia o prioridad. En el desarrollo del proyecto se utilizará el lenguaje unificado para modelado (UML) en el análisis y diseño.

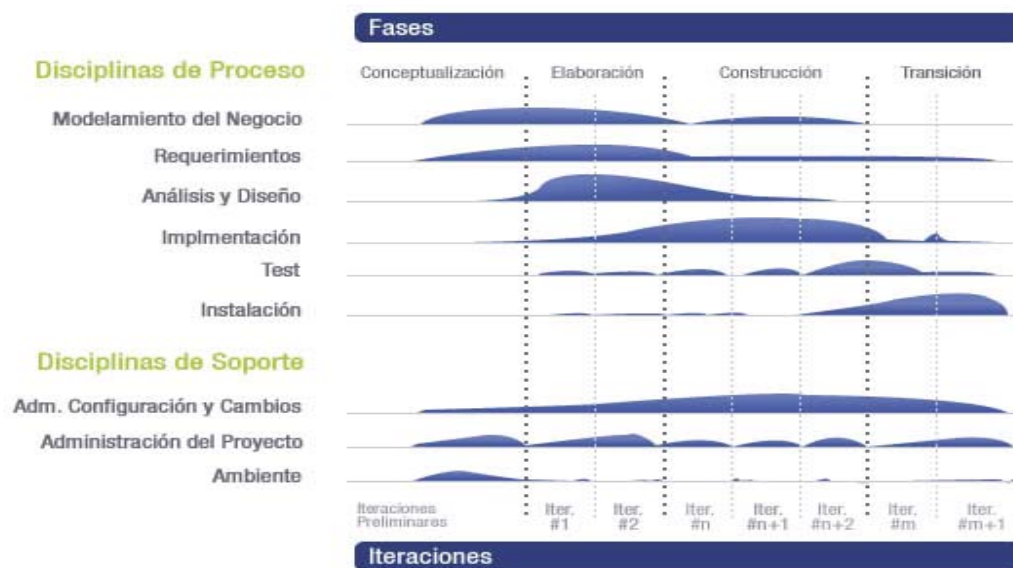


Figura 5.1 Fases e Iteraciones de la Metodología UP

A continuación se definen las fases del proceso unificado llevado al desarrollo del proyecto

1. Fase de Inicio
 - Establecimiento de los requisitos para el WS.
 - Selección de plataforma.

2. Fase de Elaboración
 - Creación del modelo de Análisis del WS.
 - Creación del Modelo de Diseño del WS.
 - Creación del primer prototipo.
 - Creación del Modelo de Diseño de una aplicación cliente.

3. Fase de Construcción
 - Creación del Modelo de Implementación.
 - Implementación del WS y la aplicación cliente.

4. Fase de Transición
 - Ubicación del WS en el servidor final.

6. Estudio de factibilidad

En este capítulo se detallará el estudio de factibilidad realizado para llevar a cabo el proyecto. En este estudio se especifican tanto la factibilidad técnica, económica y legal.

6.1. Factibilidad Técnica

En la factibilidad técnica se estudia si el trabajo para el proyecto, puede llevarse a cabo con la capacidad del recurso humano, disponibilidad de tecnologías y el equipamiento necesario.

En cuanto a la capacidad del recurso humano, el integrante del proyecto posee conocimiento en los lenguajes de programación J2ME y .NET.

Las tecnologías requeridas para la realización del proyecto están disponibles, como el equipamiento necesario para llevar un prototipo funcional

6.2. Factibilidad Económica

En el presente proyecto no se espera una tasa de retorno. Donde si, existen gastos involucrados en el mismo, en plataforma hardware y software para la implementación de un prototipo funcional.

Herramienta	Costo US\$	Costo \$	Valor US\$ observado
PHP	0	0	607,18
J2SE	0	0	09/10/2008
Visual Studio	470.60	285.738,908	
	Total	285.738,908	

Tabla 6.1. Costos Software

La plataforma hardware necesaria para el desarrollo del proyecto debe constar de características básicas para el correcto funcionamiento de la plataforma software necesaria, y para llevar a cabo un prototipo funcional.

Equipo	Modelo	Unidades	Costo
Pocket PC	Ipaq 116	1	\$ 236.200
	Inspiron	530	

Desktop PC	PC/Computadora	1	\$ 299.900
Escritorio	Modelo standart	1	\$ 26.900
		Total	\$ 563.090

Tabla 6.2. Costos Hardware

El valor total para llevar a cabo el proyecto sería de \$ 848.828,908, donde el total está dividido en costos de software (\$ 285.738,908) y hardware (\$ 563.090). Estos costos son depreciados ya que la Escuela Informática de la Pontificia Universidad Católica de Valparaíso cuenta con el equipamiento necesario para cubrir con las necesidades de desarrollo del proyecto, por lo que sería factiblemente económico llevar a cabo el proyecto.

5.3. Factibilidad legal

El proyecto no viola ninguna ley en cuanto al Turismo en sí. En cuanto a los servicios proporcionados por el proyecto, existen normas relacionadas con los agentes de viajes, los que implementarían el sistema.

Por lo tanto es factiblemente legal llevar a cabo el proyecto.

7. Análisis de Riesgos

En este capítulo se detalla el análisis de riesgos realizado previo a la desarrollo del proyecto, donde también se contemplan los planes de contingencia y mitigación.

El análisis de riesgo que se realizará para el presente proyecto es el análisis de riesgos proactivo, que a diferencia del reactivo, se evalúan los posibles riesgos potenciales, la probabilidad, impacto y se establece una prioridad según su importancia.

Los riesgos son clasificados de la siguiente forma:

- Riesgos de proyecto
Afectan la calendarización o recursos.
- Riesgos de producto
Afectan la calidad o desempeño del software que se está desarrollando
- Riesgos del negocio
Afectan la organización que desarrolla o procura el software

Riesgo	Tipo de Riesgo	Probabilidad	Impacto
Retiro del integrante del trabajo.	Negocio	Muy baja	Catastrófico
No disponibilidad del Hardware y/ o Software necesario para el desarrollo del proyecto.	Proyecto	Muy baja	Crítico
La plataforma sobre la que el sistema estará construido ha quedado obsoleta.	Negocio	Muy baja	Crítico
Incumplimiento de plazos establecidos.	Proyecto y Producto	Moderada	Catastrófico
Los miembros del equipo no dominan las tecnologías a utilizar para el diseño e implementación del proyecto.	Producto y Negocio	Alta	Crítico
Los hitos de revisión de avance no son cumplidos.	Proyecto y Producto	Baja	Crítico
Poca disponibilidad de tiempo de los integrantes	Negocio y proyecto	Media	Crítico

Tabla 7.1. Análisis de riesgos

7.1. Planes de Contingencia

Se presentan los planes de contingencia en la siguiente tabla.

Riesgo	Plan de Contingencia
Retiro del integrante del trabajo	Motivación
No disponibilidad del Hardware y/ o Software necesario para el desarrollo del proyecto.	Solicitud anticipada de los recurso
La plataforma sobre la que el sistema estará construido ha quedado obsoleta.	Estudio anticipado de la plataforma en que se implantara el sistema.
Incumplimiento de plazos establecidos.	Evaluación constante de las etapas del proyecto
Los miembros del equipo no dominan las tecnologías a utilizar para el diseño e implementación del proyecto.	Estudiar lo que sea necesario, aclarando dudas con el profesor o buscando información en distintos medios
Los hitos de revisión de avance no son cumplidos.	Revisión constante del proyecto
Poca disponibilidad de tiempo de los integrantes	Modificar la planificación

Tabla 7.2. Planes de Contingencia

7.2. Planes de Mitigación

Se presentan los planes de mitigación en la siguiente tabla.

Riesgo	Plan de Mitigación
Retiro del integran del trabajo	No se aplica
No disponibilidad del Hardware y/ o Software necesario para el desarrollo del proyecto.	Utilizar los recursos disponibles
La plataforma sobre la que el sistema estará construido ha quedado obsoleta.	Adaptar el sistema a una nueva plataforma
Incumplimiento de plazos establecidos.	No se aplica
Los miembros del equipo no dominan las tecnologías a utilizar para el diseño e implementación del proyecto.	Dedicar tiempo máximo al estudio del tema desconocido, estableciendo horarios.
Los hitos de revisión de avance no son cumplidos.	Definir nuevos hitos que permitan la realización efectiva del plazo
Poca disponibilidad de tiempo de los integrantes	Modificar la planificación

Tabla 7.3. Planes de mitigación.

8. Solución Propuesta

En este capítulo se detalla solución propuesta al problema previamente descrito. Además se describen de forma detallada el análisis y diseño del sistema, como también las herramientas y tecnologías utilizadas.

8.1. Descripción de la solución propuesta

La solución propuesta consiste en diseñar un sistema que se adapte a los diferentes dispositivos móviles, como PDAs, Smartphones, teléfonos móviles, laptops y PCs de escritorio. Para lograr esta solución se diseñará un sistema basado en Servicios Web, que permita la obtención de itinerarios de viaje, a través de las preferencias del turista, como el tiempo de viaje, precios y localización geográfica. El sistema deberá permitir la reservación de los servicios turísticos, personalizar el itinerario y notificar los cambios en los servicios.

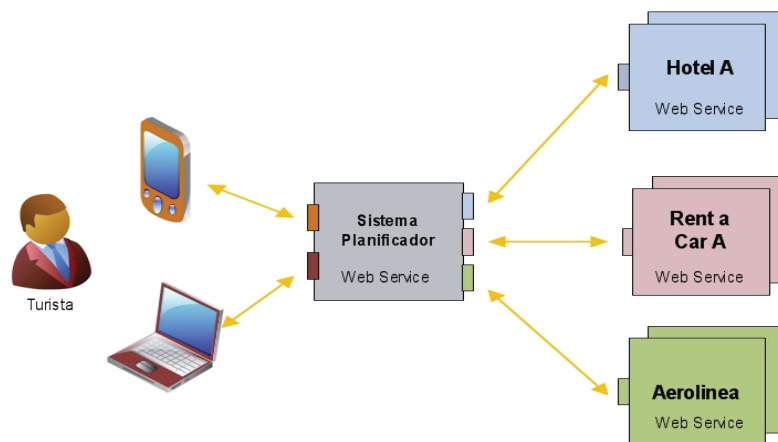


Figura 8.1 Sistema basado en Web Services

El sistema consiste en un Servicio Web, el cuál a su vez será consumidor de los servicios proporcionados por los proveedores, estos servicios, también serán Web Services. Las aplicaciones clientes, las cuales interactuarán con el usuario (turista), consumirán los servicios proporcionados por el sistema.

8.2. Análisis

Como se ha mencionado la metodología de desarrollo del proyecto estará contemplada por la del proceso unificado o UP. Ésta especifica en sus primeras etapas los requerimientos del sistema, los cuales estarán plasmados por los requerimientos funcionales. A continuación se presentan los requerimientos principales a incluir en el desarrollo del sistema:

- El turista debe crear su perfil de usuario antes de utilizar la aplicación, y acceder a todas las funcionalidades.
- El turista debe acceder mediante un nombre de usuario y contraseña.
- Un proveedor de servicios puede publicar un servicio en el sistema.
- El sistema debe ser capaz de mostrar los servicios disponibles de acuerdo a las preferencias del usuario, además de la información correspondiente a los servicios.
- El sistema debe ser capaz de listar, modificar y eliminar un perfil de usuario.
- El sistema debe ser capaz de listar, modificar y eliminar un itinerario de viaje.
- El sistema debe ser capaz de reservar los servicios disponibles deseados por el turista.
- El sistema debe cancelar las reservas, si se diese el caso.
- El sistema debe ser capaz de reconocer a un usuario ya almacenados en sus registros.
- El sistema debe ser capaz de notificar los cambios en los servicios reservados por el turista.

Un proyecto de esta envergadura debe ser capaz de enfrentarse a varios tipos de escenarios y situaciones de acuerdo a los requerimientos antes mencionados. UML proporciona diagramas para la captura de requerimientos, los cuales son complementados con los obtenidos previamente. Estos diagramas son los de casos de uso, los que aparte de dar la posibilidad de capturar requerimientos, se puede describir la interacción de los usuarios con el sistema de una forma sencilla y entendible para el lector. A continuación se presentan los siguientes diagramas de casos de uso.

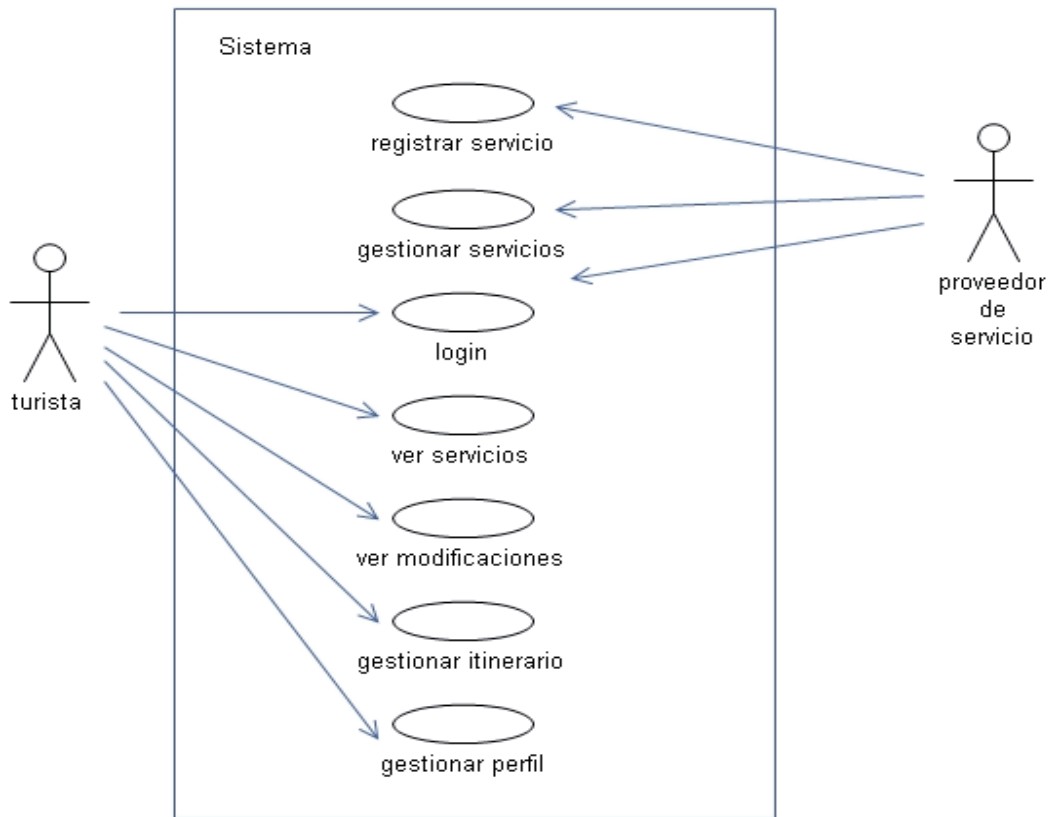


Figura 8.2 Diagrama de Casos de Uso Alto nivel

La Figura 8.2 representa el escenario principal o de alto nivel del sistema, donde se reflejan claramente las interacciones entre los actores y el sistema. A continuación se presenta una breve descripción de casos de uso.

Caso de Uso	Login
Actores	Turista, Proveedor de Servicios
Tipo	Primario
Descripción	
<p>Este caso de uso se encarga de autenticar el ingreso al sistema de parte de un turista a la aplicación, así como también, del proveedor de servicios el cual podrá suplir nuevos servicios a medida que la plataforma evolucione. Es un caso de uso primordial , pues, de esta manera se evita el ingreso al sistema de actores ajenos al mismo, así como también de agentes maliciosos que se quieran hacer control del sistema; funciona de manera tal que un turista sin un perfil previamente creado no pueda acceder a este caso de uso(viene del Caso de Uso Perfil de Usuario), así como también un proveedor no registrado en la base de datos no podrá publicar</p>	

servicios o realizar cualquier actividad que quede reflejada en el sistema. El ingreso al sistema se produce después de haber ingresado un nombre de usuario y password existente en la data store de la plataforma.

Caso de Uso	Gestionar Perfil
Actores	Turista
Tipo	Primario
Descripción	
<p>Se encargar de almacenar el perfil del usuario Turista en la aplicación, mediante la recolección de sus datos principales(Por Ej.: nombres, apellido paterno, apellido materno, dirección, mail, etc.) y logrando capturar luego los datos claves correspondientes al nombre de usuario y la contraseña del mismo, mediante la cual podrá ingresar al sistema (caso de uso resultante Login). En general se encarga de gestionar de manera simple y eficiente la información relativa al turista, la cual a posteriori es usada en los demás casos de uso de forma tal que se evite el tener que reingresar los datos del cliente que quiera realizar un itinerario de viaje, facilitando también el envío de datos y la comunicación entre la aplicación y los diversos servicios web que con ella interactúan.</p>	

Caso de Uso	Gestionar Itinerario
Actores.	Turista
Tipo	Primario
Descripción	
<p>Este es a grandes rasgos uno de los casos de uso mas importante del sistema, pues es aquel que se encarga de gestionar el perfil de usuario antes creado, así como de capturar los requisitos de planificación del turista, sirviendo de enlace entre el turista, la aplicación y los servicios disponibles(Por Ej. El turista selecciona el tipo de servicio que desea, así como los datos a enviar como lugar de destino, fecha de arribo, etc. Los cuales son enviados al sistema planificador, el cual se encarga de verificar la disponibilidad del servicio ingresado de acuerdo a los patrones antes introducidos, los cuales se traducen en una respuesta que es reflejada al turista, el cual se encarga de seleccionar alguna de las alternativas propuestas que se ajuste a sus criterios, ya sean presupuesto, cercanía a puntos de de interés, etc.). Una vez terminado el proceso de selección, se guardan los datos concernientes a las reservas hechas en una base de datos global para así tener un mayor control de las mismas, o realizar modificaciones de último momento (Por Ej. Modificación de itinerarios, o eliminación de los mismos).</p>	

Caso de Uso	Ver Modificaciones
Actores.	Turista
Tipo	Primario
Descripción	
<p>En este caso de uso el turista es informado ante posibles cambios por parte de los proveedores de servicios, a un itinerario creado con antelación, es decir, si el proveedor de servicios realiza una modificación en las condiciones de reserva, cambio de domicilio u otros datos del servicio por ejemplo, el turista tendría la opción de modificar el itinerario desde el caso de usos Itinerario o conservar el itinerario tal cual, previniendo de esta manera posibles sorpresas al turista por ejemplo al confirmar la reserva en el lugar físico y ver que el precio es diferente o que la empresa ya no se encuentra en la dirección antes mencionada; todo esto con la intención de que el itinerario realizado sea lo mas confiable y fidedigno posible.</p>	

Caso de Uso	Registrar Servicios
Actores.	Proveedor de Servicios
Tipo	:Primario
Descripción	
<p>En este escenario se refleja la necesidad de registrar nuevos servicios por parte de un proveedor , de acuerdo a los 3 grandes ítemes que contempla el sistema llámense Rent a Car, Hoteles o Tourss, dando así la posibilidad a otras empresas del rubro que mas adelante cuenten con la factibilidad tecnológica y económica de publicar sus servicios mediante el contacto con el proveedor de servicios pudiendo de esta manera diversificar la gama de servicios y opciones de itinerario que la aplicación móvil pueda consumir. Se contempla además a futuro la ampliación de los ítemes antes referidos a nuevos campos ya sean restaurantes, deporte aventura, etc.</p>	

Caso de Uso	Gestionar Servicios
Actores.	Proveedor de Servicios
Tipo	Primario
Descripción	
<p>En este escenario se encuentra la gestión de servicios registrados por parte de los proveedores de servicios en el sistema. La gestión de servicios contempla una especie de un panel de control de los servicios, es decir donde se podrá editar, mostrar e eliminar servicios registrados.</p>	

Caso de Uso	Crear Perfil
Actores	Turista.
Tipo	Primario
Descripción	
<p>Este caso de uso se encarga de la creación del perfil de usuario de la aplicación, es decir, el paso inicial para poder hacer uso de la misma. Sin la creación del perfil el usuario no puede hacer uso de la aplicación, es decir, no se puede realizar un ingreso o reservar servicios de manera anónima al sistema. Contempla el recabar la información personal introducida por el usuario (Turista) para luego ser almacenada en la base de datos local del teléfono móvil, así como un respaldo de la misma en la base de datos global de los turistas que estén registrados, lo cual permite de esta manera mediante una pregunta de seguridad, la posibilidad de que el usuario pueda recuperar su información de usuario en caso de que se olvide su password o que se extravíe su teléfono por ejemplo.</p>	

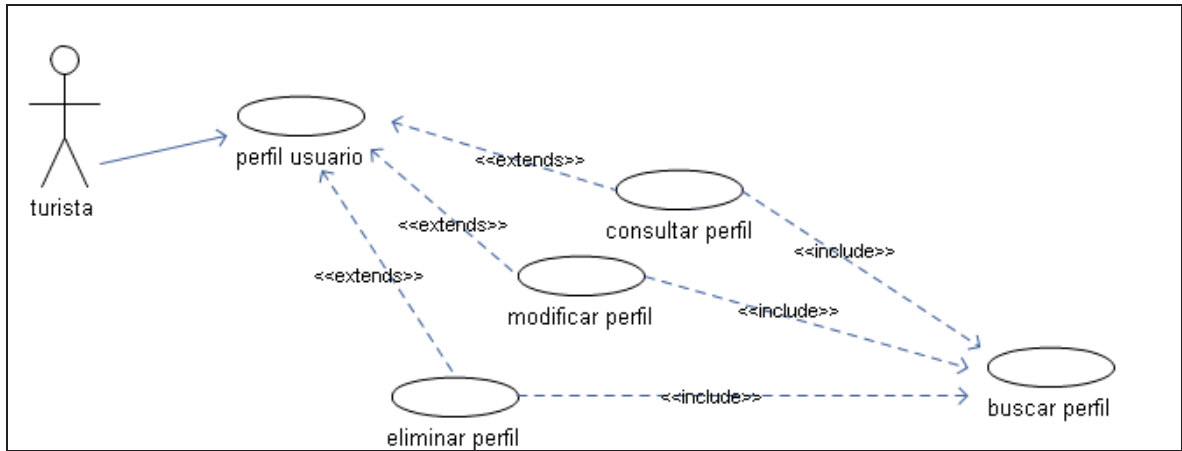


Figura 8.3. Casos de uso expandido gráfico Gestionar Perfil Usuario.

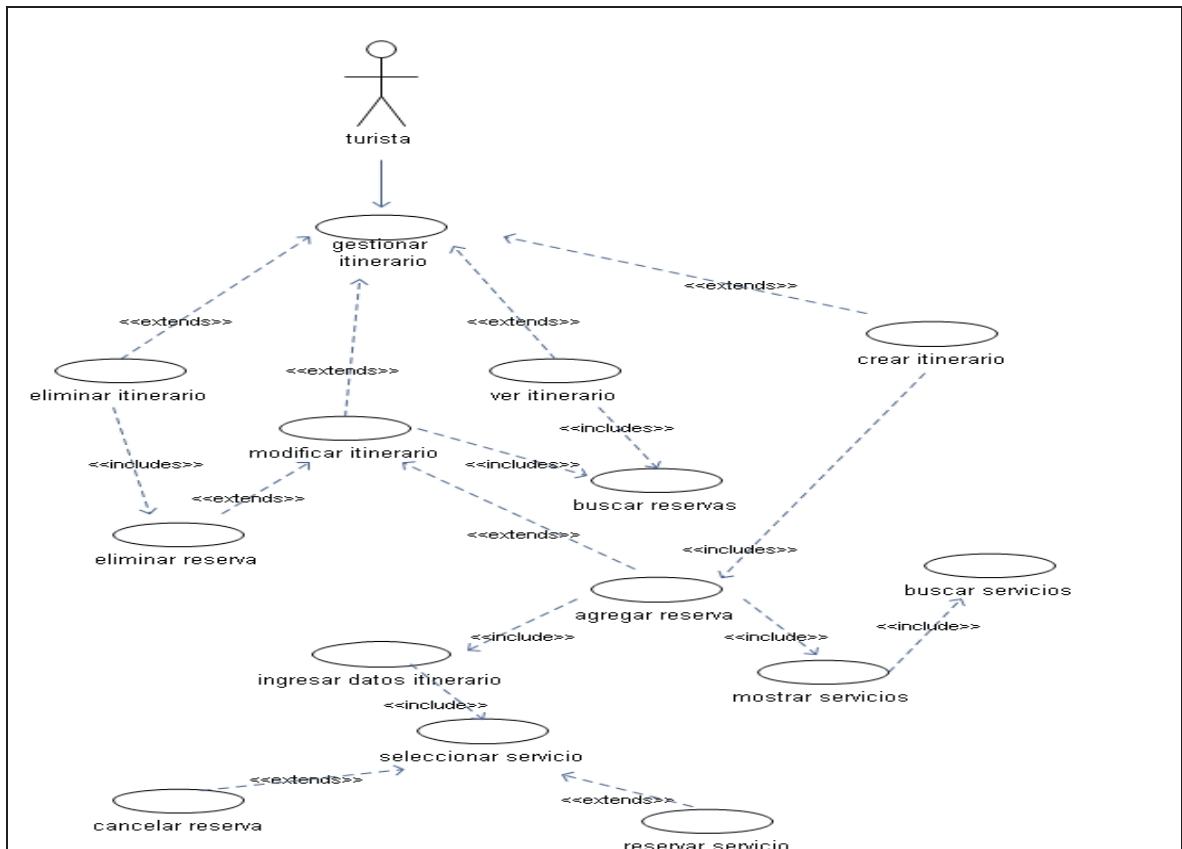


Figura 8.4. Casos de uso expandido gráfico Gestionar Itinerario.

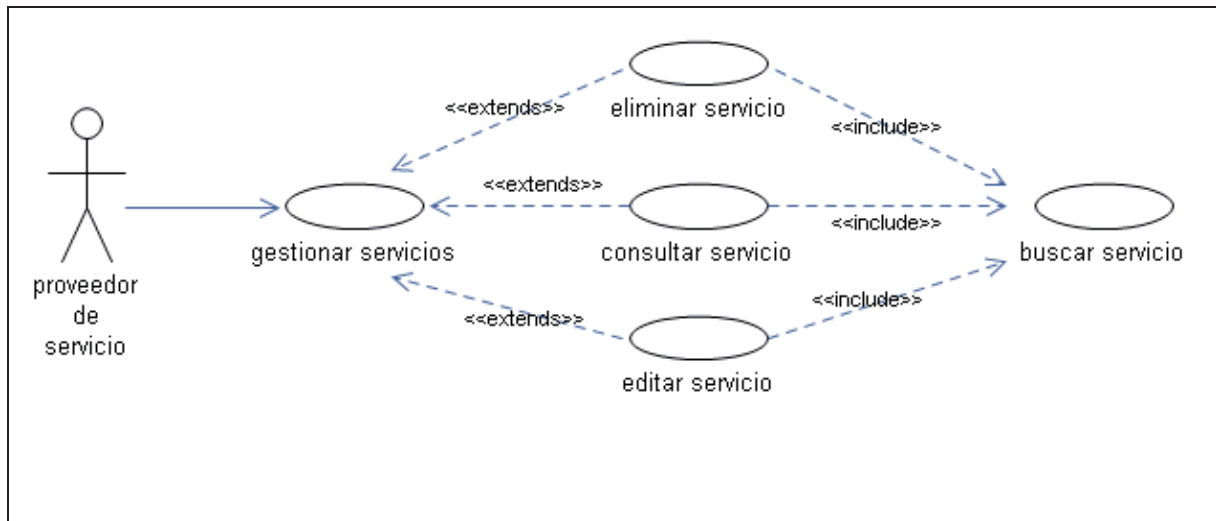


Figura 8.5. Casos de uso expandido gráfico Gestionar Servicio

8.3. Descripción de diseño

Continuando con la fase de elaboración de la metodología de proceso unificado, se pasará a describir la especificación de diseño. El sistema a implementar está dividido en 3 grandes subsistemas, los cuales corresponden a la aplicación cliente, el servicio de planificación (Servicio Web) y los servicios publicados (Servicios Web de proveedores). Se presenta a continuación el diseño del sistema. En el lenguaje de modelado unificado (UML), se encuentra el diagrama de diseño, el cual es representado en un diagrama de clases, el que presenta un acercamiento mayor a la solución propuesta. Para plasmar de mejor manera el diseño del sistema, se ha dividido los diagramas de clases en 3 partes, contemplando el sistema planificador, la aplicación cliente y los proveedores de servicios turísticos. A continuación se presenta el diagrama de diseño de clases para el sistema planificador, el cual es el más importante por implementar la lógica necesaria para la comunicación con los servicios.

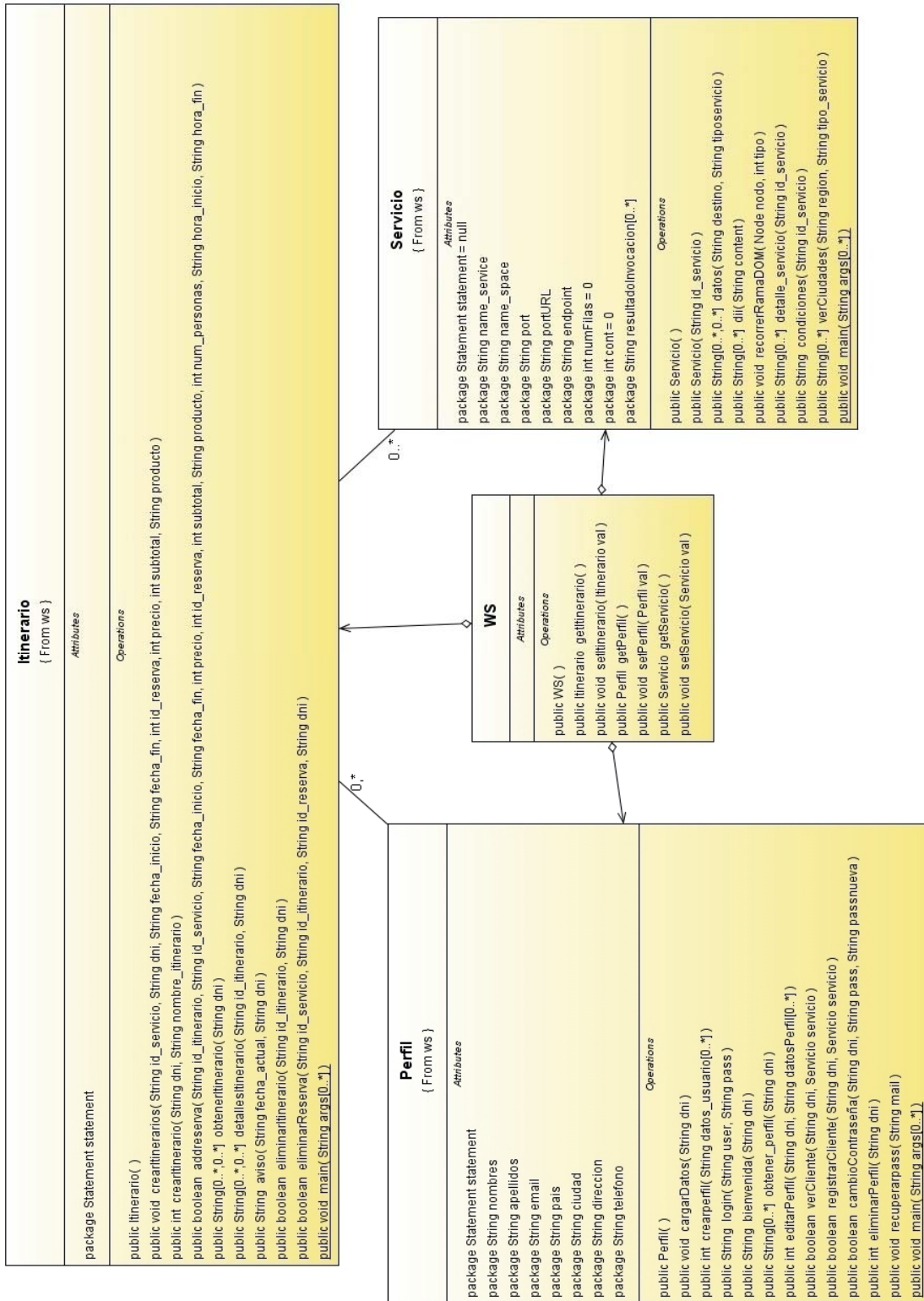


Figura 8.6. Diagrama de diseño de clases Sistema Planificador (A).

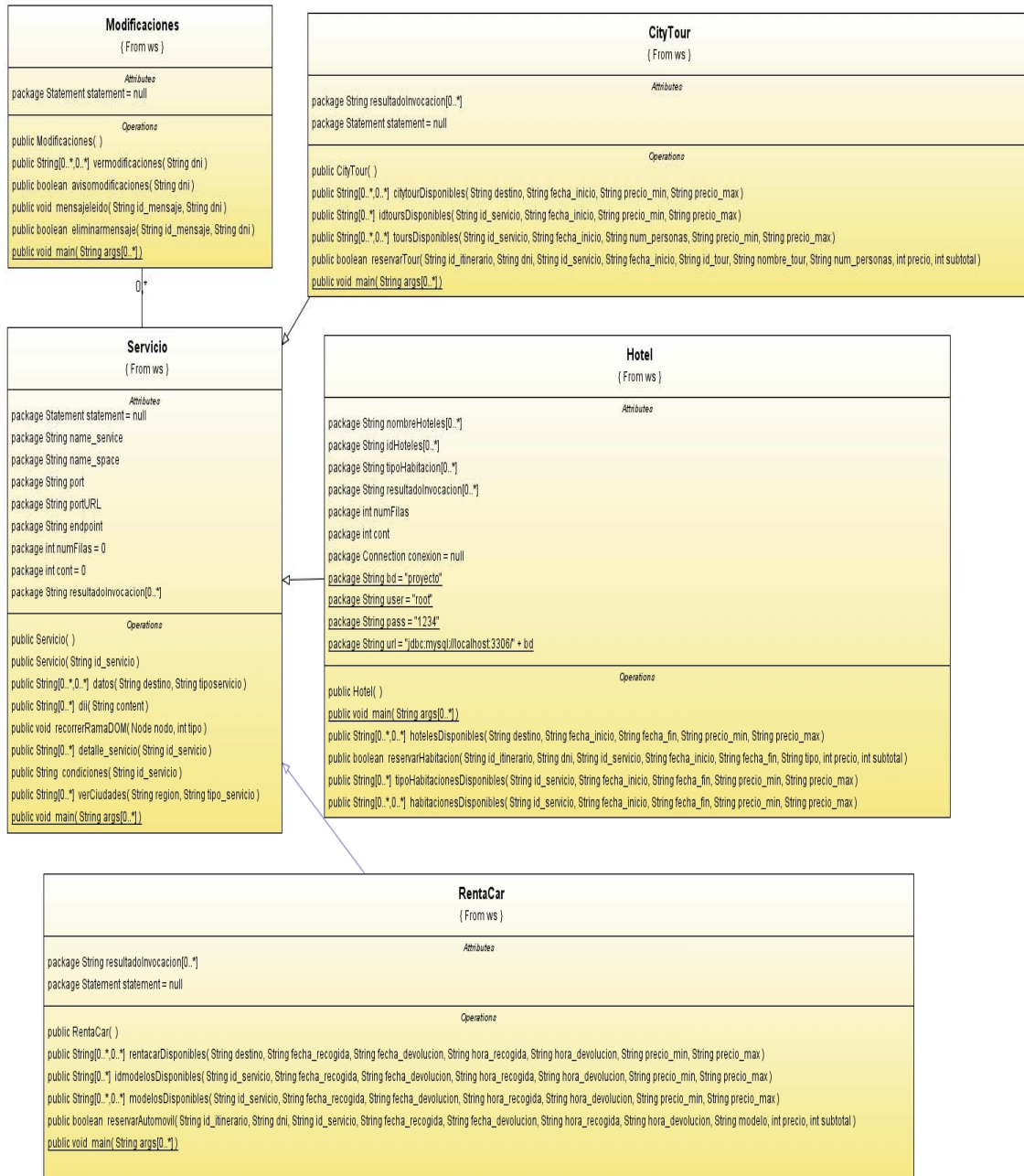


Figura 8.7. Diagrama de diseño de clases Sistema Planificador (B).

Para plasmar de forma más ordenada y dar una mejor perspectiva visual, se visualizará las clases más importantes de forma separada. Una de las clases más importantes corresponde al del servicio web del sistema panificador, el cual contempla las funciones que son consumidas por la aplicación cliente en diferentes instancias. A continuación se presentan las clases del sistema planificador, de forma más detallada.

WS { From ws }
Attributes
Operations
<pre> public boolean verEstado() public String login(String user, String pass) public String bienvenida(String dni) public boolean verDisponibilidad(String id_servicio, String fecha_inicio, String fecha_fin, String tipo_habitacion) public String[] obtener_perfil(String dni) public String[] obteneritinerario(String dni) public String[] habitacionesDisponibles(String id_servicio, String fecha_inicio, String fecha_fin, String precio_min, String precio_max) public String[] hotelesDisponibles(String destino, String fecha_inicio, String fecha_fin, String precio_min, String precio_max) public String[] hoteles(String destino) public String[] habitaciones(String id_servicio) public String[] rentacarDisponibles(String destino, String fecha_recogida, String fecha_devolucion, String hora_recogida, String hora_devolucion, String precio_min, String precio_max) public String[] modelosDisponibles(String id_servicio, String fecha_recogida, String fecha_devolucion, String hora_recogida, String hora_devolucion, String precio_min, String precio_max) public boolean reservarAutomovil(String id_itinerario, String dni, String id_servicio, String fecha_recogida, String fecha_devolucion, String hora_recogida, String hora_devolucion, String modelo, int precio, int subtotal) public int crearitinerario(String dni, String nombre_itinerario) public String[] detalleitinerario(String id_itinerario, String dni) public boolean eliminarReserva(String id_servicio, String id_itinerario, String id_reserva, String dni) public boolean reservarHabitacion(String id_itinerario, String dni, String id_servicio, String fecha_inicio, String fecha_fin, String tipo, int precio, int subtotal) public boolean cambioContraseña(String dni, String pass, String passnueva) public boolean eliminaritinerario(String id_itinerario, String dni) public String[] touroperadoresDisponibles(String destino, String fecha_inicio, String num_personas, String precio_min, String precio_max) public String[] toursoperadores(String id_servicio, String fecha_inicio, String num_personas, String precio_min, String precio_max) public boolean reservarTour(String id_itinerario, String dni, String id_servicio, String fecha_inicio, String id_tour, String nombre_tour, String num_personas, int precio, int subtotal) public String[] verModificaciones(String dni) public boolean eliminamensaje(String id_mensaje, String dni) public String condiciones(String id_servicio) public String[] verCiudad(String region, String tipo_servicio) public String[] detalle_servicio(String id_servicio) public String aviso(String fecha_actual, String dni) public boolean avisomensaje(String dni) public void mensajeleido(String id_mensaje, String dni) public void recuperapass(String email) public int crear_perfil(String datos_usuario[]) public int editarperfil(String dni, String datosperfil[]) public int eliminarperfil(String dni) </pre>

Figura 8.8. Clase Web Services Sistema Planificador.

Parte del sistema también son los servicios web de los proveedores de servicios. Se presentarán los diagramas de diseño de clases para los sistemas de Hotel, Rent a Car y Tours Operadores, correspondientes a las Figuras 8.9, 8.10 y 8.11 respectivamente.

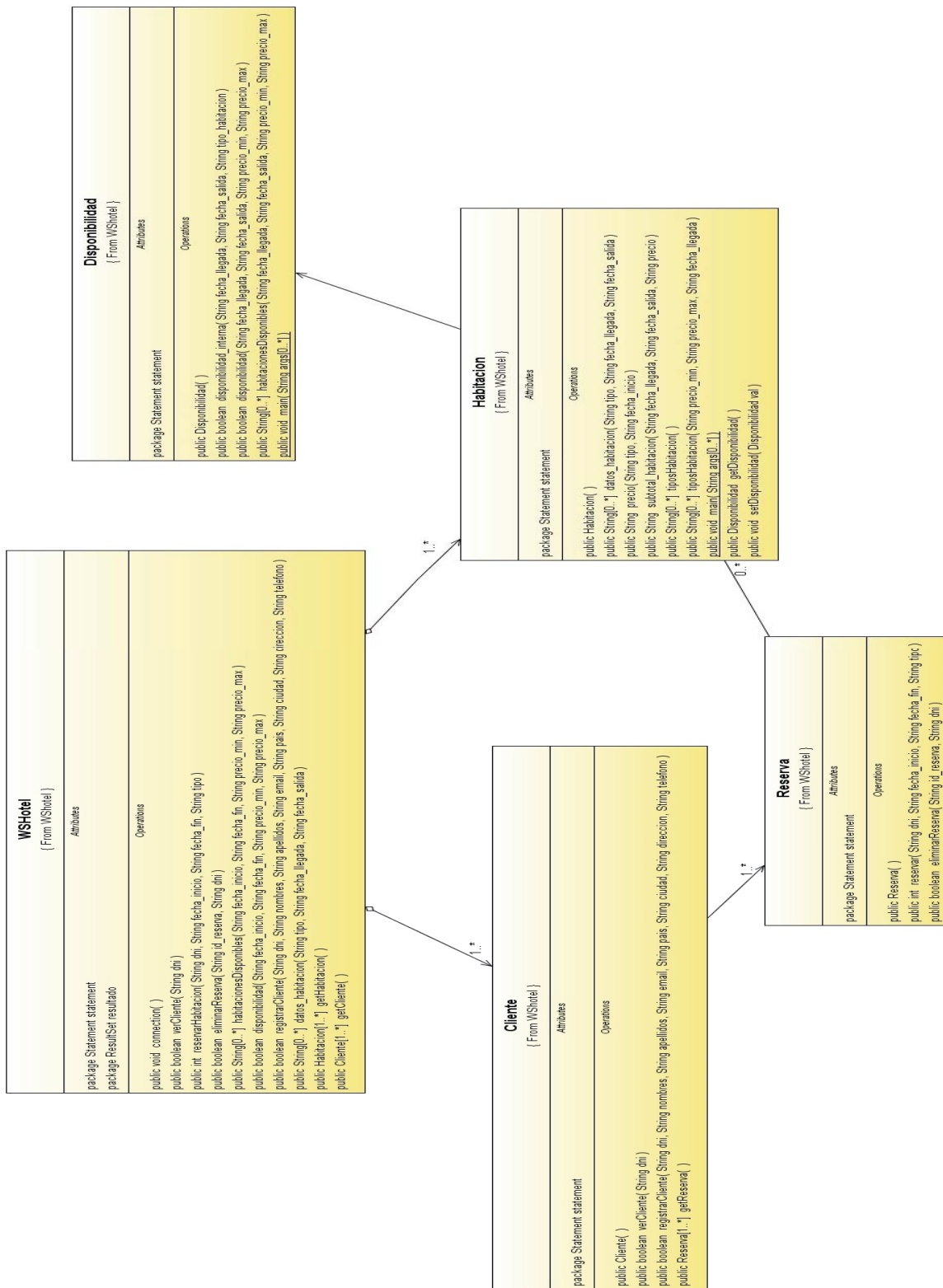


Figura 8.8 Diagrama de Diseño de clase Hotel.

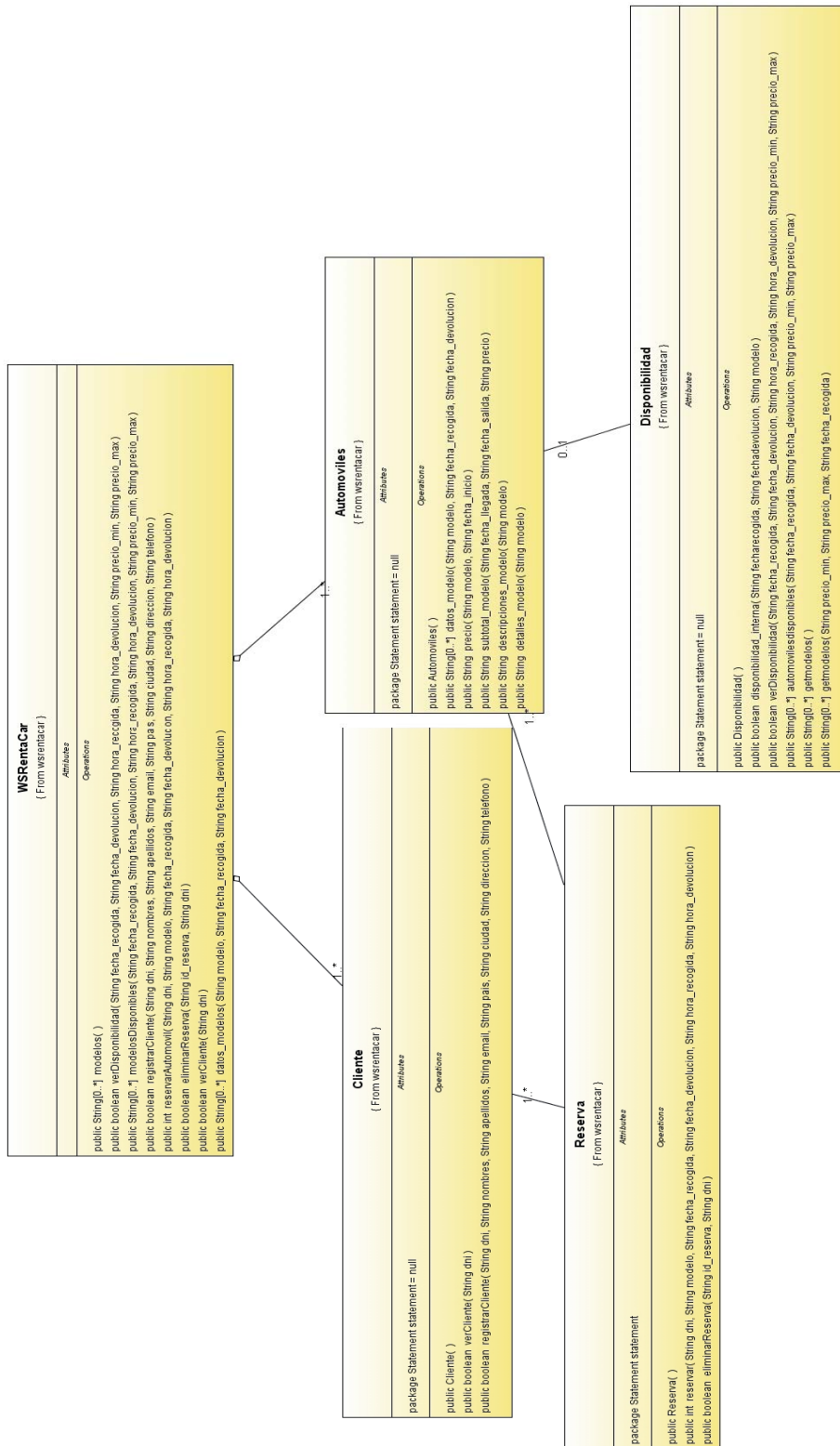


Figura 8.10. Diagrama de Diseño de clase Rent a Car.

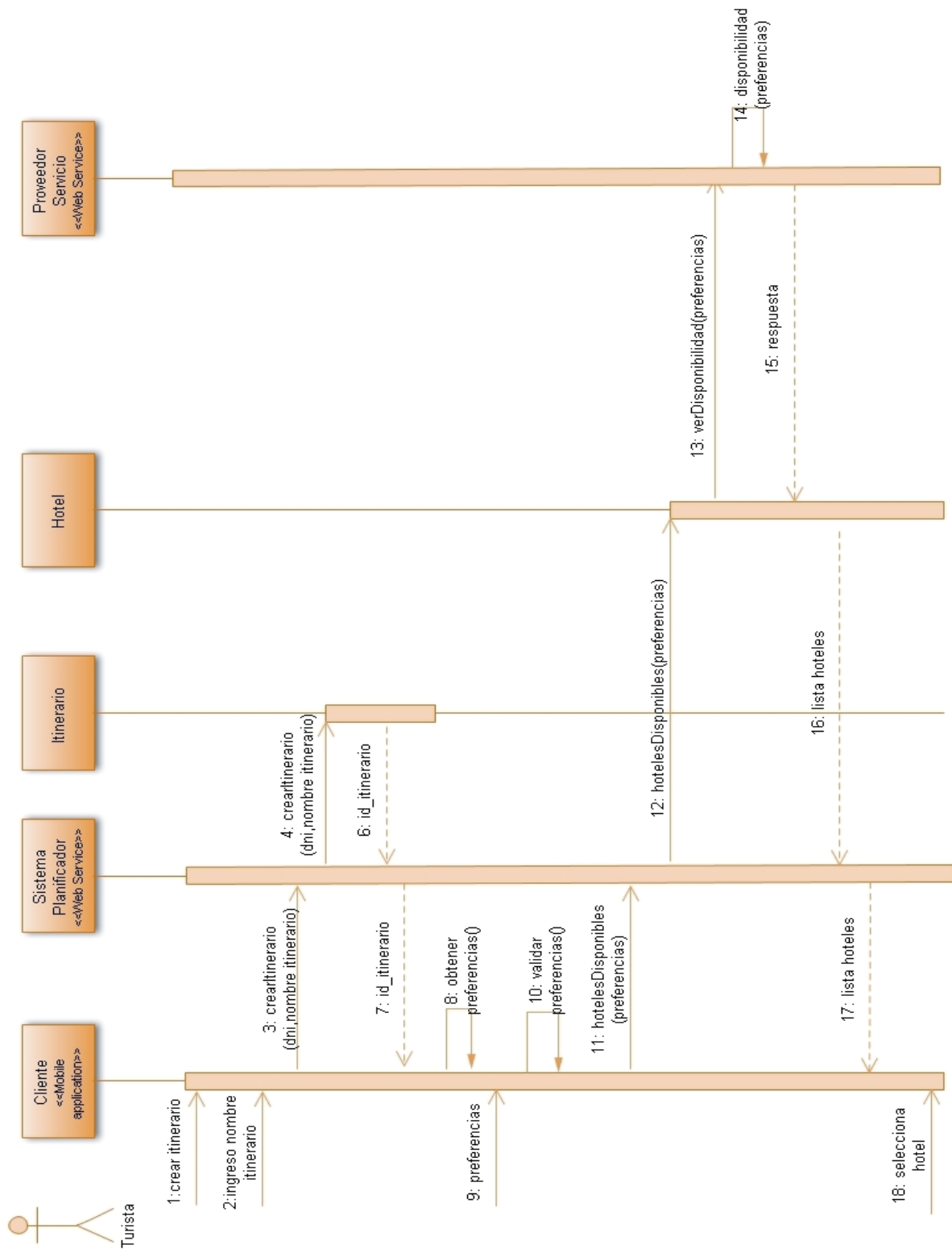


Figura 8.11 Diagrama de secuencias crear itinerario.

Se ha presentado uno de los escenarios principales correspondiente al diagrama de secuencias para la generación de itinerarios de viajes por parte del usuario, la que vendría siendo la piedra angular del presente proyecto. Para plasmar de mejor forma este diagrama, y de modo que sea más entendible al lector, se ha dividido en 2 partes abarcando las Figuras 8.11 y 8.12.

La Figura 8.11 es la inicialización del diagrama de secuencias correspondiente al escenario de creación de itinerario, éste comienza con la autenticación del turista en el sistema seguido de la elección de la opción de la creación del itinerario a través de un dispositivo móvil (Pocket PC), el cuál se comunica con el sistema planificador (Web Service). Una vez autenticado el usuario y seleccionado la opción de crear itinerario, se le solicitan las preferencias para la creación de itinerario, éstas corresponden al ingreso de ciudad de destino, fecha de llegada y fecha de salida. A continuación, se listan los tipos servicios disponibles, previamente publicados por los proveedores de servicios, en seguida el turista selecciona el tipo de servicio. El sistema planificador a través de las preferencias y el tipo de servicio busca los servicios disponibles.

En la Figura 8.12 se representa la búsqueda y la muestra de servicios relacionados con las preferencias del turista, en este caso para la búsqueda de servicios Hoteleros. En la aplicación cliente se listan tales servicios, de modo que el turista puede seleccionarlos, ver sus descripciones, precios además de poder reservarlos. Una vez reservado un servicio por parte del turista, el sistema planificador verifica si el turista está dentro de los registros del proveedor de servicio (Web Service), de no ser así, se registran los datos del turista (perfil) en la base de datos del servicio registrado. Si la reserva es exitosa, el sistema planificador almacena la reserva dentro de la base de datos interna (solo accedida por el sistema planificador), con los datos necesarios para chequear los cambios en los servicios, además de consultar, modificar y cancelar las reservas.

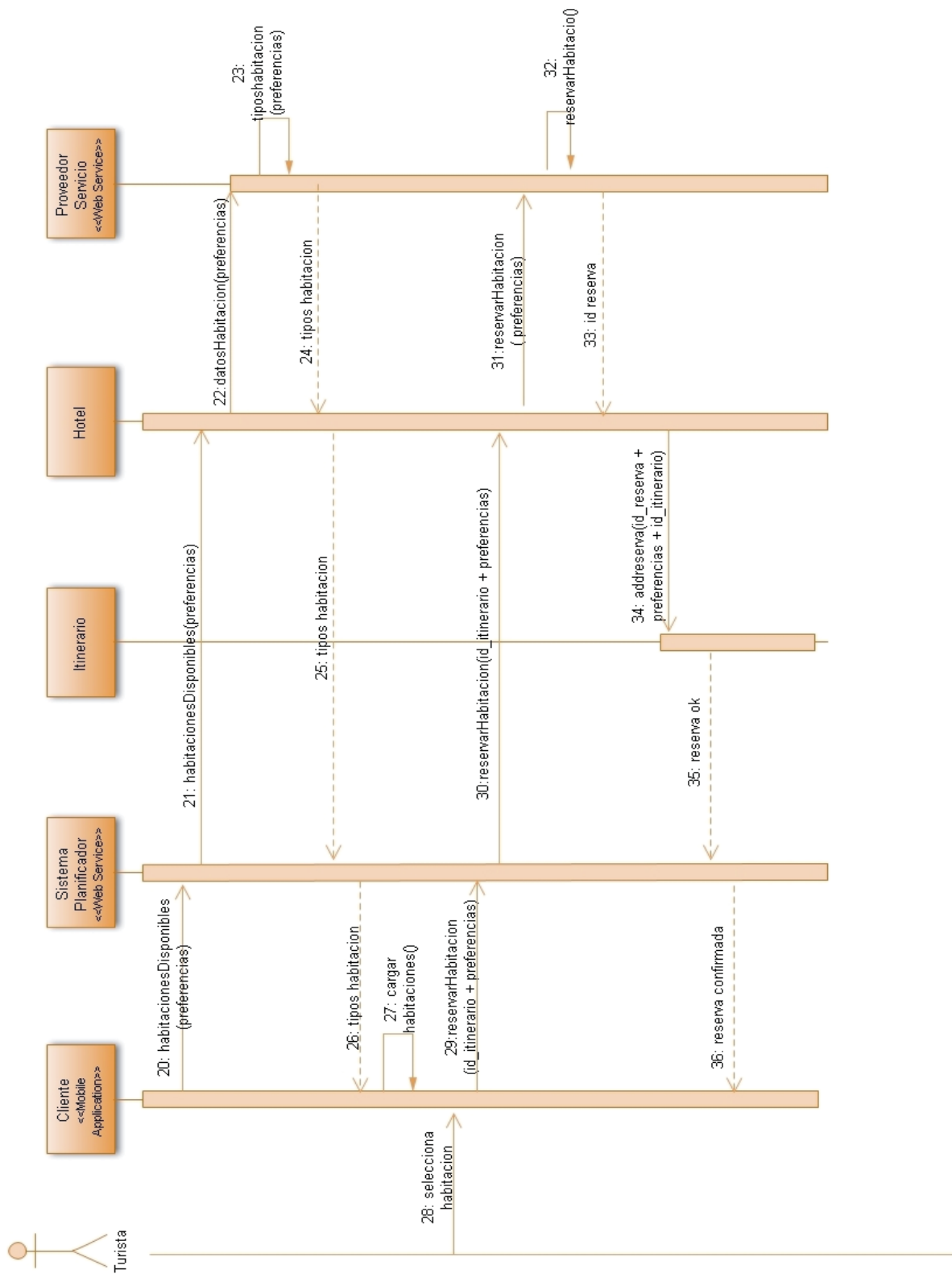


Figura 8.12. Diagrama secuencias crear itinerario (B)

Dentro de los escenarios importantes del sistema consiste en la cancelación de una reserva dentro de un itinerario de viaje. Esto porque dentro de la funcionalidad principal del sistema que es reservar servicios disponibles publicados previamente, el sistema debe ser capaz de cancelar un itinerario completo o bien parte de este, en caso que sea deseado. Para describir de una mejor forma esta funcionalidad se presenta un diagrama de secuencias, donde se cancela la reserva de un servicio parte de un itinerario, el cual corresponde a la siguiente figura.

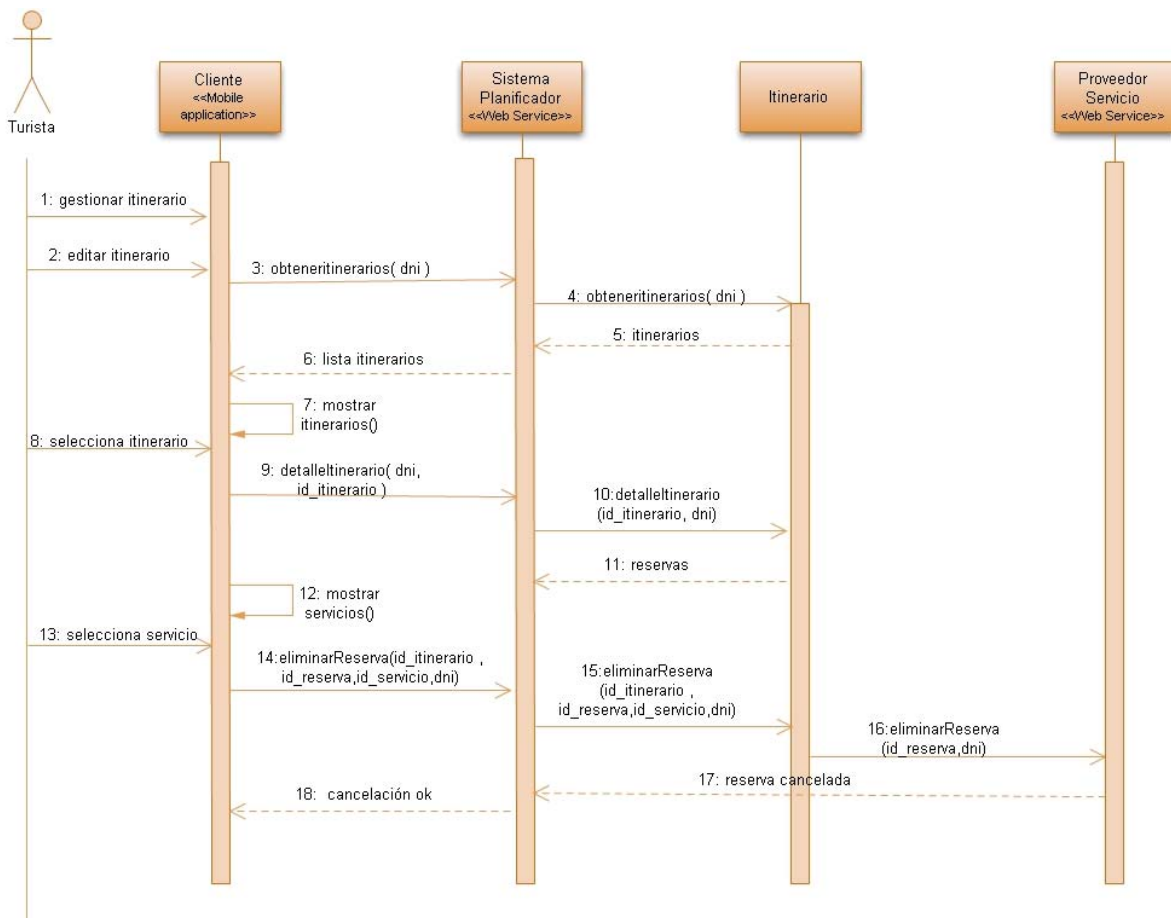


Figura 8.13 Diagrama secuencias eliminar reserva

El escenario representado por la Figura 8.13, es decir, la cancelación de una reserva dentro de un itinerario, la que comienza con la elección de itinerario dentro del menú principal por parte del turista, posteriormente la aplicación cliente solicita los itinerarios creados por el turista al sistema planificador. Éste busca dentro de sus registros en la base de datos local los itinerarios para enviarles estos a la aplicación cliente. Una vez realizado lo anterior, el turista puede seleccionar la reserva que desea eliminar, donde la aplicación cliente le envía el identificador de itinerario (para la base de datos local) y un identificador de la reserva del servicio. Con estos datos el sistema planificador es capaz de eliminar de sus registros la reservación del servicio, además de invocar el servicio web del proveedor de servicio para cancelar la reserva.

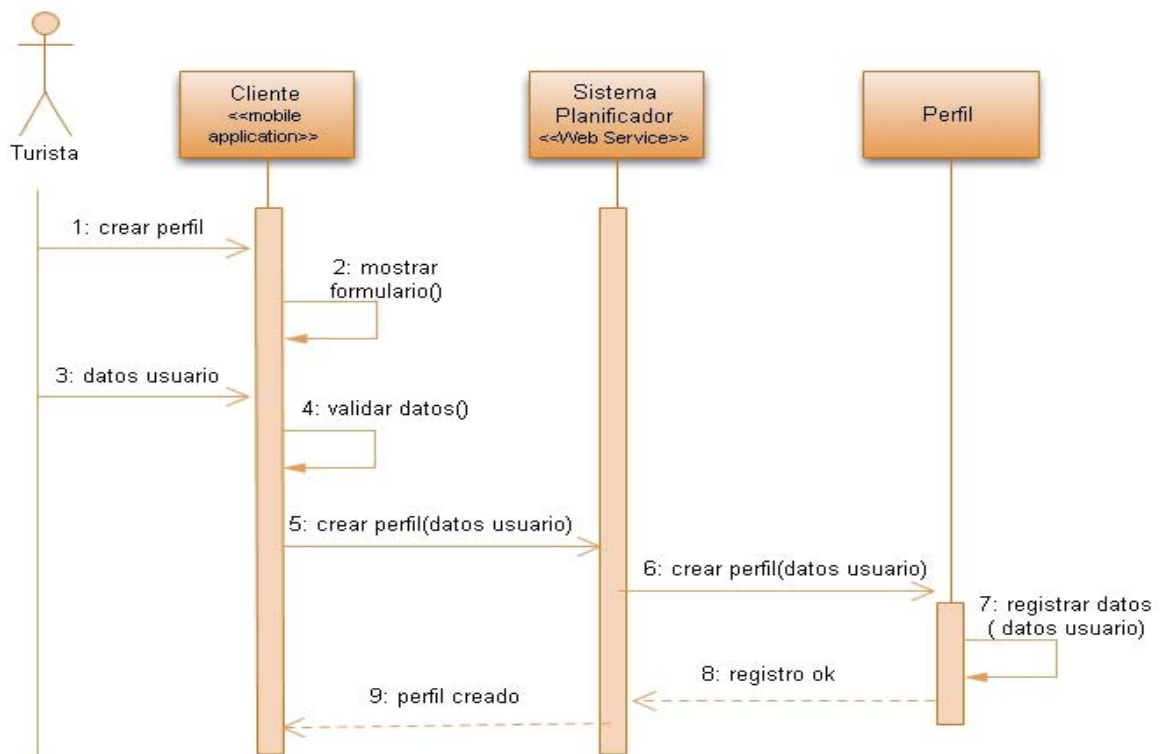


Figura 8.14 Diagrama secuencias crear perfil

La Figura 8.14 corresponde al escenario crear perfil, donde el turista puede registrarse en el sistema para poder realizar las demás actividades otorgadas por el sistema. Este escenario comienza con el ingreso de los datos de un turista en un formulario desde la aplicación cliente. Estos datos son enviados al sistema planificador el cual se encarga de registrar el cliente en caso de no estarlo.

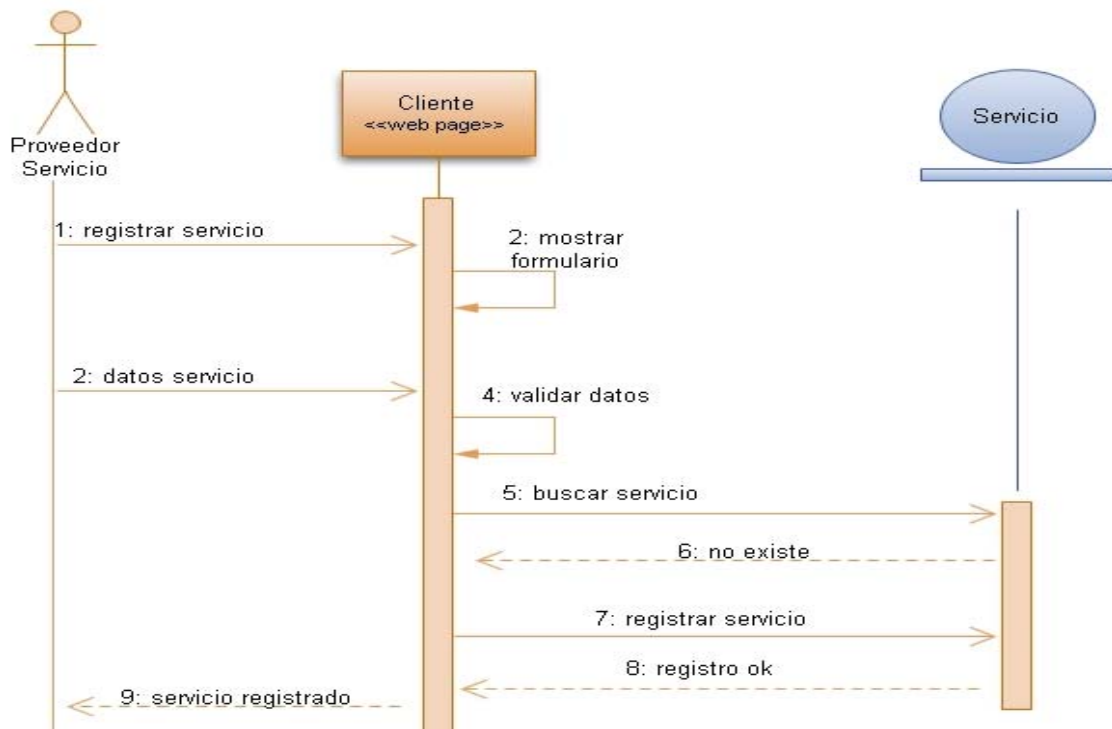


Figura 8.15 Diagrama secuencias registrar servicio

La Figura 8.15 corresponde al escenario publicar servicio, el cual comienza con el ingreso de los datos del servicio a través de un formulario. Los datos ingresados son enviados al sistema planificador el cual registra en una base de datos, por lo que estos datos son necesarios para la invocación futura del sistema planificador al servicio web del proveedor de servicios.

El modelo de composición de servicios es representado a través de un diagrama de actividad. En él se muestra también el flujo para la ejecución de un servicio del negocio pero de una manera más detallada, incluyendo los conceptos de operaciones de actividad y colaboradores del negocio.

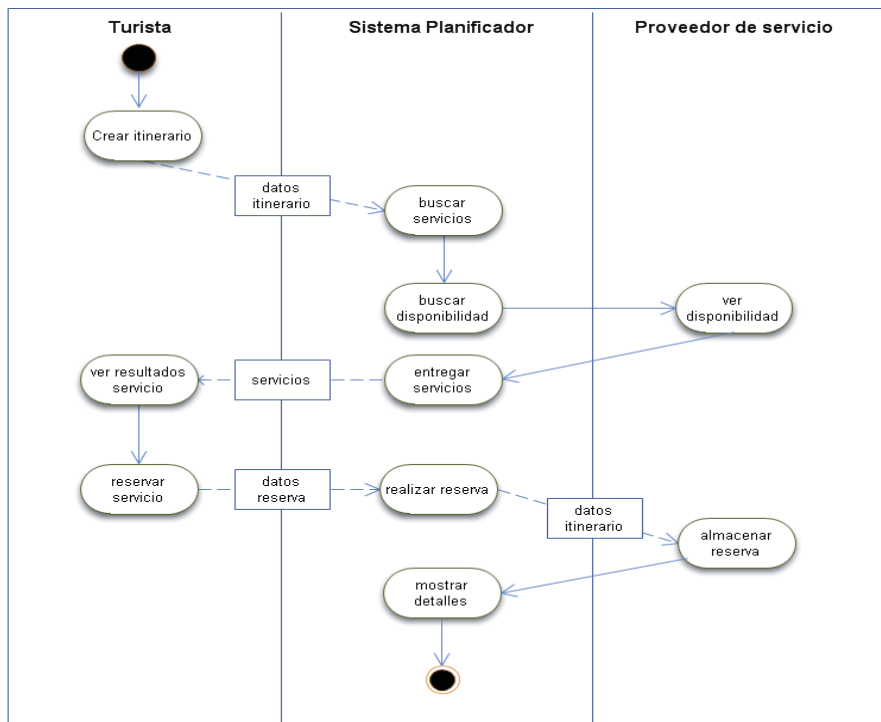


Figura 8.16 Diagrama de actividad crear itinerario.

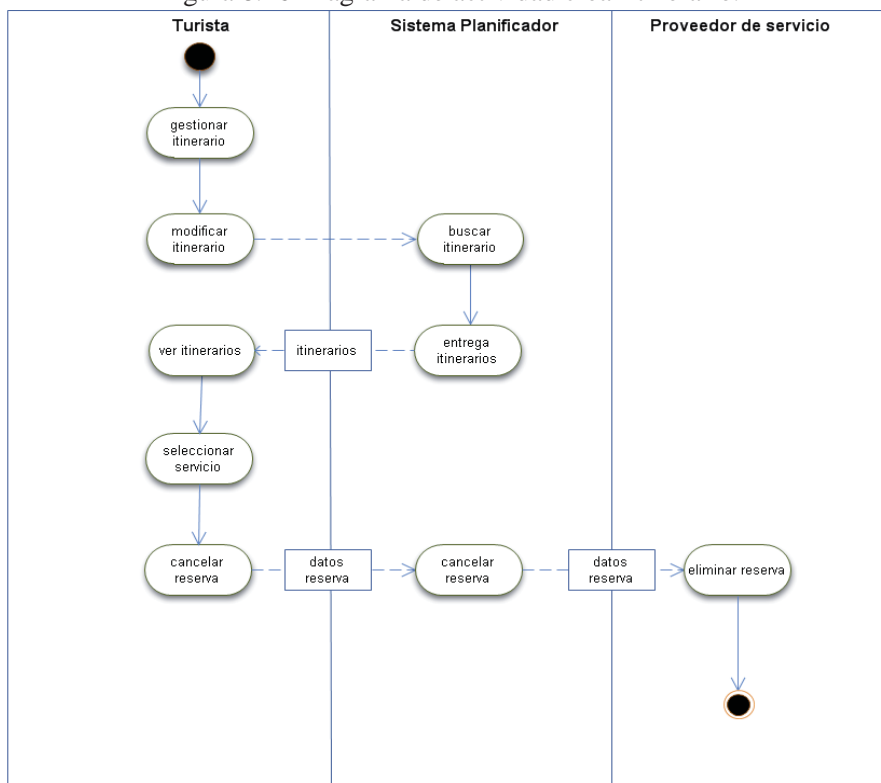


Figura 8.17 Diagrama de actividad correspondiente a cancelar reserva.

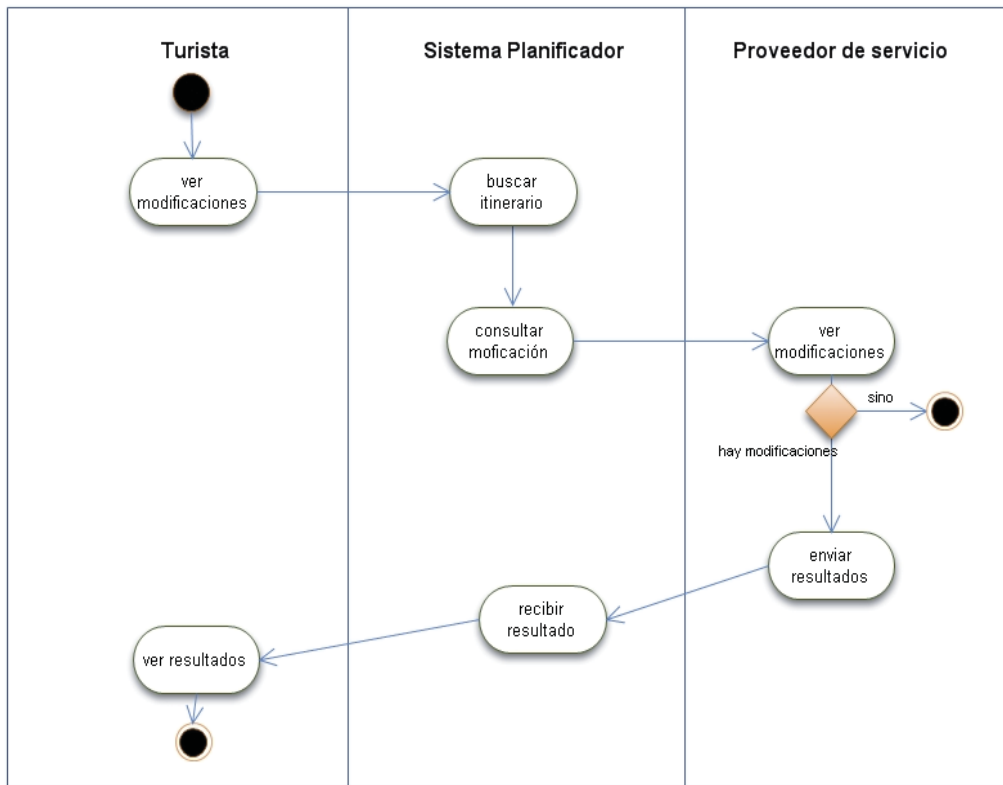


Figura 8.18 Diagrama de actividad correspondiente ver modificaciones.

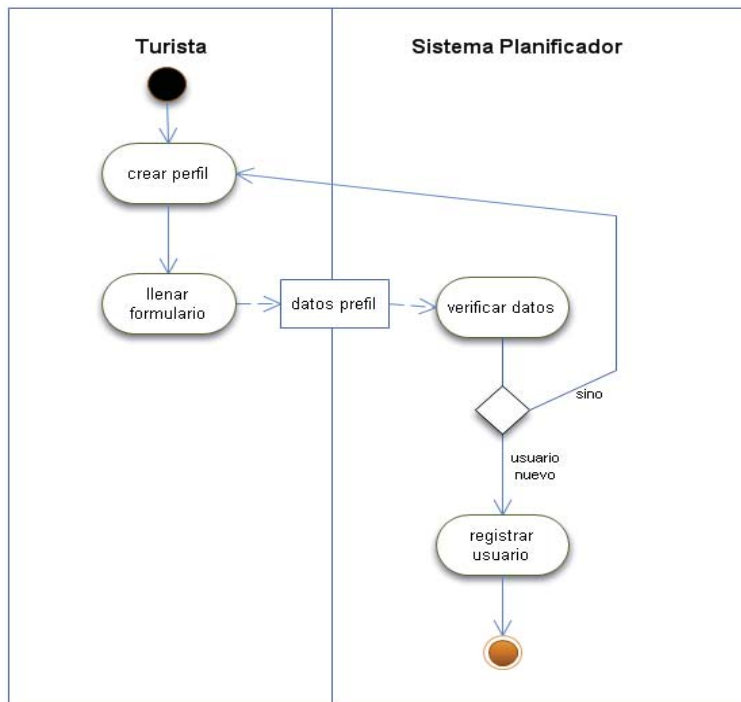


Figura 8.19 Diagrama de actividad crear perfil.

8.2.1. Arquitectura propuesta

Dado que el sistema está basado en Servicios Web comprende la arquitectura de éstos por defecto, es decir, dado del punto de vista de los roles, ésta comprende servidor, cliente y descubrimiento del servicio, los que han sido definidos anteriormente en el capítulo 3. Pero, dada la naturaleza del problema el sistema estaría basado en una arquitectura orientada a servicios o SOA, rescatando así sus principales componentes o elementos. Estos elementos corresponden a:

- Proveedor de servicios (Services Provider): proporcionadas por los proveedores de servicios, como Hoteles, Rent a Car, etc.
- Consumidor de Servicios (Services Consumer): el cual está representado por la aplicación móvil del turista.
- Localizador de Servicios (Services Locator): comprende al registro de un servicio proporcionado por un proveedor de servicios.
- Corredor de Servicio (Service Broker): este elemento corresponde al sistema en sí, es decir, es el encargado de realizar las comunicaciones con los demás servicios, de modo que sea transparente al usuario final.

Invocación de Servicios.

El sistema planificador estará implementado con la API JAX-WS de Java, la cual entrega 3 formas de invocar servicios, dentro de los cuales se destaca la generación de un Stub, Proxy Dinámico y la interface de invocación dinámica o DII, donde las 2 primeras (Stub y proxy dinámico) son de carácter estático. Por lo tanto, para alcanzar un grado de dinamismo y expansión, la invocación será a través de DII en modo PAYLOAD, es decir, se maneja el contenido del cuerpo (Body) de un mensaje SOAP.

Para lograr la invocación dinámica de servicios a través de DII, son necesarios: ServiceName, PortName y el EndPoint del servicio a invocar, los cuales estarán almacenados en una base de datos accedida por el sistema.

Publicación del Servicio.

La publicación del servicio estará comprendida por el registro del ServiceName, PortName y el EndPoint del servicio, parámetros necesarios para lograr la invocación del servicio a través de DII. Además del registro de datos necesarios en cuanto al negocio, como el tipo de servicio, ciudad, etc.

Cliente.

La aplicación cliente estará comprendida por la generación de un Stub del Servicio Web planificador, para poder invocar las funciones proporcionadas por este último.

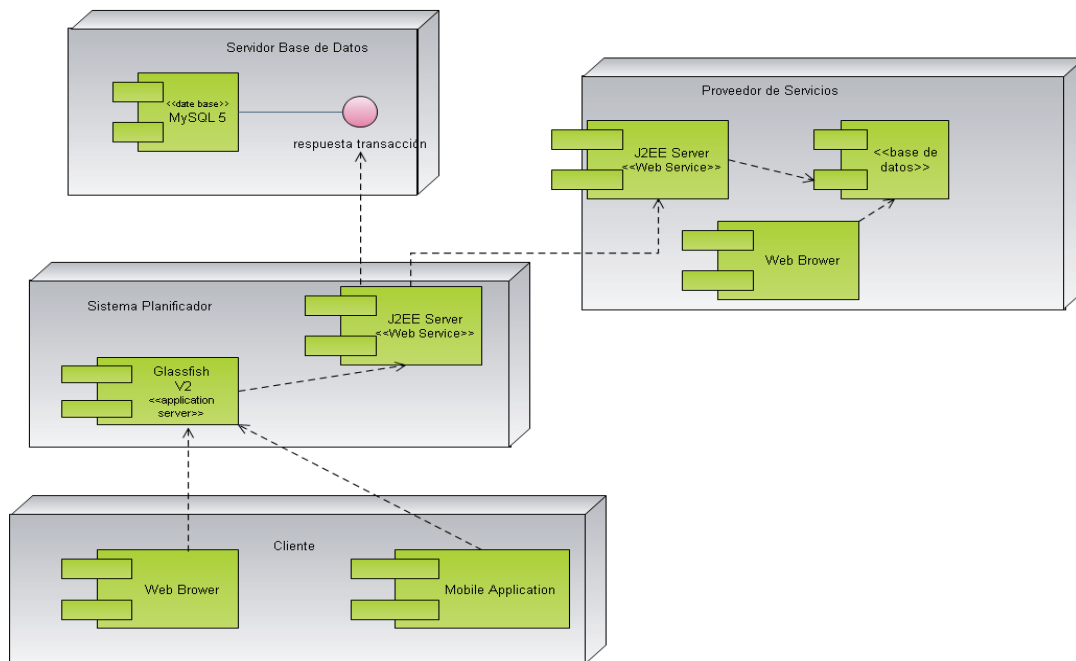


Figura 8.20 Diagrama de Despliegue.

8.3. Elección de Implementación

8.3.1. Elección del IDE

- Son numerosos los IDEs disponibles para realizar aplicaciones en Java. :
- BEA Workshop – software comercial, desarrollado por BEA Systems, integrado con BEA WebLogic.
- BlueJ – libre, desarrollado como un proyecto de investigación universitario. BlueJ es también un entorno interactivo adecuado para aprender Java.
- Eclipse – libre y de código abierto, Eclipse es desarrollado por la Fundación Eclipse.
- IntelliJ IDEA – software comercial, IntelliJ IDEA es desarrollado por JetBrains
- JBuilder – software comercial (existe una versión gratuita). JBuilder es desarrollado por Borland
- JCreator – software comercial(existe una versión gratuita) desarrollado por Xinox
- JDeveloper – IDE gratuito desarrollado por Oracle Corporation e integrado con Oracle Application Server
- NetBeans – IDE y plataforma base para aplicaciones ricas de escritorio (Rich Apps) gratuito de código abierto desarrollado por NetBeans.org
- Sun Java Studio Enterprise – software comercial (gratis para los miembros de Sun Developer Network), desarrollado por Sun Microsystems
- WebSphere Developer & Development Studio – software comercial, desarrollado por IBM, integrado con WebSphere Application Server

Se ha elegido Netbeans por ser Open Source, y por ser uno de los que ofrece más herramientas y opciones para desarrollar diferentes aplicaciones. Es uno de los IDEs más extendidos y dispone de una comunidad de desarrolladores, así como de la posibilidad de creación e instalación de los plugins que se consideren necesarios. Netbeans se ajusta a las necesidades del proyecto, ya que éste ofrece los plugins de Servicios Web, entre los cuales se encuentran JAX-RPC y JAX-WS. Además Netbeans ofrece un buen ambiente para desarrollar aplicaciones para dispositivos móviles (J2ME).

Adicionales

Junto con el IDE para desarrollar las aplicaciones, se usará el emulador de Pocket PC proporcionado por Visual Studio.

8.3.2. Parser XML

Es necesario un parser XML para manejar el contenido de un mensaje SOAP, dentro de los cuales se destacan 3 parser compatibles con java:

- SAX
- DOM
- JDOM

Para manejar el contenido de los mensajes SOAP se ha elegido el parser DOM, ya que este es un modelo de objetos estándar (propuesto por la W3C) que muestra el contenido de un documento XML. Los elementos de un documento XML son representados como nodos en DOM, lo que facilita el manejo de estos.

9. Plan de pruebas

Una de las estrategias para el desarrollo del sistema es el diseño de las pruebas futuras, las cuales ayudarán a tanto a la validación y verificación del proyecto. Por lo tanto, las pruebas servirán para saber si se está construyendo el producto correctamente en el caso de la verificación, y saber si se está desarrollando el producto correcto, de acuerdo a la validación. Si bien probar exhaustivamente el sistema completo es casi imposible debido a que no se pueden evaluar todas las posibilidades, es importante contar con un plan de pruebas para este.

Hay diversos procesos y metodologías para las pruebas de software. La mayoría de ellos distinguen las etapas de planificación, diseño e implementación. Durante la planificación de las pruebas se decide qué se probará y con qué profundidad. En la etapa de diseño de las pruebas funcionales, la especificación se analiza para derivar los casos de prueba y en la última etapa es donde se realiza la ejecución de los casos de pruebas diseñados comparando los resultados reales con los esperados, reportando los resultados.

9.1. Planificación de las pruebas.

Dentro de la planificación del plan de pruebas, se abarcará tanto las pruebas a los módulos ejecutables, y los casos de prueba, es decir, el conjunto de entradas, condiciones y resultados esperados. Dentro del diseño de los casos de prueba se utilizar los enfoques estructural o de cajas blancas, funcionales o de cajas negras.

De acuerdo a los tipos de prueba, se realizaran las pruebas unitarias, de integración y de sistema. Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado, donde los componentes corresponden a subprogramas, las subrutinas, los procedimientos individuales o las clases. Las pruebas de integración tienen como objetivo validar la correcta operación entre los diferentes módulos del sistema, es decir, el objetivo es demostrar que las interfases de cada módulo funcionan correctamente. Por otro lado, las pruebas de sistema verifican el correcto funcionamiento del sistema completo incluyendo casos de prueba que busquen los fallos del sistema; son pruebas destructivas y persiguen demostrar la robustez del sistema aun en condiciones adversas, por lo que verifican requisitos funcionales y no funcionales.

De acuerdo a los tipos de prueba, tanto unitarias, integración y de sistemas serán abordadas de acuerdo a los módulos que comprenden el sistema. De esta forma, las pruebas unitarias se aplicarán tanto a las funciones, clases del sistema planificador, servicios de los proveedores de servicios y aplicaciones cliente, donde se aplicarán los casos de prueba de caja blanca y caja negra. Las pruebas de integración comprenderán la interacción del sistema planificador con los servicios de los proveedores turísticos, y las aplicaciones cliente con el sistema planificador. Por lo tanto, las pruebas del tipo sistema, se probará la interacción de la aplicación cliente con el sistema planificador, y este con los servicios turísticos.

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Usuario	Alfanumérico(15) Formato: xxxxxxxx-y	<ul style="list-style-type: none"> ▪ x entre 0-9 ▪ y entre 0-9 ▪ y valor = 'k' 	<ul style="list-style-type: none"> ▪ En blanco ▪ $x > 9$ ó $x < 0$ ▪ $y > 9$ ó $y < 0$ ▪ y [a- z] - k
Password	Alfanumérico(15)	<ul style="list-style-type: none"> ▪ Cualquier cadena de caracteres alfanumérico mayor de 8 posiciones 	<ul style="list-style-type: none"> ▪ En blanco ▪ Cualquier cadena de caracteres alfanumérico menor de 8 posiciones
DNI	Alfanumérico(15) Formato: xxxxxxxx-y	<ul style="list-style-type: none"> ▪ x entre 0-9 ▪ y entre 0-9 ▪ y valor = 'k' 	<ul style="list-style-type: none"> ▪ En blanco ▪ $x > 9$ ó $x < 0$ ▪ $y > 9$ ó $y < 0$ ▪ y [a- z] - k
ID de Servicio	Alfanumérico(15)	<ul style="list-style-type: none"> ▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones 	<ul style="list-style-type: none"> ▪ En blanco
Fecha Inicio	Alfanumérico(10) Formato: yyyy/mm/dd	<ul style="list-style-type: none"> ▪ yyyy > 1950 ▪ mm entre 1-12 ▪ dd entre 1-31 	<ul style="list-style-type: none"> ▪ En blanco ▪ yyyy < 1950 ▪ mm > 12 ó mm < 0 ▪ dd > 31 ó dd < 0
Fecha Inicio	Alfanumérico(10) Formato: yyyy/mm/dd	<ul style="list-style-type: none"> ▪ yyyy > 1950 ▪ mm entre 1-12 ▪ dd entre 1-31 	<ul style="list-style-type: none"> ▪ En blanco ▪ yyyy < 1950 ▪ mm > 12 ó mm < 0 ▪ dd > 31 ó dd < 0

Tipo Habitación	Alfanumérico(15)	<ul style="list-style-type: none"> ▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones 	<ul style="list-style-type: none"> ▪ En blanco
ID de Itinerario	Numérico(15)	<ul style="list-style-type: none"> ▪ Cualquier valor numérico > a 0. 	<ul style="list-style-type: none"> ▪ En blanco ▪ Valor < 0.
ID de Reserva	Numérico(15)	<ul style="list-style-type: none"> ▪ Cualquier valor numérico > a 0. ▪ Valor tipo entero 	<ul style="list-style-type: none"> ▪ En blanco ▪ Valor < 0. ▪ Distinto tipo entero
Destino	Alfanumérico(20)	<ul style="list-style-type: none"> ▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones 	<ul style="list-style-type: none"> ▪ En blanco
Precio Mínimo	Numérico	<ul style="list-style-type: none"> ▪ Valor > a 0 ▪ Valor tipo entero 	<ul style="list-style-type: none"> ▪ En blanco ▪ Valor < 0. ▪ Distinto tipo entero
Precio Máximo	Numérico	<ul style="list-style-type: none"> ▪ Valor > a 0 ▪ Valor tipo entero 	<ul style="list-style-type: none"> ▪ En blanco ▪ Valor < 0. ▪ Distinto tipo entero
Nombres	Alfanumérico(30)	<ul style="list-style-type: none"> ▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones 	<ul style="list-style-type: none"> ▪ En blanco ▪ Posea algún valor numérico
Apellidos	Alfanumérico(30)	<ul style="list-style-type: none"> ▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones 	<ul style="list-style-type: none"> ▪ En blanco ▪ Posea algún valor numérico
País	Alfanumérico(20)	<ul style="list-style-type: none"> ▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones 	<ul style="list-style-type: none"> ▪ En blanco ▪ Posea algún valor numérico

Ciudad	Alfanumérico(20)	<ul style="list-style-type: none">▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones	<ul style="list-style-type: none">▪ En blanco▪ Posea algún valor numérico
Dirección	Alfanumérico(40)	<ul style="list-style-type: none">▪ Cualquier cadena de caracteres alfanumérico mayor de 1 posiciones	<ul style="list-style-type: none">▪ En blanco
Email	Alfanumérico(40) Formato: x@y.z	<ul style="list-style-type: none">▪ x, y, z cualquier cadena de caracteres alfanumérico mayor de 1 posiciones	<ul style="list-style-type: none">▪ En blanco

9.1. Tabla Análisis de Límites.

10. Detalles de la solución

En este capítulo se presentarán los detalles generales de la aplicación final, donde ya se da por hecho que se han realizado y representados prototipos de la aplicación en los informes de avances anteriores correspondientes a la fase de construcción de la metodología de proceso unificado.

La solución total está dividida en 3 partes importantes a destacar, las cuales se dividen en la aplicación cliente que está implementada en ambiente móvil, el sistema planificador y, por último los servicios web correspondientes a los proveedores de servicios turísticos como servicios de Hotelería, alquiler de automóviles y tours. A continuación se detallarán los 3 entes principales que componen el sistema desarrollado.

10.1. Sistema Coordinador

Como se ha mencionado anteriormente, una parte del sistema es el sistema coordinador, el cual, contiene las funciones para la autenticación de usuario, creación y administración de perfil de usuario, funciones necesarias para la reserva de servicios turísticos disponibles, además del manejo de Itinerarios. Esta parte del sistema corresponde a un servicio web que cuenta con el acceso a una base de datos propio donde se almacenan los perfiles de usuario, itinerarios y servicios publicados.

El sistema coordinador es la piedra angular del sistema total, ya que está encargado de realizar las funciones principales. Es el encargado de filtrar las preferencias de un turista para encontrar productos turísticos disponibles, además de establecer la comunicación con los servicios web de los proveedores de forma dinámicamente.

A continuación se presenta una Figura donde se aprecia un testing de servicio webs proporcionado por el contenedor de aplicaciones de JAVA Glassfish, en el cual se puede invocar a las funciones desplegadas.

WSService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String ws.WS.login(java.lang.String,java.lang.String)

(,)

public abstract java.util.List ws.WS.nombreHoteles(java.lang.String)

()

public abstract boolean ws.WS.crearPerfil(java.util.List)

()

public abstract boolean ws.WS.verEstado()

()

public abstract java.lang.String ws.WS.bienvenida(java.lang.String)

Figura 10.1. Tester de Web Service Coordinador.

10.1.1. Invocación Dinámica de Servicios

La invocación o consumo de servicios web disponibles de los proveedores de servicios a través del sistema coordinador se realiza de forma dinámica. Para ello se necesitan ciertos parámetros del proveedor de servicio como el endpoint, namespace, y el port del servicio web a consumir.

La invocación dinámica de servicios web, se lleva a cabo a través de la construcción de mensajes de SOAP de solicitud por parte del programador. Para ello, hay 2 formas de manejar la construcción de mensajes SOAP, los cuales son MESSAGE y PAYLOAD. La construcción de mensajes SOAP de solicitud para este proyecto se realizará a través del modo PAYLOAD, donde se maneja solo el cuerpo (<<body>>) del mensaje SOAP. A continuación se presenta el código fuente.

```

import java.io.*;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import javax.xml.namespace.QName;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.stream.StreamSource;
import javax.xml.ws.Dispatch;
import javax.xml.ws.Service;
import javax.xml.ws.soap.SOAPBinding;
import org.w3c.dom.*;
import org.xml.sax.*;
import javax.xml.parsers.*;
import java.sql.*;

public class Servicio {
    String name_service,name_space,port,portURL,endpoint;

    public String[] dinamycInterfaceInvocation(String content){

        try {
            // Se define el servicio.
            QName svcQname = new QName(name_space,name_service);
            QName portQName = new QName(portURL,port);
            Service svc = Service.create(svcQname);
            svc.addPort(portQName,SOAPBinding.SOAP11HTTP_BINDING,endpoint);

            // Se Crea el objeto de invocación dinámica desde el servicio (svc).
            Dispatch<Source> dispatch =
svc.createDispatch (portQName,Source.class,Service.Mode.PAYLOAD);

            ByteArrayInputStream bais = new ByteArrayInputStream(content.getBytes());
            Source input = new StreamSource(bais);

            // Se invoca la operacion.
            Source output = dispatch.invoke(input);

            // Se procesa la respuesta.
            StreamResult result = new StreamResult(new ByteArrayOutputStream());
            Transformer trans = TransformerFactory.newInstance().newTransformer();
            trans.transform(output, result);
            ByteArrayOutputStream baos = (ByteArrayOutputStream) result.getOutputStream();

            // Se parsea el mensaje de respuesta.
            DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            Document doc = builder.parse(new InputSource(new StringReader(new
            String(baos.toByteArray()))));

            numFilas=0;cont=0;
            recorrerRamaDOM(d,1);
            resultadoInvocacion = new String[numFilas];
            recorrerRamaDOM(d,2);

```

```
return resultadoInvocacion;
} catch (Exception e) {}
return null;
}

// se recorre el mensaje SOAP de respuesta a través de una parser DOM

public void recorrerRamaDOM(Node nodo, int tipo) {
    if (nodo != null) {
        if (nodo.getNodeName().equals("#text")) {
            if (tipo == 1) numFilas++;
            if (tipo == 2) {
                resultadoInvocacion[cont] = nodo.getNodeValue();
                cont++;
            }
            System.out.println("valor nodo: " + nodo.getNodeValue());
        }
        // System.out.println("nombre nodo: " + nodo.getNodeName());
        NodeList hijos = nodo.getChildNodes();
        for (int i = 0; i < hijos.getLength(); i++) {
            Node nodoNieta = hijos.item(i);
            recorrerRamaDOM(nodoNieta, tipo);
        }
    }
}
```

10.1.2. Invocación Asíncrona de Servicios

Además de la invocación dinámica de servicios por parte del sistema coordinador, para optimizar el consumo e invocación a los servicios web de los proveedores de servicios se ha implementado la invocación asíncrona de servicios. Con esta modalidad se reduce de forma considerable los tiempos de respuestas del planificador hacia a la aplicación cliente. La invocación asíncrona corresponde a que las respuestas de los servicios web de los proveedores turísticos se procesan mientras se realizan otras actividades como el llamado a otros servicios. A continuación se presenta un extracto del código fuente mediante el cual se ejecutan las llamadas a servicios de manera asíncrona.

```

public void dii(int index){
    if(index < servicios.length){

        try {
            // Se define el servicio.
            QName svcQname = new QName(servicios[index][8],servicios[index][7]);
            QName portQName = new QName(servicios[index][10],servicios[index][9]);
            Service svc = Service.create(svcQname);
            svc.addPort(portQName,SOAPBinding.SOAP11HTTP_BINDING,servicios[index][11]);

            // Se Crea el objeto de invocación desde el servicio (svc).

            Dispatch<Source> dispatch =
            svc.createDispatch(portQName,Source.class,Service.Mode.PAYLOAD);
            String contenido;
            if(this.tipo_servicio.equals("hotel")) contenido = "<ns2:disponibilidad
xmlns:ns2=\""+servicios[index][8]+"\">\" + content;
            else contenido = "<ns2:verDisponibilidad xmlns:ns2=\""+servicios[index][8]+"\">\" + content;
            ByteArrayInputStream bais = new ByteArrayInputStream(contenido.getBytes());
            Source input = new StreamSource(bais);

            // Se invoca la operacion.

            Response<Source> response = dispatch.invokeAsync(input);

            while (!response.isDone()){
                int anterior = index +1;
                if (anterior < servicios.length)dii(anterior);
            }
            try {
                //get the actual result
                Source output = (javax.xml.transform.Source)response.get();
                //do something with the result
                StreamResult result = new StreamResult(new ByteArrayOutputStream());
                Transformer trans = TransformerFactory.newInstance().newTransformer();
                trans.transform(output, result);
                ByteArrayOutputStream baos = (ByteArrayOutputStream) result.getOutputStream();

                // Se parsea el mensaje de respuesta.
                DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
                Document d = builder.parse(new InputSource(new StringReader(new String(baos.toByteArray()))));
                System.out.println("index :"+index+ " id_ servicio :"+servicios[index][0]);

                recorrerRamaDOM(d,index);

                //return resultadoInvocacion;
            } catch (Exception ex){
                resultadoInvocacion[index] = "false";
                int anterior = index + 1; dii(anterior);
            }
        } catch (Exception ex){

```

```
        resultadoInvocacion[index] = "false";
        int anterior = index + 1; dii(anterior);}

    }

}

public void recorrerRamaDOM(Node nodo,int index){
    if(nodo!=null){
        if(nodo.getNodeName().equals("#text")){
            resultadoInvocacion[index] = nodo.getNodeValue();

            System.out.println("valor nodo: "+nodo.getNodeValue());
        }
        //System.out.println("nombre nodo: "+nodo.getNodeName());
        NodeList hijos = nodo.getChildNodes();
        for(int i=0; i< hijos.getLength();i++){
            Node nodoNieto = hijos.item(i);
            recorrerRamaDOM(nodoNieto,index);
        }
    }
}
```

10.2. Aplicación cliente

En cuanto a la aplicación cliente utilizada por el turista, se comunicará con el servicio web coordinador invocando las funciones de éste. El consumo de funciones del servicio web coordinador por parte de la aplicación cliente está dado por la construcción de una referencia estática (stub) del servicio, donde el constructor de este, está dado por la dirección del documento de descripción (WSDL) del servicio web coordinador. El stub generado a partir del WSDL del servicio, se puede instanciar como una clase cualquiera, es decir, se puede construir objetos a través de ella.

Siguiendo la especificación de diseño, el turista podrá contar con un menú principal que abarque las funcionalidades principales, y representadas de una forma simple.

La Figura 10.2 corresponde al menú principal de la aplicación cliente, los cuales se dividen en más submenús.



10.2. Interfaz de menú principal de aplicación móvil.

▪ Menú Itinerarios

La creación de itinerarios por parte del turista se realiza a través de preferencias. Estas preferencias están dadas por un rango de fechas del que el turista desee ver y/o reservar un servicio disponible, además de seleccionar el destino de donde se encuentre dichos servicios. Por otro lado, para hacer un filtrado de preferencias del usuario más fino, el turista puede seleccionar un rango de precios de los cuales estén dentro de presupuesto o deseo de los servicios disponibles a ver y/o reservar. La siguiente Figura representa un formulario de preferencias para la búsqueda de hoteles disponibles en un destino requerido.

The screenshot shows the 'Buscar Servicios' (Search Services) form in the mobile application. The title bar shows 'Planificador Movil' and the time 11:59. The form has a dark blue header with the title 'Buscar Servicios'. Below the header, there are three dropdown menus: 'tipo servicio' (set to 'Hotel'), 'Region' (set to 'X Region'), and 'Ciudad' (set to 'Puerto Montt'). Below these are two date pickers: 'check in' (set to '13/07/09') and 'check out' (set to '14/07/09'). There is a checkbox labeled 'buscar por precio' which is currently unchecked. At the bottom, there is a blue bar with 'volver' and 'salir' buttons, and a keyboard icon.

10.3. Interfaz de formulario de búsqueda de servicios.

Para el listado de productos turísticos, estos se mostraran por pantalla, con sus respectivas características, tales como precios y servicios extras asociados al producto mostrado, además de una imagen del producto. Por ejemplo, para el caso de un Hotel, los productos son las habitaciones, las cuales se mostraran de acuerdo el tipo de habitación, acompañado de su respectiva descripción, precio e imagen como lo muestra la siguiente Figura. Estos productos son mostrados de acuerdo a las preferencias ingresadas anteriormente por el turista, por lo cual podrían no aparecer los productos no disponibles en las fechas requeridas, o las que no estén en un rango de precio que haya estipulado el usuario. Una vez seleccionado un producto turístico como una habitación, automóvil o un Tour, se podrán reservar dichos productos, los cuales se asociarán a un itinerario previamente creado por el usuario o turista en este caso.

A continuación se presentara las interfaces correspondientes a los listados de productos turísticos como habitaciones para el caso de Hoteles, automóviles para el caso de Rent a Cars y Tours para los Tours operadores.



10.4. Interfaz de productos.

El turista podrá ver los itinerarios creados, modificarlos o eliminarlos. Cada itinerario se divide como una carpeta con un nombre específico en la interfaz, en la cual contiene las reservas realizadas para tal itinerario, y además se enumeran la cantidad de reservas por itinerario. Las reservas son presentadas con íconos diferentes de acuerdo al tipo de producto seleccionado, además de poder ver los detalles las reservas como el precio, rango de fechas y destino. También, puede verse los detalles de los proveedores de los productos turísticos que han sido seleccionados, ésta información corresponde a los datos de la empresa como el nombre, dirección, teléfono, email, página web y políticas de reservas.

A continuación se presenta la interfaz de itinerarios creados.



10.5. Interfaz de itinerarios creados.

Por otro lado, el turista podrá eliminar las reservas hechas previamente, las que se podrán realizar de 2 formas. Estas formas corresponden a la eliminación de reservas de forma unitaria o eliminar un itinerario que contienen reservas registradas, pasando así a eliminar todas las reservas asociadas a dicho itinerario.



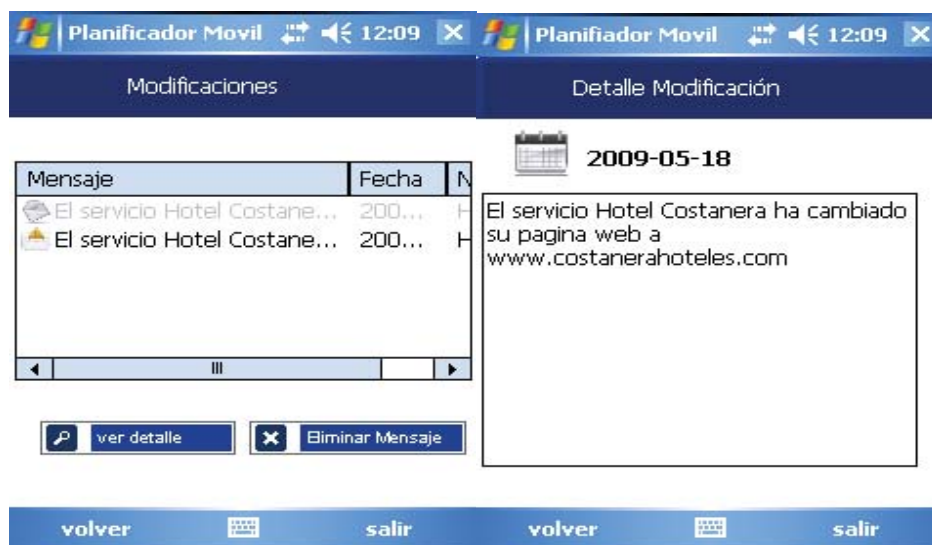
10.6. Interfaz detalle de reservas.

- **Menú Perfil Usuario.**

Este submenú corresponde a la gestión de los datos del usuario, en el cual se podrá consultar los datos personales, editar estos datos personales, modificar la contraseña de para la autenticación en el sistema y poder darse de baja en el sistema.

- **Menú Modificaciones.**

El menú de modificaciones corresponde al listado de posibles modificaciones realizadas por los proveedores turísticos que estén asociados a algún producto reservado por el turista. Las modificaciones están representadas en forma de bandeja de entrada de un servidor de correo normal. Donde las modificaciones no leídas por el usuario son representadas por un sobre vacío, y las modificaciones leídas por un sobre abierto como lo presenta la siguiente Figura.

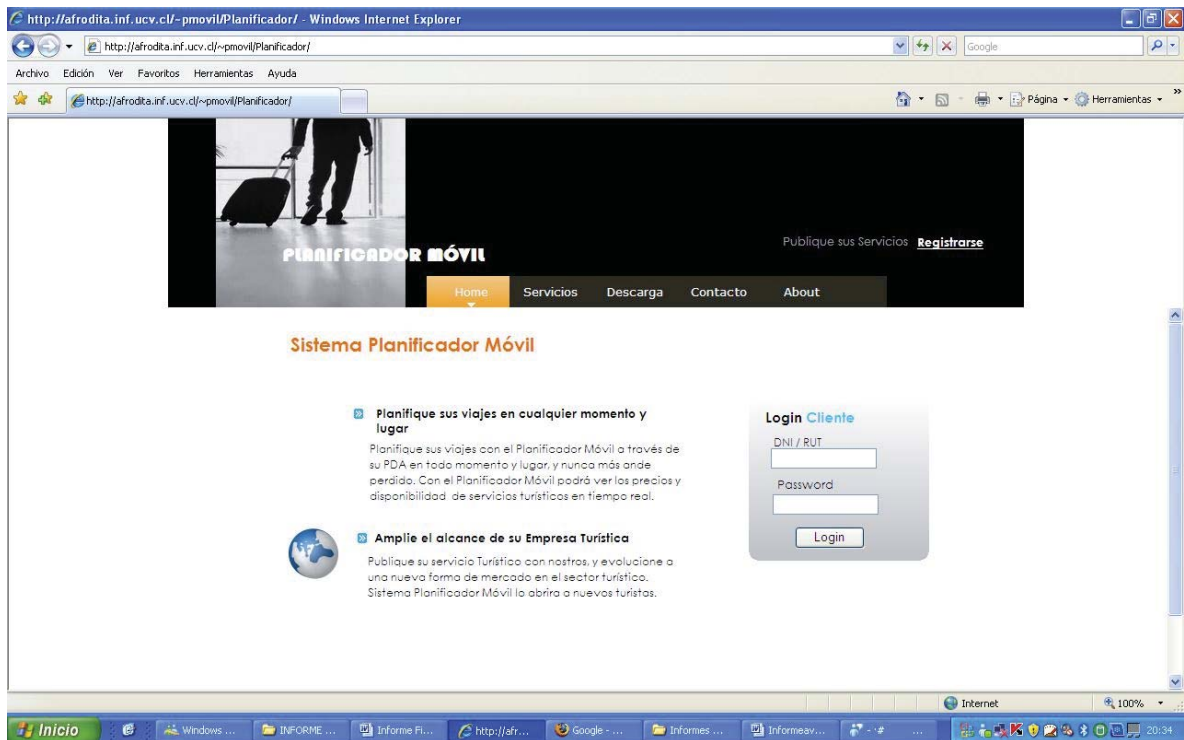


10.7. Interfaz modificaciones

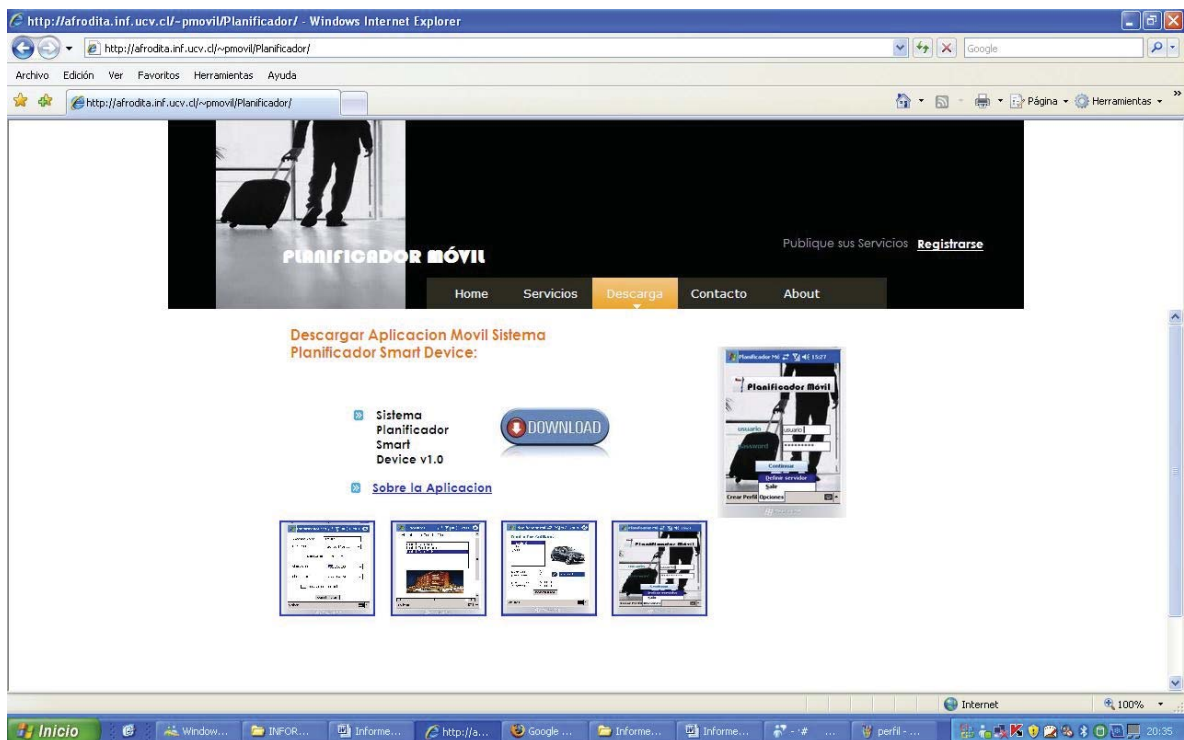
10.3. Registro de Servicios

El registro de servicios web por parte de un proveedor de servicios estará previsto mediante una página web. Esta página web comparte la base de datos con el sistema planificador para poder captar los datos de los servicios y realizar la invocación a los servicios web cuando sea necesario.

Además la página web proporciona información de los servicios que entrega a los potenciales usuarios, agregado de una zona de descarga, la que provee la descarga de la aplicación cliente, para poder instalarla en los dispositivos móviles de los usuarios, como se muestra en la Figura 10.8.



10.8. Interfaz de página web.



10.9. Interfaz de página web para descarga de aplicación móvil.

10.4. Modelo de datos

El sistema planificador tiene acceso a una base de datos relacional, donde almacena los datos necesarios para la gestión de itinerarios, perfil y usuarios. Esta base de datos ya está implementada de forma local para su funcionamiento. Se presenta a continuación el modelo lógico de la base de datos utilizada por el sistema planificador.

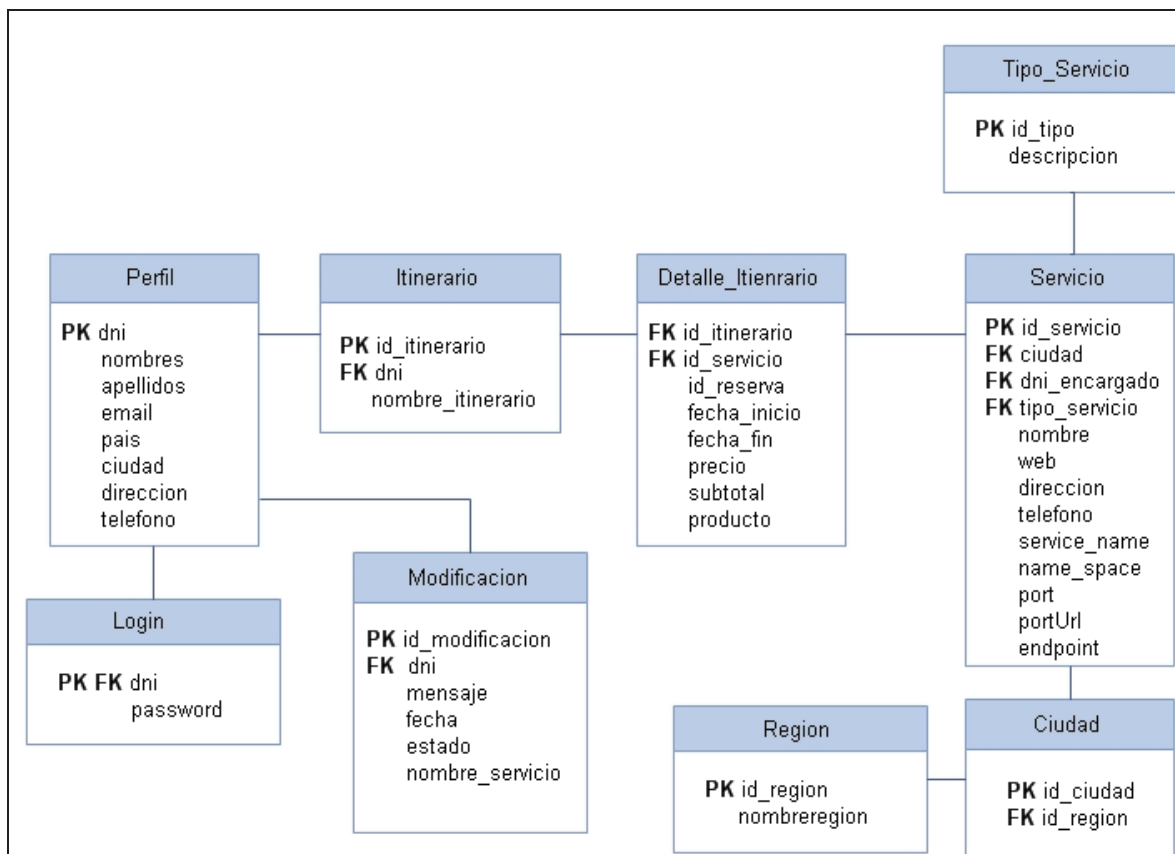
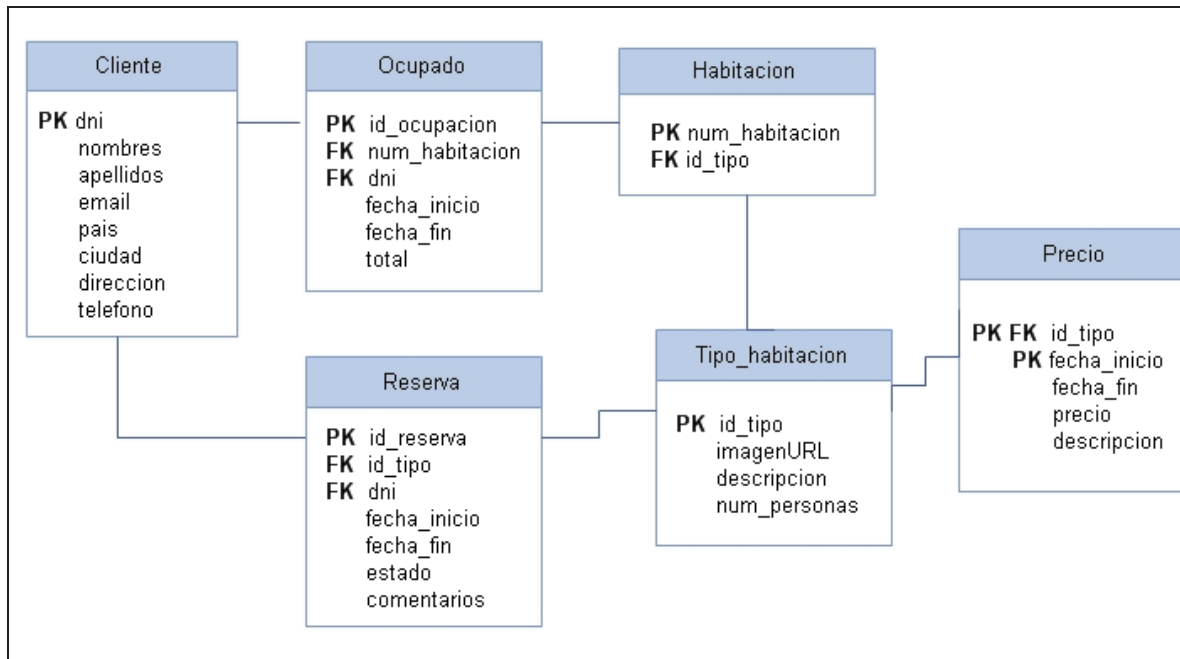
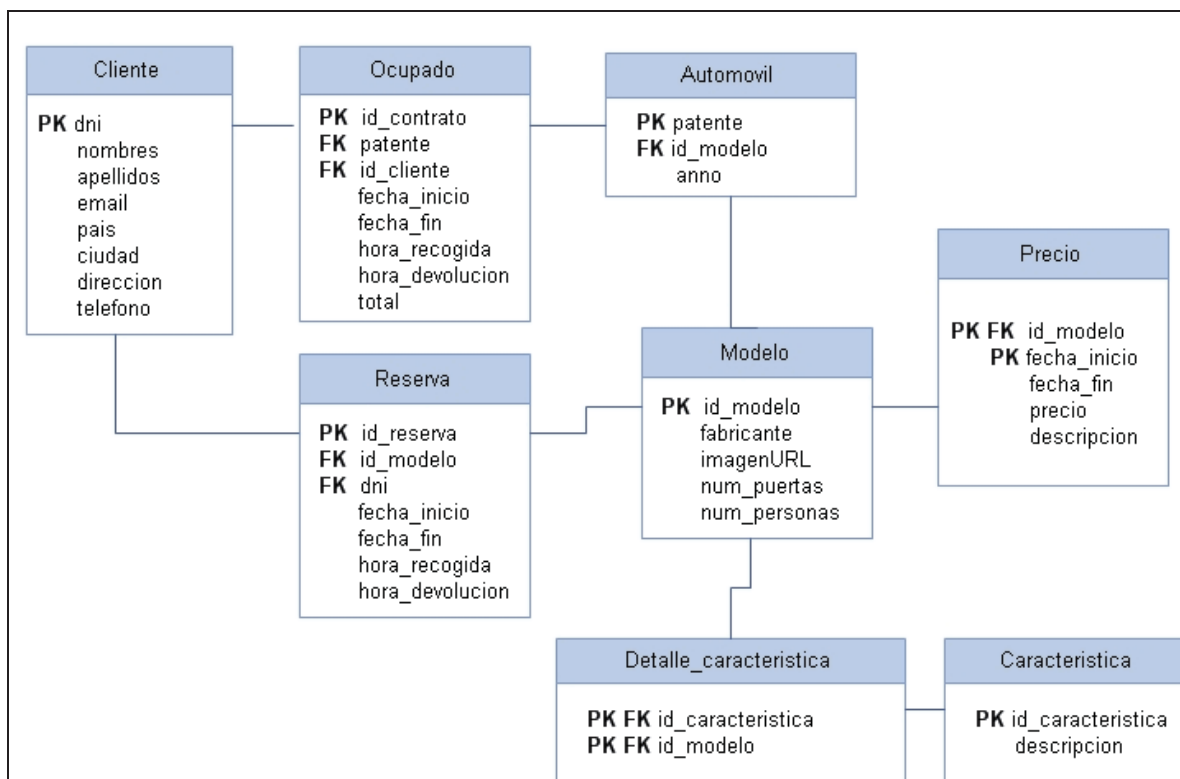


Figura 10.10. Modelo lógico base de datos sistema planificador

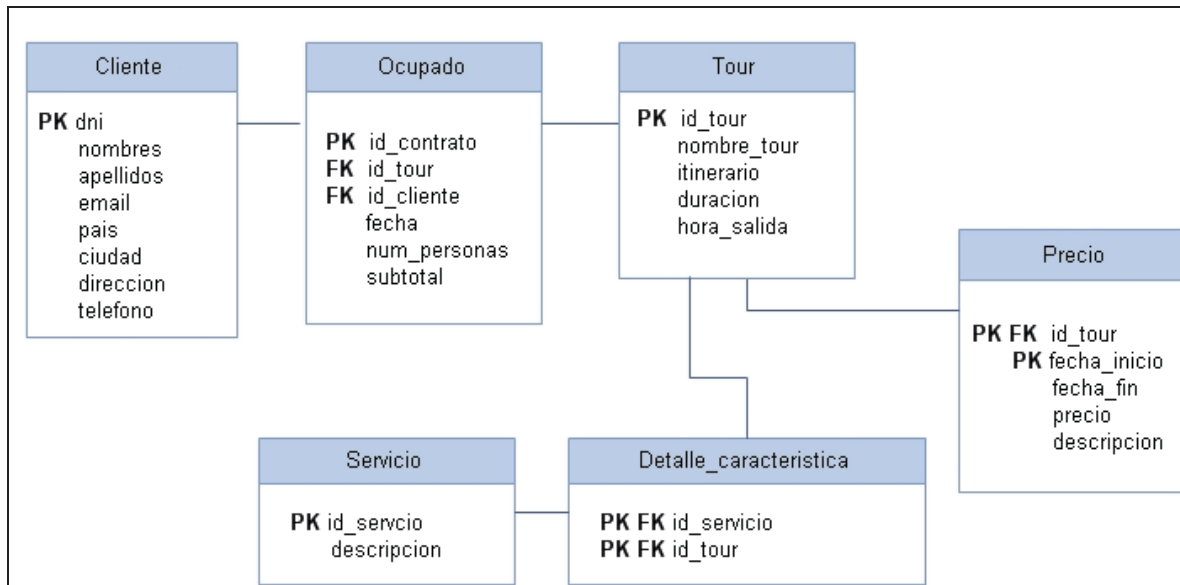
Por otro lado, también se ha implementado una base de datos para el servicio web del hotel que se ha desarrollado, la que es utilizada para el manejo de reservas de habitaciones, precios, y gestión de clientes de tal proveedor. La siguiente Figura representa el modelo lógico de la base de datos utilizada por el servicio web del hotel implementado, además de Rent a Car y City Tour.



10.11. Modelo lógico de base datos servicio web hotel



10.12. Modelo lógico de base datos servicio web Rent a Car



10.13. Modelo lógico de base datos servicio web city Tour

10.5. Descripción de Servicios,

En esta sección se presenta la descripción de los principales servicios desarrollados en el proyecto, la descripción de los servicios restantes estarán detallados en el Anexo A.

Crear Perfil

Crea un perfil de usuario en el sistema planificador, es decir, registra los datos de un nuevo usuario en el sistema con sus datos personales respectivos.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombres	Requerido	Nombres del Turista	Alfabético(50)
Apellidos	Requerido	Apellidos del Turista	Alfabético(60)
Email	Requerido	Email del Turista	Alfanumérico(50)
País	Requerido	País del Turista	Alfabético(40)
Ciudad	Requerido	Ciudad del Turista	Alfabético(40)
Dirección	Requerido	Dirección de residencia del turista	Alfabético(40)
Usuario	Requerido	Es nombre de usuario del cliente	Alfanumérico(15)
Password	Requerido	Password del usuario	Alfanumérico(20)

CrearItinerario

Función para crear un itinerario por un usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombre Itinerario	Requerido	Es el nombre del nuevo itinerario a crear	Alfanumérico(40)

ReservarAutomovil

Reserva un automóvil de un Rent a Car seleccionado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Itinerario	Requerido	Es el identificador del Itinerario	Alfanumérico(13)
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
Fecha Recogida	Requerido	Fecha de recogida de un vehículo deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Devolución	Requerido	Fecha de devolución de un vehículo deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Hora Recogida	Requerido	Hora de recogida de vehículo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Hora Devolución	Requerido	Hora de devolución de vehículo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Modelo	Requerido	Id del modelo de automóvil deseado	Alfanumérico(20)
Precio	Requerido	Precio del vehículo por día	Numérico
Subtotal	Requerido	Subtotal de la reserva de acuerdo al modelo y a los días.	Numérico

ReservarHabitacion

Reserva una habitación de un Hotel seleccionado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)

Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)

Eliminar Reserva

Elimina una reserva de un producto reserva. Esta eliminación se realiza tanto en el sistema planificador, como en el sistema del proveedor e servicios donde ha sido reservado el producto.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
ID de Itinerario	Requerido	Identificador de un itinerario en el sistema planificador	Alfanumérico(30)
ID de Reserva	Requerido	Identificador de una reserva en el sistema de un Proveedor de servicios	Alfanumérico(30)
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

11. Pruebas de Software

Como se ha mencionado en el Capítulo 8, durante el proceso y fase de transición se han desarrollado las respectivas pruebas de software, tales como de caja negra y blanca. Además de los tipos de pruebas unitarias, de integración y de sistema. Cabe destacar, que si bien se han realizado pruebas relativamente exhaustivas, siempre cabe el riesgo de que haya algo que no se tomó en cuenta.

En este anexo se presentará algunas de las pruebas funcionales más importantes realizadas al software final. Estas pruebas se irán dividiendo por funciones específicas, donde se presentarán los datos válidos y no válidos.

Crear Perfil

Proceso de registro de un usuario en el sistema.

Paso	Casos	Resultados
Ingresar datos en el formulario entregado, correspondiente a un registro de un nuevo usuario	Datos Correctos	Sistema valida los datos, y registra los datos del usuario en el sistema.
	DNI en blanco	Sistema despliega un mensaje "ingrese Dni"
	Nombres en blanco	Sistema despliega un mensaje "ingrese Nombres"
	Apellidos en blanco	Sistema despliega un mensaje "ingrese Apellidos"
	Ciudad en blanco	Sistema despliega un mensaje "ingrese Ciudad"
	País en blanco	Sistema despliega un mensaje "ingrese País"
	Email en blanco	Sistema despliega un mensaje "ingrese Email"
	Email inválido	Sistema despliega un mensaje "ingrese Email Válido"
	Teléfono en blanco	Sistema despliega un mensaje "ingrese Teléfono"
	Contraseña en blanco	Sistema despliega un mensaje "ingrese Contraseña"
	Contraseña diferente a la contraseña de confirmación.	Sistema despliega un mensaje "contraseñas no coinciden"
	DNI existe en el sistema	Sistema despliega un mensaje "Usuario Ya existe"
	Email existente	Sistema despliega un mensaje "email usado por otro usuario"

Crear Itinerario

Proceso de creación de un itinerario de viaje por un usuario registrado.

Paso	Casos	Resultados
Ingresar datos en el formulario entregado, correspondiente a la creación de un itinerario de viaje	Datos Correctos	Sistema valida los datos, y crea un nuevo itinerario para el usuario.
	Nombre de Itinerario en blanco	Sistema despliega un mensaje "ingrese nombre del nuevo itinerario"
	Nombre de Itinerario existe para el mismo usuario.	Sistema despliega un mensaje "el itinerario ya existe"

Recuperar Contraseña

Proceso de obtener la contraseña para identificación el sistema.

Paso	Casos	Resultados
Ingresar datos en el formulario entregado, correspondiente a la petición de obtención de contraseña.	Datos Correctos	Sistema valida los datos, y envía a un email al usuario con la contraseña.
	Email en blanco	Sistema despliega un mensaje "ingrese mail"
	Email inválido	Sistema despliega un mensaje "ingrese mail válido"

Cambio de Contraseña

Proceso de cambiar la contraseña actual por una nueva.

Paso	Casos	Resultados
Ingresar datos en el formulario entregado, correspondiente a la petición de obtención de contraseña.	Datos Correctos	Sistema valida los datos, cambia el valor de la contraseña, y envía a un email al usuario con la nueva contraseña contraseña.
	Contraseña actual en blanco	Sistema despliega un mensaje "ingrese contraseña actual"
	Contraseña nueva en blanco	Sistema despliega un mensaje "ingrese nueva contraseña"
	Confirmar contraseña en blanco	Sistema despliega un mensaje "contraseñas no coinciden"
	Contraseña nueva distinta de confirmar contraseña	Sistema despliega un mensaje "contraseñas no coinciden"

Login desde móvil

Proceso de autenticarse en el sistema.

Paso	Casos	Resultados
Ingresar datos en el formulario entregado, correspondiente al login en la aplicación móvil.	Datos Correctos	Sistema valida los datos, y redirecciona al menú principal si se ha logrado comunicar con el servicio web planificador.
	Login no es correcto	Sistema despliega un mensaje "contraseña o nombre de usuario incorrecto"
	Identificador en blanco.	Sistema despliega un mensaje "ingrese user"
	Contraseña en blanco	Sistema despliega un mensaje "ingrese contraseña"
	Error al conectarse con el servicio web planificador.	Sistema despliega un mensaje "Error de Conexión"

Registrar Servicio

Proceso de registro de un servicio por un usuario registrado en la página web.

Paso	Casos	Resultados
Ingresar datos en el formulario entregado, correspondiente a un registro de un nuevo servicio	Datos Correctos	Sistema valida los datos, y registra los datos del servicio en el sistema.
	Id Servicio en blanco	Sistema despliega un mensaje "Id servicio"
	Nombre en blanco	Sistema despliega un mensaje "ingrese Nombre servicio"
	Región inválida	Sistema despliega un mensaje "ingrese Región"
	Ciudad inválida	Sistema despliega un mensaje "ingrese Ciudad"
	Ciudad en blanco	Sistema despliega un mensaje "ingrese Ciudad"
	Dirección en blanco	Sistema despliega un mensaje "ingrese Dirección"
	Teléfono en blanco	Sistema despliega un mensaje "ingrese Teléfono"
	ID de Servicio Existente	Sistema despliega un mensaje "Servicio ya existe"

12. Conclusiones

Los Servicios Web hoy en día son de una gran importancia para las empresas, ya que proveen de interoperabilidad entre aplicaciones heterogéneas, por lo que, para el objetivo del proyecto es una tecnología adecuada, además de la incorporación de tecnologías móviles.

La metodología de desarrollo para este proyecto fue el proceso unificado (up), lo que facilita que en etapas tempranas se comience a indagar en las partes más complejas para el desarrollo del sistema, dejando al final los detalles menos relevantes a considerar. Por otra parte, la metodología permite la utilización de notación UML, la cual es bien aceptada en el ambiente laboral y universitario, lo que no hace necesario aprender una nueva forma de notación para su utilización.

La utilización de Servicios Web ha ayudado a entender los sistemas distribuidos, además de la utilización de tecnologías móviles ha dado como resultado un prototipo basado en la arquitectura orientada a servicios. También cabe destacar que los dispositivos móviles poseen cada vez más poder de procesamiento y capacidad, además de ser más accesibles a las personas lo que se plantea como una nueva forma de negocio.

El proyecto realizado, desde el punto de vista personal ha ayudado para entender y aprender la forma de trabajo para la construcción de un sistema siguiendo una metodología y sus pasos de una forma ordenada. También se ha logrado manipular una cantidad de herramientas y lenguajes de programación como JAVA, C# y PHP. Por otro lado, el manejo de sistemas de gestión de base de datos como MySQL. Al utilizar la tecnología de Web Services se ha manejado la forma de utilizar documentos de tipo XML y parsers.

Para el proyecto en el futuro se pretende aumentar la cantidad de servicios y productos turísticos al sistema, como también servicios de utilidad pública de cada localidad como información de hospitales, bomberos y otros. Además cabe destacar, que se pretende expandir el alcance territorial, es decir, que se incluyan servicios de países extranjeros e incorporar idiomas deseados por los turistas.

13. Referencias

1. Cerami, E. "Web Services Essentials". Sebastopol: O'Reilly & Associates, 2002.
2. Snell, J.; Tidwell, D.; Kulchenko, P. "Programming Web Services with SOAP", O'Reilly & Associates, Inc., 2002.
3. Valdés, M. A, "Diseño de un Sistema Multiagente sobre dispositivos móviles para la planificación de viajes turísticos". Tesis. Ingeniería Civil Informática. Pontificia Universidad Católica de Valparaíso. Escuela de Ingeniería Informática. Valparaíso. Chile, 2008.
4. Barry, D. K., "Service-Oriented Architecture and Web Services: The Savvy Manager's Guide", Morgan Kaufmann, 2003.
5. Iversen, W. "Real World Web Services". O'Reilly. 2004.
6. Arguedas, R and Nogueras, M. T. "Planificación, Dirección y gestión financiera de empresas turísticas". Madrid: Editorial Universitaria Ramón Areces, 2007.
7. Garrigós, F. J., and Narangajavana, Y. "Capacidades directivas y nuevas tecnologías en el sector turístico". Universitat Jaume I, 2007.
8. W3C Working Group Note. "Web Services Architecture". <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>. 2004.
9. Gómez, B and López P. "Regionalización turística del mundo". Edicions Universitat de Barcelona, Barcelona. 2001.
10. Rufin, R. "Las empresas turísticas en la sociedad de la información". Ed. Centro de Estudios Ramón Areces. Madrid. 2003.
11. Borja, L.; Casanovas J. A. and Bosch R. "El consumidor turístico". ESIC, Madrid. 2007
12. Texier, E. "Una oportunidad para la innovación turística". En: VII Congreso de Turismo y las Tecnologías de la Información y Comunicaciones. Centro de Innovación Turística de Andalucía, Malaga. España, 25 al 26 de noviembre 2008.
13. Herzum, P. "Case Study in Tourism: F.E.T.I.S.H". En: Workshop on Modeling, Architectures, Infrastructures, and Standart for Business Collaboration. San José, Estados Unidos. 4 al 7 de marzo 2002
14. Alberola, J. "CINNTA". En: VII Congreso de Turismo y las Tecnologías de la Información y Comunicaciones. Centro de Innovación Turística de Andalucía, Malaga. España, 25 al 26 de noviembre 2008.
15. Pais, J. "Interoperabilidad y distribución de contenidos y productos turísticos". En: Seminario Turismo y nuevas Tecnologías. Santiago de Compostela. España, 17 de octubre 2007.

Anexo A

En este anexo se presenta la información detallada de los servicios web que componen el sistema. Se presentarán los datos de entrada de los servicios web del sistema coordinador y del servicio web de Hotel, Rent a Car y Tour Operador.

A.1 Sistema Coordinador

Login

Autenticación de un usuario en el sistema.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Usuario	Requerido	Es nombre de usuario del cliente	Alfanumérico(15)
Password	Requerido	Password del usuario	Alfanumérico(20)

Bienvenida

Mostrar nombre del usuario en la aplicación móvil una vez logueado en el sistema.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

VerDisponibilidadHotel

Ver la disponibilidad de un servicio, en este caso un Hotel, donde los datos de filtrado son las fechas de entrada y fecha de término.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)

ReservarHabitacion

Reserva una habitación de un Hotel seleccionado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)

Eliminar Reserva

Elimina una reserva de un producto reserva. Esta eliminación se realiza tanto en el sistema planificador, como en el sistema del proveedor e servicios donde ha sido reservado el producto.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
ID de Itinerario	Requerido	Identificador de un itinerario en el sistema planificador	Alfanumérico(30)
ID de Reserva	Requerido	Identificador de una reserva en el sistema de un Proveedor de servicios	Alfanumérico(30)
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

Hoteles Disponibles

Busca la lista de Hoteles disponibles correspondiente al filtrado de preferencias del turista. Estas preferencias son un rango de fechas, las cuales son las fechas de entrada y fecha de término; además de un rango de precios deseados por el turista.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Destino	Requerido	Destino deseado donde se encuentran los servicios	Alfanumérico(15)
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico

Precio Máximo	Requerido	Precio Máximo del rango de precios deseado por el cliente	Numérico
---------------	-----------	---	----------

Habitaciones Disponibles

Busca la lista de habitaciones disponibles en un Hotel seleccionado previamente por el turista. La búsqueda se realiza a través al filtrado de preferencias del turista. Estas preferencias son un rango de fechas, las cuales son las fechas de entrada y fecha de término; además de un rango de precios deseados por el turista.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico
Precio Máximo	Requerido	Precio Máximo del rango de precios deseado por el cliente	Numérico

Crear Perfil

Crea un perfil de usuario en el sistema planificador, es decir, registra los datos de un nuevo usuario en el sistema con sus datos personales respectivos.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombres	Requerido	Nombres del Turista	Alfabético(50)
Apellidos	Requerido	Apellidos del Turista	Alfabético(60)
Email	Requerido	Email del Turista	Alfanumérico(50)
País	Requerido	País del Turista	Alfabético(40)
Ciudad	Requerido	Ciudad del Turista	Alfabético(40)
Dirección	Requerido	Dirección de residencia del turista	Alfabético(40)
Usuario	Requerido	Es nombre de usuario del cliente	Alfanumérico(15)
Password	Requerido	Password del usuario	Alfanumérico(20)

Editar Perfil

Edita el perfil o datos personales de un usuario registrado en el sistema planificador.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombres	Requerido	Nombres del Turista	Alfabético(50)
Apellidos	Requerido	Apellidos del Turista	Alfabético(60)
Email	Requerido	Email del Turista	Alfanumérico(50)
País	Requerido	País del Turista	Alfabético(40)
Ciudad	Requerido	Ciudad del Turista	Alfabético(40)
Dirección	Requerido	Dirección de residencia del turista	Alfabético(40)

ObtenerItinerario

Función para obtener los itinerarios creados por un usuario

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

RentacarDisponibles

Busca la lista de Rent a Cars disponibles correspondiente al filtrado de preferencias del turista. Estas preferencias son un rango de fechas, las cuales son las fechas de recogida y fecha de devolución; además de un rango de precios deseados por el turista.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Destino	Requerido	Destino deseado donde se encuentran los servicios	Alfanumérico(15)
Fecha Recogida	Requerido	Fecha de recogida de un vehículo deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Devolución	Requerido	Fecha de devolución de un vehículo deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Hora Recogida	Requerido	Hora de recogida de vehiculo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Hora Devolución	Requerido	Hora de devolución de vehiculo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico
Precio Máximo	Requerido	Precio Máximo del rango de precios deseado por el cliente	Numérico

CrearItinerario

Función para crear un itinerario por un usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombre Itinerario	Requerido	Es el nombre del nuevo itinerario a crear	Alfanumérico(40)

ReservarAutomovil

Reserva un automóvil de un Rent a Car seleccionado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Itinerario	Requerido	Es el identificador del Itinerario	Alfanumérico(13)
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
Fecha Recogida	Requerido	Fecha de recogida de un vehículo deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Devolución	Requerido	Fecha de devolución de un vehículo deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Hora Recogida	Requerido	Hora de recogida de vehiculo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Hora Devolución	Requerido	Hora de devolución de vehiculo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Modelo	Requerido	Id del modelo de automóvil deseado	Alfanumérico(20)
Precio	Requerido	Precio del vehiculo por día	Numérico
Subtotal	Requerido	Subtotal de la reserva de acuerdo al modelo y a los días.	Numérico

DetalleItinerario

Función que devuelve todas las reservas relacionadas a un itinerario de viaje.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID Itinerario	Requerido	Es el identificador del itinerario seleccionado por el usuario	Alfanumérico(40)

CambioContraseña

Función que actualiza la contraseña de un usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Password	Requerido	Es la password actual de un usuario para identificarse en el sistema.	Alfanumérico(40)
NuevoPassword	Requerido	Es la password nueva de un usuario que se actualizara en el sistema.	Alfanumérico(40)

EliminarItinerario

Función que elimina un itinerario de viaje creado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID Itinerario	Requerido	Es el identificador del itinerario seleccionado por el usuario	Alfanumérico(40)

TouroperadoresDisponibles

Busca la lista de Tour Operadores disponibles correspondiente al filtrado de preferencias del turista. Estas preferencias son un rango de fechas, las cuales son las fechas de entrada y fecha de término; además de un rango de precios deseados por el turista.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Destino	Requerido	Destino deseado donde se encuentran los servicios	Alfanumérico(15)
Num Personas	Requerido	Es el número de personas.	Numérico
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico
Precio Máximo	Requerido	Precio Máximo del rango de precios deseado por el cliente	Numérico

ToursDisponibles

Busca la lista de tours disponibles de un Tour Operador seleccionado previamente por el turista. La búsqueda se realiza a través al filtrado de preferencias del turista. Estas preferencias son un rango de fechas, las cuales son las fechas de entrada y fecha de término; además de un rango de precios deseados por el turistay un número de personas.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Servicio	Requerido	Es el identificador de	Alfanumérico(15)

		un servicio registrado	
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Num Personas	Requerido	Es el número de personas	Numérico
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico
Precio Máximo	Requerido	Precio Máximo del rango de precios deseado por el cliente	Numérico

ReservarTour

Reserva un tour de un Tour Operador seleccionado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID de Servicio	Requerido	Es el identificador de un servicio registrado	Alfanumérico(15)
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
ID tour	Requerido	Es el id de un tour de un tour operador seleccionado	Alfanumérico(20)
Nombre tour	Requerido	Es el nombre del tour seleccionado	Alfanumérico(20)
Num Personas	Requerido	Es el número de personas	Numérico
Precio	Requerido	Es el precio del tour de acuerdo al número de personas y fecha seleccionada.	Numérico

VerModificaciones

Función retorna todas las modificaciones realizadas por los proveedores de servicios que estén relacionados con un itinerario de viaje creado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

EliminarModificacion

Función retorna una modificación de un usuario, las cuales son realizadas por los proveedores de servicios que estén relacionados con un itinerario de viaje creado por el usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
ID Modificacion	Requerido	Es el identificador de una medicación relacionada con el usuario	Alfanumérico(20)

Condiciones

Función retorna las condiciones y politicas de reserva asociada a un proveedor específico seleccionado por el usuario cuando quiera reservar un producto o tenga reservado un producto de dicho proveedor.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID Servicio	Requerido	Es el identificador del servicio seleccionado	Alfanumérico(13)

VerCiudad

Función retorna las ciudades correspondientes a una región seleccionada por el usuario y donde hayan tipos de servicios registrados en una ciudad de la región.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Region	Requerido	Es la Región seleccionada por el usuario	Alfanumérico(13)
TipoServicio	Requerido	Es el tipo de servicio seleccionada por el usuario	Alfanumérico(20)

DetalleServicio

Función que retorna los datos de un servicio, como la dirección, teléfono, etc.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID Servicio	Requerido	Es el identificador del servicio seleccionado	Alfanumérico(13)

RecuperarPass

Función que envía la password a la casilla de correos ingresada relacionada con un usuario.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Email	Requerido	Es el email donde se enviara la contraseña	Alfanumérico(100)

EliminarCuenta

Función que elimina la cuenta de un usuario registrado en el sistema.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

A.2 Servicio Web Hotel.

Registra cliente

Registrar un cliente nuevo en el sistema del Hotel. El registro de un nuevo cliente corresponde a un cliente ya registrado en el sistema planificador. Esta función es necesaria para la reserva de habitaciones desde la aplicación cliente móvil.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombres	Requerido	Nombres del Turista	Alfabético(50)
Apellidos	Requerido	Apellidos del Turista	Alfabético(60)
Email	Requerido	Email del Turista	Alfanumérico(50)
País	Requerido	País del Turista	Alfabético(40)
Ciudad	Requerido	Ciudad del Turista	Alfabético(40)
Dirección	Requerido	Dirección de residencia del turista	Alfabético(40)

VerCliente

Verifica si un cliente esta dentro del registro del sistema del Hotel.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

Eliminarreserva

Se elimina una reserva ya existe dentro del sistema del Hotel. Esta función es invocada desde el sistema planificador cuando el turista desea eliminar una reserva.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Reserva	Requerido	Identificador de una reserva en el sistema de un Proveedor de servicios	Alfanumérico(30)
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

Disponibilidad

Verifica si hay habitacones disponibles entre las fechas de entrada y término ingresadas por el tusrista. Además esta búsqueda puede ser realizada a través de un rango de precios ingresados por el turista.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico
Precio Máximo	Requerido	Precio Máximo del rango de precios deseado por el cliente	Numérico

ReservarHabitacion

Función para reservar una habitación del Hotel.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para el check in.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para el check out.	Alfanumérico(10) (YYYY/MM/DD)
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)

DatosHabitacion

Función que retorna los datos e una habitación del Hotel.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha de Término	Requerido	Fecha de término deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)

A.3 Servicio Web Rent a Car.

Registra cliente

Registrar un cliente nuevo en el sistema del Rent a Car. El registro de un nuevo cliente corresponde a un cliente ya registrado en el sistema planificador. Esta función es necesaria para la reservar un modelo de automóvil desde la aplicación cliente móvil.

Parámetro	Obligatoriedad	Observación	Tipo Dato
-----------	----------------	-------------	-----------

DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombres	Requerido	Nombres del Turista	Alfabético(50)
Apellidos	Requerido	Apellidos del Turista	Alfabético(60)
Email	Requerido	Email del Turista	Alfanumérico(50)
País	Requerido	País del Turista	Alfabético(40)
Ciudad	Requerido	Ciudad del Turista	Alfabético(40)
Dirección	Requerido	Dirección de residencia del turista	Alfabético(40)

VerCliente

Verifica si un cliente esta dentro del registro del sistema del Rent a Car.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

Eliminarreserva

Se elimina una reserva ya existe dentro del sistema del Rent a Car. Esta función es invocada desde el sistema planificador cuando el turista desea eliminar una reserva.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Reserva	Requerido	Identificador de una reserva en el sistema de un Proveedor de servicios	Alfanumérico(30)
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

Disponibilidad

Verifica si hay modelos disponibles entre las fechas de entrada y término ingresadas por el turista. Además esta búsqueda puede ser realizada a través de un rango de precios ingresados por el turista.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Fecha Recogida	Requerido	Fecha de recogida deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Fecha Devolucion	Requerido	Fecha de devolución deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Hora Recogida	Requerido	Hora de recogida de vehiculo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Hora Devolucion	Requerido	Hora de devolución deseada por el usuario	Alfanumérico(10) (HH:MM)
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico
Precio Máximo	Requerido	Precio Máximo del rango de precios	Numérico

		deseado por el cliente	
--	--	------------------------	--

ReservarModelo

Función para reservar un modelo de automóvil.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Fecha Recogida	Requerido	Fecha de recogida del automovil deseado por el usuario .	Alfanumérico(10) (YYYY/MM/DD)
Fecha Devolucion	Requerido	Fecha de devolucion deseado por el usuario para entrega del vehiculo.	Alfanumérico(10) (YYYY/MM/DD)
Hora Recogida	Requerido	Hora de recogida de vehiculo deseada por el usuario.	Alfanumérico(10) (HH:MM)
Hora Devolucion	Requerido	Hora de devolucion de vehiculo deseado por el usuario.	Alfanumérico(10) (HH:MM)
ID Modelo	Requerido	Modelo de automóvil del rent a Car	Alfanumérico(20)

DatosModelo

Función que retorna los datos de un modelo de automóvil de un rent a Car.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Fecha Recogida	Requerido	Fecha de recogida del automovil deseado por el usuario.	Alfanumérico(10) (YYYY/MM/DD)
Fecha Devolucion	Requerido	Fecha de devolucion deseado por el usuario para entrega del vehiculo.	Alfanumérico(10) (YYYY/MM/DD)
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)

A.4 Servicio City Tour.

Registra cliente

Registrar un cliente nuevo en el sistema del City Tour. El registro de un nuevo cliente corresponde a un cliente ya registrado en el sistema planificador. Esta función es necesaria para la reserva de habitaciones desde la aplicación cliente móvil.

Parámetro	Obligatoriedad	Observación	Tipo Dato
-----------	----------------	-------------	-----------

DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Nombres	Requerido	Nombres del Turista	Alfabético(50)
Apellidos	Requerido	Apellidos del Turista	Alfabético(60)
Email	Requerido	Email del Turista	Alfanumérico(50)
País	Requerido	País del Turista	Alfabético(40)
Ciudad	Requerido	Ciudad del Turista	Alfabético(40)
Dirección	Requerido	Dirección de residencia del turista	Alfabético(40)

VerCliente

Verifica si un cliente esta dentro del registro del sistema del City Tour.

Parámetro	Obligatoriedad	Observación	Tipo Dato
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

Eliminarreserva

Se elimina una reserva ya existe dentro del sistema del City Tour. Esta función es invocada desde el sistema planificador cuando el turista desea eliminar una reserva.

Parámetro	Obligatoriedad	Observación	Tipo Dato
ID de Reserva	Requerido	Identificador de una reserva en el sistema de un Proveedor de servicios	Alfanumérico(30)
DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)

Disponibilidad

Verifica si hay tours disponibles disponibles entre las fechas de entrada y término ingresadas por el turista. Además esta búsqueda puede ser realizada a través de un rango de precios ingresados por el turista, y el número de personas.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Número de Personas	Requerido	Número de personas que irían el tour.	Numérico
Precio Mínimo	Requerido	Precio Mínimo del rango de precios deseado por el cliente	Numérico
Precio Máximo	Requerido	Precio Máximo del rango de precios deseado por el cliente	Numérico

ReservarTour

Función para reservar un tour del tour Operador.

Parámetro	Obligatoriedad	Observación	Tipo Dato
-----------	----------------	-------------	-----------

DNI	Requerido	Es el identificador del cliente	Alfanumérico(13)
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para el check in.	Alfanumérico(10) (YYYY/MM/DD)
Número de Personas	Requerido	Número de personas que irían el tour.	Numérico
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)

DatosTour

Función que retorna los datos de un tour del sistema.

Parámetro	Obligatoriedad	Observación	Tipo Dato
Fecha Inicio	Requerido	Fecha de inicio deseado por el usuario para ver servicios.	Alfanumérico(10) (YYYY/MM/DD)
Número de Personas	Requerido	Número de personas que irían el tour.	Numérico
Tipo de Habitación	Requerido	Tipo de habitación de un Hotel seleccionado	Alfanumérico(20)