

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

**SISTEMA DE PLANIFICACIÓN
TURISTICO HÍBRIDO MEDIANTE
TECNOLOGÍA DE DISPOSITIVOS
MÓVILES, AGENTES INTELIGENTES Y
SERVICIOS WEB**

DANTE JULIÁN MEDINA VENEGAS

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

JULIO 2010

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**SISTEMA DE PLANIFICACIÓN TURÍSTICO
HÍBRIDO MEDIANTE TECNOLOGÍA DE
DISPOSITIVOS MÓVILES, AGENTES
INTELIGENTES Y SERVICIOS WEB**

DANTE JULIÁN MEDINA VENEGAS

Profeso Guía: **Claudio Cubillos Figueroa**

Profesor Co-referente: **José Miguel Rubio León**

Carrera: **Ingeniería de Ejecución en Informática**

JULIO 2010

*A mi Familia mi Madre Edita, mi Padre Luis y mi Hermano Fabián
por ser un pilar fundamental en todo lo que soy y por su apoyo incondicional*

*A Catalina que nos apoyamos mutuamente en nuestra
formación profesional y que hasta ahora seguimos juntos*

*Finalmente los que ya no están porque partieron
y estuvieron en esta etapa de mi vida*

Resumen

En la actualidad el término sistema centralizado está desapareciendo, con la aparición de las redes entre computadores podemos compartir información sin preocuparnos de la ubicación geográfica. La planificación turística es cada vez más necesaria debido a la demanda y los ingresos que genera el turismo en una región, país o zona, los servicios como alojamiento, entretenimiento, restaurantes y transporte que son los más requeridos. Debido a esto es necesario contar con soluciones que ayuden a satisfacer esta necesidad, es por eso que los sistemas distribuidos están siendo impulsados en parte por una industria del software que está madurando a grandes pasos. Es así como los Sistemas Multiagente y Servicios Web forman parte de soluciones descentralizadas. Frente a esta afirmación nace la necesidad de construir un sistema turístico que mezcle ambas arquitecturas de sistemas distribuidos.

Palabras Clases: *Sistemas multiagente, Web Services, Distribuido, Turismo.*

Abstract

Today the term centralized system is disappearing, with the emergence of networks can share information between computers without worrying about the location. Tourism planning is becoming increasingly necessary due to demand and revenue generated by tourism in a region, country or area, services such as accommodation, entertainment, restaurants and transport are the most requested. Because it is necessary to have solutions to help meet this need, this is why distributed systems are being driven in part by a software industry that is maturing by leaps and bounds. Thus Multiagent Systems and Web Services are part of decentralized solutions. Given this statement comes the need to build a touristic system that blends both distributed systems architectures.

Keywords: Multiagent Systems, Web Services, Distributed, Tourism.

INDICE

Descripción del Proyecto.....	1
1.1 Introducción	1
1.2 Problemática	2
1.3 Definición de Objetivos	3
1.3.1 Objetivo Principal	3
1.3.2 Objetivos Específicos.....	3
1.4 Metodología.....	3
1.4.1 Proceso Unificado (UP)	4
1.4.1.1 Descripción UP.....	4
1.4.1.2 Fases UP	5
1.4.2 PASSI.....	5
1.4.2.1 Descripción PASSI.....	5
1.4.2.2 Pasos de PASSI	6
1.5 Plan de Trabajo	6
1.6 Resumen.....	10
Marco Teórico de Sistemas Multiagente	11
2.1 Agente.....	11
2.2 Sistemas multiagente	12
2.3 Arquitectura de agentes.....	13
2.3.1 Deliberativas.....	13
2.3.2 Reactivas	13
2.3.3 Híbridas	13
2.3.4 FIPA	13
2.4 Comunicación entre Agentes	15
2.4.1 FIPA CAL (Communicative Act Library)	15
2.4.2 FIPA ACL	17
2.4.3 KQML	18
2.4.4 Protocolos de comunicación.....	19
2.5 Metodologías de desarrollo Orientado Agentes.....	20
2.5.1 GAIA.....	20
2.5.2 Tropos.....	21
2.5.3 INGENIAS	21
2.5.4 PASSI.....	22
Marco Teórico Web Services.....	24
3.1 Web Services	24
3.2 Arquitectura Web Services	25
3.2.1 Conceptos	25
3.2.2 Pila de Protocolos y Lenguajes.	26
3.2.2.1 HTTP	26
3.2.2.2 SMPT	27
3.2.2.3 FTP	27
3.2.2.4 BEEP	27
3.2.2.5 XML	27
3.2.2.6 SOAP.....	28
3.2.2.7 WSDL.....	29

3.2.2.8 XML Schema.....	29
3.2.2.9 UDDI	29
3.2.2.10 DISCO	30
3.3 Arquitectura Orientada a Servicios.....	30
3.3.1 Características SOA	30
3.3.2 Beneficios SOA.....	31
3.4 Tecnología en Web Services.....	31
3.4.1 Tecnología .NET	31
3.4.1.1 Visual Studio	31
3.4.2 Tecnología JAVA.....	32
3.4.2.1 JAX-RPC	32
3.4.2.2 JAX-WS	32
Marco Teórico Turismo	33
4.1 Conceptos Básicos	34
4.2 Clasificación del Turismo	34
4.2.1 Tipo de Turismo	34
4.2.2 Temporada Turística	34
4.3 El Turista.....	35
4.4 Empresa Turística	35
4.5 Servicios del Turismo	36
4.5.1 Servicio de Alojamiento.....	36
4.5.2 Servicio de Restauración.....	37
4.5.3 Servicio de Transporte	37
4.5.4 Servicio de Entretención	37
4.6 Efectos Turísticos	38
4.6.1 Efectos Sociales.....	38
4.6.2 Efectos Económicos	38
4.6.3 Efectos Políticos	38
4.6.4 Efectos Culturales	38
4.6.5 Efectos Educativos.....	39
4.7 Turismo y Sistemas de Información	39
Solución Propuesta	40
5.1 Descripción de la Solución Propuesta	40
5.2 Análisis	41
5.2.1 Casos de Uso de Servicios Web.....	42
5.2.1.1 Caso Uso Alojamiento.....	42
5.2.2 Modelo Descripción del Dominio	51
5.3 Especificación del Diseño.....	52
5.3.1 Diagrama de Clases.....	52
5.3.2 Diagrama de Secuencia	56
5.3.3 Diagrama de Actividad.....	57
5.3.4 Modelo de Identificación de Agentes.....	59
5.3.5 Modelo de especificación de Tareas	61
5.3.6 Modelo de descripción de Ontología del dominio	64
5.3.7 Modelo de descripción estructura Sistema Multiagente	65
5.3.8 Modelo de Estructura de Agentes	66
5.3.9 Modelo Despliegue	71

5.4 Implementación del Código	71
5.5 Modelo de Datos	74
Prototipo de Sistema.....	79
6.1 Lanzamiento RMA	79
6.2 Lanzamiento Prototipo Dispositivo Móvil	81
6.3 Prototipo Dispositivo Móvil	82
6.4 Prototipo Servicios Web	84
Pruebas	87
7.1 Descripción Plan de Pruebas.....	87
7.1.1 Proceso de Prueba	87
7.1.2 Elementos a Probar.....	87
7.1.3 Criterios Evaluar	87
7.1.4 Enfoque	87
7.1.5 Materiales para las Pruebas	88
7.2 Pruebas Unitarias	88
7.3 Resultados Pruebas Unitarias.....	89
Pruebas de Performance	91
8.1 Descripción de la Prueba de Performance	91
8.2 Elementos a Utilizar.....	91
8.3 Escenarios de Pruebas.....	92
8.4 Resultados Prueba de Performance.....	93
Conclusiones.....	97
9.1 Conclusiones del Proyecto	97
9.2 Trabajo Futuro.....	97
Referencias	99
Casos de Usos	101
Resultados Pruebas Performance	107
Manual de Usuario	111
Diagramas y Diccionario de Datos.....	116

ÍNDICE DE ILUSTRACIONES

ILUSTRACIONES

Ilustración 1.1 Ciclo Vida Proceso Unificado con PASSI	4
Ilustración 1.2 Ciclo de Vida del Proceso Unificado	5
Ilustración 2.1 Modelo Referencia Agente FIPA	14
Ilustración 2.2 Fases Metodología GAIA	21
Ilustración 3.1 Modelo básico de Web Services	25
Ilustración 3.2 Bloques construcción aplicación remota	26
Ilustración 3.3 Elementos Fundamentales de un mensaje SOAP	28
Ilustración 5.1 Sistema Híbrido basado en el dominio turístico.	40
Ilustración 5.2 Diagrama de Casos de Uso alto nivel	43
Ilustración 5.3 Diagrama de Casos de Uso Buscar Alojamiento	44
Ilustración 5.4 Diagrama de Casos de Uso Administrar Usuario	45
Ilustración 5.5 Diagrama de Casos de Uso Administrar Servicio Turístico	48
Ilustración 5.6 Modelo Descripción del Dominio	51
Ilustración 5.7 Diagrama de Clases Alojamiento	53
Ilustración 5.8 Diagrama de Clases Restauración	54
Ilustración 5.9 Diagrama de Secuencia: Buscar Alojamiento	56
Ilustración 5.10 Diagrama de Actividad: Buscar Preferencias de Alojamiento	57
Ilustración 5.11 Diagrama de Actividad: Registrar Servicio de Alojamiento	58
Ilustración 5.12 Modelo de identificación de Agentes	59
Ilustración 5.13 Diagrama de Especificación de Tareas: Comunicar Web Services	61
Ilustración 5.14 Código Fuente: Invocación Servicio Web Alojamiento	62
Ilustración 5.15 WSDL: Servicio Web Alojamiento	62
Ilustración 5.16 Diagrama de Especificación de Tareas: Generar Itinerario	63
Ilustración 5.17 Modelo descripción de ontología del dominio	64
Ilustración 5.18 Modelo de Definición de Estructura Sistema Multiagente	65
Ilustración 5.19 Diagrama Definición Estructura de Agente: Interfaz	66
Ilustración 5.20 Diagrama Definición Estructura de Agente: IntermediarioMóvil	67
Ilustración 5.21 Diagrama Definición Estructura de Agente: IntermediarioWS	68
Ilustración 5.22 Diagrama Definición Estructura de Agente: Planificador	69
Ilustración 5.23 Diagrama Despliegue: Sistema Planificación Turística Híbrido	71
Ilustración 5.24 Código Fuente: Behaviour Buscar Alojamiento	72
Ilustración 5.25 Código Fuente: Behaviour Guardar Selección Planificador	74
Ilustración 5.26 Modelo Datos: Turista	74
Ilustración 5.27 Modelo Datos: Restauración	75
Ilustración 5.28 Modelo Datos: Transporte	76
Ilustración 5.29 Modelo Datos: Alojamiento	77
Ilustración 5.30 Modelo Datos: Atracción	78
Ilustración 6.1 RMA Sistema de planificación Turística	79
Ilustración 6.2 Despliegue gráfico del Sistema de planificación Turística	80
Ilustración 6.3 Lanzamiento dispositivo Móvil	81
Ilustración 6.4 Login de Usuario y menú principal de la Aplicación	82
Ilustración 6.5 Nuevo Itinerario	83
Ilustración 6.6 Selección de Preferencias	83

Ilustración 6.7 Selección resultado Preferencias	84
Ilustración 6.8 Login Proveedor alojamiento	85
Ilustración 6.9 Registro de nuevo proveedor	86
Ilustración 8.1 Performance Cuatro Escenarios Consulta dos	94
Ilustración 8.2 Performance Cuatro Escenarios Consulta nueve	95
Ilustración A.1 Diagrama de Casos de Uso Administrar Empresa Turística	101
Ilustración A.2 Diagrama de Casos de Uso Registrar Proveedor	103
Ilustración B.1 Promedio consultas cuatro Servicios Web en cuatro estaciones	109
Ilustración B.2 Promedio consultas cuatro Servicios Web en dos estaciones	109
Ilustración B.3 Promedio consultas cuatro Servicios Web en una estación	110
Ilustración B.4 Caso Estudio: cuatro servicios web en LocalHost	110

TABLAS

Tabla 1.1 Plan de Trabajo Proyecto 1	6
Tabla 1.2 Plan de Trabajo Proyecto 2	8
Tabla 2.1 Elementos mensaje FIPA ACL	17
Tabla 2.2 Elementos reservados de un mensaje KQML	18
Tabla 8.1 Estructura de búsqueda de servicios	91
Tabla 8.2 Promedio de cada Consulta en los cuatro despliegues	93
Tabla 8.3 Variables de Estadística Descriptiva	94
Tabla B.1 Cuatro Servicios Web en Cuatro estaciones de Trabajo	107
Tabla B.2 Cuatro Servicios Web en dos estaciones de Trabajo	107
Tabla B.3 Cuatro Servicios Web en una estación de Trabajo	108
Tabla B.4 Local Host	108

Capítulo 1

Descripción del Proyecto

1.1 Introducción

En la actualidad, el avance de las tecnologías y la necesidad de interconectar procesos, personas e información presentan problemas que se desean resolver. La falta de flexibilidad de los sistemas de información genera costos y reduce la capacidad de respuesta para los usuarios. El trabajo en red con componentes software y hardware localizados en computadores que se comunican mediante paso de mensajes forman un sistema distribuido, es así como encontramos arquitecturas de trabajo como sistemas multiagente y servicios web que ayudan a solucionar problemas de control de pasajeros, control de tráfico, control de procesos, planificación de viajes, comercio electrónico, entre otros. Esto demuestra como estas arquitecturas pueden ser aplicadas en distintas áreas, sin que se vea afectada la naturaleza del problema al cual se desea implementar una solución.

La arquitectura FIPA es un estándar utilizado para el desarrollo de sistemas multiagente desde ahora SMA, que define un modelo para la administración de agentes que provee una plataforma para el funcionamiento para que dos o más de ellos se puedan localizar y cooperar. Se comunican intercambiando mensajes que representen actos de habla codificados en un lenguaje de comunicación.

Por otro lado la arquitectura que se utiliza para desarrollar servicio web está orientada a servicios (SOA), que establece un marco de diseño ágil y flexible reutilizando aplicaciones existentes, además permite re-aprovechar plataformas ya creadas. Los servicios web prestan información al usuario dispersa por distintos sistemas desacoplando el desarrollo de servicios y procesos.

Una arquitectura Multiagente con servicio web (híbrida) ofrece una ventaja comparativa, en donde a un sistema que posee todas las ventajas que tiene una arquitectura SOA puede ser además, dotada de cierta capacidad de autonomía que caracteriza a los SMA, y que hoy en día en un medio tan competitivo, vale la pena evaluar a la hora de construir un sistema distribuido en desmedro de una solución centralizada.

Por todo lo mencionado anteriormente, nace la inquietud de elaborar una arquitectura híbrida que mezcle ambas tecnologías: SMA y servicios web. Para que de esta manera, se exploren más las ventajas que ofrecen juntas, todo esto pensado en una aplicación móvil que navegue por internet y permita acceder a las necesidades del usuario.

El contexto del problema a solucionar bajo estas tecnologías, es el área del Turismo, la cual aumenta día a día en lo referido a la oferta de servicios y sus respectivos proveedores. Los que compiten por satisfacer las demandas de sus clientes, ya que el periodo de trabajo no es estrictamente en verano sino que durante todo el año.

1.2 Problemática

La industria del turismo es una de las grandes del mundo en el 2009 genero 852 mil millones dólares EE.UU. (€611 mil millones) en ingresos por exportación [3], aportando al producto interno bruto de muchos países. La economía del turismo en Chile ha adquirido un peso específico, que se ha consolidado en el último decenio, en ello incidió el rápido crecimiento y desarrollo económico que vivió el país, donde la favorable imagen de democratización, y estabilidad frente a la comunidad internacional, promocionó la oferta turística chilena tanto al interior como al exterior del país.

Sin duda las tecnologías de información han revolucionado el panorama de los negocios y esta industria no es la excepción. Los servicios turísticos que se ofrecen cada día son más sofisticados y van desde una simple reserva hotelera, hasta complejos itinerarios con distintos tipos de servicios: como renta de vehículos, alojamiento, transporte, recreación y espectáculos. En todos ellos, la planificación de un itinerario es una tarea fundamental a la hora de realizar un viaje, ya que éste se inicia a la hora que se empieza a planificar.

De las tecnologías, una que toma ventaja y cada día tiene más adeptos es la celular o teléfonos inteligentes, todo esto gracias a los grandes avances tecnológicos que se han obtenido en torno a ella y lo fácil y sencillo que es usarla.

Por eso encontrar soluciones que se enfoquen al área del turismo, es algo normal donde se ofrecen soluciones de planificación de itinerarios de viaje, en que el usuario es capaz de ingresar sus preferencias y al mismo tiempo encontrar solución a sus requerimientos de viaje con un par de clicks.

De lo mencionado anteriormente nos encontramos con varias soluciones propuestas que apoyan la tarea de planificar un itinerario. Una propuesta bajo la arquitectura de SMA y otra bajo una arquitectura SOA, se podría pensar erróneamente que ambas soluciones ofrecen los mismos resultados, pero no es así ya que cada solución tiene distintos desempeños dados los diferentes escenarios.

La arquitectura SOA ofrece soluciones que reducen los costos, tiempo de desarrollo y ofrece mantención de forma más flexible, escalable y ágil siendo utilizada con mayor frecuencia en el entorno Industrial. Por otro lado, la arquitectura de agentes basada en el estándar FIPA que no goza de mucha popularidad en un entorno Industrial como los servicios web, se presenta como una alternativa a la hora de proponer una arquitectura con capacidad de acción autónoma en un entorno para lograr sus objetivos específicos.

Con estas dos arquitecturas, se busca adaptar una nueva aplicación con agentes y servicios web, para así explorar las ventajas de ambas en la solución de un sistema de planificación de viajes turísticos. De lo anterior, se crea la necesidad de desarrollar un sistema que las una, ya que en la actualidad no se encuentran disponibles sistemas que mezclen los servicios web con agentes inteligentes y no se tiene literatura de referencia.

En síntesis de todo lo esgrimido hasta ahora, nace la inquietud de construir un sistema de planificación turística que mezcle ambas tecnologías, para conocer las ventajas y desventajas de ellas. Además se desea explorar su performance en distintos despliegues en la misma solución y evaluar cual es la mejor alternativa.

1.3 Definición de Objetivos

1.3.1 Objetivo Principal

Generar un sistema de planificación turística mediante tecnologías para dispositivos móviles, agentes inteligentes y servicios web.

1.3.2 Objetivos Específicos

Para lograr el objetivo principal se deben llevar a cabo una serie de objetivos específicos que ayudaran a la consecución del objetivo final.

- ✓ Diseñar los Agentes que estén acorde a la solución del problema.
- ✓ Diseñar los Servicios Web que sean acorde a la solución del problema.
- ✓ Generar un prototipo del sistema turístico que cumpla con los requerimientos.
- ✓ Validar el sistema turístico verificando que cumpla los requerimientos mediante un plan de pruebas
- ✓ Generar un caso de estudio que compare las prestaciones de los servicios web

1.4 Metodología

Las metodologías que se utilizan hoy para el desarrollo de una investigación son múltiples, y una elección adecuada de ella está relacionada con el tipo de investigación que se esté llevando a cabo. En este caso, existe una cantidad abundante de información e investigaciones relevantes para la realización del proyecto.

Para el desarrollo del trabajo se utilizan dos metodologías: una de ellas es el proceso unificado (UP), y la otra es PASSI. La primera encapsulará a la segunda, es decir, UP contiene a PASSI, de esta forma se utilizarán pasos de PASSI en las distintas fases e iteración de UP. Todo esto debido a que:

- ✓ Una guía ordenada para realizar todas las tareas.
- ✓ Especifican los artefactos que hay que desarrollar.
- ✓ Reduce la posibilidad del no funcionamiento de proyecto final.

A continuación en la Ilustración 1.1 se muestra un mapeo de cómo se trabajo con ambas metodologías.

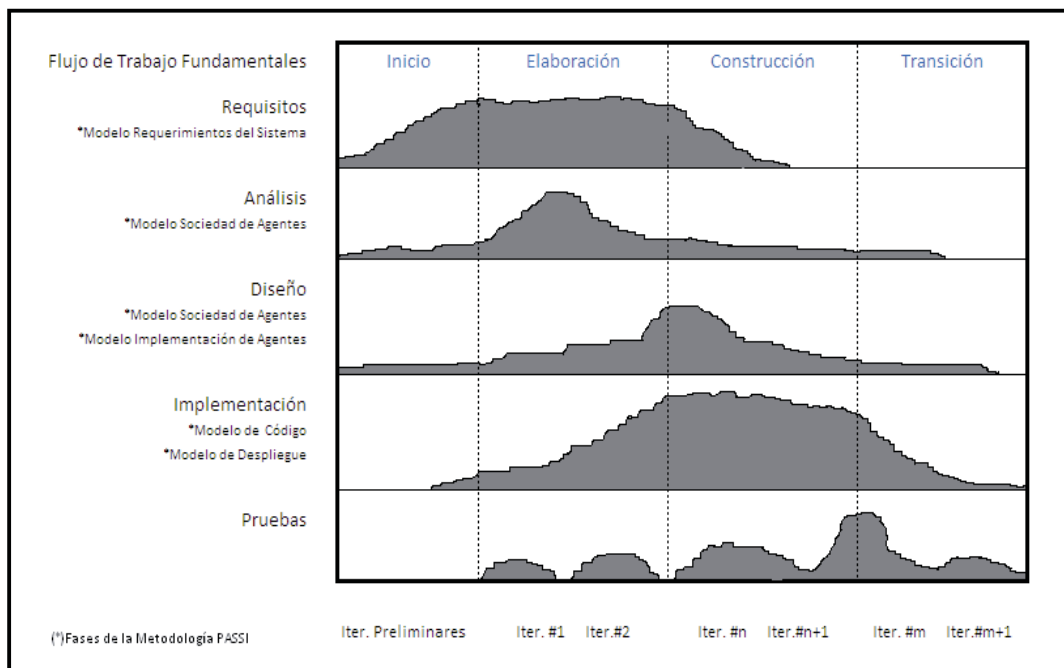


Ilustración 1.1 Ciclo Vida Proceso Unificado con PASSI

La Ilustración 1.1 describe la forma en que se implementó la metodología UP y PASSI, donde las fases de PASSI fueron adjuntas a los flujos de trabajos fundamentales que posee UP para que de esta forma se ejecute en cada iteración según corresponda a la fase del UP.

1.4.1 Proceso Unificado (UP)

1.4.1.1 Descripción UP

El proceso unificado (UP) es un proceso de desarrollo de software, es un conjunto de actividades necesarias para transformar los requerimientos del usuario en un sistema de software. [1]

Las principales características del UP son:

- ✓ **Manejado por Casos de Usos:** Pieza de funcionalidad en el sistema que entrega al usuario un resultado de valor. Los casos de usos capturan requerimientos funcionales, el conjunto de ellos forma el modelo de casos de usos.
- ✓ **Centrado en la Arquitectura:** Es cierto que los casos de usos manejan el proceso, pero estos no lo hacen solos, sino que son desarrollados en conjunto con la arquitectura del sistema. La arquitectura del sistema y los casos están estrechamente relacionadas, por eso maduran a medida que el ciclo de vida continúa.
- ✓ **Iterativo e incremental:** La vida del sistema está dividida en ciclos. Cada ciclo concluye con el lanzamiento de diferentes modelos del producto. Un ciclo posee fases,

cada fase esta subdivida en iteraciones que concluyen con algún documento o implementación interna.

1.4.1.2 Fases UP

El ciclo consiste en cuatro fases:

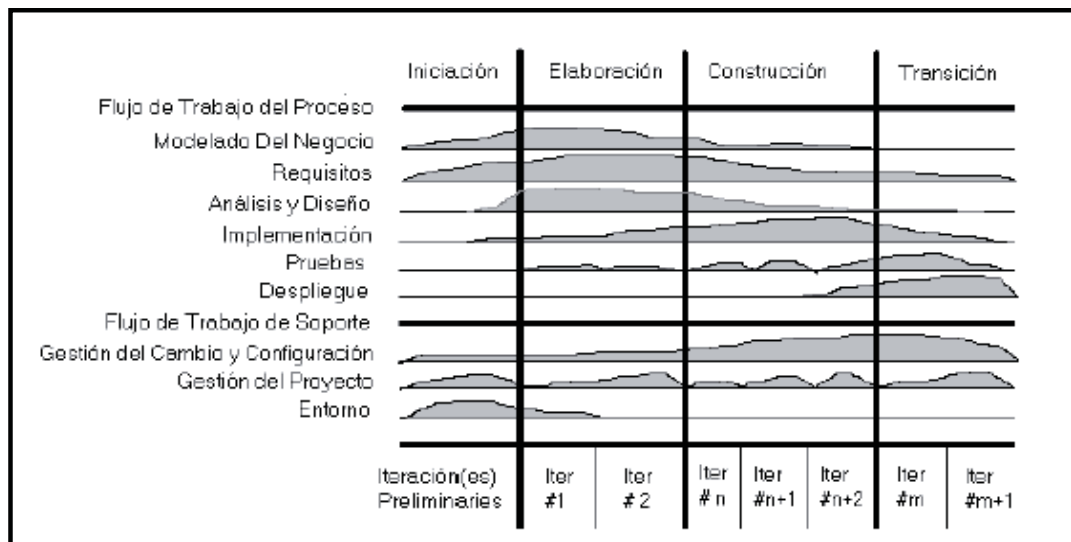


Ilustración 1.2 Ciclo de Vida del Proceso Unificado [2]

- ✓ **Fase de Inicio:** Suele ser la fase más corta del desarrollo, y no debería alargarse demasiado en el tiempo. Se desarrolla una descripción del producto final, además de identificar y priorizar los riesgos más importantes asociados a cada etapa.
- ✓ **Fase de Elaboración:** Fase de captura de la mayoría de los requerimientos del sistema, aunque los objetivos principales son: tratar los riesgos ya identificados y establecer y validar la base de la arquitectura del sistema.
- ✓ **Fase de Construcción:** Fase más larga del proyecto, contempla la implementación del sistema en base a la arquitectura, donde se deben implementar todos los casos de uso.
- ✓ **Fase de Transición:** Fase final del proyecto donde se lleva a cabo el despliegue del producto en un entorno de usuario, el cual, evoluciona de una versión Beta a una Final y se resuelven incidencias de implantación e integración. Esta Fase finaliza con la publicación del producto.

1.4.2 PASSI

1.4.2.1 Descripción PASSI

Metodología para el diseño y desarrollo de SMA pasó a paso. Utiliza la notación entregada por UML, lenguaje de modelado universal. PASSI abarca el diseño análisis e

implementación con un enfoque bottom-up, ella será descrita más en detalle en el capítulo 4 que trata del marco teórico de los SMA. [7]

1.4.2.2 Pasos de PASSI

A continuación sólo se nombraran las etapas más relevantes de la metodología PASSI:

- ✓ Fase descripción del Dominio.
- ✓ Fase Identificación de Agentes.
- ✓ Fase Identificación de Roles.
- ✓ Fase Especificación de Tareas.
- ✓ Fase Descripción de la Ontología del Dominio.
- ✓ Fase Descripción de la Ontología de Comunicación.
- ✓ Fase Descripción de Roles.
- ✓ Fase Descripción de Protocolos.
- ✓ Fase Definición Estructura de Agentes.
- ✓ Fase Descripción Comportamiento de Agentes.
- ✓ Modelo de Código
- ✓ Configuración del Despliegue.

1.5 Plan de Trabajo

El plan de trabajo se basó en dos metodologías UP y PASSI, todo esto para poder elaborar el trabajo en el ambiente de servicios web y Agente de manera conjunta. La metodología que se utilizó se describe de la siguiente forma: donde UP es el molde en que se construyen los distintos artefactos de las fases de PASSI en sus iteraciones, además de los propios de la metodología (UP).

A continuación se exponen los hitos más importantes realizados tanto en el Proyecto 1 y Proyecto 2.

Tabla 1.1 Plan de Trabajo Proyecto 1

Fase	Iteración	Hito	F. Inicio	F. Termino
Inicio	1	Estudiar Estado del Arte Servicios Web	10 de Agosto	28 de Agosto
		Estudiar Estado del Arte Sistemas multiagente	10 de Agosto	28 de Agosto
		Estudiar Estado del Arte Turismo	10 de Agosto	28 de Agosto
		Estudiar Trabajos relacionados	20 de Agosto	30 de Agosto

		Estudiar Plataformas Híbridas	05 de Septiembre	08 de Septiembre
		Definir requerimientos	11 de Septiembre	17 de Septiembre
Elaboración	1	Generar Casos de usos servicios web	15 de Septiembre	20 de Septiembre
		Generar Modelo descripción del Dominio	15 de Septiembre	20 de Septiembre
		Generar Modelo Identificación de Roles	21 de Septiembre	23 de Septiembre
		Generar Escenarios de Pruebas dado los distintos Sistemas	21 de septiembre	23 de septiembre
		Entrega Informe Avance Proyecto 1	25 de Septiembre	25 de Septiembre
		Corregir faltas o errores encontradas en la exposición de avance	28 de Septiembre	09 de Octubre
Elaboración	2	Generar Diagramas de Clases de los servicios web	09 de Octubre	18 de Octubre
		Generar Modelo de Datos servicios web	09 de Octubre	18 de Octubre
		Generar Modelo Especificación de Tareas	12 de Octubre	18 de Octubre
		Definir Set de Pruebas	12 de Octubre	18 de Octubre
		Generar Diagramas de Secuencia servicios web	18 de Octubre	23 de Octubre
		Planificación de modelo de Pruebas	16 de Octubre	28 de octubre
		Revisar Iteración antes del Incremento	26 de Octubre	27 de Octubre
Elaboración	3	Generar Modelo Ontología Dominio	27 de Octubre	01 de Noviembre
		Generar Diagramas de actividad servicios web	28 de Octubre	2 de Noviembre
		Generar Modelo Estructura Multiagente	29 de Octubre	3 de Noviembre
		Generar Modelo Estructura Agente	30 de Octubre	4 de Noviembre
		Generar Prototipo de Pantalla Administrador de Servicios	28 de Octubre	15 de Noviembre
		Generar Prototipo de Pantalla Ingreso Preferencias	29 de Octubre	16 de Noviembre
		Generar Prototipo de Pantalla Administrador de Servicios	30 de Octubre	17 de Noviembre

		Revisar Iteración antes del Incremento	15 de Noviembre	18 de Noviembre
		Entrega Informe Final	20 de Noviembre	20 de Noviembre
Construcción	1	Realizar Pruebas de Comunicación Modulo Multiagente y Servicios Web	17 de Noviembre	22 de Noviembre

Tabla 1.2 Plan de Trabajo Proyecto 2

Fase	Iteración	Hito	F. Inicio	F. Termino
Construcción	1	Revisar Correcciones Presentación Final P1	08 de Marzo	11 de Marzo
		Mejorar Diseño Dispositivo Móvil	11 de Marzo	16 de Marzo
		Mejorar Diseño Servicios Web	12 de Marzo	17 de Marzo
		Evaluar Factibilidad Tecnologías	16 de Marzo	18 de Marzo
		Buscar Herramientas Implementación (SDK)	18 de Marzo	20 de Marzo
		Construir Modelo de Código	22 de Marzo	30 de Marzo
Construcción	2	Corregir Iteración Anterior	01 de Abril	04 de Abril
		Comunicar Dispositivo Móvil BackEnd	05 de Abril	12 de Abril
		Ejecutar Dispositivo Móvil en el Emulador	12 de Abril	17 de Abril
		Construir Mantenedores Servicio Web Hotel	12 de Abril	17 de Abril
		Construir Mantenedores Servicio Web Atracción	12 de Abril	17 de Abril
		Construir Mantenedores Servicio Web Alojamiento	12 de Abril	17 de Abril
Construcción	3	Revisar Iteración Anterior	18 de Abril	18 de Abril
		Construir Mantenedores Servicio Web Transporte	18 de Abril	23 de Abril
		Generar Base de datos Usuario	18 de Abril	23 de Abril
		Generar Base de datos Atracción	18 de Abril	23 de Abril

		Generar Base de datos Alojamiento	18 de Abril	23 de Abril
		Generar Base de datos Transporte	18 de Abril	23 de Abril
		Generar Base de datos Restauración	18 de Abril	23 de Abril
		Llegar a cabo plan de pruebas	23 de Abril	29 de Abril
		Entrega Avance Proyecto 2	23 de Abril	23 de Abril
Construcción	4	Comunicar Main Container con Servicios Web	25 de Abril	01 de Mayo
		Comunicar BackEnd con Main-Container	26 de Abril	03 de Mayo
		Integrar Módulos de proveedor y cliente	04 de Mayo	14 de Mayo
		Probar Conexión Dispositivo Móvil – Intermediario	08 de Mayo	16 de Mayo
		Completar Código dispositivo Móvil	06 de Mayo	20 de Mayo
		Completar Código Servicios Web	06 de Mayo	21 de Mayo
Transición	1	Probar Aplicación Ambiente Real	25 de Mayo	04-jun
		Realizar Mejoras Servicios Web	06 de Junio	10 de Junio
		Realizar Mejoras Aplicación Móvil	06 de Junio	10 de Junio
		Realizar Mejoras Integración	06 de Junio	10 de Junio
Transición	2	Corregir Errores omisiones interacción Anterior	10 de Junio	13 de Junio
		Preparan Ambiente revisión Software	20 de Junio	22 de Junio
		Revisión Final de Software	23 de Junio	23 de Junio
		Evaluación de las correcciones	23 de Junio	25 de Junio
		Entrega del Informe Final Proyecto 2	25 de Junio	25 de Junio
		Implementación del Caso de Estudio	26 de Junio	26 de Junio
		Implementación Mejoras Corrección del Software	26 de Junio	26 de Junio

		Presentación Final	27 de Junio	08 de Julio
		Mejoras y Correcciones presentación Final	08 de Julio	31 de Julio

1.6 Resumen

Este primer capítulo cumple la función de presentación del tema del proyecto de título.

El capítulo dos presenta el marco teórico correspondiente a los SMA, donde se describe que son los agentes, los SMA, sus arquitecturas y las metodologías más relevantes al momento de trabajar orientado hacia los agentes.

En el capítulo tres se presenta el marco teórico correspondientes a los servicios web, donde se describen los servicios web, su arquitectura, protocolos, arquitectura SOA y tecnologías de desarrollo.

En el capítulo cuatro se plantea el marco teórico entorno al área del turismo, donde se exponen conceptos básicos, clasificaciones, la empresa turística, los servicios turísticos y sus efectos.

En el capítulo cinco se expone la solución propuesta al proyecto de título dando una descripción de ella, adjuntando los diagramas de la etapa de análisis y diseño para los servicios web, los diagramas PASSI para los agentes y el modelo de datos.

En el capítulo seis se muestra la aplicación desde el lanzamiento del prototipo del SMA hasta los servicios web involucrados en la solución y la mantención de ellos.

En el capítulo siete se realizaron las pruebas unitarias del prototipo y con esto validar el correcto funcionamiento.

En el capítulo ocho se plantearon las pruebas de performance donde se evaluaron el rendimiento de la solución del sistema en distintos despliegues, con el fin de validar la mejor solución.

En el capítulo nueve se exponen las conclusiones finales del proyecto de título, además del trabajo futuro que se puede realizar.

Como último elemento al final existen una serie de anexo que vienen a profundizar aún más el trabajo expuesto en este proyecto de título.

Capítulo 2

Marco Teórico de Sistemas Multiagente

2.1 Agente

Hoy en la actualidad no es sencillo definir lo que es un agente, ya que existen múltiples definiciones que reflejan lo que es un agente, como por ejemplo:

“Toda entidad que de forma autónoma perciba su entorno (real o simulado) mediante sensores y que actúe en el mismo mediante efectores”. [5]

“Un agente es un sistema informático que se sitúa en un cierto ambiente y que es capaz de acción autónoma en este ambiente para lograr sus objetivos de diseño” [6]

De las citas anteriores ninguna es una definición formal de lo que en realidad es un agente, a pesar de ello estos poseen algunos atributos tales como:

- **Autonomía:** Característica de control de un agente donde debe ser capaz de decidir qué acciones ejecutar o qué objetivos elegir sin ningún control exterior explícito.
- **Reactividad:** Capacidad del agente percibir cambios en el entorno y responder a ellos.
- **Pro-Actividad:** Iniciativa de un agente un comportamiento orientado a sus metas.
- **Habilidad Social:** Capacidad interactuar tanto con otros agentes y seres humanos para alcanzar un objetivo o meta.

Con estos atributos obtenemos una definición que cumple con las nociones débiles de un agente, a continuación presentaremos algunas de las nociones fuertes de un agente, que son opcionales al conformar un agente.

- **Movilidad:** Capacidad de un agente de moverse en un entorno (de un host a otro) sin perder su estado interno.
- **Veracidad:** El agente no comunicara con conocimiento información falsa.
- **Benevolencia:** Los agentes no poseen metas en discusión, por ende siempre intentará hacer lo que se le pide.
- **Racionalidad:** El agente debe ser capaz de aprender, y con ello mejorar junto con la experiencia obtenida.

2.2 Sistemas multiagente

Tomando en cuenta lo que es un agente, se puede definir lo que forma un Sistema Multiagente (SMA): que es una red de agentes que interactúan en conjunto para resolver problemas que van más allá de las capacidades individuales de cada entidad. Sus características son: [7]

- Cada agente posee capacidades o información incompleta para resolver un problema, por ende posee un punto de vista limitado.
- No existe un control global de sistema.
- Los datos se encuentran descentralizados.
- La comunicación se efectúa de manera asíncrona.

Además de características un SMA nos entrega una serie de beneficios a la hora de construir un sistema orientado a agentes. Los cuales son:

- Velocidad y eficiencia: tareas realizadas de manera asíncrona y paralela.
- Robustez y confiabilidad: existencia de múltiples agentes presenta tolerancia a fallas.
- Escalabilidad y flexibilidad: Sistema escalado naturalmente agregando nuevos agentes.
- Costos: Descomposición del problema en múltiples subsistemas con costo unitario menor.
- Desarrollo y reusabilidad: Agentes extendidos, reutilizados, probados y mantenidos fácilmente.

Sin embargo, los SMA también cuentan con algunas limitaciones que poseen todos los sistemas descentralizados. Como son:

- Overhead: al comparar resultados frente a otras tecnologías, seleccionar otra.
- Ausencia Control Central: No existe de manera explícita un control central.
- Ausencia perspectiva global: Cada agente toma decisiones basado en sus conocimientos.

2.3 Arquitectura de agentes

Tanto la estructura interna de un agente como la de un SMA están determinadas por alguna arquitectura. Ésta determina los mecanismos que utiliza un agente para reaccionar a los estímulos, actuar, comunicarse, etc.

2.3.1 Deliberativas

Arquitectura que utiliza modelos de representación simbólica del conocimiento. Los agentes poseen un sistema de planificación que se encarga de llevar a cabo los objetivos que tienen. Por lo tanto, una arquitectura deliberativa contiene un modelo simbólico del mundo, donde las decisiones se toman usando un mecanismo de razonamiento lógico de correspondencia entre patrones y manipulación simbólica, para alcanzar sus metas.

Para implementar una arquitectura deliberativa se debe encontrar una descripción simbólica adecuada al problema. [7]

2.3.2 Reactivas

Los numerosos problemas que lleva utilizar una representación simbólica, han conducido a estudios más efectivos de representación del conocimiento. Las arquitecturas reactivas se caracterizan por no tener como elemento central de razonamiento un modelo simbólico y no utilizar razonamiento simbólico complejo. [7]

Esta arquitectura se basa en las hipótesis de que la inteligencia es una propiedad emergente de ciertos sistemas complejos y con ello, generar comportamientos inteligentes sin necesidad de un modelo simbólico.

La mayor parte de la aplicación de este tipo de arquitectura se ha centrado en el desarrollo de controladores en robótica.

2.3.3 Híbridas

Arquitecturas deliberativas como reactivas poseen limitaciones. Por ello, han propuesto sistemas híbridos que combinan aspectos de ambos modelos. En primera instancia se pretende construir dos subsistemas: uno deliberativo que utilice un modelo simbólico y otro reactivo, centrado a reaccionar ante eventos que tengan lugar en el entorno. [7]

2.3.4 FIPA

Con toda nueva tecnología dos de los problemas que hay que resolver son: la interoperabilidad y la apertura. Para ello la organización que más ha trabajado en ese sentido es FIPA (*Foundation for Intelligent Physical Agents*). En las especificaciones FIPA, se definen varias características que deben cumplir las plataformas de gestión de sistemas multiagente. Aparte de implementaciones concebidas a partir de este estándar como FIPA OS y JADE, donde la mayoría de los entornos de desarrollo y ejecución de agentes tienden a ser compatibles con FIPA.

Al definir una plataforma de agentes FIPA cumple con definir sólo el comportamiento externo (interfaz), dejando las decisiones de diseño a cada desarrollador. El modelo FIPA establece un modelo lógico referente a la creación, destrucción, registro, localización y comunicación de agentes.

La Plataforma agente proporciona la infraestructura para el desarrollo y uso de agentes, esta plataforma contiene todos los recursos hardware y software (sistema operativo, software de comunicación, middleware y software de gestión de agentes).

El estándar FIPA define los servicios que debe proporcionar toda plataforma de agentes: [7]

- **Sistema de Gestión de Agentes** (*Agent Management System, AMS*): Elemento de gestión central que controla el estado y acceso a la plataforma. También proporciona un servicio de páginas blancas que asocia un nombre que identifica a un agente, de tal forma que proporciona un método básico para buscar un agente dentro de la plataforma.
- **Facilitador de Directorio** (*Directory Facilitator, DF*): Complemento del servicio de páginas blancas, incluye un servicio de páginas amarillas, que permite buscar un agente no sólo por su nombre sino por sus capacidades. Los agentes se registran en DF indicando los servicios que ofrecen.
- **Canal de Comunicación de Agentes** (*Agent Communication Chanel, ACC*): Todos los agentes tiene acceso al menos a un ACC, canal de comunicación por defecto entre agentes de diferentes plataformas.
- **Sistema encargado de Transporte de Mensajes** (*Internal Platform Message Transport, IPMT*): Método de intercambio de mensajes dentro de la misma plataforma el cual depende de la implementación.

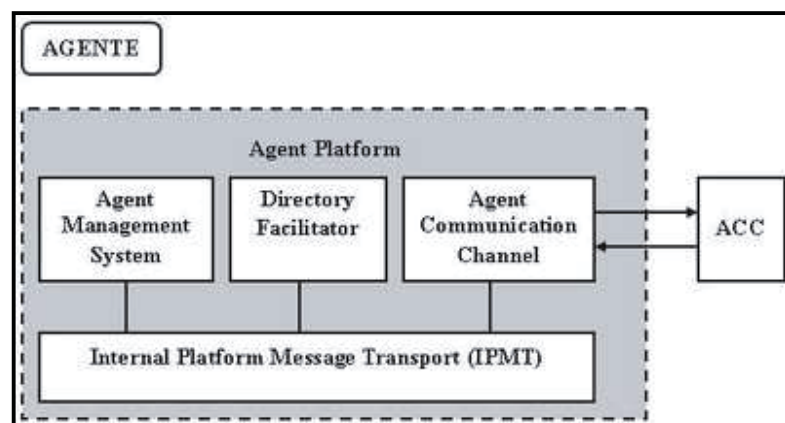


Ilustración 2.1 Modelo Referencia Agente FIPA [8]

2.4 Comunicación entre Agentes

La solución a un problema de forma distribuida requiere de coordinación de actividades, para solucionar este problema han aparecido una serie de iniciativas que han tratado, de especificar una serie de estándares que consigan una interacción real y homogénea entre sistemas tan distintos como son los SMA.

2.4.1 FIPA CAL (Communicative Act Library)

Librería compuesta de 22 acciones comunicativas que proporcionan la siguiente información: [7]

- Descripción detallada del lenguaje natural de la acción y sus consecuencias.
- Modelo formal escrito en SL de la semántica de la acción, sus precondiciones formales y efectos.

Ejemplos de uso del acto comunicativo

Agruparemos las acciones comunicativas en cuatro clases:

- Información.
- Realización de Acciones
- Negociación
- Intermediación

En las acciones intervienen dos agentes un emisor y receptor. El emisor puede tomar dos papeles, a los cuales denominaremos P1 (ofrece información) y P2 (solicita información).

Acciones de información en función de P1:

- *Inform(x)*: Emisor informa al receptor que la proposición es verdadera.
- *Inform-if(x)*: Marco de una acción contenida dentro de otra acción donde el emisor pide que el receptor informe si la afirmación es verdadera o falsa.
- *Inform-ref(Ref x g(x))*: Contenido de la acción expresión referencial. Emisor pide que se le informe del objeto de conjunto de objeto que coincida con la expresión referenciada.
- *Confirm(x)*: El emisor informa al receptor que x es cierta, sabiendo que receptor tiene incertidumbre de la certeza de x .
- *Disconfirm(x)*: El emisor informa al receptor que x es falsa, sabiendo que receptor tiene incertidumbre de la certeza de x .
- *Not-understood(a,x)*: el indica que la acción ha sido percibida pero no comprendida.

Acciones de información en función de P2:

- *Query-if(x)*: emisor pide al receptor que comunique con un acto *inform* se la proposición es cierta.
- *Query-ref(Ref x g(x))*: emisor pide receptor comunique con un *inform* todo el conjunto de objetos cuya descripción coincide con la expresión referenciada.
- *Suscribe(Ref x g(x))*: versión persistente de *query-ref*. Continuará enviando más actos *inform* cada vez que dicho valor cambie.

Ahora diremos que el emisor desempeña el ROL1 si es el agente interesado en realizar una acción, ROL2 si es el agente que realizará la acción.

Acciones de realización en función de P1:

- *Request(a)*: emisor requiere que receptor lleve a cabo alguna acción *a*.
- *Request-when(<j ; act>, x)*: emisor quiere que el receptor *j* lleve a cabo alguna acción *act*, cuando *j* sea una proposición verdadera.
- *Request-whenever(<j ; act>, x)*: emisor quiere que el receptor *j* lleve a cabo alguna acción *act*, tan pronto como *j* crea que la proposición *x* es verdadera.

Acciones de realización en función de P2:

- *Agree(<i ; act>, x)*: el emisor comunica al receptor *i* que está de acuerdo en llevar a cabo en el futuro la acción *act*, cuando la preposición *x* es verdadera.
- *Refuse(<i ; act>, x)*: el emisor comunica al receptor *i* que está de acuerdo en llevar a cabo en el futuro la acción *act*, cuando la preposición *x* es verdadera.
- *Cancel(a)*: esta acción permite al agente emisor cancelar o detener la acción llevada a cabo o de continuar ejecutando alguna acción con *a*.
- *Failure(a, x)*: esta acción permite al agente emisor comunicar al receptor que la ejecución de la acción *a* fue realiza pero fue fallida debido a la razón *x*.

El emisor desempeña un P1 cuando el agente relacionado en la acción no conoce las condiciones bajo las cuales se realiza la acción, por el contrario P2 cuando el agente coopera en que la acción se realice.

Acciones de negociación en función de P1:

- *Cfp(<j ; act>. Ref x g(x))*: acción de propósito general para iniciar un proceso de negociación haciendo una llamada para realizar la acción dada.
- *Accep-proposal(<j ; act>, x)*: el emisor comunica *j* se te acepta la propuesta dada la acción *act* y que se llevara a cabo la proposición *x*.
- *Reject-proposal(<j ; act>, x, y)*: acción de rechazar un propuesta. El emisor informa al receptor *j* que debido a la proposición y la acción *act* con precondición *x* no será llevada a cabo.

Acciones de negociación en función de P2:

- *Propose(<i ; act>, x)*: emisor propone llevar a cabo una acción *act*, siempre y cuando la proposición *x* se verdadera además de su precondición.

Aquí diremos que el agente emisor representa P1 cuando solicita intermediación. R2 corresponde al agente cuando realiza de puente entre el primer agente y todos los demás.

Acciones de intermediación en función de P2:

- *Propagate(Ref x g(x), <i; cact>, x)*: emisor quiere que el agente receptor actúe de intermediario, que reciba el mensaje *cact* y además propague dicho mensaje a los agentes referenciados, siempre y cuando verifique la restricción *x*. Y que los posibles receptores propaguen el mensaje.
- *proxi(Ref x g(x), <i; cact>, x)*: ídem al anterior con la diferencia que el mensaje *j* no va dirigido al intermediario, y además los posibles receptores no han propagado el mensaje.

2.4.2 FIPA ACL

En FIPA encontramos la definición del lenguaje de comunicación entre agentes FIPA ACL. Corresponde a la estructura que deben tener los mensajes. Un mensaje ACL debe contar con un conjunto de elementos de mensajes de los que no todos son obligatorios. El único elemento obligatorio es del tipo performativa (*performative*), dicho elemento denota el tipo de acto comunicativo. [7]

Tabla 2.1 Elementos mensaje FIPA ACL [7]

Elemento de Mensaje	Categoría de elementos
Performative	Tipo de acto comunicativo
Sender	Participante en la comunicación
Receiver	Participante en la comunicación
Reply-to	Participante en la comunicación
Content	Contenido mensaje
Language	Descripción del contenido
Encoding	Descripción del contenido
Ontology	Descripción del contenido
Protocol	Control de la conversación
Conversation-id	Control de la conversación
Reply-with	Control de la conversación
in-reply-to	Control de la conversación
Reply-by	Control de la conversación

Este es un ejemplo de mensaje donde podemos ver el agente *j* responde con un error a una acción previamente pedida por el agente *i* (abrir el archivo). La etiqueta *performative* se ha omitido ya que el mensaje en su conjunto está asociado a ese elemento. [7]

```
(failure
  :sender ( agent-identifier :name j)
  :receiver (set (agent-identifier : name i))
  :content
    ((action (agent-identifier :name j)
      (open "foo.txt")
      (error-message "No such file: "foo.txt")))
  :language FIPA-SL)
```

2.4.3 KQML

Utiliza una sintaxis similar a ACL. Pero KQML está muy asociado al lenguaje de representación desarrollado por KIF (*Knowledge Interchange Format*). KIF es una implementación de lógica de predicados de primer orden, donde sus principales ventajas está la de que incluye mecanismos para traducción semiautomática desde y hacia los principales lenguaje de representación de conocimiento.

Tabla 2.2 Elementos reservados de un mensaje KQML [7]

Elemento de mensaje	Descripción
:sender	nombre del emisor del mensaje
:receiver	El receptor del mensaje
:in-reply-to	La etiqueta para la respuesta a un mensaje anterior
:reply-with	Si el emisor espera respuesta, la etiqueta para dicha respuesta
:language	Nombre del lenguaje de representación del parámetro :content
:ontology	Nombre de la ontología usada en el parámetro :content
:content	Información sobre la que la performativa expresa una actitud

A continuación un ejemplo de mensaje *inform* en KQML.

```
(tell
  :language Prolog
  :ontology motors
  :in-reply-to j1
  :content weather(today, raining)
  :receiver j
  :sender i )
```

2.4.4 Protocolos de comunicación

La estructura de comunicación entre agentes corresponde a menudo con patrones típicos de secuencia de mensajes. A esta secuencia se denomina protocolos de interacción (IP). En la especificación FIPA encontramos protocolos muy comunes basados en aplicaciones basadas en agentes FIPA IPL (*Interaction Protocol Library*).

Para la definición de protocolos FIPA ha definido un lenguaje de representación de protocolos llamado AUML, basado en los diagramas de secuencia de UML que se denominan diagramas de protocolo. Donde se distinguen los siguientes elementos:

- *Roles de los agentes*: Papel de cada agente dentro de la comunicación.
- *Línea de vida*: Define la vida en que existe el agente.
- *Hilos de interacción*: Periodo de tiempo en que el agente está realizando una tarea.
- *Mensajes*: Representación de la comunicación de agente a otro.

El siguiente cuadro muestra un resumen de protocolos de comunicación en el estándar FIPA:

Tabla 2.3 Protocolos de comunicación estándar FIPA [7]

Nombre Protocolo	Descripción
Request	A un agente se le pide que realice cierta acción.
Request when	A un agente se le pide que realice cierta acción siempre que se cumpla la precondición.
Query	A un agente se le pide que informe sobre algo.
Contract net	Un agente pide la realización de cierta tarea a un conjunto de agentes. Estos dan su propuesta basada en unos costes y el iniciador elige quien la realiza finalmente
Brokering	Un agente (broker) ofrece las funcionalidades de otros agentes o reenvía las peticiones al agente apropiado.
English auction	Varios agentes participan en una subasta que se inicia con un precio más bajo y progresivamente se va subiendo.
Dutch auction	Varios agentes participan en una subasta que se inicia con un precio más alto y progresivamente se va bajando.
Recruiting	En como el <i>brokering</i> , pero las respuestas sobre el servicio van directamente al agente que lo necesita.
Propose	El iniciador propone a una serie de agentes la realización de una tarea y estos aceptan o no.
Subscribe	Un agente pide ser notificado si cierta condición se vuelve verdadera.

2.5 Metodologías de desarrollo Orientado Agentes

2.5.1 GAIA

Metodología que da pautas para seguir un desarrollo de SMA, surge como respuesta a la programación orientada a agentes. [9]

Añade nuevas abstracciones que deben ser definidas y agregadas a las ya existentes:

- **Entorno:** modelado de todas las entidades y/o recursos del entorno que un SMA pueda explorar, consumir y controlar para conseguir su objetivo.

Dichas entidades puede ser:

- ✓ Físicas: Temperatura, velocidad, etc.
- ✓ Virtuales: base de datos, servicios web, etc.
- **Rol:** El rol de un agente define la forma intrínseca de la posición sobre la organización, asociado a unos comportamientos establecidos.
- **Interacciones:** definen los distintos protocolos utilizados entre los diferentes roles para su comunicación.
- **Reglas:** Identifican funciones y protocolos, especifican posibles limitaciones entre diversos roles y protocolos.
- **Estructuras:** Modelo de estructuras inspirados del modelo del mundo real.

La metodología GAIA se definen tres fases de definen un SMA.

- Análisis.
- Diseño Arquitectura.
- Diseño en detalle.

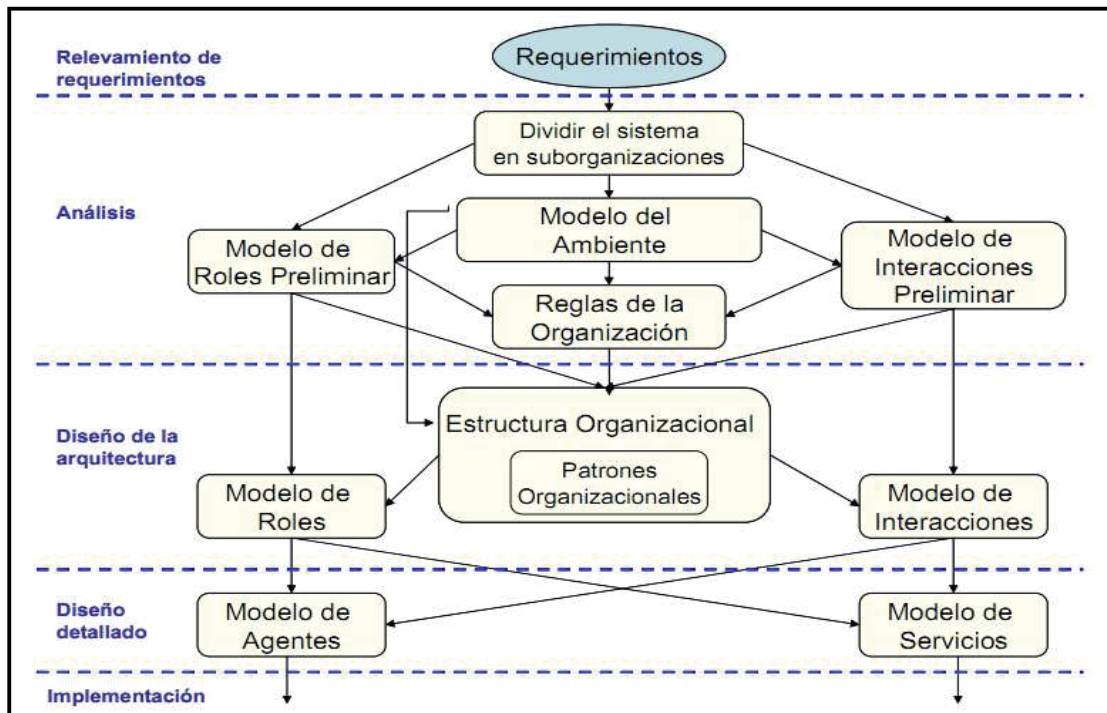


Ilustración 2.2 Fases Metodología GAIA [9]

2.5.2 Tropos

Metodología orientada al proceso de desarrollo dirigido por requerimientos, donde el análisis de éstos, tomados de manera temprana ayuda a tener mayor comprensión del entorno donde estará el sistema en desarrollo. La metodología TROPOS incluye cinco fases descritas a continuación. [10]

- **Requerimientos Iniciales:** Identificación de actores relevantes y sus objetivos.
- **Requerimientos Posteriores:** Sistema a desarrollar se considera un actor, a partir de las dependencias con otros actores se identifican sus obligaciones.
- **Diseño de Arquitectura:** Se introducen más actores del sistema mediante la descomposición de tareas y objetivos del sistema.
- **Diseño Detallado:** Definición de los actores del sistema en detalle, incluyendo protocolos de coordinación y comunicación.
- **Implementación:** Transformación de las especificaciones en plantillas adaptadas a una plataforma de programación de agentes.

2.5.3 INGENIAS

Metodología que propone una serie de meta-modelos que capturan las distintas vistas de un SMA. [10]

- *Organización*
- *Tareas / Objetivos*

- *Agente / Rol*
- *Interacción*
- *Dominio*

INGENIAS aborda con mayor profundidad la especificación y notación de los elementos que constituyen un SMA, tales como: agentes, roles, organización, servicios, objetivos, tareas, entorno, etc. Además ofrece un modelo de ciclo de desarrollo de SMA dividido en: etapas, actividades, productos y métodos.

INGENIAS ofrece una serie de modelos para el desarrollo de un SMA:

- *Modelo de Organización:* Estructura del SMA, roles, relaciones de poder, workflows.
- *Modelo de Agente:* Lo agentes realizan tareas o persiguen objetivos y poseen control y estado mental.
- *Modelo de objetivos y tareas:* Identificación de objetivos generales y específicos.
- *Modelo de Interacción:* Interacciones entre agentes/roles.
- *Modelo Entorno:* Entidades y relaciones con el entorno SMA.

2.5.4 PASSI

Metodología que da pautas para seguir un desarrollo de SMA, surge como respuesta a la programación orientada agentes. PASSI utiliza UML como lenguaje de modelado debido que es ampliamente aceptado en el ámbito industrial como académico. El ciclo de desarrollo de software es iterativo (excepto código y despliegue), abarca el análisis, diseño e implementación desde un enfoque bottom-up. Está compuesto por cinco modelos: [7]

- 1) *Modelo de requerimientos del Sistema:* Este modelo está compuesto por cuatro fases que buscan dar forma a los requerimientos del sistema.
 - **Descripción del Dominio:** Descripción general del sistema utilizando caso de usos convencionales.
 - **Identificación de Agentes:** Se separan las responsabilidades de los agentes encapsulando los casos de uso mediante paquetes UML estereotipado, si existe alguna comunicación entre paquetes dicha comunicación es un estereotipo llamado <<communicate>>
 - **Identificación de Roles:** Esta fase utiliza diagramas de secuencia para explorar los distintos roles que realizan los agentes en distintos escenarios.
 - **Especificación de Tareas:** Última fase del modelo de requerimientos, utiliza diagramas de actividad, en la cual se describen todas las actividades que realizan los agentes para conseguir sus objetivos.
- 2) *Modelo de Sociedad de Agentes:* Modelo de interacciones sociales y dependencias de los agentes que forman la solución.

- **Descripción de la Ontología del Dominio:** Utiliza diagramas de clases para representar la ontología del dominio que se describe en términos de conceptos, predicados y acciones.
 - **Descripción de la Ontología de Comunicación:** Se utilizan diagramas de clases, para representar la comunicación entre agentes. Cada agente se representa en: término de conocimiento y roles durante la comunicación.
 - **Descripción de Roles:** Uso de diagramas de clases para representan los distintos roles, colaboraciones y comunicaciones en que participa un agente.
 - **Descripción de Protocolos:** Existen protocolos de comunicación estándar, pero se pueden generar nuevos donde se usan diagramas de secuencias para especificar la gramática en términos de performative.
- 3) *Modelo de implementación de Agentes:* Modelo de la arquitectura de la solución en términos de clases y métodos.
- **Definición de Estructura de Agentes:** Diagramas de clases que especifican la estructura interna del agente, atributos, métodos, además de las tareas. Involucra la arquitectura del sistema en general en el cual ve a los agentes y las tareas que realizan.
 - **Descripción Comportamiento de Agentes:** Diagrama de actividad que especifican el comportamiento de los agentes individuales y el flujo de acontecimientos entre los agentes.
- 4) *Modelo de Código:* Presenta la solución a nivel de código, se emplea la reutilización de código y el código fuente propiamente tal.
- 5) *Modelo de Despliegue:* Distribución de las partes del sistema a través de procesamiento de hardware y migración de unidades de proceso.
- **Configuración de Despliegue:** Uso de diagramas de despliegue para describir las decisiones de los agentes a través de las plataformas y sus movimientos

Capítulo 3

Marco Teórico Web Services

3.1 Web Services

Existen múltiples definiciones sobre lo que son los Servicios Web (WS), lo que muestra lo complejo que es dar una descripción que englobe todo lo que implica. Una definición será hablar de los WS como un conjunto de tecnologías o aplicaciones con la capacidad de interactuar con la web. Estas aplicaciones o tecnologías intercambian datos con el fin de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios de los servicios solicitan estos procedimientos a través de la Web. [11]

Según la W3C, define un Servicios Web como:

“Un servicio Web es una aplicación software identificada mediante una URI, cuya interfaz (y uso) es capaz de ser definido, descrito y descubierto mediante artefactos XML, y soportar interacciones directas con otras aplicaciones software usando mensajes basados en XML y protocolos basados en Internet” [12]

En sencillas palabras un Servicios Web es un procedimiento remoto que se ejecuta donde un proveedor de servicios responde a la llamada de un solicitador de servicios, todo esto bajo ciertos estándares de comunicación que permiten que plataformas de software distribuido no tengan problemas de comunicación a la hora de establecer un llamado procediendo remoto (RPC) entre el solicitador y el proveedor.

A continuación se presentan algunas de las características más destacadas de un WS:

- *Interoperabilidad:* Los servicios web se pueden solicitar por clientes de otras plataformas.
- *Acceso externo desde Internet:* Los servicios web gestionan de buena forma los accesos de internet que solicitan los clientes.
- *Tipos de datos de las Interfaces:* Tipo de datos definidos por el servicio web que corresponde con los tipos de datos definidos en la mayoría de los lenguajes.
- *Uso de los estándares de Internet:* Los WS utilizan los estándares de internet.
- *Soporte de cualquier Lenguaje:* La implementación de un WS no está ligado a un lenguaje de programación en particular.

- *Soporte para cualquier arquitectura de componentes Distribuidos:* los WS no están ligados a una arquitectura de componentes en particular.

3.2 Arquitectura Web Services

Antes de analizar la pila de protocolos en la arquitectura de los Servicios Web es importante considerar una serie de conceptos para comprender con mayor claridad lo que implica la estructura de ellos.

3.2.1 Conceptos

- *Cliente del Servicio:* Es el que solicita la ejecución del servicio web, por lo tanto es quien lo consume.
- *Proveedor del Servicio:* Encargado de implementar el Servicio y ofrecerlo a los clientes.
- *Registro del Servicio:* Directorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

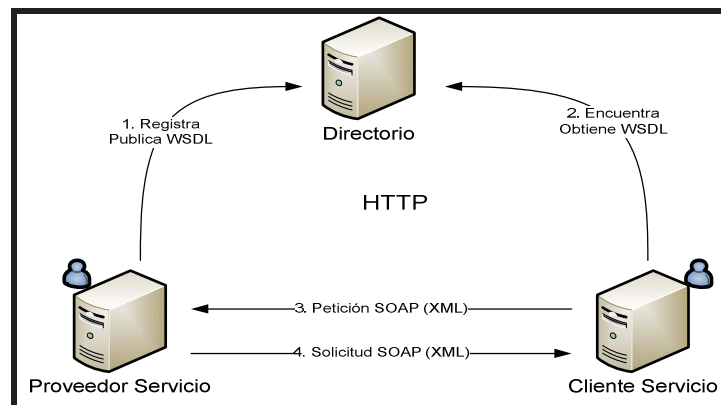


Ilustración 3.1 Modelo básico de Web Services [13]

Dada la Ilustración 3.1 la secuencia de ejecución es la siguiente:

1. El proveedor del servicio da de alta el servicio web en el directorio. Para realizar esto, el proveedor almacena en el directorio el documento de descripción del servicio web.
2. El cliente del servicio busca en el directorio el servicio web que pueda adaptarse a sus necesidades.
3. Una vez seleccionado el servicio, el cliente lo invoca mediante el envío del mensaje SOAP, el cual indica la acción a realizar y los datos de entrada.
4. El servicio web recibe la petición y ejecuta la funcionalidad. Para finalizar, envía el mensaje SOAP al cliente con los resultados obtenidos.

3.2.2 Pila de Protocolos y Lenguajes.

Lo bloques necesarios para construir una aplicación remota se detallan con una breve descripción de cada bloque y su función.

- **Discovery:** Permite al cliente conocer la ubicación del WS.
- **Description:** Proporciona al cliente información para que pueda interactuar con el WS. Estructura de metadatos de su interfaz (WSDL) hasta documentación detallada sobre la funcionalidad y ejemplos de uso.
- **Message Format:** Especifica formato de codificación de los mensajes para que el cliente y el servidor puedan comunicarse.
- **Encoding:** Codificación de los datos que se transmiten en el cuerpo del mensaje.
- **Transport:** Realiza la transferencia del mensaje entre cliente y servidor mediante un protocolo de transporte.

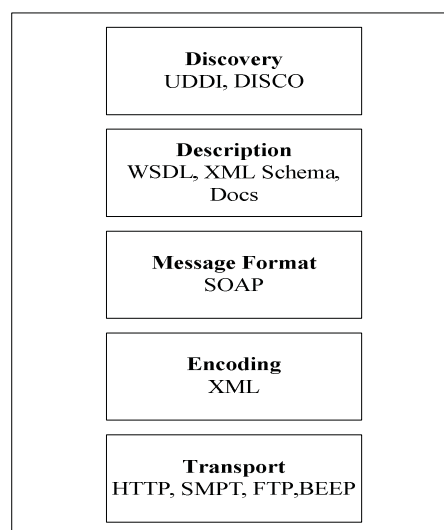


Ilustración 3.2 Bloques construcción aplicación remota [14]

3.2.2.1 HTTP

Protocolo usado en cada transacción web, es considerado un protocolo sin estado es decir no guarda ninguna información sobre conexiones anteriores. Utilizado para la transferencia de las transacciones XML.

3.2.2.2 SMPT

Protocolo de la capa de aplicación utilizado ampliamente para el envío de mensajes de correo electrónico.

3.2.2.3 FTP

Protocolo de red para la transferencia de archivos entre sistemas conectados a una red, permite realizar modificaciones en equipos remotos evitando el uso de archivos en el cliente.

3.2.2.4 BEEP

Protocolo también conocido como BXXP, el que está diseñado para la interacción asíncrona punto a punto sobre una red TCP/IP.

3.2.2.5 XML

XML es un lenguaje de etiquetado extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de datos. Es un lenguaje muy similar a HTML pero su función principal no es mostrar datos sino que describirlos. XML permite la lectura de datos a través de diferentes aplicaciones. [15]

La estructura de un documento XML son:

- **Prólogo:** Elemento opcional del documento que puede empezar con unas líneas que describen la versión XML, el tipo de documento, entre otras cosas, como la codificación de caracteres, la estructura del documento y las hojas de estilos.
- **Cuerpo:** No es opcional, el cual debe contener un y solo un elemento raíz.
- **Elementos:** Pueden tener contenido o bien estar vacíos, pueden ser caracteres, elementos XML o ambos.
- **Atributos:** Los elementos pueden tener atributos, que incorporan características o propiedades. Deben ir entre comillas.

Un ejemplo de documento XML se muestra a continuación: [15]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<libro>
  <titulo></titulo>
  <capitulo>
    <titulo></titulo>
    <seccion>
      <titulo></titulo>
    </seccion>
  </capitulo>
</libro>
```

XML ofrecen un conjunto de servicios útiles para los usuarios, además sirve para estructurar, almacenar e intercambiar información.

- **XSL:** Lenguaje extensible de hojas de estilo, cuyo fin es mostrar cómo debería estar estructurado el contenido, diseñado y paginado en un medio de presentación como lo puede ser un navegador web.
- **XPath:** Lenguaje de ruta XML, permite acceder a partes de un documento XML.
- **XLink:** Lenguaje de enlace XML, permite insertar elementos en documentos XML, para crear enlaces entre recursos XML.
- **XPointer:** Lenguaje direccionamiento XML, permite el acceso a la estructura interna de un documento XML. Esto es a sus elementos, atributos y contenidos.
- **XQL:** Lenguaje de consulta, facilita la extracción de datos de un documento XML.

3.2.2.6 SOAP

Protocolo que proporciona una definición de información basada en XML que puede ser usada para el intercambio de información estructurada, y de tipos concretos entre puntos entre un entorno descentralizado, distribuido.

SOAP es fundamentalmente un paradigma de intercambio de mensajes en un solo sentido, sin estado, pero las aplicaciones pueden tener relaciones de comunicación más complejas combinando tales intercambios de un solo sentido con características proporcionadas por el protocolo utilizados y/o información específica de la aplicación. [16]

Un mensaje SOAP es una transmisión en una dirección entre nodos SOAP, se espera que los mensajes SOAP sean combinados por las aplicaciones para implementar patrones de interacción más complejos. Los elementos que componen un mensaje SOAP son:

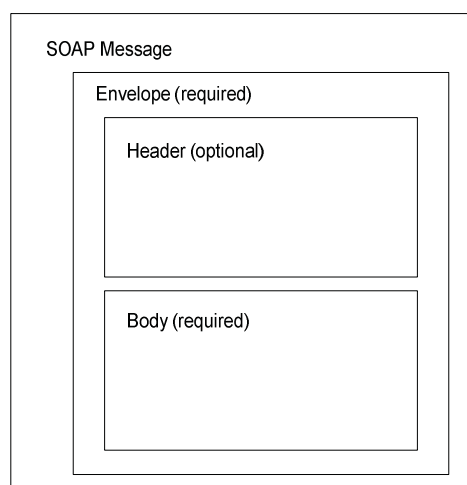


Ilustración 3.3 Elementos Fundamentales de un mensaje SOAP [17]

- **Envelope:** Elemento raíz de un mensaje SOAP contiene los elementos *header*, *body* y *fault*.
- **Header:** Elemento opcional ofrece marco de trabajo flexible para una especificación adicional de la aplicación.
- **Body:** Obligatorio, contiene el mensaje propiamente tal, almacena el documento XML que el receptor final procesara.

3.2.2.7 WSDL

WSDL es el lenguaje de descripción de servicios web basado en XML, describiendo la interfaz ofrecida por un Web Services, operaciones que un servicio puede soportar y sus parámetros de entrada y salida. [21]

El elemento principal que posee un documento WSDL es el de *definitions* que define el nombre del servicio web, además admite otras extensiones opcionales que componen el bloque.

- **Types:** Contiene definición de los mensajes que pueden ser enviados o recibidos por un servicio.
- **Message:** Proporciona las asociaciones entre los mensajes y su definición en el esquema.
- **PortType:** Define un conjunto de interfaces que ofrece el WS. Cada interfaz está asociada a uno o más mensajes. Se especifica una interfaz por cada uno de estos métodos de acceso que se deben utilizar SOAP, HTTPGET o HTTPPOST.
- **Binding:** Asocia la definición portType con un protocolo concreto.
- **Service:** Define los puertos ofrecidos por el WS, dichos puertos son cada elemento de Binding con su correspondiente URL.

3.2.2.8 XML Schema

Lenguaje utilizado para describir la estructura y restricciones de los documentos XML de una forma más precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Fue desarrollado por la W3C.

3.2.2.9 UDDI

UDDI (Universal Description, Discovery and Integration) proporciona un medio de directorio para publicar acerca de WS. En el directorio se publican los ficheros WSDL que describen los WS que se ofrecen.

La información publicada en los directorios se puede dividir en tres categorías:

1. **Paginas Blancas:** Incluye información específica de un servicio. Ejemplo: nombre, descripción, información de contacto, etc.

2. **Páginas Amarillas:** Permite buscar servicio por categoría o clasificación.
3. **Páginas Verdes:** Contiene información técnica del servicio web. Generalmente, contiene un puntero a una especificación externa y una dirección para invocar el servicio web.

3.2.2.10 DISCO

Alternativa a UDDI; la diferencia es que en UDDI es complicado averiguar que WS ofrece una determinada máquina. DISCO ofrece un acceso a los WS de manera similar a lo que es un enlace HTML, presentando una lista de WS y referencia a otros directorios DISCO. [12]

3.3 Arquitectura Orientada a Servicios

La arquitectura SOA ofrece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red se pueda acceder a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es a través de WS.

La estrategia de la orientación a servicios permite la creación de éstos y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. Puesto que los servicios están diseñados para ser autónomos e independientes pueden combinarse y re-combinarse con suma facilidad, en aplicaciones más complejas que respondan a las necesidades de cada momento en una organización.

3.3.1 Características SOA

SOA como arquitectura de sistemas descentralizado, distribuido posee las siguientes características:

- **Orientada al Mensaje:** servicio definido en torno a los mensajes enviados ente el solicitante y el proveedor.
- **Vista Lógica:** Imagen que muestra los componentes que forman el sistema y explicando que hace cada uno de ellos.
- **Orientado a red:** Los servicios web deben ser usados sobre la red, aunque no es obligación.
- **Orientado a la Descripción:** Un servicio web es descrito por metadatos procesables por la máquina.
- **Granularidad:** servicios con pequeñas cantidad de operaciones, pero con gran cantidad de mensajes.
- **Plataforma neutral:** Mensajes creados en una plataforma neutral, utilizando un lenguaje estándar (XML).

3.3.2 Beneficios SOA

Los beneficios de SOA para una organización se manifiestan en dos niveles: usuario y organizacional. [18]

- **Mejorar en toma de decisiones:** Integrando servicios e información de un conjunto de aplicaciones dinámicas compuestas.
- **Mejorar productividad:** Un acceso a los sistemas de información y la posibilidad de mejorar los procesos de productividad individual.
- **Potenciar relaciones cliente proveedor:** las ventajas de SOA trascienden las fronteras de la organización. Mejora la calidad de respuesta de los clientes y flexibiliza los procesos críticos de compras y gestión de pedidos.
- **Aplicaciones más productivas y flexibles:** Conseguir mayor productividad de los recursos ya existentes sin tener que aplicar soluciones de integración, además de independencia de plataformas y lenguajes de programación.
- **Desarrollo de aplicaciones más rápido y económico:** diseño de servicios reutilizables que se pueden combinar con servicios de mayor nivel y aplicaciones compuestas. Con ello se reduce el costo del desarrollo de soluciones.

3.4 Tecnología en Web Services

Para el desarrollo de aplicaciones WS existen varios lenguajes que soportan el desarrollo, los que más destacan son .NET y Java.

3.4.1 Tecnología .NET

En los Servicios Web, .NET Framework es el modelo de programación de código gestionado por Microsoft para la creación de aplicaciones sobre la plataforma Windows. .NET Framework indica soluciones pre-codificadas que anteriormente han sido generadas mediante lenguajes de programación y herramientas de distintos tipos, además .NET Framework proporciona un soporte apropiado para los Servicios Web de forma que los desarrolladores puedan codificar, descubrir, depurar, instalar y consumir servicios web.

La versión 3.5 apareció el 2007 amplía las interfaces de programación de la 2.0 con nuevas tecnologías para la creación de aplicaciones.

3.4.1.1 Visual Studio

Entorno de desarrollo profesional para aplicaciones sobre la plataforma Windows, Visual permite el consumo de servicios web en aplicaciones de Windows, Office, aplicaciones web y dispositivos móviles. Además facilita la publicación y localización de nuevos servicios web en una organización.

3.4.2 Tecnología JAVA

La tecnología JAVA (J2EE) proporciona una potente y completa plataforma orientada al desarrollo de aplicaciones corporativas distribuidas y a los Servicios Web. La tecnología J2EE integra un conjunto de APIs, framework y patrones de programación que permiten responder de forma robusta y flexible.

3.4.2.1 JAX-RPC

API estándar en Java que permite implementar e invocar operaciones de servicios web SOAP mediante el paradigma RPC, forma parte del API de JavaEE, como cualquier otro API está formado por un conjunto de interfaces (javax.xml.rpc). Además, posee múltiples implementaciones por ejemplo: Apache Axis, entre otras. [19]

3.4.2.2 JAX-WS

Sustituto de JAX-RPC, en lo que concierne a soporte para RPC, conceptualmente es equivalente a JAX-RCP, posee anotaciones para facilitar el desarrollo y proporciona soporte para invocaciones asíncronas, aunque actualmente la mayor parte de las aplicaciones Java usan JAX-RPC. La implementación de referencia forma parte del proyecto Glassfish de Sun y conoce como JAX-WS RI. [20]

Capítulo 4

Marco Teórico Turismo

El turismo es una actividad cuyo crecimiento progresivo tiene un impacto cada vez más importante en el ámbito social, cultural y económico de un país. Esto produce efectos en las inversiones y en la construcción, el mejoramiento del transporte, lo que a su vez repercute en el bienestar de la comunidad.

El turismo hoy en día según la OMT (Organización Mundial del Turismo, dependiente de la ONU) lo define: “como todas las actividades realizadas por las personas durante sus viajes a lugares diferentes de su residencia habitual, por un periodo de tiempo inferior a un año y para ir de vacaciones, trabajar u otras actividades”. [4]

El turismo posee dos elementos económicos que permiten medir el valor real del turismo en un país y su aporte a la actividad económica nacional y regional, ellos son la oferta turística y la demanda turística:

- La demanda turística está compuesta por:
 - ✓ Perfil de los visitantes/turistas.
 - ✓ Medio de transporte utilizado.
 - ✓ Tipo de alojamiento utilizado.
 - ✓ Destinos elegidos.
 - ✓ Periodo y duración de estadía.
 - ✓ Actividades realizadas en el lugar visitado.
 - ✓ Finalidad o periodo del viaje.
 - ✓ Impacto económico en las diferentes ramas de actividad, en la ocupación e inversión.

- La oferta turística se define en torno a la satisfacción del consumo turístico, los elementos que la integran son:
 - ✓ Hoteles y establecimientos de alojamiento.
 - ✓ Inmuebles en alquiler para el turismo.
 - ✓ Restaurant, cafés.
 - ✓ Transporte.
 - ✓ Agencia de viajes y operadores de turismo.

4.1 Conceptos Básicos

El turismo es un conjunto de actividades realizadas por personas que viajan a lugares fuera de su entorno habitual, por motivos de ocio, negocio u otros motivos. Es una actividad multisectorial muy compleja, que genera de forma directa e indirecta una serie de beneficios a distintos niveles sociales.

Se podría decir entonces, que el turismo lo componen los siguientes elementos:

1. **Turista:** Persona que viaja y/o hospeda fuera de su domicilio habitual.
2. **Servicio Turístico:** Actividad realizada por el turista que permiten satisfacer las necesidades del turista.
3. **Proveedor Turístico:** Persona o empresa que presta el servicio turístico.
4. **Intermediario Turístico:** Rol que desempeña un agente de viaje para acercar los servicios turísticos a los turistas.

4.2 Clasificación del Turismo

El turismo se puede clasificar según dos elementos: el tipo de turismo que se lleva a cabo como tal el turista esto significa de donde viene y hacia dónde se dirige, y por temporada turística que caracterizan la situación de la oferta.

4.2.1 Tipo de Turismo

De acuerdo a cada país el turismo se reconoce de la siguiente forma: [22]

- **Turismo interno:** Los residentes del país viajan dentro del país.
- **Turismo receptor:** Los no residentes viajan dentro de un país dado.
- **Turismo emisor:** Los residentes de un país dado viajan a otro país.

Con estas tres formas básicas de turismo, pueden combinarse produciendo entonces las siguientes categorías:

- **Turismo Interior:** Incluye turismo interno y receptor.
- **Turismo Nacional:** Incluye turismo interno y emisor.
- **Turismo Internacional:** Incluye turismo receptor y emisor.

4.2.2 Temporada Turística

La temporada turística dentro de un año se divide en tres partes, y va depender del país en que el turista se encuentre para saber en cuál periodo de la temporada se encuentra. La

temporada turística siempre depende de un factor económico y se mueve según el comportamiento de la oferta y la demanda. Las temporadas son: baja, alta y media y se definen de la siguiente manera:

- **Temporada Alta:** Corresponde y coincide generalmente al acceso masivo de turistas a una zona, lo que provoca un incremento en los precios de los productos y servicios, también conocida por temporada estival, porque se asocia al verano de cada país.
- **Temporada Media:** Aún existen buenas condiciones climáticas favorables, pero los turistas poseen limitaciones de tiempo que impiden que se concrete un viaje, esta temporada se utiliza para realizar promociones y así realzar el atractivo turístico.
- **Temporada Baja:** Periodo donde los atractivos turísticos han perdido sus características y condiciones. Por ende provoca una baja en los precios de los servicios o productos ofrecidos y en algunos casos el cierre temporal de algunas empresas.

4.3 El Turista

El turista se puede clasificar de dos maneras, una es el viajero que pernocta y el otro es el visitante del día o excursionista.

El tipo de turista se puede identificar en detalle de la siguiente manera:

- **Visitante que pernocta o turista:** Es aquel que permanece al menos una noche en un medio de alojamiento colectivo o privado en un destino visitado, dentro de los tipos de visitantes encontramos visitantes extranjeros como nacionales. [4]
- **Visitante del día o excursionista:** Es aquel que no pernocta en un alojamiento del destino visitado, es decir, se encuentra menos de un día en el lugar. También, se encuentran visitantes tanto nacional como extranjero. [4]

4.4 Empresa Turística

El sector turístico está formado por empresas de distintos tipos que ofrecen mayoritariamente sus productos a los demandantes: los turistas. [23]

La relación entre la empresa turística y el turista está estrechamente ligada y se compone de los siguientes actores que forman parte del sector económico turístico:

- **Empresas de alojamiento:** Hoteles y similares (moteles, pensiones, apartamentos turísticos, etc.), paradores, casas rurales, balnearios (que ofrezcan servicio de hospedaje), etc.
- **Empresas de restauración:** Restaurantes, cafeterías, salas de fiestas, café-conciertos, etc.
- **Empresas distribuidoras:** Agencias de viajes (mayoristas, minoristas, mixtas, central de reservas).

- **Empresas de transporte y auxiliares:** Compañías aéreas, de autobuses, ferroviarias y navales, agencias de alquiler de vehículos y taxis.
- **Empresas de esparcimiento, recreo y cultura públicos o privados:** Museos, espectáculos, parques naturales, oficinas de información, centros deportivos, zoológicos, etc.
- **Empresas de servicios financieros y seguros especializados en el sector turístico:** Empresas que ofrecen cheques de viaje o seguro de viajes.
- **Empresas comerciales y auxiliares, dedicadas a la producción comerciales o bienes dirigidos a los turistas:** souvenirs, productos tipos y artesanías locales, guías de viajes, etc.

De la definición anterior se desprenden varios aspectos. En primer lugar, los productos turísticos son mayoritariamente servicios: servicios de alojamiento, servicio de transporte.

4.5 Servicios del Turismo

En cualquier sector existen empresas que ofrecen servicios y el turismo no es la excepción. Dichas empresas tienen como función principal actuar como intermediarios entre cliente/turista frente al servicio prestado. Los servicios turísticos ofrecidos pueden ser muchos dependiendo del lugar geográfico, el nivel socio económico, etc., pero los principales o los más comunes a la hora de viajar son: servicios de alojamiento, servicios de transporte, servicios de restauración, servicios de entretenimiento y servicios públicos.

4.5.1 Servicio de Alojamiento

El servicio de alojamiento ofrece un producto básico que es precisamente ese, el alojamiento u hospedaje en un edificio que tiene dedicada toda o una parte independiente del resto para prestar el servicio. El servicio de alojamiento se puede complementar por ejemplo con el servicio de restauración.

El servicio de alojamiento según la OMT se divide en hoteleros y extra-hoteleros:

- **Hoteles:** ofrece alojamientos con o sin servicios complementarios que se clasifican desde 1 a 5 estrellas.
- **Hoteles-Apartamentos:** cuenta con todos los servicios de un hotel, además cuenta con instalaciones y equipamiento para conservación, preparación y consumo de alimentos. Van desde 3 a 5 estrellas.
- **Moteles:** Establecimientos ubicados generalmente fuera de los lugares urbanos y las proximidades de las carreteras, en el que se facilita alojamiento en departamentos con entradas independientes.
- **Hostal o pensión:** Establecimiento que cumple con las funciones de hotel, pero que no alcanza con las condiciones mínimas necesarias para cumplir como tal.

4.5.2 Servicio de Restauración

El servicio de restauración implica aquellos establecimientos que expenden comidas preparadas o no preparadas en el lugar que se ofrece a los visitantes. Las condiciones del servicio son señaladas según el reglamento de restauración de cada país, que busca supervigilar las condiciones mínimas para ofrecer un buen servicio. Hay que nombrar que estos establecimientos varían su ambiente, tamaño, especialidad entre otras cosas según el público objetivo al cual este orientado o dirigido. A continuación algunos ejemplos de servicios de restauración:

- **Restaurants:** Establecimiento de carácter público que ofrece comidas, bebidas de carácter alcohólicas en algunos casos y no alcohólicas mediante un precio fijado por el proveedor, para ser consumidas en el local.
- **Cafeterías:** Establecimiento donde se despacha café y otras bebidas, complementando con aperitivos y comidas no muy sofisticadas.
- **Local de Comida Rápida:** Establecimiento donde se preparan alimentos y sirven para comer en el local o a pie en la calle.

4.5.3 Servicio de Transporte

El servicio de transporte en el área turística cumple con el traslado de pasajeros y/o turistas, donde encontramos tres sub-categorías: terrestre, aéreo y marítimo. La categoría que se dispondrá a detallar corresponde al transporte terrestre, que es provisto por compañías de autobuses que proveen desplazamiento total al turista, no obstante, existen compañías que ofrecen complementos de servicios al ya ofrecido, como lo son servicios de alimentación, o mayor calidad en el servicio de transporte.

Dentro de los servicios de transporte encontramos el alquiler de vehículos particulares. A continuación algunos ejemplos de servicios de transporte:

- **Transporte Urbano:** Corresponde al servicio prestado al interior de una zona urbana que es utilizada tanto por las personas que viven en la zona como por los turistas.
- **Transporte Inter-Urbano:** Corresponde al servicio que traslada viajeros que se desplazan entre zonas urbanas. Ejemplo. Santiago - Valparaíso.

4.5.4 Servicio de Entretención

Servicios que vienen a complementar las preferencias naturales hechas por el turista, son de carácter cultural, entretenimientos, misceláneas, aventura, etc. Los servicios de atracción se pueden describir en los siguientes elementos.

- **Museos:** Lugar donde se exponen objetos artísticos, científicos, cultural, o de otra índole.
- **Deporte Aventura:** Agencias que ofrecen servicios aventura, generalmente extremos.

- **Teatro y Cine:** Recintos donde se exponen obras de teatro o películas.
- **Recreativos:** Se ofrecen servicios de entretenimiento y diversión. Ejemplo. Parque de diversiones.

4.6 Efectos Turísticos

El turismo es una actividad que incorpora varios sectores a la vez, lo que genera de forma indirecta o directamente una serie de beneficios a la sociedad. Dada las múltiples implicancias, el turismo se considera un fenómeno propio de la sociedad actual. El hecho de ser social es dado que va dirigido a satisfacer necesidades de las personas. También posee una naturaleza económica, ya que es capaz de generar divisas al país que lo produce. La administración pública es otro punto, debido a que responde a las políticas y lineamientos económicos de un país en desarrollo. Otro punto importante es la cultura y la educación, la primera corresponde a la idiosincrasia de las personas de diferentes realidades geográficas, y la segunda va dada como un medio de formación intelectual.

4.6.1 Efectos Sociales

El turismo dentro de la sociedad se entiende como una forma de relajación y entretenimiento que trasciende transversalmente a todas las clases sociales en un país, que va dirigido a satisfacer las necesidades individuales y colectivas de la misma. La repercusión que tiene en la sociedad se ve reflejada en todos los elementos que forman un sistema social, ya sea político, económico, cultural y educacional. La sociedad es la punta angular de la pirámide de efectos que produce el turismo.

4.6.2 Efectos Económicos

Dentro de los efectos económicos que tenga dentro de un país, va la entera dependencia de la situación económica general que atraviesa el país, puesto que la demanda turística está ligada fuertemente a la situación actual de un país. Por otra parte el turismo requiere una fuerte mano de obra, sobre todo el mantener una reserva de trabajadores especializados. Por todo esto, es una fuente importante de empleo donde el volumen turístico es directamente proporcional a la cantidad de empleo.

4.6.3 Efectos Políticos

Los efectos políticos que produce el turismo poseen relación directa con los dos puntos nombrados anteriormente, debido a que las políticas públicas apuntan a apoyar y fomentar el turismo. El poder ejecutivo con sus órganos propios de gobierno se encarga de fomentar y representar al turismo frente al entorno. Por otro lado poder legislativo norma y regula todo este marco para que no existan excesos.

4.6.4 Efectos Culturales

Los efectos se ven tanto a nivel de culturas como la local o receptora como la foránea o emisora, lo que genera un intercambio de conductas, hábitos de distinta índole, y formas de

vida. Este efecto, en ocasiones positivo, puede llegar a ser perjudicial para la población autóctona, ya que puede perder su identidad propia.

4.6.5 Efectos Educativos

Por un lado el efecto educacional que se produce es el mejor conocimiento de otras sociedades y mentalidades, favorece el aprendizaje de idiomas extranjeros. Esto ocurre generalmente con estudiante de pre y post grado que realizan estudios de intercambio donde buscan tener más experiencia y conocer otras culturas, además de aprender otro idioma.

4.7 Turismo y Sistemas de Información

Las tecnologías de información y comunicación, como el comercio electrónico y las plataformas web ofrecen una perspectiva de economía digital.

Los sistemas de información en el turismo, especialmente los de planificación ayudan:

- Planificación.
- Dirección
- Organización y control

Las tecnologías de información y la irrupción del internet proponen un nicho de trabajo que está disponible las 24 horas del día y los 365 días del año desde cualquier parte del mundo agrupados en servidores web y aplicaciones.

Pero cuáles son los beneficios que ofrece el comercio electrónico desde un punto de vista turístico del proveedor son:

- Obtención de datos detallados en tiempo real.
- Reducción de costos de marketing.
- Menor costo operativo debido a las transacciones electrónicas.
- Acceso nuevos mercados y nuevas líneas de negocio.
- Mayor orientación asía el consumidor.

Por otro lado, los beneficios que ofrece el comercio electrónico al turista como tal:

- Obtención de más y mejor información sobre los productos.
- Posibilidad de buscar información o realizar compras en cualquier momento.
- Facilidad de comparación entre productos.
- Orientación comercial personalizada gracias a funcionalidades del sistema.
- Menores precios. Al obtener menores costos operacionales y de marketing por parte del proveedor.

Capítulo 5

Solución Propuesta

5.1 Descripción de la Solución Propuesta

La solución propuesta consiste en diseñar un sistema que se adapte a los diferentes dispositivos móviles, como PDA, Smartphone y teléfonos móviles. Para lograr esto se implemento un sistema híbrido (SMA y Servicios Web), que permite obtener un itinerario de viaje dada las preferencias del turista, como tipo de servicio, lugar geográfico y precios, además de permitir modificar el itinerario en el caso que se estime conveniente.

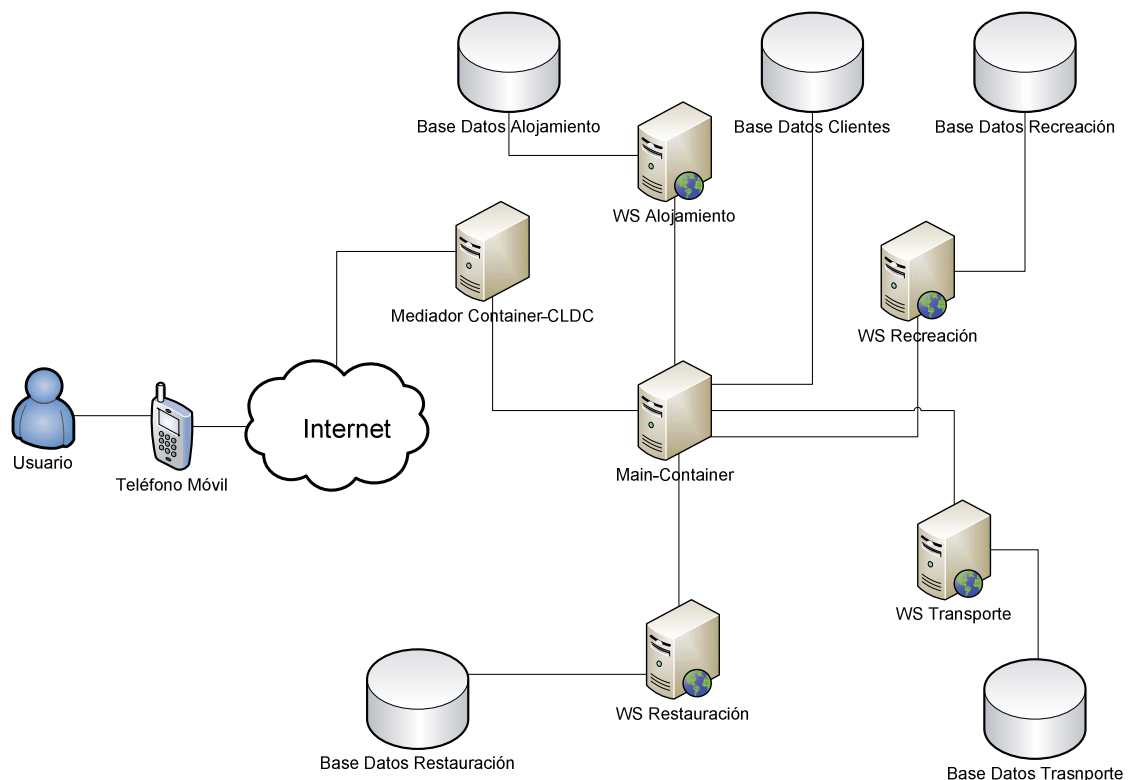


Ilustración 5.1 Sistema Híbrido basado en el dominio turístico.

El sistema consta de un dispositivo móvil y un mediador que provee la conexión para los agentes con el celular a través de internet, un main-container que proporciona los agentes principales del sistema y los servicios web donde los proveedores ofrecen sus servicios y son consumidos por el main-container.

Antes de comenzar con la descripción detallada hay que entender una serie de término. El primero es el “*perfil de usuario*”, se refiere a todos los datos personales que posee el

turista, segundo lugar “*Preferencias*”, nos habla de las necesidades del turista como por ejemplo: un viaje a la región del Biobío, donde requiere alojamiento, y transporte, además de una fecha tentativa. Para finalizar el “*Itinerario*”, que corresponde a los servicios ya seleccionados por el Turista para ser contratados por el mismo.

Los tipos de agentes que forman el sistema son cuatro:

- **GuiAgent:** Es el responsable de crear la interfaz del dispositivo móvil, mostrar los datos del usuario y desplegar las búsquedas que el realice, además es el impulsor de la comunicación del dispositivo móvil con el contenedor principal de los agentes.
- **IntermediarioMovil:** Interactúa con la base de datos del usuario, además recibe todas las peticiones del usuario y valida que se cumplan, en caso contrario se encarga de reportar el o los errores.
- **IntermediarioWS:** Procesa las peticiones de búsqueda de servicios web, se encuentra en modo escucha donde al recibir una petición, esta es resuelta por el servicio web para posteriormente retornar dicha respuesta y devolverla a su solicitante.
- **Planificador:** Realiza las validaciones de las preferencias, además de generar y administrar el uso del itinerario.

Los Servicios Web que componen la solución son cuatro:

- **Servicio Web Alojamiento:** Almacena la información relacionada a los servicios de alojamiento de todo el país.
- **Servicio Web Entetención:** Relacionado con los servicios misceláneos que puedan existir en torno al país, estos son guardados en este servicio web.
- **Servicio web Restauración:** Almacena la Información pertinente a lugares y tipo de alimentación disponible en el país por región.
- **Servicio web Transporte:** Guarda la información de transporte interurbano en torno al país. Además de los recorridos que estos pueden tener.

5.2 Análisis

Como se ha descrito anteriormente, la metodología de desarrollo del proyecto estuvo compuesta por un híbrido que consta de proceso unificado (UP) y PASSI, donde UP contiene a PASSI. Esto consiste en que a los flujos de trabajo se agregan fases de PASSI para de esta forma construir tanto los servicios web como los agentes sin alejarse demasiado de la solución a la problemática.

A continuación se describen los requisitos principales que componen el sistema híbrido:

- El turista debe crear su perfil, antes de tener acceso a todas las funcionalidades.
- El turista puede modificar su perfil.
- El turista debe acceder mediante usuario y contraseña.

- El turista debe ingresar sus preferencias.
- El turista debe seleccionar servicios ofrecidos para generar un itinerario.
- El turista puede cancelar servicios adquiridos.
- El turista puede eliminar los itinerarios.
- El sistema debe mostrar una lista de servicios ofrecidos previo ingreso de preferencias.
- El sistema debe mostrar el itinerario previa selección del turista.
- Los proveedores deben crear perfil en los distintos servicios.
- Los proveedores deben acceder mediante usuario y contraseña.
- Los proveedores pueden modificar perfil y servicios.

Un proyecto híbrido de este tipo posee dos metodologías contenidas en una sola, con ello debe ser capaz de enfrentar distintos escenarios de acuerdo a los requerimientos, para todo aquello se utilizó una notación UML que soporta ambas metodologías, además de proporcionar los diagramas necesarios para la captura de requerimientos. Los diagramas correspondientes al análisis de proyecto serán descritos a continuación.

5.2.1 Casos de Uso de Servicios Web

5.2.1.1 Caso Uso Alojamiento

Descripción Breve

El objetivo principal de este caso de uso es describir las funcionalidades básicas que tiene el Sistema de proveedor de servicios. Lo conforman el módulo de alojamiento, entretenimiento, restauración y transporte, que son pieza fundamental para el sistema de planificación.

El caso de uso propuesto es una muestra genérica de los escenarios para: alojamiento, entretenimiento, restauración y transporte que se implementaron para cada uno de estos servicios.

A continuación se presentan los Casos de Uso más relevantes tomando como referencia el uso del Módulo de Alojamiento, como es un mantenedor principalmente de servicios se construyeron genéricos para los demás casos en que solo existe un cambio de atributos. Los demás casos de usos serán adjuntados en el Anexo A Casos de Usos.

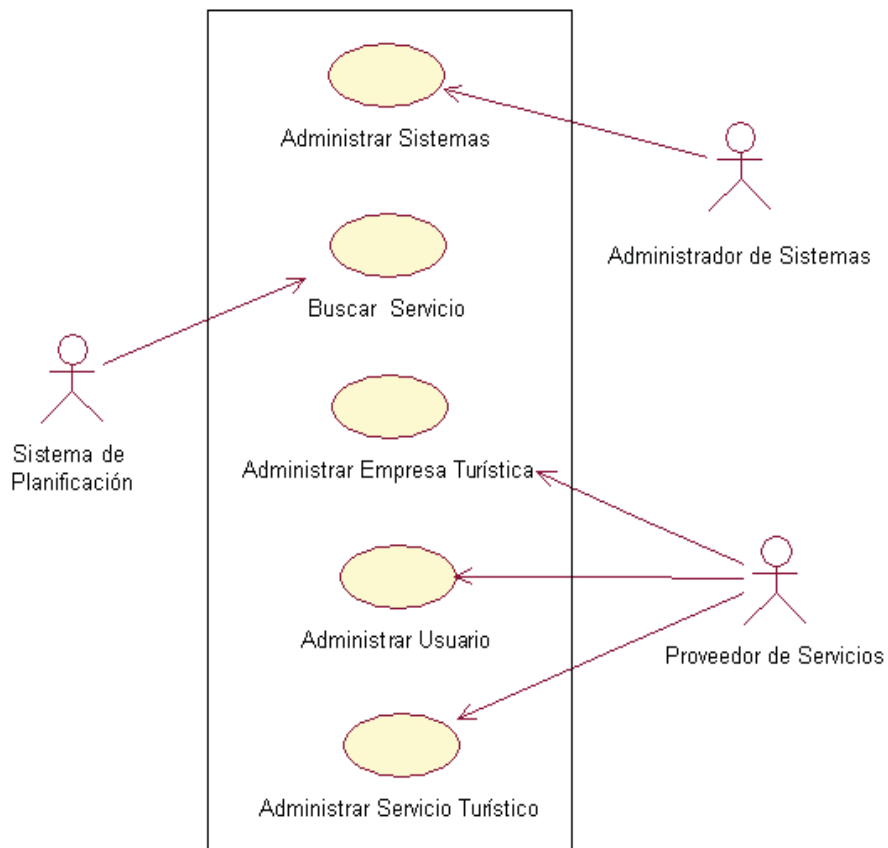


Ilustración 5.2 Diagrama de Casos de Uso alto nivel

Definiciones, acrónimos y abreviaturas

Sistema Planificación	Se define Sistema de Planificación al dispositivo móvil que accede a los distintos servicios disponibles para realizar una búsqueda de preferencias.
Proveedor de Servicios	Es el usuario responsable de la definición de productos y servicios disponibles en cada modulo de servicios.
Servicio Turístico	Se define como un conjunto de prestaciones que vienen a satisfacer las necesidades de un cliente.
Administrador Sistema	Es el usuario responsable y con todos los permisos posibles para gestionar el aplicativo de proveedores de servicios.

Detalle del Caso de Uso: Buscar Alojamiento

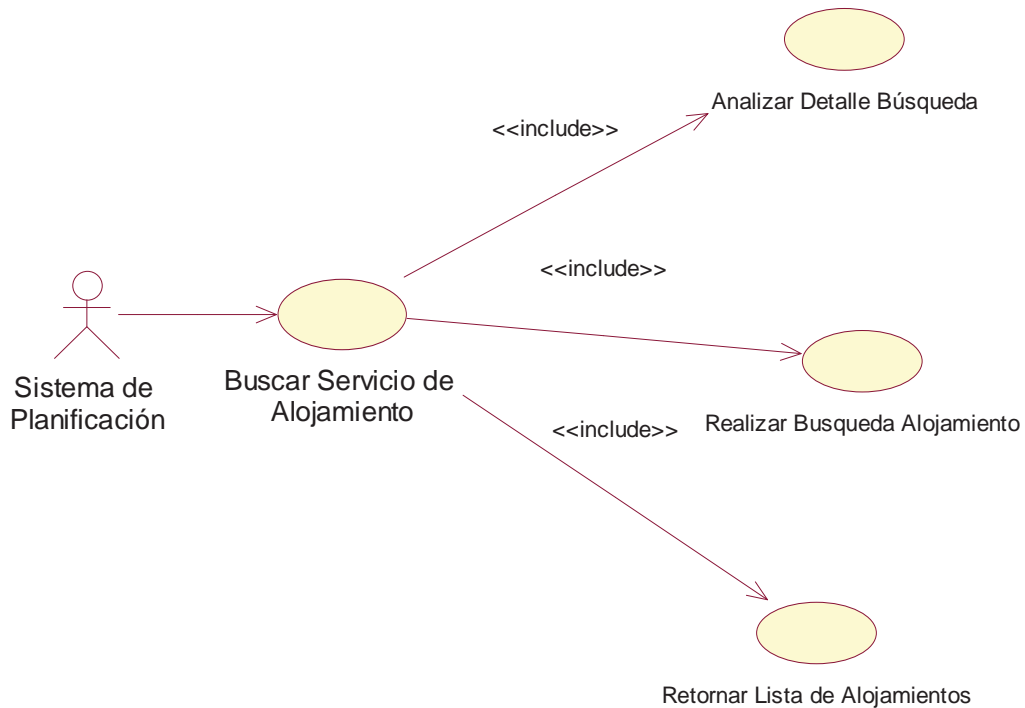


Ilustración 5.3 Diagrama de Casos de Uso Buscar Alojamiento

Nombre de Caso de Uso	CUS-MDAL01: Buscar Alojamiento
Descripción de Caso de Uso	Este caso de uso posibilita la búsqueda de servicios asociados al alojamiento.
Actores Directos	Sistema Planificación
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El sistema de planificación solicita la búsqueda de servicios de alojamiento 2. El sistema recibe las preferencias ingresadas por el usuario. 3. El sistema descompone el mensaje de las preferencias y genera los patrones de búsqueda. 4. El sistema busca los servicios de alojamiento asociados a los intervalos. 5. El sistema compone la respuesta con todos los servicios que cumplan con las preferencias. 6. Sistema retorna un arreglo con los servicios que coinciden con las preferencias.
Flujo Alternativo	<ol style="list-style-type: none"> 4.1 El sistema no encuentra servicios dada las preferencias. 5.1 El sistema retorna el valor null al sistema de planificación.
Escenario de Extensiones Alternativas	No aplica en este nivel.

Pre-requisitos	El usuario del sistema de planificación debe estar debidamente logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Detalle del Caso de Uso: Administrar Usuario

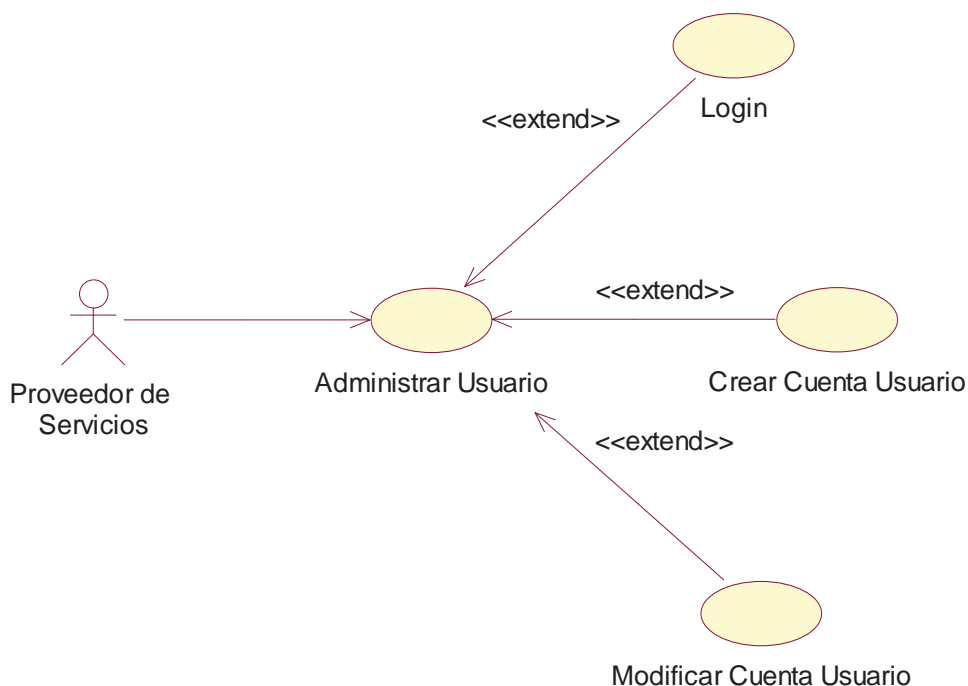


Ilustración 5.4 Diagrama de Casos de Uso Administrar Usuario

Nombre de Caso de Uso	CUS-MDAL02.1: Login
Descripción de Caso de Uso	Este caso de uso posibilita el ingreso del Usuario al Sistema de Proveedores
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa al Sistema de Proveedores. 2. El Proveedor de Servicios ingresa su Usuario y contraseña. 3. El Proveedor de Servicios presiona un botón para ingresar al Sistema. 4. El Sistema de Proveedores valida que el Usuario y Contraseña estén correctos.

	5. El Sistema da acceso al menú principal del Sistema de Proveedores
Flujo Alternativo	3.1 El Sistema encuentra que los campos de usuario o contraseña esta vacios. 3.2 El Sistema muestra una alerta para que se completen el o los campos faltantes. 4.1 El Sistema encuentra que el usuario y/o contraseña no coinciden. 4.2 El Sistema muestra una alerta con el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	No aplica en este nivel.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL02.2: Administrar Usuario: Crear Usuario
Descripción de Caso de Uso	Este caso de uso posibilita la creación de nuevos Usuarios al Sistema de Proveedores.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de creación de Usuarios 2. El Usuario completa los siguientes campos. <ul style="list-style-type: none"> ▪ Nombre ▪ Apellido ▪ Seudónimo ▪ Fecha Nacimiento ▪ Email ▪ Contraseña ▪ Duplicado Contraseña 3. El Sistema verifica que el ingreso de los datos sean correctos. 4. El Sistema valida que el seudónimo y el email no existan. 5. El Sistema guarda los datos ingresados generando un nuevo usuario en el. 6. El Sistema muestra un mensaje de ingreso exitoso.
Flujo Alternativo	3.1 El Sistema encuentra que uno o más de los campos ha sido mal ingresado o en blanco. 3.2 El Sistema muestra un alerta con los errores encontrados. 4.1 El Sistema valida el seudónimo y/o Email ya existen. 4.2 El Sistema muestra un alerta indicando el error que ocurrió.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	No aplica en este nivel.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL02.3: Modificar Usuario
Descripción de Caso de Uso	Este caso de uso posibilita la edición de datos del usuario.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de modificación del Usuario. 2. El Proveedor puede editar los siguientes campos. <ul style="list-style-type: none"> ▪ Nombre ▪ Apellido ▪ Fecha nacimiento ▪ Contraseña 3. El Proveedor de Servicios presiona el botón “Modificar”. 4. El Sistema de Proveedores valida que los datos sean validos. 5. El Sistema muestra un mensaje de operación realizada exitosamente.
Flujo Alternativo	<ol style="list-style-type: none"> 4.1 El Sistema de Proveedores encuentra campos incorrectos y/o en blanco. 4.2 El Sistema de Proveedores muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Detalle del Caso de Uso: Administrar Servicio Turístico

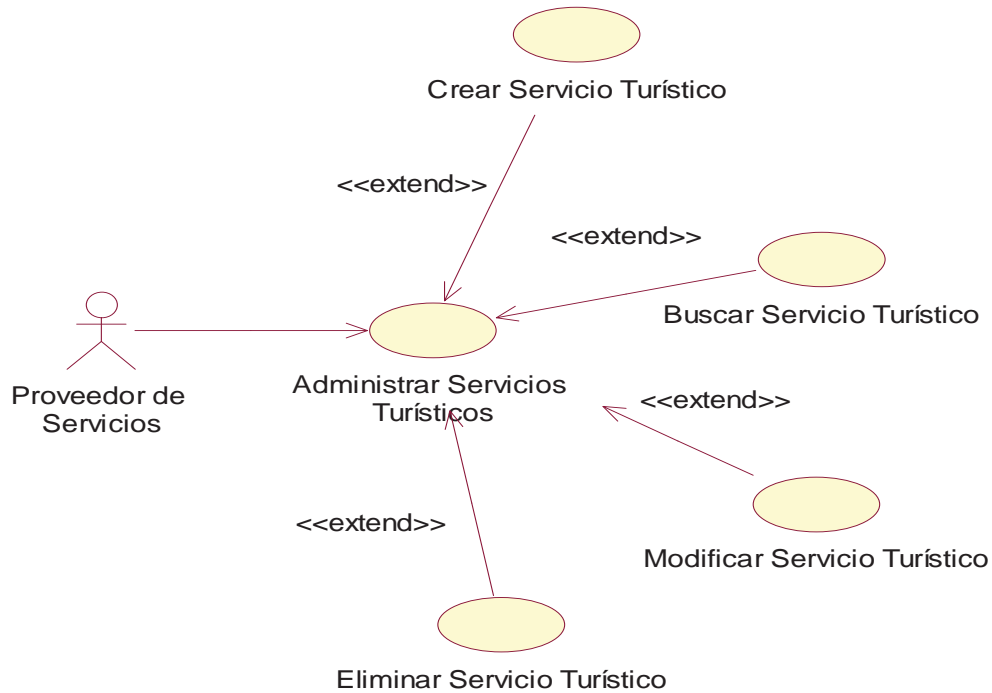


Ilustración 5.5 Diagrama de Casos de Uso Administrar Servicio Turístico

Nombre de Caso de Uso	CUS-MDAL03.1: Crear Servicio Turístico
Descripción de Caso de Uso	Este caso de uso posibilita la creación de Servicios Turísticos.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de creación de Servicios Turísticos. 2. El Proveedor debe completar los siguientes campos. <ul style="list-style-type: none"> ▪ Nombre Empresa ▪ Nombre Servicio ▪ Valor ▪ Descripción ▪ Vigencia 3. El Proveedor de Servicios presiona el botón “Crear”. 4. El Sistema de Proveedores valida que los datos sean validos. 5. El Sistema muestra un mensaje de operación realizada exitosamente.
Flujo Alternativo	<ol style="list-style-type: none"> 4.1 El Sistema de Proveedores encuentra campos incorrectos y/o en blanco. 4.2 El Sistema de Proveedores muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.

Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL03.2: Buscar Servicio Turístico
Descripción de Caso de Uso	Este caso de uso posibilita la búsqueda de Servicios Turísticos.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de búsqueda Servicios Turísticos. 2. El Proveedor puede buscar por. <ul style="list-style-type: none"> ▪ ID ▪ Nombre Servicio 3. El Proveedor de Servicios presiona el botón “Buscar”. 4. El Sistema de Proveedores valida que los datos sean validos. 5. El Sistema muestra una lista con los Servicios que coincidan con los criterios.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1 El Sistema de Proveedores encuentra campos incorrectos. 4.2 El Sistema de Proveedores muestra una alerta indicando el error. 2.1 El Sistema no encuentra resultado a los criterios de búsqueda.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL03.3: Modificar Servicio Turístico
Descripción de Caso de Uso	Este caso de uso posibilita la edición de Servicios Turísticos.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de edición para Servicios Turísticos. 2. El Proveedor puede modificar. <ul style="list-style-type: none"> ▪ Nombre Servicio ▪ Valor ▪ Vigencia ▪ Descripción 3. El Proveedor de Servicios presiona el botón “Modificar”. 4. El Sistema de Proveedores valida que los datos sean validos. 5. El Sistema muestra un mensaje indicando que la modificación se realizo de manera correcta.
Flujo Alternativo	<ol style="list-style-type: none"> 4.1 El Sistema de Proveedores encuentra campos incorrectos. 4.2 El Sistema de Proveedores muestra una alerta indicando el error.

Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL03.4: Eliminar Servicio Turístico
Descripción de Caso de Uso	Este caso de uso posibilita la eliminación lógica de Servicios Turísticos.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de eliminación para Servicios Turísticos. 2. El Proveedor Selecciona el servicio turístico. 3. El Proveedor de Servicios presiona el botón “Eliminar”. 4. El Sistema de Proveedores valida que no existan reservas. 5. El Sistema muestra un mensaje indicando que la eliminación se realizo de manera correcta.
Flujo Alternativo	<ol style="list-style-type: none"> 4.1 El Sistema de Proveedores encuentra reservas vigentes. 4.2 El Sistema de Proveedores muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

5.2.2 Modelo Descripción del Dominio

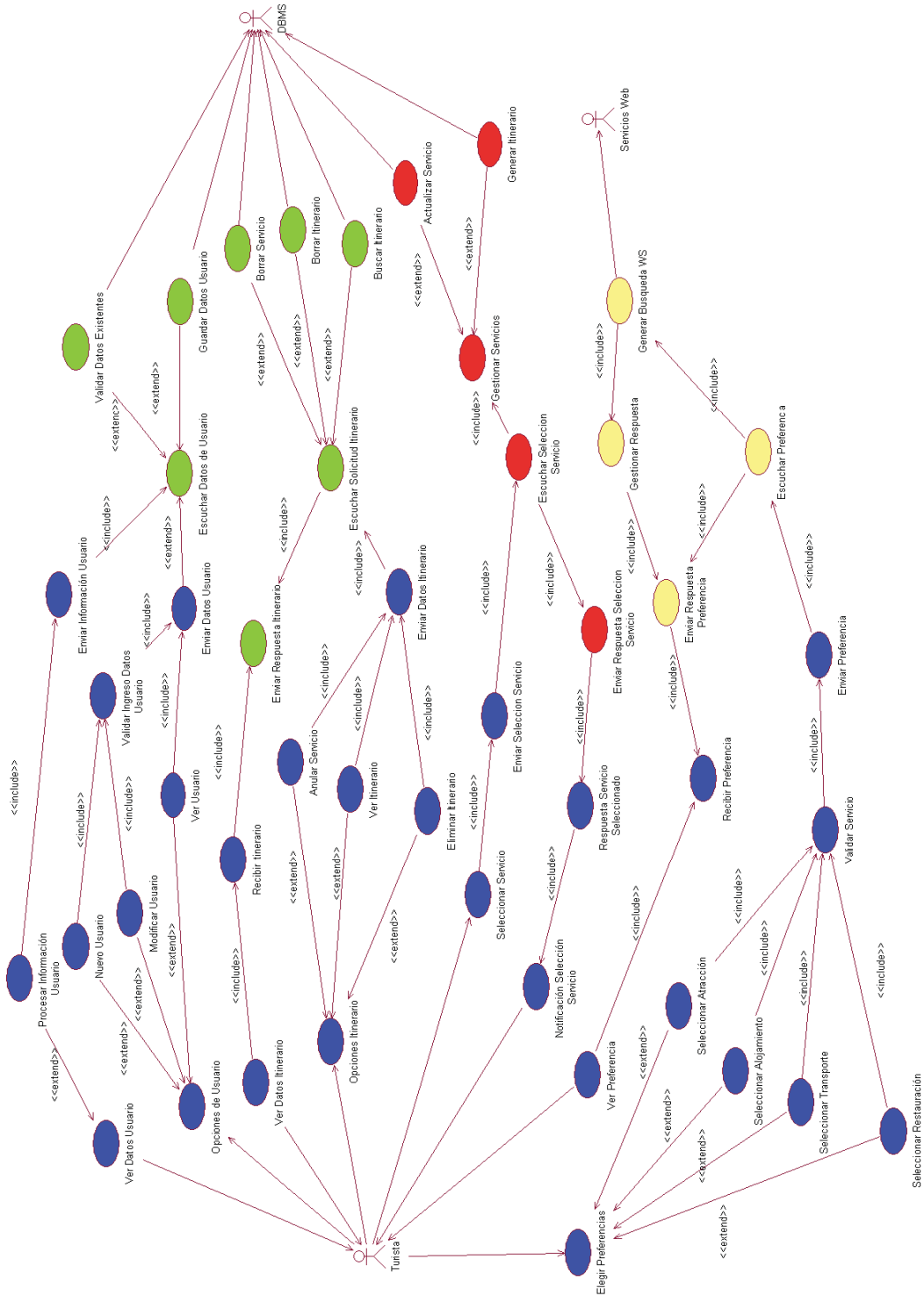


Ilustración 5.6 Modelo Descripción del Dominio

En la Ilustración 5.6 podemos distinguir tres actores que interactúan con sistema: el Turista quien es el que desea viajar; la DBMS donde se almacenan los datos relacionados con el turista y su itinerario; por otro lado están los Servicios Web que son las opciones que posee el turista dada sus necesidades para realizar un viaje.

Dentro de estos actores se reconocen cuatro tareas importantes para cumplir el objetivo de planificar un viaje, los cuales son la Interfaz que se encarga de tomar los datos válidos ingresados por el turista, el intermediario móvil que se encarga de manipular la información ya ingresada por el turista, el planificador que cumple con validar los servicios adquiridos y el intermediario Web Services que se encarga de consumir los servicios ofrecidos.

5.3 Especificación del Diseño

Continuando con los flujos de trabajo que ofrece la metodología se plasma el diseño del sistema. Este lo componen cuatro Servicios Web que conforman la oferta de servicios que dispone el turista y la aplicación móvil que administra la información del turista y sus preferencias.

El modelado que se presenta a continuación en los servicios web está compuesto de diagramas de clases y de secuencia, además de los de actividad que pretenden profundizar aun más la solución para el sistema de planificación. En la aplicación móvil se exponen el diagrama de identificación de agentes, el modelo de descripción ontológica del dominio, el diagrama de estructura multiagente, el diagrama de estructura de agentes y el diagrama de despliegue, contemplados para la aplicación del turista la cual es la más relevante ya que implementa la lógica de comunicación entre la aplicación y los servicios web.

5.3.1 Diagrama de Clases

Continuando con la fase de elaboración se pasa a describir el siguiente diagrama de clases donde los servicios ofrecidos por el sistema están divididos en cuatro elementos que forman distintos Servicios Web, los cuales corresponden a la aplicación del proveedor, el servicio de alojamiento, atracción, restauración y transporte.

A continuación se presentan dos diagramas de clases que pretenden acercar a la solución del sistema de planificación propuesto, donde por un tema de espacio los otros dos diagramas estarán adjuntos en el Anexo D de proyecto.

Los diagramas que se plasman en este punto son el de clases de Alojamiento y el de Restauración.

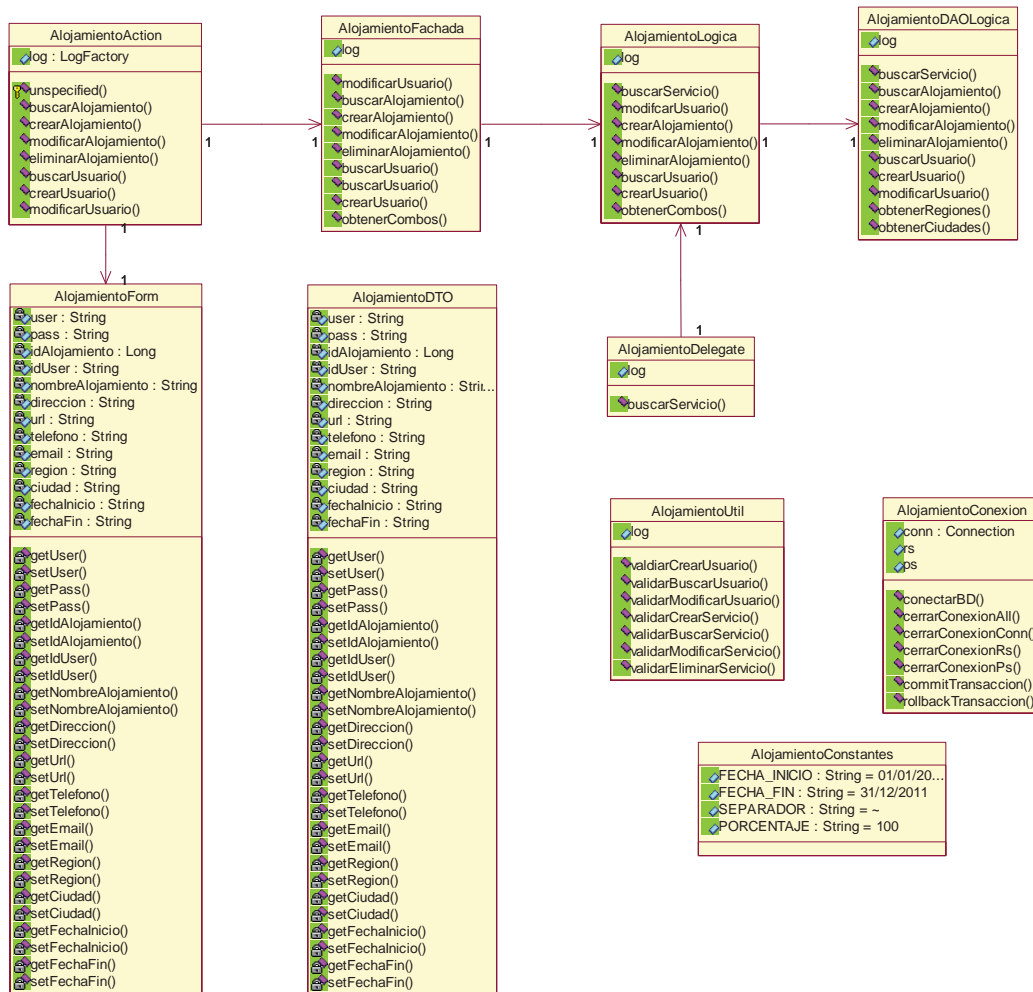


Ilustración 5.7 Diagrama de Clases Alojamiento

AlojamientoAction: Clase maneja la lógica de la presentación del sistema de alojamiento, se encarga de capturar los datos y procesarlos según las acciones que el sistema permita.

AlojamientoForm: Clase formulario, encargada de rescatar y setear las variables utilizadas en la aplicación.

AlojamientoFachada: Clase interfaz encargada de separar la presentación de la lógica de la aplicación.

AlojamientoDelegate: Clase interfaz entre el sistema de planificación y el servicio web implementado para buscar servicios.

AlojamientoLogica: Clase encargada del procesamiento de la lógica que algún proceso, encargada de manejar la conexión además de la transaccionalidad con la base de datos.

AlojamientoDAOLogica: Clase encargada del realizar la consulta en la base de datos.

AlojamientoUtil: Clase encargada de la validaciones de sistema en la presentación de la aplicación.

AlojamientoDTO: Clase que almacena y transporta la información del modelo de datos de alojamiento.

AlojamientoConexion: Clase encargada de los procesos de apertura y cierre de la conexión.

AlojamientoConstantes: Clase almacena las variables declaradas como constantes en el sistema.

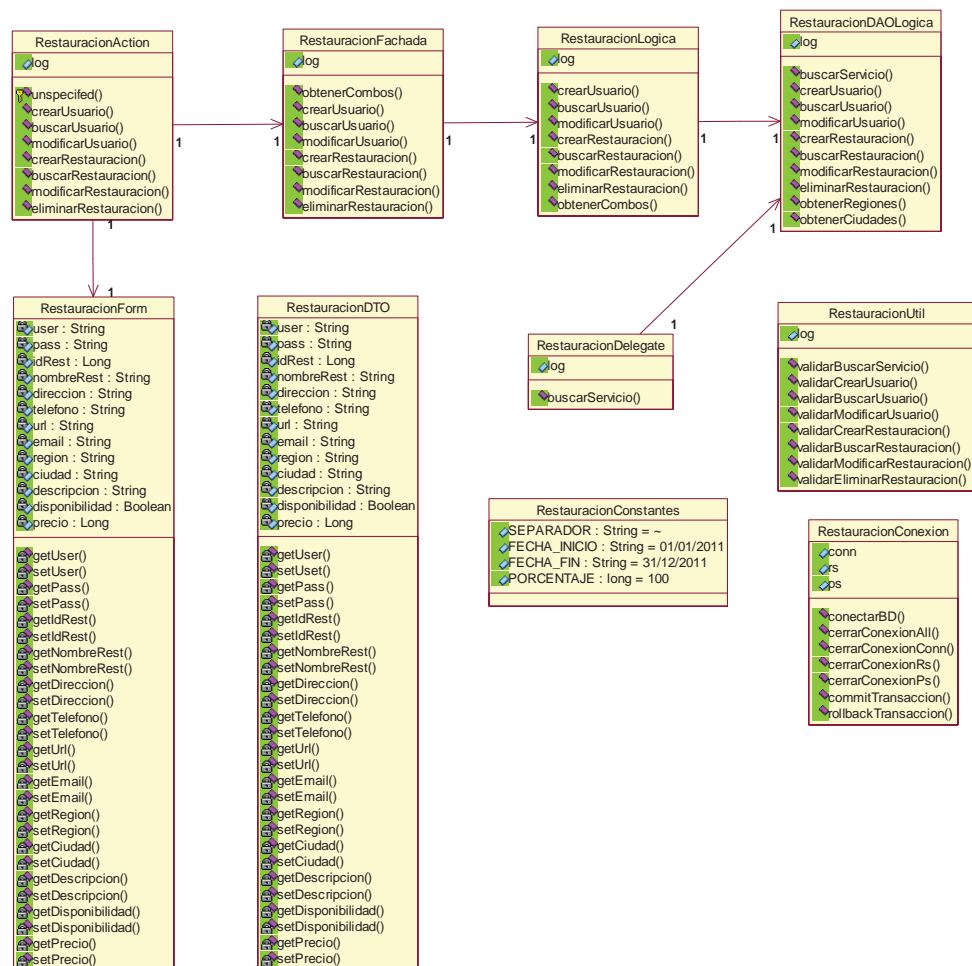


Ilustración 5.8 Diagrama de Clases Restauración

RestauracionAction: Clase que maneja la lógica de la presentación del sistema de restauración, se encarga de capturar los datos y procesarlos según las acciones que el sistema permita.

RestauracionForm: Clase formulario, encargada de rescatar y setear las variables utilizadas en la aplicación.

RestauracionFachada: Clase interfaz encargada de separar la presentación de la lógica de la aplicación.

RestauracionDelegate: Clase interfaz entre el sistema de restauración y el servicio web implementado para buscar servicios.

RestauracionLogica: Clase encargada del procesamiento de la lógica de las acciones, maneja la conexión además de la transaccionabilidad con la base de datos.

RestauracionDAOLogica: Clase encargada del realizar la consulta en la base de datos.

RestauracionUtil: Clase encargada de las validaciones del sistema en la presentación de la aplicación.

RestauracionDTO: Clase que almacena y transporta la información del modelo de datos de restauración.

RestauracionConexion: Clase encargada de los procesos de apertura y cierre de la conexión.

RestauracionConstantes: Clase almacena las variables declaradas como constantes en el sistema.

5.3.2 Diagrama de Secuencia

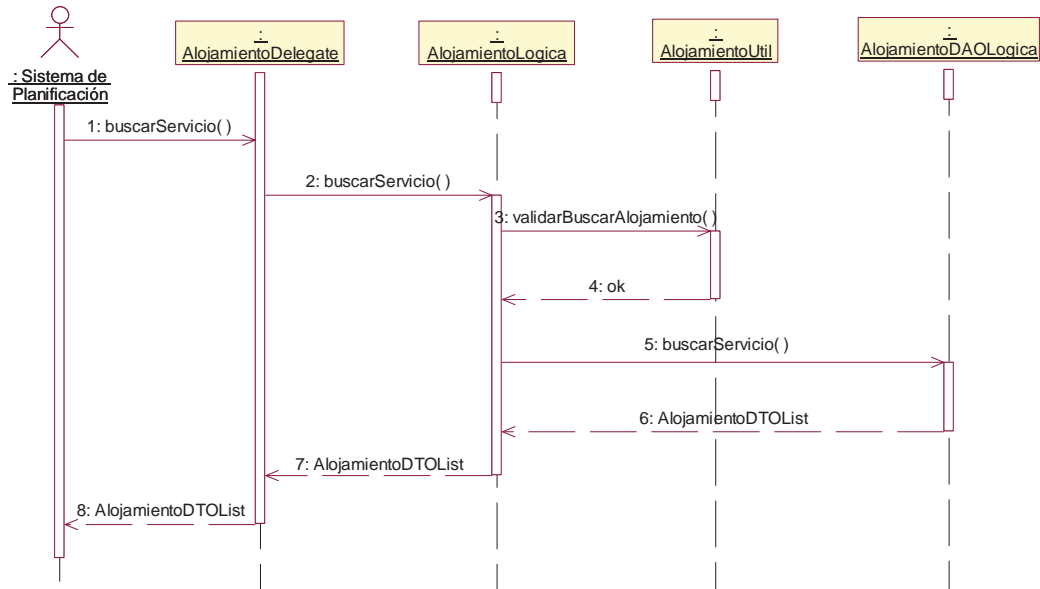


Ilustración 5.9 Diagrama de Secuencia: Buscar Alojamiento

En la Ilustración 5.9 se muestra el escenario principal para la búsqueda de alojamiento como parte de la generación de un itinerario de viajes para el turista. En esta etapa se muestra principalmente la interacción de las clases y los objetos involucrados como el “*AlojamientoDelegate*” que funciona como interfaz para separar la lógica del proceso de llamada desde el Servicio Web, el “*AlojamientoLogica*” maneja el flujo principal de la búsqueda de alojamiento realizando llamadas a funciones de utilidad y acceso a la base de datos. Para aquello la Ilustración 5.10 muestra en detalle la secuencia de pasos a seguir por parte del Servicio Web de alojamiento, iniciado por la llamada de un procedimiento remoto a través del planificador de servicios turísticos, que envía una lista de preferencias para que el Servicio Web procese en detalle, y genere una lista que se acomode a las indicaciones dadas y las retorne para que el usuario pueda elegir.

5.3.3 Diagrama de Actividad

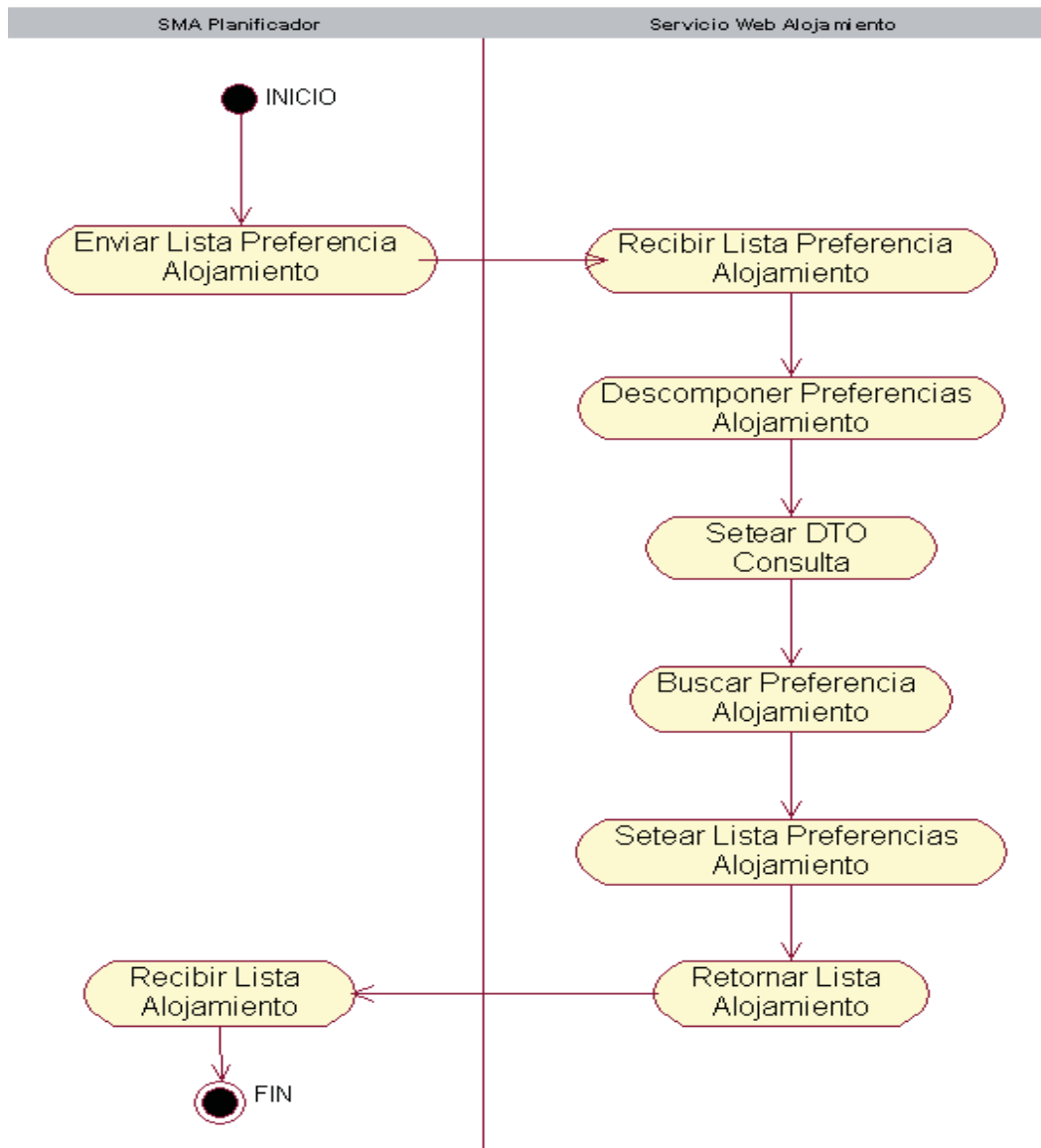


Ilustración 5.10 Diagrama de Actividad: Buscar Preferencias de Alojamiento

En la Ilustración 5.10 se muestra las principales actividades que cumple el servicio web de alojamiento al momento de buscar una lista de preferencias que se acomoden al usuario, donde se describe el flujo de ejecución de forma detallada entre el intermediarioWS y el servicio web alojamiento. Para partir se recibe una cadena de *String* que posee los parámetros de búsqueda para el alojamiento, estos se separan y se setea en el DTO de Alojamiento, se realiza la búsqueda luego se recibe el *resultset* con el resultado de la búsqueda y se vacía en una lista para su posterior retorno.

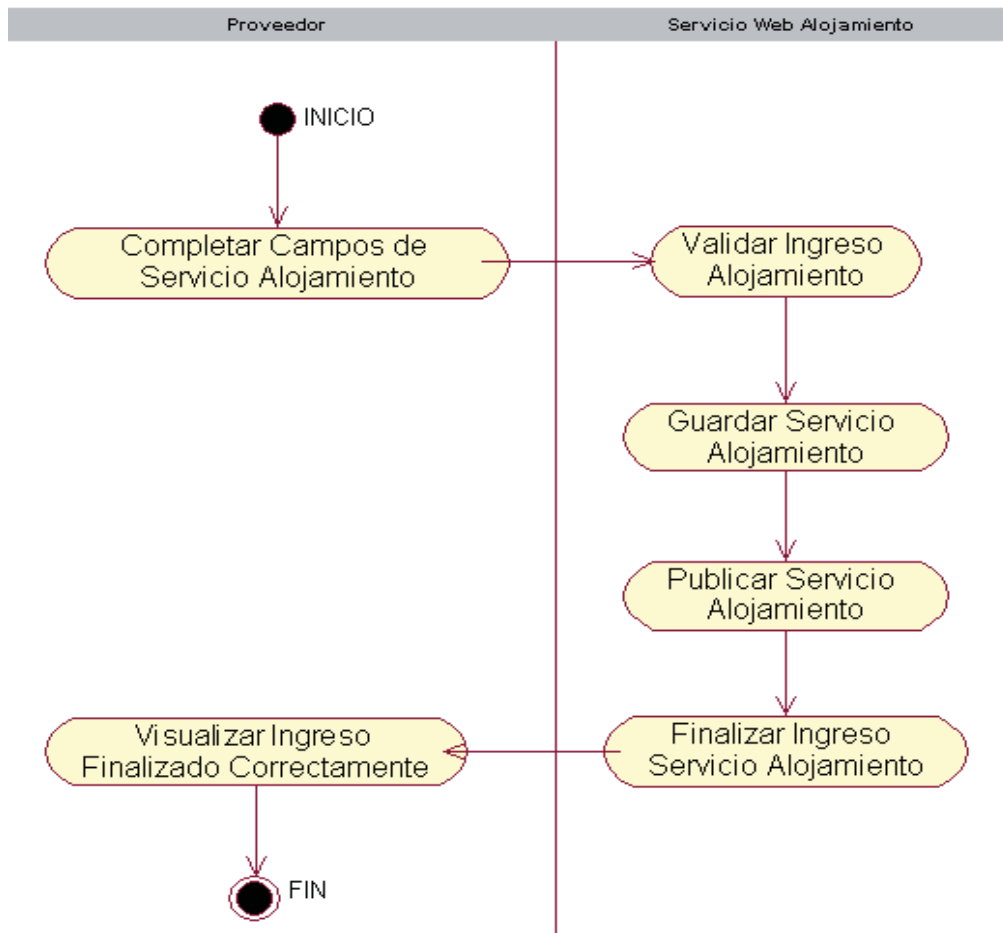


Ilustración 5.11 Diagrama de Actividad: Registrar Servicio de Alojamiento

En la Ilustración 5.11 participa el proveedor del servicio en este caso registrando un alojamiento para su posterior publicación donde se muestran todos los pasos que describen el flujo de ejecución de éxito en la tarea. En este diagrama participa el proveedor del servicio como iniciador, posteriormente se valida el ingreso de tipo de datos sea el correcto de acuerdo al campo requerido con este paso validado se procede a guardar y publicar el servicio para que esté disponible para los turistas, al finalizar el ingreso de un nuevo servicio de alojamiento se visualiza un mensaje de ingreso exitoso.

Para finalizar, en la Ilustración 5.10 e Ilustración 5.11 se describen escenarios de actividades más relevantes para el sistema, se destaca que por un tema de orden se eligió el mismo servicio web para ejemplificar. Se realiza hincapié que son casos genéricos para los servicios debido a que el diagrama de clases propuesto es el mismo en todos los servicios donde solo cambian los atributos y sus propiedades.

5.3.4 Modelo de Identificación de Agentes

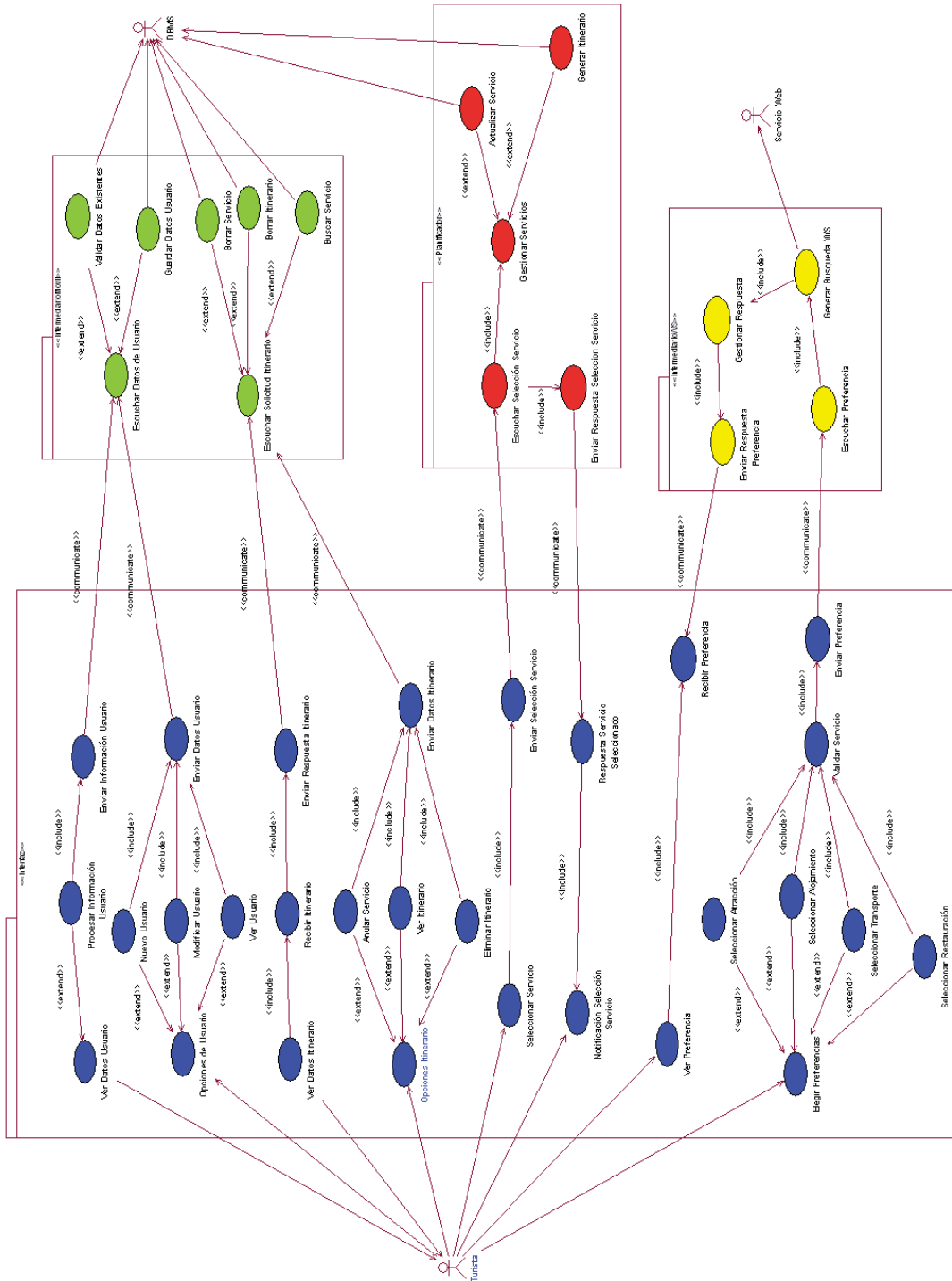


Ilustración 5.12 Modelo de identificación de Agentes

La Ilustración 5.12 muestra el diagrama de identificación de agentes que toma como base el diagrama de descripción del dominio donde se identifican cuatro agentes, cada uno se encuentra agrupado en paquetes, además aparece un nuevo tipo de relación *communicate*, que representa una correspondencia de comunicación entre agentes que reemplaza las relaciones que existían en el diagrama de descripción del dominio. De ésta vista del sistema, se puede obtener un indicio de cuáles son los agentes más importantes.

Bajo la explicación dada anteriormente podemos identificar tres agentes:

- **Interfaz:** Agente encargado de recibir los datos que el turista ingresa, visualiza la información procesada por los agentes IntermediarioMovil, Planificador y IntermediarioWS.
- **IntermediarioMovil:** Agente a cargo de la administrar los contenidos del turista, su tarea es mantener disponible la información del turista para cuando sea solicitada.
- **Planificador:** Maneja la lógica de planificación y control del itinerario para el turista. Su tarea principal es validar la selección de servicios sea coherente con el desplazamiento y el tiempo del turista, además se encarga de generar y editar el itinerario.
- **IntermediarioWS:** Corresponde al agente que comunica el sistema multiagente con los servicios web. Se encarga de buscar la oferta de toda la gama de servicios disponibles sin excepción que se ajuste a las preferencias que ingrese el Turista. Otra función es enviar la información a la Interfaz para que ella la presente al Turista.

5.3.5 Modelo de especificación de Tareas

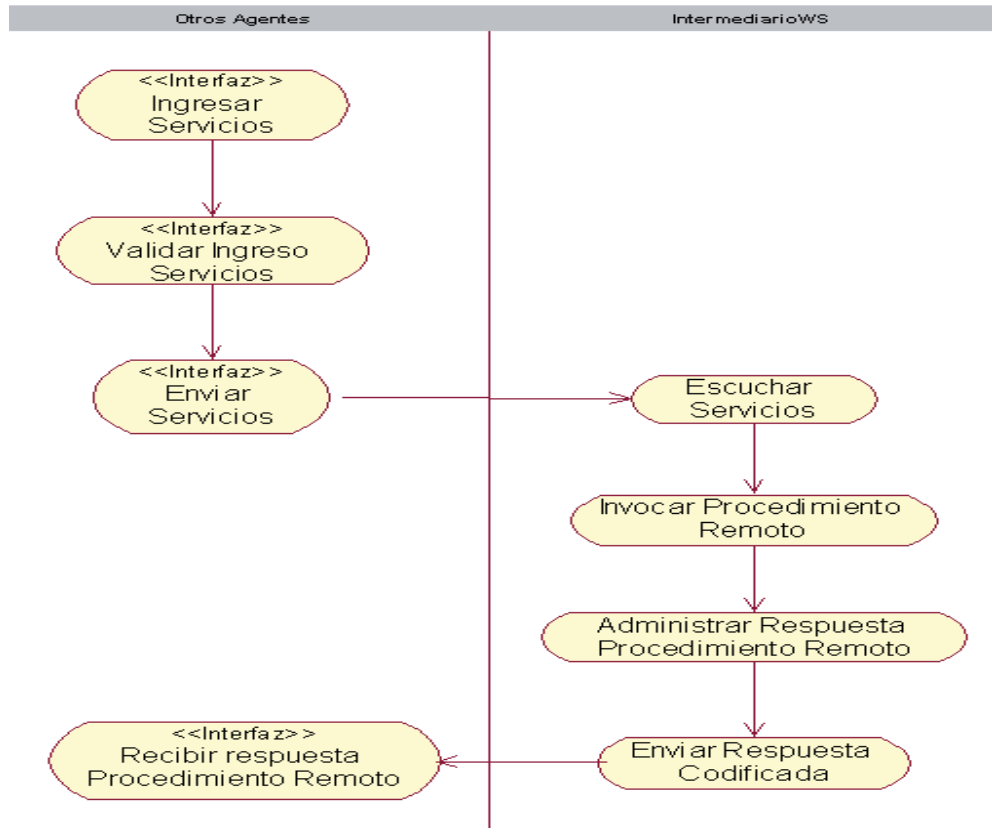


Ilustración 5.13 Diagrama de Especificación de Tareas: Comunicar Web Services

La Ilustración 5.13 muestra las actividades que realiza el agente: IntermediarioWS para enviar y recibir una solicitud de preferencias dado un cliente que busca servicios disponibles. Esto se realizó mediante la invocación de un procedimiento remoto a través de una wsdl estática que posee el método remoto a consumir, posteriormente la respuesta es codificada en un *String* para poder ser interpretada por la interfaz del dispositivo móvil.

A continuación se puede observar en la Ilustración 5.14 parte del código fuente del Behaviour maneja la invocación a los servicios web. Lo primero que tiene que hacer es rescatar el contenido de las preferencias del Usuario (Línea 6, 11, 16, 21), de existir la selección carga el “*Sub Behaviour*” de servicio al “*ParallelBehaviour*”, además este posee la condición de terminar cuando todos los “*Sub Behaviour*” estén completos “*WHEN ALL*”. Luego instancia un “*SequentialBehavior*” (Línea 27) con el objetivo de ordenar primero realizar la búsqueda (Línea 29) y segundo mostrar la Pantalla con el resultado de esta (Línea 31). Para finalizar carga el Behaviour (Línea 33) y este se ejecuta.

```

1 public void buscarPref(String regI, String regF,String aloja,
2     String atracc, String resta, String transp,
3     ArrayList servAtrac){
4     ParallelBehaviour s = new ParallelBehaviour(this, ParallelBehaviour.WHEN_ALL);
5     // Valida si desea buscar Alojamiento
6     if(aloja != null){

```



```

7      //Carga Behaviour de Alojamiento
8      s.addSubBehaviour(new prefeAlojaMovil(this,regI,regF,aloja));
9  }
10     // Valida si desea buscar Atracciones
11     if(atracc != null){
12         //Carga Behavior de Atraccion
13         s.addSubBehaviour(new prefeAtracMovil(this,regI,regF,atracc,servAtrac));
14     }
15     // Valida si desea buscar Restauración
16     if(resta != null){
17         //Carga Behavior Restauración
18         s.addSubBehaviour(new prefeRestaMovil(this,regI,regF,resta));
19     }
20     // Valida si desea buscar Transporte
21     if(transp!=null){
22         //Carga Behavior Transporte
23         s.addSubBehaviour(new prefeTransMovil(this,regI,regF,transp));
24     }
25
26     // Genera Secuencial Behavior para manejar la búsqueda y la nueva pantalla
27     SequentialBehaviour se = new SequentialBehaviour(this);
28
29     se.addSubBehaviour(s);
30
31     se.addSubBehaviour(new pantalla());
32
33     addBehaviour(se);
34 }
35 }

```

Ilustración 5.14 Código Fuente: Invocación Servicio Web Alojamiento

La invocación o consumo de los servicios web disponibles de proveedores de servicios turísticos se realiza a través del agente “*IntermediarioWS*”, mediante una wsdl estática como muestra la Ilustración 5.15. Para optimizar el consumo e invocación de los servicios web se implemento una invocación concurrente a través de un “*ParallelBehaviour*” con la condición de que concluya cuando todos sus behaviour han terminado.

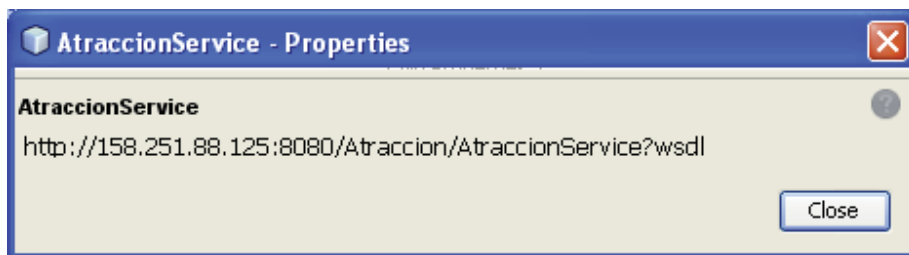


Ilustración 5.15 WSDL: Servicio Web Alojamiento

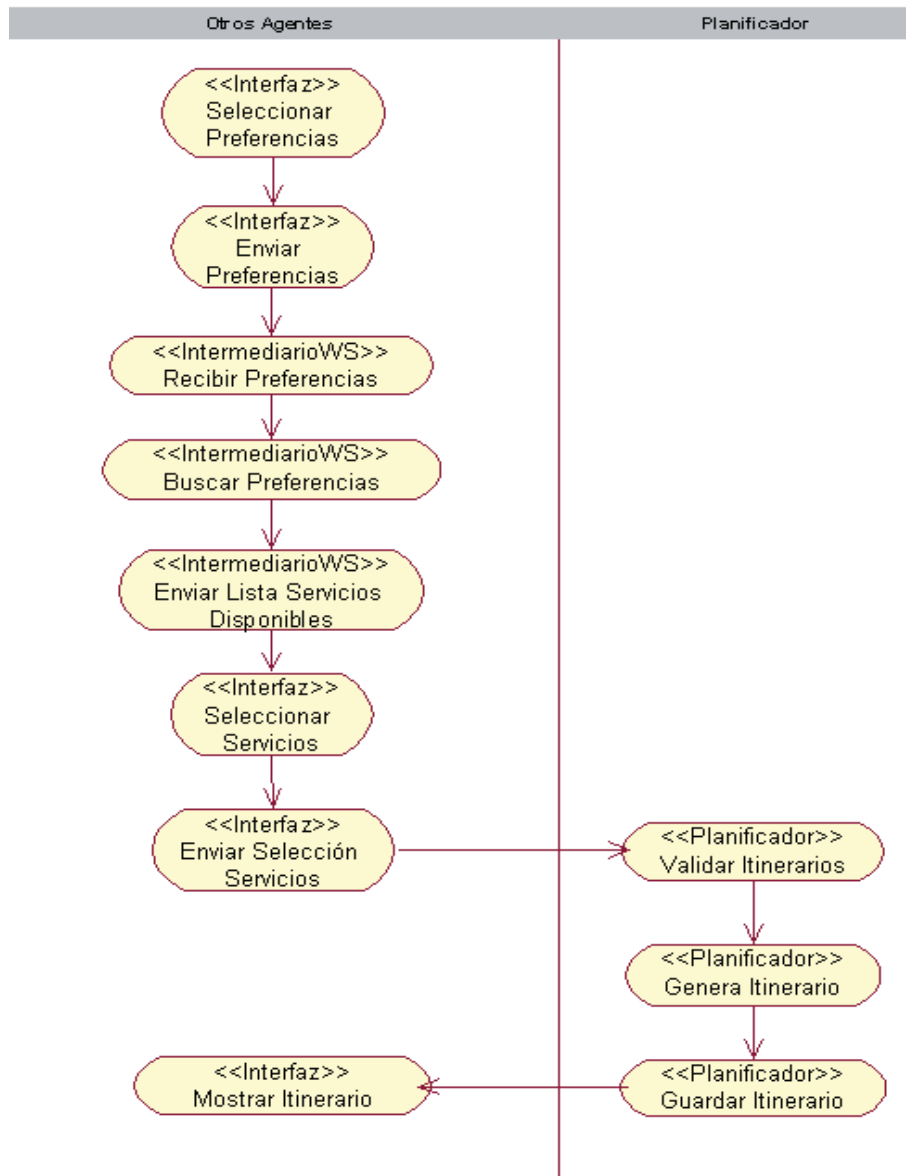


Ilustración 5.16 Diagrama de Especificación de Tareas: Generar Itinerario

Una de las tareas más relevantes se describe en el siguiente diagrama Ilustración 5.16 donde se muestra la secuencia de tareas que debe realizar el programa para generar un itinerario. Cabe considerar que esta tarea está relacionada desde el punto de vista del planificador como principal actor ya que es el que está a cargo de validar y generar el itinerario donde participan los demás agentes aunque los demás agentes también participan.

Dentro de la Ilustración 5.16 en el agente Planificador se valida el ingreso correcto de los parámetros requeridos para generar un itinerario y que con las distancias en función del tiempo sea posible cumplir con el itinerario. Posterior a esto se guarda en la base de datos y se muestra al turista que el Itinerario esta completado.

5.3.6 Modelo de descripción de Ontología del dominio

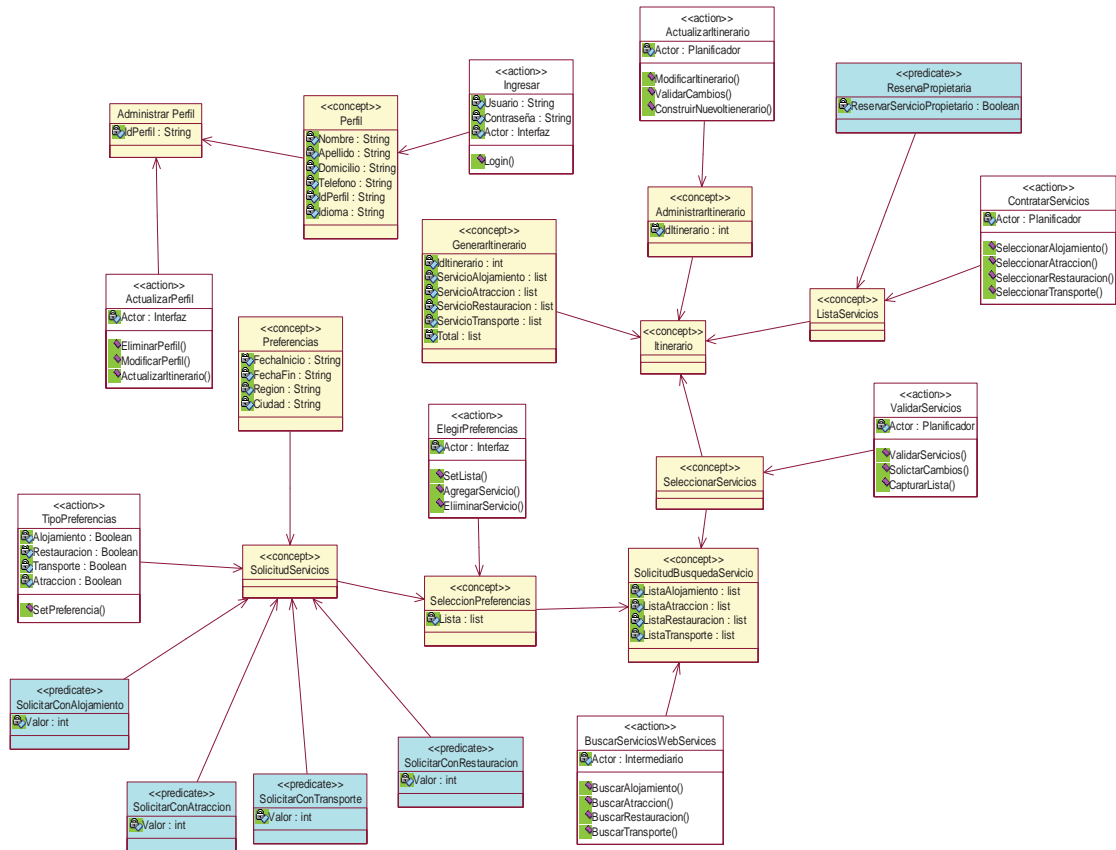


Ilustración 5.17 Modelo descripción de ontología del dominio

El presente diagrama descripción de ontología del dominio se presenta en la Ilustración 5.17, y es una descomposición de conceptos que componen el dominio del problema, donde se encuentran los conceptos, acciones y predicado que define PASSI.

5.3.7 Modelo de descripción estructura Sistema Multiagente

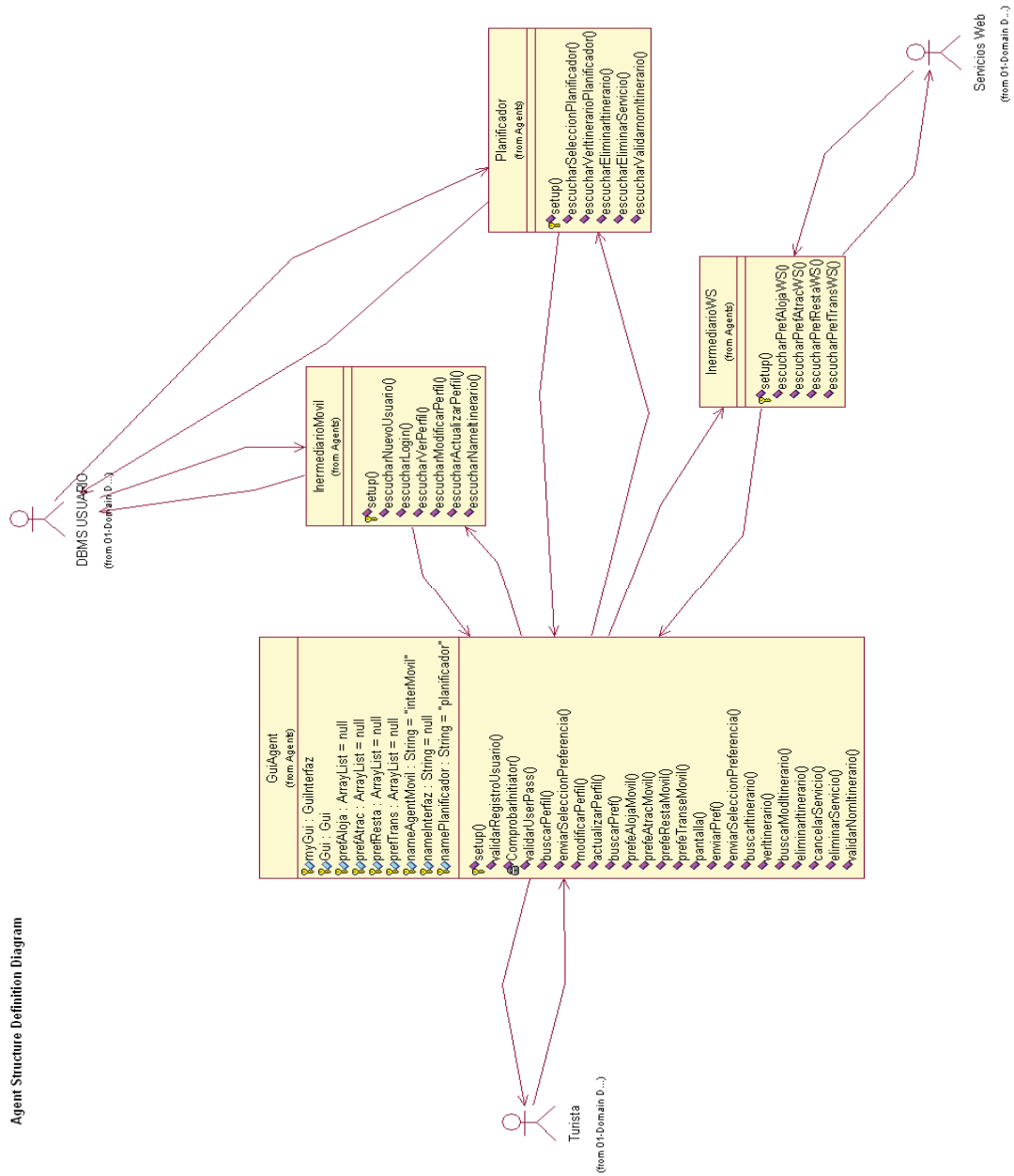


Ilustración 5.18 Modelo de Definición de Estructura Sistema Multiagente

Otro artefacto de PASSI es el diagrama de estructura Multiagente, donde se busca definir la estructura interna de las clases que componen el SMA. En la Ilustración 5.18 se puede apreciar los diversos actores y sus relaciones con todos los agentes del sistema, que indican el flujo de comunicación existente entre agentes. En este diagrama se pone especial énfasis en mostrar los atributos y métodos de cada agente en el sistema.

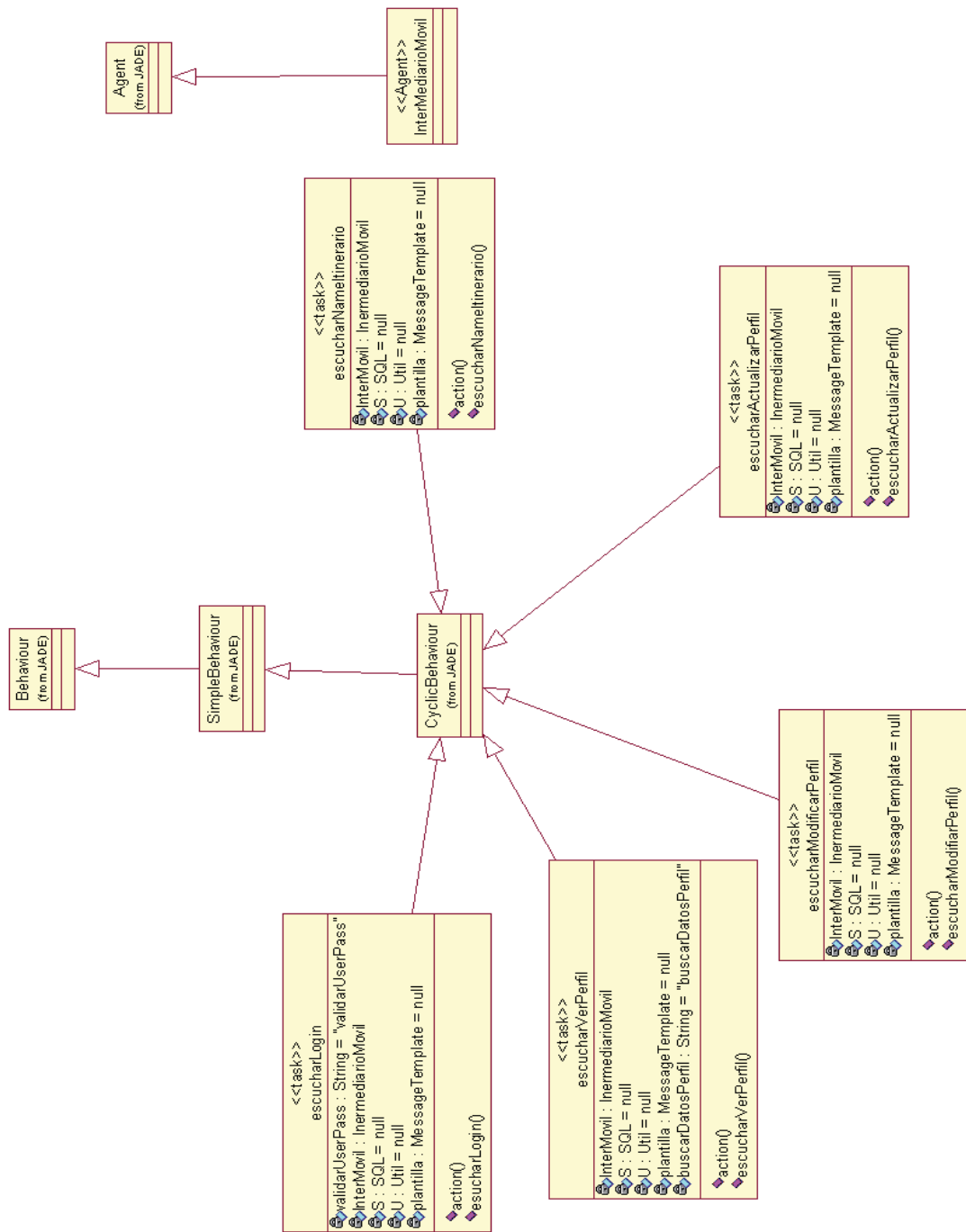


Ilustración 5.20 Diagrama Definición Estructura de Agente: IntermediarioMovil

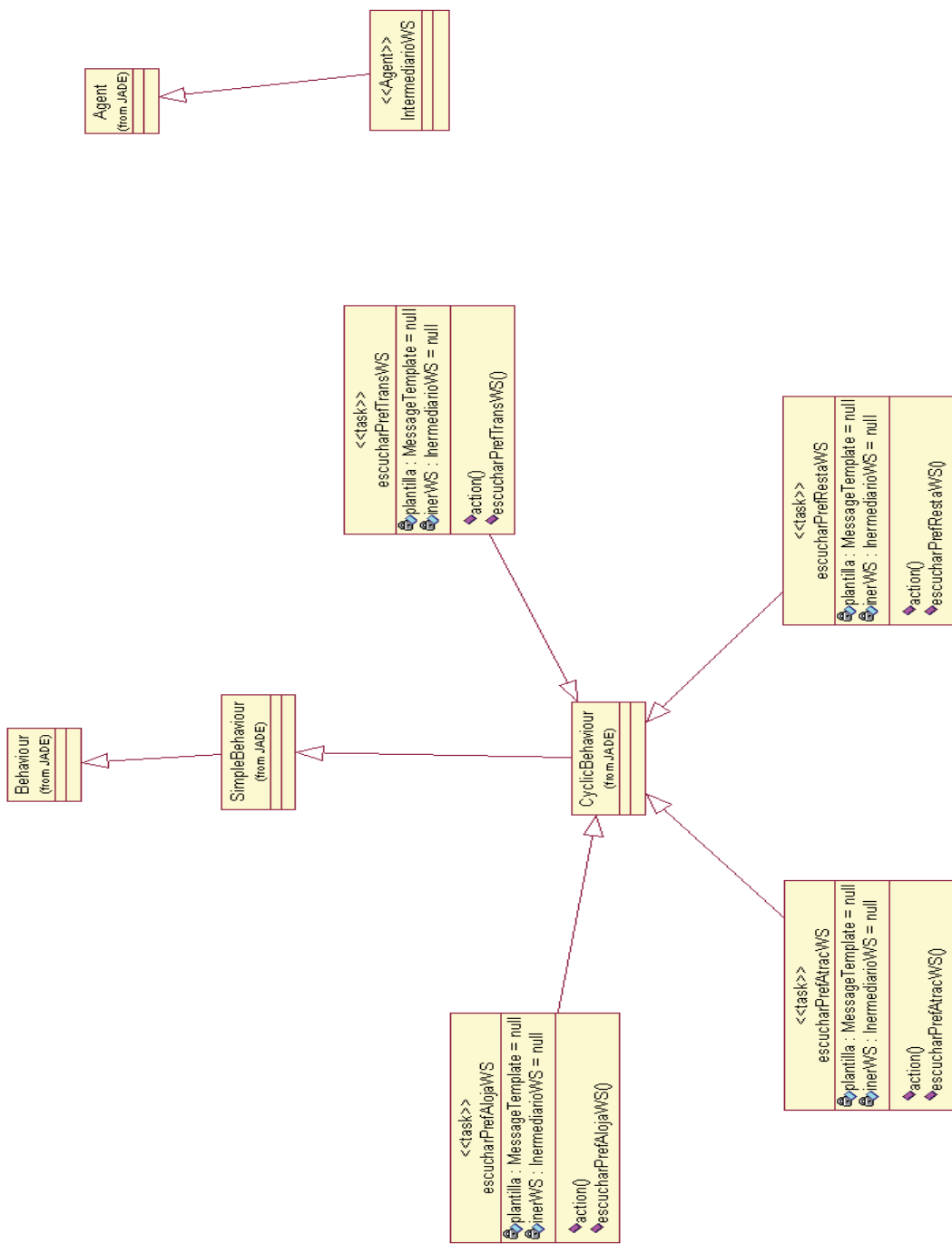


Ilustración 5.21 Diagrama Definición Estructura de Agente: IntermediarioWS

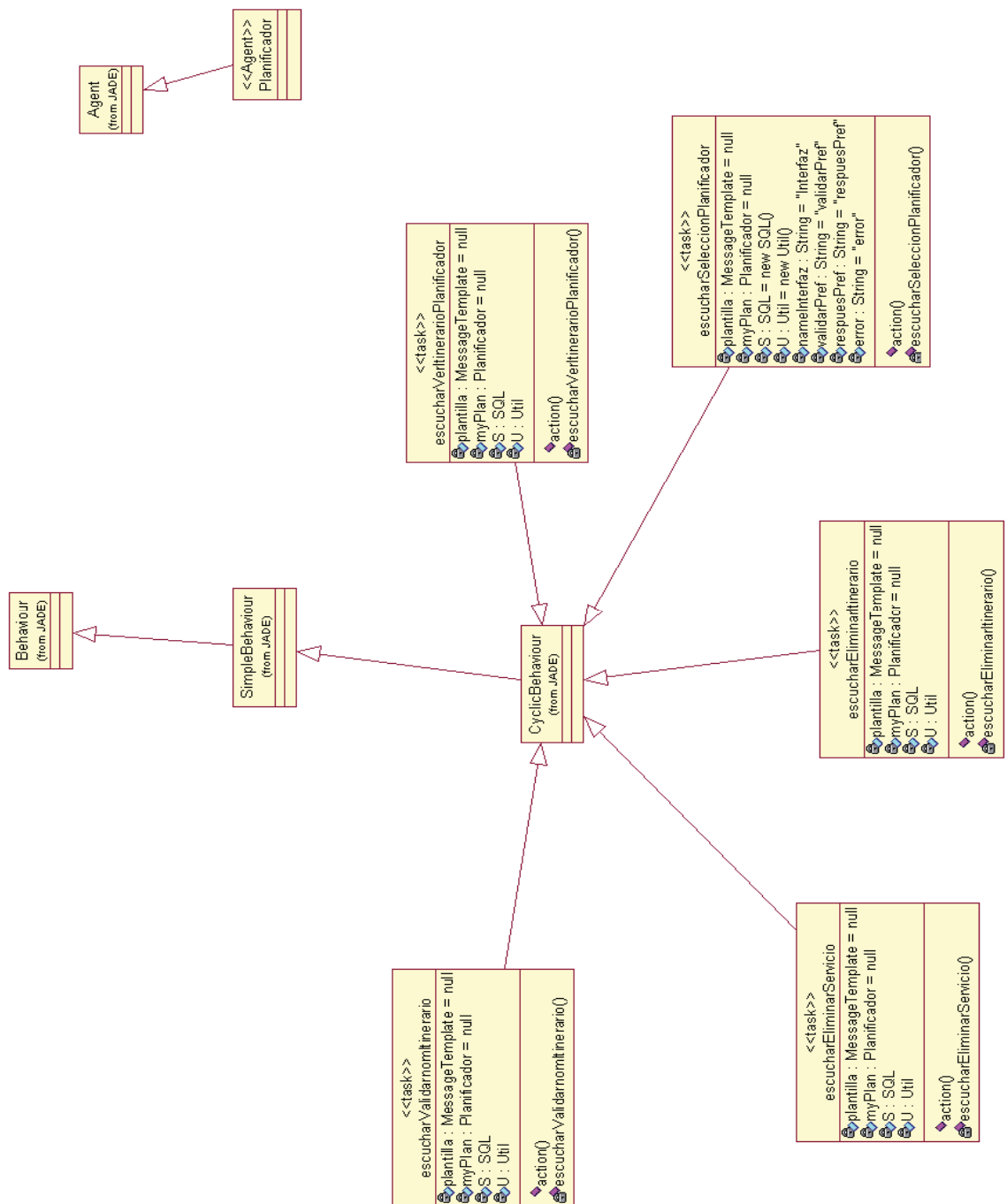


Ilustración 5.22 Diagrama Definición Estructura de Agente: Planificador

Para finalizar los diagramas de definición de estructura de agente en las Ilustraciones 5.19, 5.20, 5.21 y 5.22 describen en forma detallada el comportamiento individual de los agentes, donde sus tareas han sido definidas como comportamiento de la plataforma JADE-

LEAP. Se definió la estructura de todos los agentes identificados, Interfaz, Planificador, IntermediarioMóvil e IntermediarioWS.

La Ilustración 5.19 muestra los comportamientos que poseen el agente Interfaz, compuesto en su mayoría por “*OneShotBehaviour*”, un “*SequentialBehaviour*” y una composición de un “*ParallelBehaviour*”. Este agente es su mayoría es reactivo y reacciona a la acción del turista sobre el dispositivo móvil.

En la Ilustración 5.20 los comportamientos del agente intermediarioMóvil son cuatro “*cyclicBehavior*”, que corresponde a la forma de centinela que posee el sistema que se encuentra a modo escucha de cualquier solicitud por parte del usuario para ver sus datos.

La Ilustración 5.21 del agente intermediarioWS maneja cuatro comportamientos *cyclicBehavior* que se encargan de buscar dada las preferencias del usuario los servicios que se adecuen a ellas.

Para finalizar la Ilustración 5.22 los comportamientos del agente Planificador están compuestos por cinco *cyclicBehavior* (escuchan los mensajes que están dirigidos a él), donde estos se encargan de ver todas las tareas relacionadas con el itinerario, validarlo, cancelar servicios, eliminar itinerario, ver itinerario, del se puede manejar todo lo relacionado al itinerario en sus tareas pre y post generación.

5.3.9 Modelo Despliegue

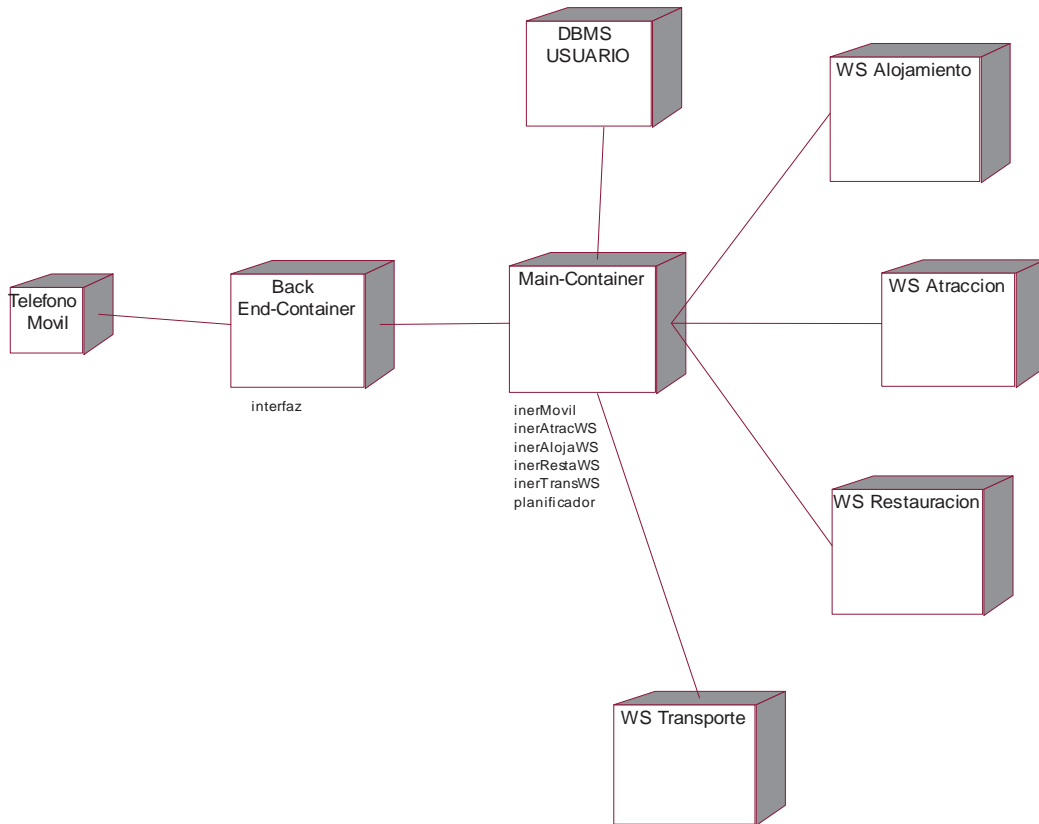


Ilustración 5.23 Diagrama Despliegue: Sistema Planificación Turística Híbrido

Una aplicación de planificación disponible vía web que posee una sociedad de agentes, para este caso un host mediador que se encarga de la interfaz y la información a visualizar además otro host con el contenedor principal que se encarga de toda la planificación de itinerario y se comunica tanto con los servicios web, además del dispositivo móvil. La Ilustración 5.23 muestra el diagrama de despliegue del sistema, la comparación final mostrara que tan efectiva es la distribución de esta plataforma respecto a una orientada netamente agentes y otra orientada a servicios web.

5.4 Implementación del Código

Dentro de lo desarrollado en el proyecto existen funcionalidades interesantes que merecen ser detalladas. A continuación una muestra del código de las funciones más relevantes.

```
1 public class escucharPrefAlojaWS extends CyclicBehaviour{
2
3     private MessageTemplate    plantilla    = null;
4     private IntermediarioWS    inerWS      = null;
5
6     public escucharPrefAlojaWS(IntermediarioWS a){
```

```

7
8     ...
9
10 }
11 public void action() {
12     String contenido = "";
13
14     //Filtro Mensaje segun Plantilla
15     ACLMessage mensaje = inerWS.receive(plantilla);
16
17     if (mensaje != null) {
18         ...
19
20
21     try {
22         java.util.List<java.lang.Object> result = null;
23         wshotel.HotelService service = new wshotel.HotelService();
24         wshotel.Hotel port = service.getHotelPort();
25
26         //Obtiene Contenido del Mensaje
27         String mensajeAgente = mensaje.getContent();
28
29         //Declara e instancia llamada a procedimiento remoto
30         result = port.buscarAlojamiento(mensajeAgente);
31
32         //Transforma resultado a contenido deseado
33         for(int j=0;j<result.size();j++){
34             if(j+1!=result.size())
35                 contenido = contenido+(String)result.get(j)+"~";
36             else
37                 contenido = contenido+(String)result.get(j);
38         }
39     } catch (Exception ex) {
40         System.err.println("error aloja "+ex);
41     }
42
43     //Setea Contenido de la respuesta
44     respuesta.setContent(contenido);
45     //Envia respuesta
46     inerWS.send(respuesta);
47
48 }else{
49     System.out.println("EscucharPrefAlojaWS inerWS: esperando mensajes");
50     block();
51 }
52 }
53 }
54 }
55 }

```

Ilustración 5.24 Código Fuente: Behaviour Buscar Alojamiento

Los servicios de alojamiento disponibles se buscan según la preferencia del Usuario (ver Ilustración 5.24), que no es más que un CyclicBehaviour que recibe las solicitudes de búsqueda de servicios de alojamiento, este Behaviour dependiendo del contenido del mensaje recibido (línea 15) es aceptado generando una búsqueda (línea 30) según el contenido del mensaje (línea 27) y realiza una llamada a un procedimiento remoto de un servicio web, donde el resultado es procesado y almacenado en un String separado por el carácter '~' (línea 33), para finalmente ser enviado como respuesta (línea 47).

En la Ilustración 5.24 se muestra la implementación del Behaviour en la clase “*escucharPrefAlojaWS*”, la cual corresponde a la búsqueda de alojamiento que es la misma estructura que se implemento para la búsqueda de transporte, entretenición y restauración.

```

1 public class escucharSeleccionPlanificador extends CyclicBehaviour{
2
3     private      MessageTemplate      plantilla      = null;
4     private      Planificador          myPlan        = null;
5
6     public escucharSeleccionPlanificador(Planificador a){
7
8         ...
9     }
10    public void action() {
11
12        ACLMessage mensaje = myPlan.receive(plantilla);
13        String contenido = null;
14        String idIti = null;
15        ...
16        boolean flag = false;
17
18        if(mensaje != null){
19
20            contenido = mensaje.getContent();
21
22            ACLMessage respuesta = new ACLMessage(ACLMessage.INFORM);
23            respuesta.setLanguage(respuesPref);
24            respuesta.addReceiver(mensaje.getSender());
25
26            if(contenido != null && contenido.length() > 5){
27
28                conetenidoMsjList = UtilUsuarios.SepararMensaje(contenido, "~");
29
30                ...
31
32                basicoList = UtilUsuarios.SepararMensaje(mensaje.getEncoding(), ",");
33                itinerarioDTO = new ItinerarioDTO();
34                itinerarioDTO.setIdUser(basicoList.get(0));
35                itinerarioDTO.setNombreItinerio(basicoList.get(1));
36                itinerarioDTO.setRegIni(basicoList.get(2));
37                itinerarioDTO.setRegFin(basicoList.get(3));
38                itinerarioDTO.setFechaIni(basicoList.get(4));
39                itinerarioDTO.setFechaFin(basicoList.get(5));
40                itinerarioDTO.setCantidad(basicoList.get(6));
41
42                flag = ItinerarioDAOLogica.guardarItinerario(itinerarioDTO);
43
44                if (flag) {
45
46                    itinerarioRespDTO = ItinerarioDAOLogica.
47                        buscarItinerarioIdUsuario(itinerarioDTO);
48
49                    for (String servicio : conetenidoMsjList) {
50                        servicioMsjList = UtilUsuarios.SepararMensaje(servicio, "-");
51                        servicioDTO = new ServicioDTO();
52                        servicioDTO.setIdIti(itinerarioRespDTO.getIdItine());
53                        servicioDTO.setNomServ(servicioMsjList.get(0));
54                        servicioDTO.setDireccion(servicioMsjList.get(1));
55                        servicioDTO.setTelefono(servicioMsjList.get(2));
56                        servicioDTO.setCiudad(servicioMsjList.get(3));
57                        servicioDTO.setDescripcion(servicioMsjList.get(4));
58                        servicioDTO.setPrecio(servicioMsjList.get(5));
59
60                        flag = ServicioDAOLogica.guardarServicio(servicioDTO);
61                    }
62
63                    ...
64                }
65                myPlan.send(respuesta);
66
67            }else{
68                respuesta.setContent(error);
69                myPlan.send(respuesta);

```

```

70     }
71   }else{
72     System.out.println(" EscucharSeleccionPlanificador: esperando mensajes");
73     block();
74   }
75 }
76 }
77 }
78 }

```

Ilustración 5.25 Código Fuente: Behaviour Guardar Selección Planificador

En la Ilustración 5.25 se muestra parte del código del Agente Planificador. Lo primero es que corresponde al Behaviour “*escucharSeleccionPlanificador*” encargado de guardar un Itinerario nuevo al momento de recibir una solicitud, este Behaviour recibe un mensaje dependiendo del contenido (línea 26) lo acepta y separa almacenándolo en dos Array “*conetenidoMsjList*” y “*basicoList*”, el primero corresponde a los servicios contratados y el segundo a la cabecera del Itinerario, estos arreglos son almacenados en sus respectivos DTO, primero es guardado el Itinerario con su cabecera y posteriormente los Servicios adquiridos uno por uno, al finalizar este proceso envía una respuesta indicando su término (línea 65).

5.5 Modelo de Datos

Estos modelos se han desarrollado, con la finalidad de plasmar las tablas que sustentan el proyecto completo.

Su propósito viene dado a:

- Establecer el alcance entre las tablas.
- Dar visibilidad sobre las relaciones existentes entre las tablas.
- Proveer la información para su posterior desarrollo.

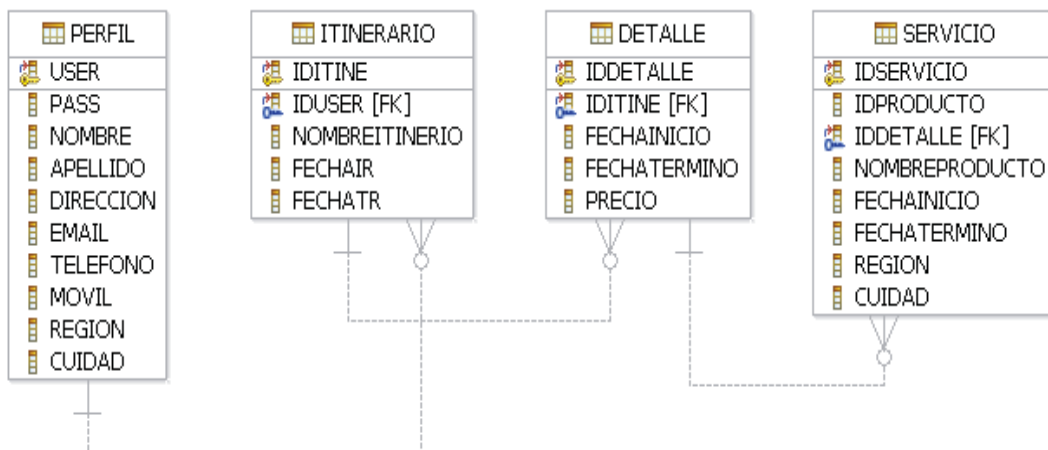


Ilustración 5.26 Modelo Datos: Turista

La Ilustración 5.26 corresponde al modelo de datos del usuario de la aplicación, es decir, el turista. Dependiendo de cuando corresponda almacenar el itinerario y sus servicios contratados.

- **Perfil:** Esta tabla provee y mantiene la información relacionada con el turista. Se relaciona con Itinerario.
- **Itinerario:** Esta tabla provee y mantiene la información relacionada con el plan general del turista. Se relaciona con Detalle.
- **Detalle:** Esta tabla provee y mantiene la información relacionada fechas de servicios contratados. Se relaciona con Servicio
- **Servicio:** Esta tabla provee y mantiene la información relacionada con el detalle del servicio contratado.

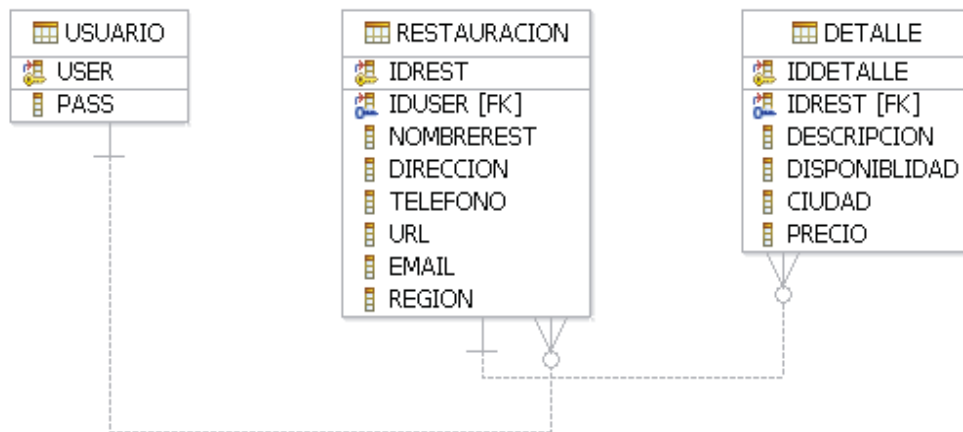


Ilustración 5.27 Modelo Datos: Restauración

La Ilustración 5.27 corresponde al modelo de datos de Restauración del Sistema de Proveedores, donde se almacenan los datos de estos proveedores.

- **Usuario:** Esta tabla provee y mantiene la información relacionada al proveedor de restauración. Se relaciona con Restauración.
- **Restauración:** Esta tabla provee y mantiene la información relacionada, con los restaurantes que posee un usuario. Se relaciona con Detalle.
- **Detalle:** Esta tabla provee y mantiene la información relacionada con la oferta de productos que posee el restaurant.

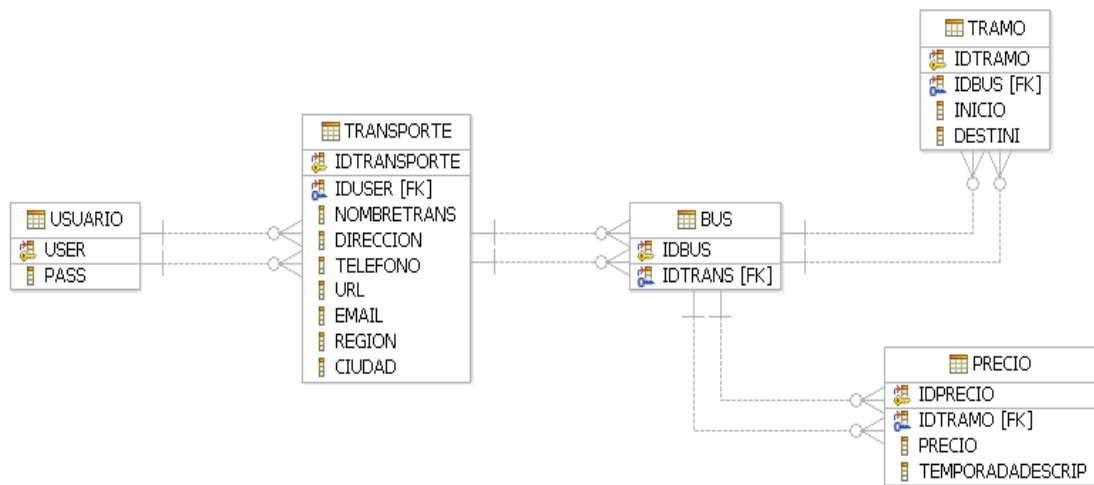


Ilustración 5.28 Modelo Datos: Transporte

La Ilustración 5.28 corresponde al modelo de datos de Transporte del Sistema de Proveedores, donde se almacenan los datos de estos proveedores.

- **Usuario:** Esta tabla provee y mantiene la información relacionada al proveedor de transporte y es administrador de sus servicios. Se relaciona con Transporte.
- **Transporte:** Esta tabla provee y mantiene la información relacionada con el nombre de la empresa de transporte además de su número de contacto, dirección e informaciones varias. Se relaciona con Bus
- **Bus:** Esta tabla se relaciona con Tramo y Precio.
- **Tramo:** Esta tabla provee y mantiene la información relacionada con los recorridos de la maquina.
- **Precio:** Esta tabla provee y mantiene la información relacionada con los precios que posee un Bus.

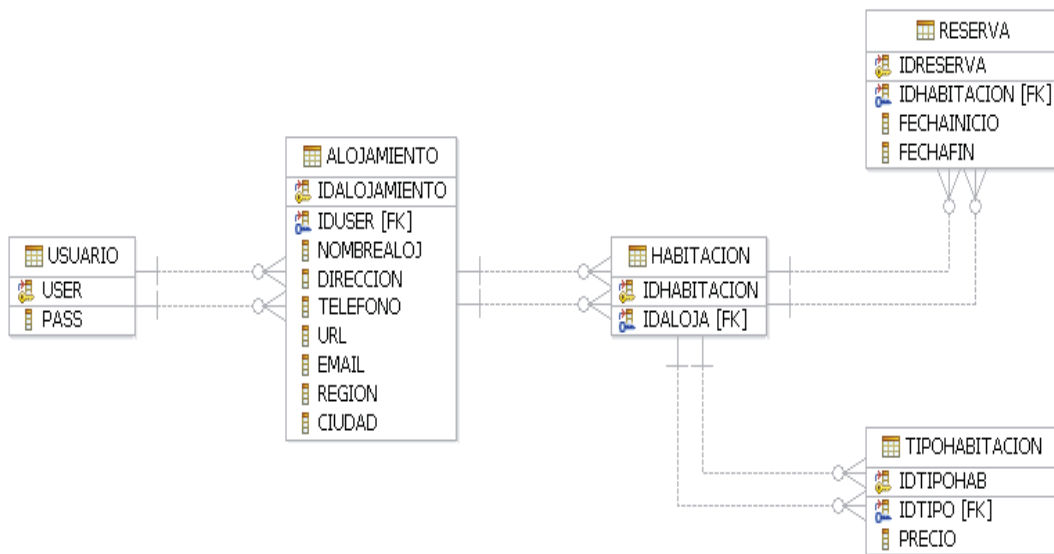


Ilustración 5.29 Modelo Datos: Alojamiento

La Ilustración 5.29 corresponde al modelo de datos de Alojamientos del Sistema de Proveedores, donde se almacenan los datos de estos proveedores y se administran sus servicios.

- **Usuario:** Esta tabla provee y mantiene la información relacionada al proveedor de alojamiento y administrador de servicios. Se relaciona con Alojamiento.
- **Alojamiento:** Esta tabla provee y mantiene la información relacionada con el nombre de la empresa de alojamiento además de su número de contacto, dirección e informaciones varias. Se relaciona con Habitación.
- **Habitación:** Esta tabla se relaciona con Reserva y TipoHabitacion.
- **Reserva:** Esta tabla provee y mantiene la información relacionada con las reservas previas a una habitación.
- **TipoHabitacion:** Esta tabla provee y mantiene la información relacionada con los precios que posee una habitación además de su tipo.

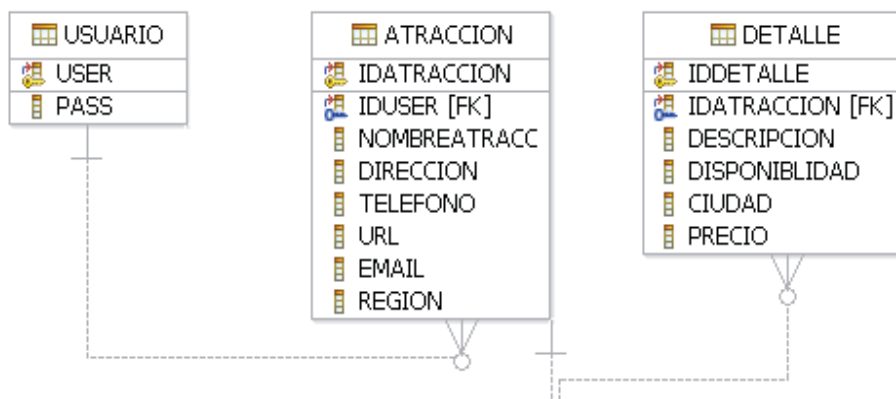


Ilustración 5.30 Modelo Datos: Atracción

La Ilustración 5.30 corresponde al modelo de datos de Atracciones del Sistema de Proveedores, donde se almacenan los datos de y se administran sus servicios.

- **Usuario:** Esta tabla provee y mantiene la información relacionada al proveedor de atracciones quien es el que administra sus servicios. Se relaciona con Atracción.
- **Atracción:** Esta tabla provee y mantiene la información relacionada con el nombre del contrato de la atracción ofrecida. Se relaciona con Detalle.
- **Detalle:** Esta tabla provee y mantiene la información relacionada a la descripción de la atracción.

Capítulo 6

Prototipo de Sistema

Para construir el sistema propiamente tal, se utilizara un lenguaje de programación JAVA en su plataforma JADE LEAP para lo que es el planificador de itinerarios en el dispositivo móvil por otro lado se utilizara Java Web para los Servicios Web.

Con respecto al IDE de trabajo se utilizó Netbeans 6.8 y en el caso de las bases de datos se utilizó DB2 9.5 para trabajar.

Para implementar la solución se trabaja en el prototipo en Java usando el framework JADE-LEAP soportado bajo el estándar FIPA. Las pruebas e implementación se realizan en dos emulador de Series 40 6th Edition SDK Nokia y en Sun Java™ Wireless Toolkit for CLDC Version 2.5.2_01.

6.1 Lanzamiento RMA

La implementación del main-container corresponde a la base del proyecto porque contiene los agentes que manejan la lógica principal de programa, además posee un contenedor secundario que provee el intermediario Front End que es necesario para operar agentes en un dispositivo móvil de bajo poder de procesamiento.

A continuación una captura pantalla del RMA, implementando todos los agentes que posee el sistema.

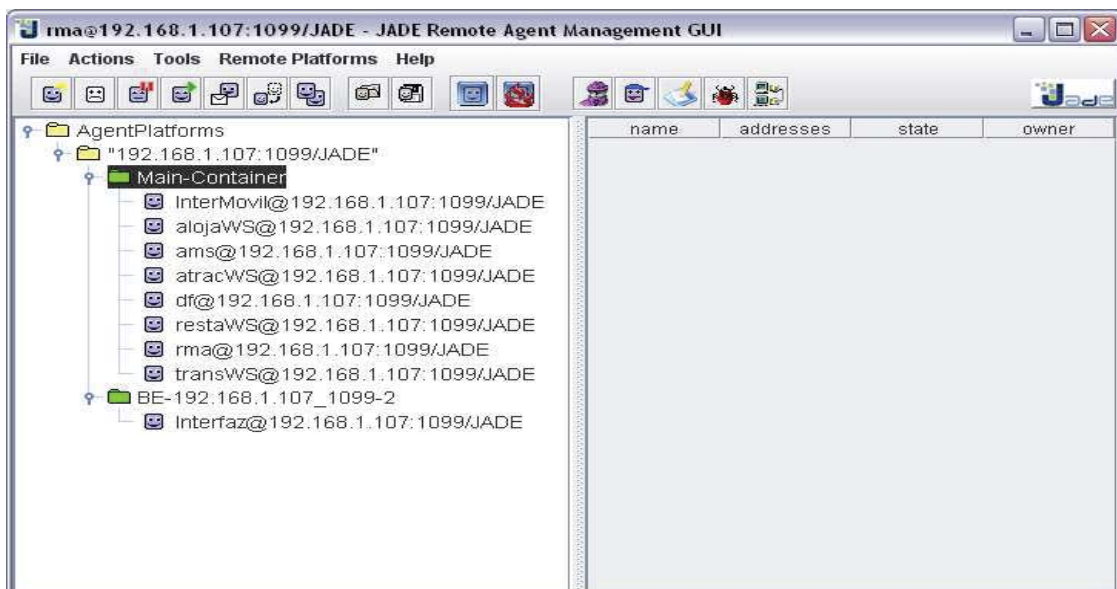


Ilustración 6.1 RMA Sistema de planificación Turística

En la Ilustración 6.1 se pueden ver dos contenedores; el Main-Container y el BE-192.168.1.107_1099-2. Del primero se puede observar a tres agentes “df”, “ams” y “rma” que son los agentes que se necesitan para ejecutar la plataforma JADE. A continuación se presenta una descripción grafica más en detalle de los agentes que componen el sistema y sus tareas a cargo.

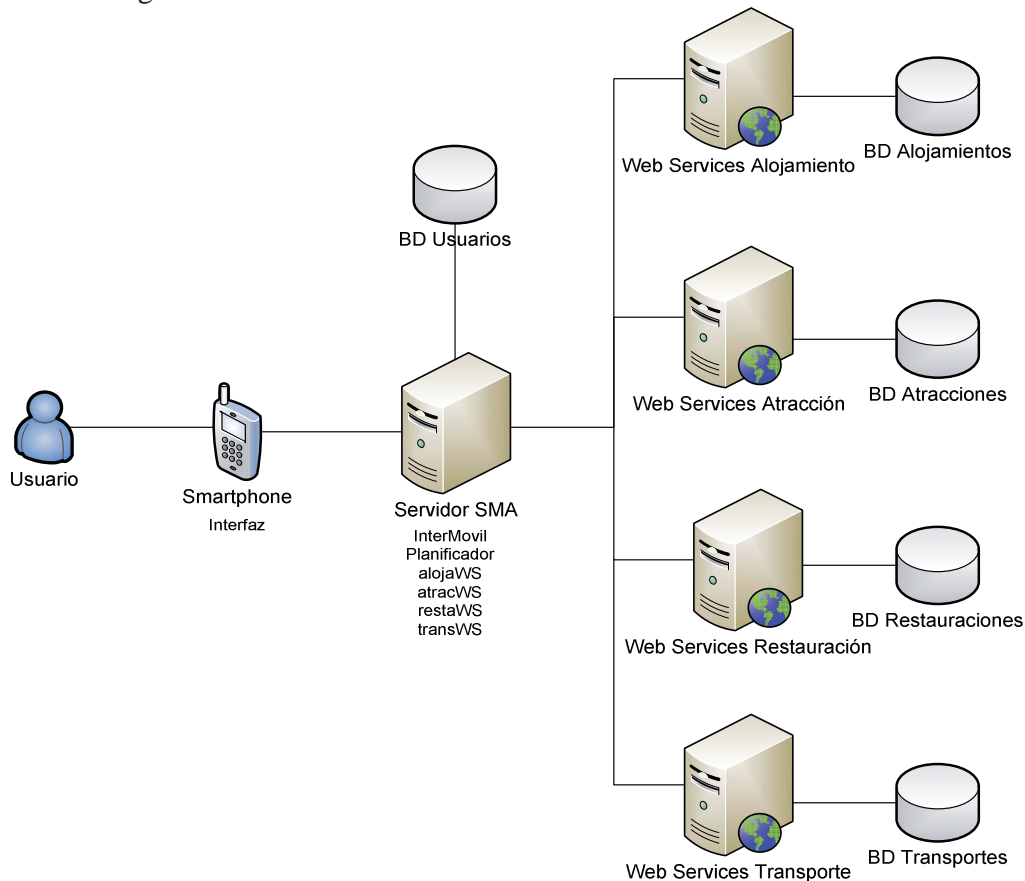


Ilustración 6.2 Despliegue grafico del Sistema de planificación Turística

El sistema consiste en un Smartphone, un Servidor SMA, el primero corresponde al Front End y el segundo al Back End, esto es muy útil para usar menos memoria. El Front End es el que captura los datos ingresados por el Usuario y el Back End es el contenedor de la lógica y de la comunicación con los Web Services, a continuación una descripción de cada agente y su función en el sistema.

- **Interfaz:** Corresponde al agente que actúa de intermediario entre el dispositivo móvil y el BackEnd. Además maneja la información que se visualiza y envía al turista para que seleccione o disponga.
- **InterMovil:** Maneja la comunicación entre desde el main-container con el agente Interfaz y se comunica con los demás agentes del Main-Container entregando información de las preferencias como del itinerario seleccionado.

- **Planificador:** Corresponde al agente a cargo de validar que las preferencias del seleccionadas por el usuario sean factibles.
- **alojaWS:** Aquí el agente se encarga de solicitar los servicios web correspondientes los servicios de alojamiento, los debidos Web Services dada selección de atributos que el turista ha señalado.
- **atracWS:** Este agente la tarea que tiene es solicitar que servicios están disponibles respecto a las atracciones turísticas disponibles dadas la selección del turista.
- **restaWS:** Cumple con solicitar los servicios de comida disponibles, dada la selección del turista.
- **transWS:** Le corresponde enviar solicitudes a los servicios web de transporte, para buscar en ellos proveedores de transportes que más se acomoden a las solicitudes del turista

Los últimos cuatro agentes que consumen servicios web forman parte de los consumidores de servicios es decir son la parte del cliente que conforma una arquitectura de servicios web.

6.2 Lanzamiento Prototipo Dispositivo Móvil

Después de lanzar la RMA en el contenedor secundario en el RMA debería ejecutarse la aplicación en el dispositivo móvil a través del agente Interfaz:GuiAgent donde al ejecutarse debería mostrar dos opciones (Ver Figura 6.3) de MIDlet, la primera “Agent” y la segunda “Crear Usuario” en este caso se selecciona la primera que corresponde a la aplicación del turista.

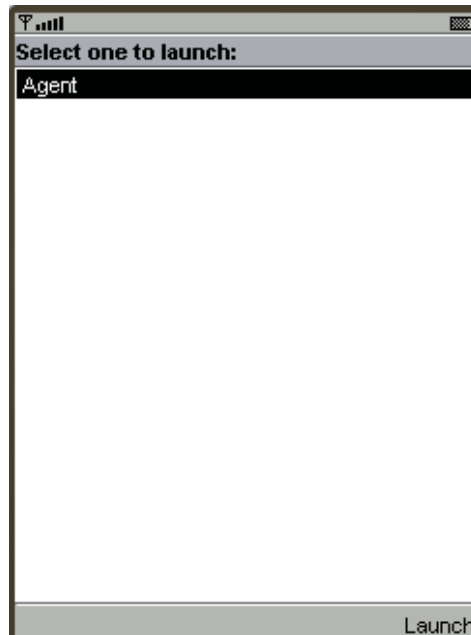


Ilustración 6.3 Lanzamiento dispositivo Móvil

Como se observa en la Ilustración 6.3 se puede ver el dispositivo móvil con la aplicación funcionando, al seleccionar la primera opción aquí se permitirá acceder a todas funcionalidades del sistema de planificación turística.

Cabe destacar que por un tema de espacio solo se mostrarán las pantallas sin el dispositivo móvil.

6.3 Prototipo Dispositivo Móvil

La siguiente secuencia corresponde a la manera de operar de la aplicación móvil de acuerdo a un flujo correcto de datos.

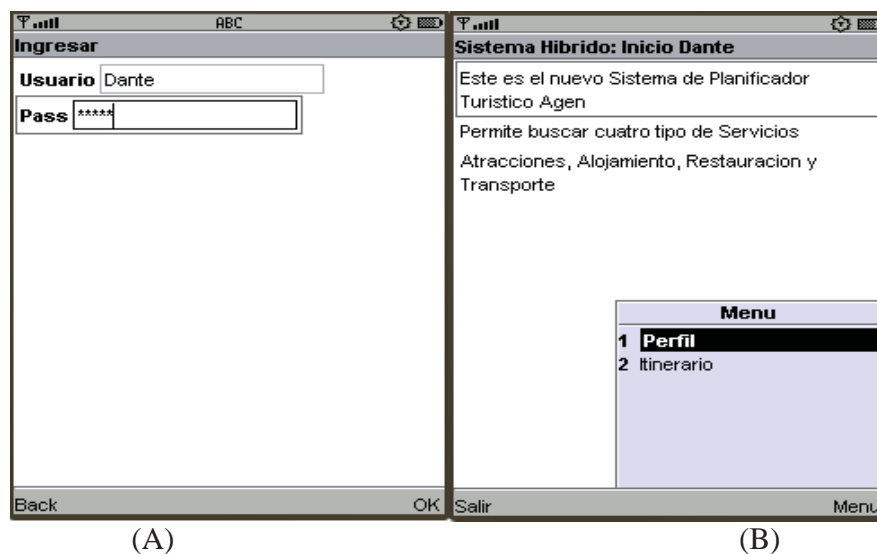


Ilustración 6.4 Login de Usuario y menú principal de la Aplicación

La Ilustración 6.4 en la captura (A) corresponde al ingreso del turista mediante usuario y contraseña que se encuentran alojadas en una base de datos. Realizada la validación correspondiente al usuario se le permite acceder al menú principal de la aplicación, donde existen dos opciones: *Perfil* e *Itinerario* las cuales son las acciones que permite realizar la aplicación.

En la captura (A) participan los agentes Interfaz e InterMovil donde el primero se encarga de visualizar la interfaz del usuario y el segundo comunica al main-container donde se encuentra la base de datos de usuarios registrados.

De la captura (B) participa el agente Interfaz que visualiza la gui en el dispositivo móvil a través de los agentes.

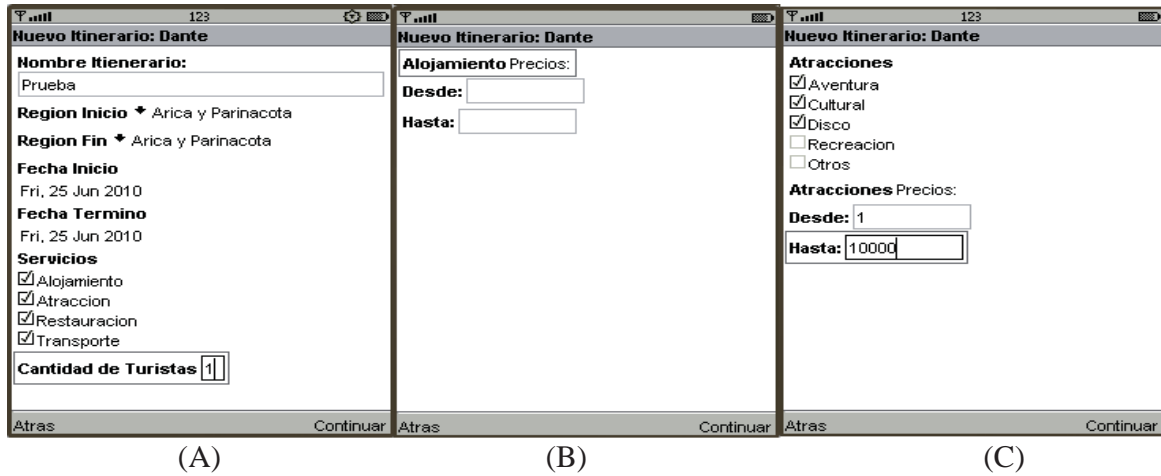


Ilustración 6.5 Nuevo Itinerario

En la Ilustración 6.5 (A) corresponde preferencias básicas del usuario respecto a la “región de inicio y destino”, la “fecha de inicio y término” los servicios que desea consultar y la “cantidad de turistas”, posteriormente se marca siguiente.

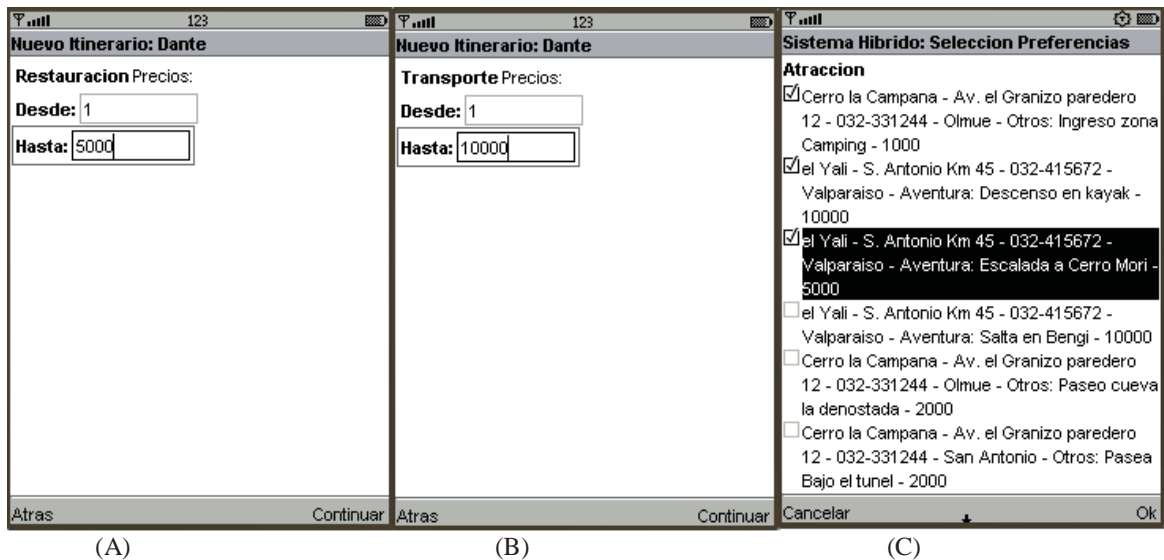


Ilustración 6.6 Selección de Preferencias

De la misma forma que en la selección de nuevo itinerario se seleccionan en la Ilustración 6.6 los distintos servicios contratados en la captura (C) se seleccionan.



Ilustración 6.7 Selección resultado Preferencias

En la Ilustración 6.7 se completa la selección y se visualizan un mensaje que cumplen con las preferencias del turista, donde avisa que se realizó sin ningún contratiempo.

Con esta actividad al presionar “ok” se genera se completa la tarea aunque ya están guardados los datos del itinerario en la base de datos.

Para completar la explicación del uso del software se adjunta un Anexo C que comprende ese apartado explicado de manera secuencial en su uso para que no exista ninguna duda y quien lo consulte comprenda su funcionamiento.

6.4 Prototipo Servicios Web

En este punto se muestra la forma en que operan los cuatro servicios web: atracciones, alojamiento, restauración y transporte que componen las actividades o servicios que consulta el sistema de planificación turística.

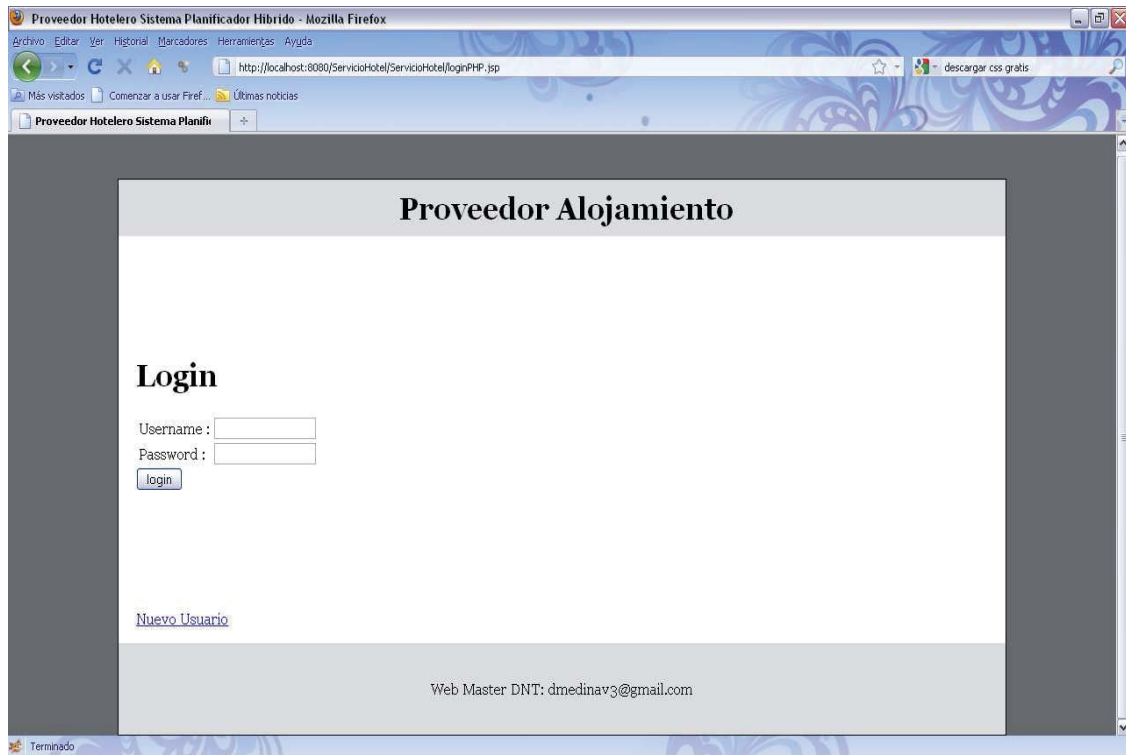


Ilustración 6.8 Login Proveedor alojamiento

La Ilustración 6.8 corresponde al menú de ingreso del servicio web de alojamiento donde un proveedor previamente registrado puede acceder.

El ingreso y acceso a todas las funcionalidades del sistema están validadas por usuarios registrados donde las tareas comunes que se registran son: crear nuevo proveedor que es la principal para acceder al sistema web Services, después de ellos se puede agregar servicios dependiendo de cada uno y su nivel de abstracción con el turista.

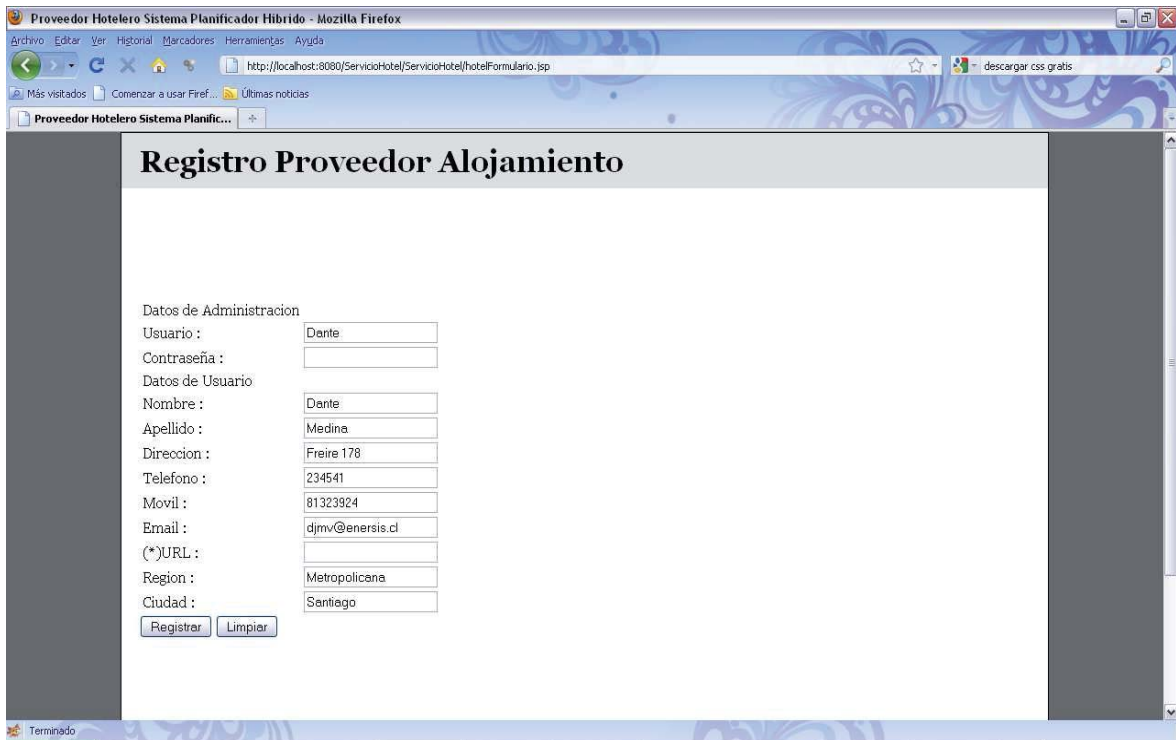


Ilustración 6.9 Registro de nuevo proveedor

En la Ilustración 6.9 se muestra la actividad de cómo se mantiene en ingreso de nuevos proveedores al sistema de servicios web que dispone el alojamiento, en este caso se implementa de la misma forma en todos los servicios registrando los datos básicos para posteriormente administrar los contenidos del web Services asociado, los servicios web más complejos de mantener son el alojamiento y el transporte, los otros dos son más simples, esto se debe a que los dos primeros manejan servicios con cupos limitados en este caso reservas.

Para finalizar con la explicación del Prototipo de servicios web, este consiste en registrar servicios en cada módulo (cuatro) para que los turistas que accedan la aplicación obtengan la información que ellos necesitan saber para planificar correctamente un viaje.

Capítulo 7

Pruebas

7.1 Descripción Plan de Pruebas

El plan de pruebas se enfocó principalmente a realizar pruebas de defectos, se buscaron las inconsistencias entre la especificación y el sistema. Al realizar las pruebas unitarias se utilizó un enfoque de caja negra, ingresando entradas posibles y simulando todas las funcionalidades que ofrece el sistema,

De acuerdo al tipo de prueba se realizaron: modulares por componente y de integración. Las pruebas modulares cumplieron con el objetivo de validar individualmente la funcionalidad de cada componente una vez codificado. Las pruebas de integración tuvieron como función validar la correcta operación entre el módulo de agentes y el de servicios web.

7.1.1 Proceso de Prueba

El sistema se ejecutó con datos ficticios de prueba, donde se evaluaron las salidas que efectúa el software y su comportamiento, con estas pruebas se buscó encontrar alguna inconsistencia posible, esperando encontrar algún defecto para de esta forma poder corregirlo.

7.1.2 Elementos a Probar

Los elementos a probar dentro del sistema fueron principalmente los requerimientos funcionales que están compuestos por Agentes, Behaviour, métodos que forman módulos del sistema. Respecto a cómo se seleccionaran los escenarios de pruebas, estos serán de acuerdo a la importancia y complejidad, dejando de lado los más simples.

7.1.3 Criterios Evaluar

Los criterios a evaluar se definieron en cada prueba considerando el set de datos a ingresar y la respuesta que se espera tener como resultado correcto es un éxito, en caso contrario la prueba se definirá como falla y se realizó la corrección más adecuada para resolver dicho problema.

7.1.4 Enfoque

El enfoque de prueba en primera instancia fue de caja negra para encontrar los errores de las funcionalidades rápidamente. En el caso de éxito la prueba será catalogada como exitosa, de lo contrario se realizará una inspección de código para encontrar la falla y corregirla.

Se realizaron pruebas unitarias para evaluar partes de códigos y así mostrar que funcionan correctamente, en caso contrario se corrigieron con inspecciones como se menciono anteriormente.

7.1.5 Materiales para las Pruebas

Las pruebas se realizaron con el emulador que dispone la herramienta de desarrollo y en un dispositivo móvil con conexión wifi Nokia 5530 Xpress Music.

7.2 Pruebas Unitarias

A continuación se describen los escenarios de pruebas de cada prueba unitaria el detalle y resultado de cada una de ellas se encuentra en el apartado final Resultados Pruebas Unitarias.

- **Nuevo ID Usuario**

En este caso el Usuario ingresa todos los campos requeridos para ingresar un nuevo usuario al sistema donde los campos no deben poseer menos de 3 ni más 30 caracteres, para posteriormente presionar “*registrar*”.

- **Login de Usuario**

En este caso El usuario debe ingresar su identificador y su contraseña los campos son alfanuméricos y el sistema debe responder ingresando en el caso que los campos ya mencionados correspondan y con un mensaje de error en el caso contrario. Esto se llevara a cabo con la previa carga de datos en la base de datos del usuario para que cuando la consulte de las formas correctamente e incorrectamente esta responda adecuadamente.

- **Selección de tipo de Preferencias**

Previo ingreso al sistema se selecciona la opción itinerario, nuevo itinerario para entrar a selección básica del sistema, dentro de este formulario seleccionar el o los tipos de servicios a consultar. Este parte no requiere carga de datos en las bases de datos eso si requiere mayor rigurosidad en las secuencias de ejecución que se realizan.

- **Nombre Itinerario**

En el formulario de nuevo Itinerario ingresar un nombre al itinerario para posteriormente el sistema realice una búsqueda que muestre si ese nombre de itinerario existe ya que no deben existir dos itinerarios con el mismo nombre.

- **Enviar Preferencias**

Ya realizada la selección de preferencias se valida que se consulte correctamente los tipos de servicios consultados, esto significa que se preparara un set de datos cargados en los servicios web y se esperara que la respuesta sea la que se preparo.

- **Otras Pruebas**

Se realizaron múltiples pruebas validando los campos de completado por el usuario cuando correspondiera llenar o completar formularios ya sean alfanuméricos y simplemente numéricos, también existió una preocupación por los correos electrónicos que posean el arroba (@).

7.3 Resultados Pruebas Unitarias

A continuación se describen los resultados obtenidos para el plan de pruebas, dando una descripción de la prueba, los componentes a probar, los datos de entrada y salida, además de los resultados esperados.

- **Nuevo ID Usuario**

Nombre de Método: validarRegistroUsuario.

Objetivo Método: capturar los datos ingresados por el nuevo Usuario y pasarlos al behaviour.

Behaviour: ComprobarInitiator

Clase: Gui

Agente: GuiAgent - InmediarioMovil

Conceptos a probar: Que no existan dos ID de usuario con el mismo nombre

Especificaciones de entrada: Los ID deben medir al menos 4 caracteres de largo alfanumérico y máximo 12 caracteres alfanuméricos

Especificaciones de salida: mensaje a través de Inform que especifique que si el resultado de la operación es correcto o incorrecto.

Resultados esperados: En el caso que exista un ID igual mensaje que lo indique en el caso contrario saltar a interfaz de login.

- **Login de Usuario**

Nombre de Método: validarUserPass.

Objetivo Método: capturar los datos ingresados por el Usuario y pasarlos al behaviour.

Behaviour: validarUserPass

Clase: Gui

Agente: GuiAgent - InmediarioMovil

Conceptos a probar: Validar que ID y password son correctos.

Especificaciones de entrada: Los ID deben medir al menos 4 caracteres de largo alfanumérico y máximo 12 caracteres alfanuméricos, lo mismo para el password.

Especificaciones de salida: en el caso exista retorna un mensaje que lo indique en el caso contrario también

Resultados esperados: En el caso de ser correcto ingresar al menú principal del sistema, de lo contrario mantenerse en la pantalla de login.

- **Selección de tipo de Preferencias**

Nombre de Método: validarUserPass.

Objetivo Método: capturar los datos ingresados por el Usuario y pasarlos al behaviour.

Behaviour: ninguno

Clase: Gui

Agente: ninguno

Conceptos a probar: Validar que ID y password son correctos.

Especificaciones de entrada: Los ID deben medir al menos 4 caracteres de largo alfanumérico y máximo 12 caracteres alfanuméricos, lo mismo para el password.

Especificaciones de salida:

Resultados esperados: En el caso de ser correcto ingresar al menú principal del sistema, de lo contrario mantenerse en la pantalla de login.

- **Nombre Itinerario**

Nombre de Método: validarNomItinerario.

Objetivo Método: capturar los datos ingresados por el Usuario y los entrega al behaviour.

Behaviour: validarNomItinerario

Clase: Gui

Agente: GuiAgent - Planificador

Conceptos a probar: Que un Usuario no posea dos itinerarios con el mismo nombre

Especificaciones de entrada: Largo de la cadena mayor a 0 y menor a 30 caracteres alfanuméricos.

Especificaciones de salida: mensaje con la respuesta de confirmación de verdadero o falso.

Resultados esperados: Caso nombre aceptado continua la ejecución, caso contrario vuelve al menú de selección con un mensaje de error.

- **Enviar Preferencias**

Nombre de Método: enviarPref.

Objetivo Método: capturar los datos ingresados por el Usuario y los entrega al behaviour.

Behaviour: enviarSeleccionPreferencia

Clase: Gui

Agente: GuiAgent - InermediarioWS

Conceptos a probar: Que se busquen los servicios de acuerdo set de datos ofrecido

Especificaciones de entrada: Largo de la cadena mayor a 0 y menor a 30 caracteres alfanuméricos en el nombre del itinerario, seleccionar al menos un elemento de choicegroup de servicios completar numéricamente la cantidad de turistas y completar los formularios de servicios numéricamente y llenar los choicegroup con una opción como mínimo cuando sea requerido.

Especificaciones de salida: mensaje con la respuesta de confirmación de verdadero o falso.

Resultados esperas: Una interfaz que indique los resultados de la búsqueda ya sea positiva o negativa listando los datos de la búsqueda o expresando que no se encontró nada.

Capítulo 8

Pruebas de Performance

8.1 Descripción de la Prueba de Performance

El caso de las pruebas de performance se elaboro para validar el despliegue del sistema e implementar la mejor solución. En esta evaluación se busco medir la respuesta en tiempo (milisegundos) que poseen los servicios web bajo distintos escenarios que se componen de una serie de consultas.

Debido a que es un tema de performance se modifiko el despliegue del sistema obviando la interfaz y creando un acceso directo a la solicitud de preferencias por el Usuario, esto permite realizar de forma continua una serie de solicitudes de servicios sin mediar interacción de él.

La prueba fue concebida de la siguiente forma, se armaron diez sets de consultas aleatorias con los cuatro servicios cada una. Por cada consulta se ejecuto un proceso paralelo que realizo la búsqueda los cuatro servicios de manera simultánea y se daba por terminado cuando el agente que inicio el proceso recibe la respuesta de todos los servicios por las diez consultas. Este proceso se repitió diez veces.

8.2 Elementos a Utilizar

Los elementos utilizados en esta prueba por parte del software involucraron una modificación en el despliegue del sistema obviando la interfaz, esto con el objeto de generar un acceso directo a la solicitud de preferencias de parte del Usuario. La modificación se realizo en el código del intermediario WS, donde para simular las búsquedas de preferencias se crearon cuatro hashtable que almacenaron las preferencia de los cuatro servicios que dispone el sistema.

Los datos para simular el acceso desde la interfaz se armaron con la siguiente estructura, se definió como consulta una petición de cuatro de los cuatro servicios que posee el sistema, la cantidad de consultas a evaluar fueron diez y los datos se describen a continuación emulan una consulta.

Tabla 8.1 Estructura de búsqueda de servicios

Tipo Servicio	Valor Min.	Valor Max.	Origen	Destino
Alojamiento	20000	40000	Valparaíso	Valparaíso
Atracción	400	8000	Valparaíso	Valparaíso
Restauración	3500	10000	Valparaíso	Valparaíso
Transporte	7000	9000	Santiago	Valparaíso

La Tabla 8.1 muestra la estructura que se armó para realizar la búsqueda de servicios en las diez consultas, como cada consulta consta de los cuatro servicios se puso un identificador de servicio “*Tipo Servicio*”, el valor mínimo a cancelar “*Valor Min.*”, el valor máximo a ofrecer “*Valor Max.*”, como existen servicios que requieren lugar de partida y lugar de llegada se decidió llenar este campo en todos los servicios aunque no sean requeridos los dos “*Origen*” y “*Destino*”.

Debido al tipo de prueba el set de datos que se utilizó fue de cien ofertas por “*Tipo de Servicio*”, donde, por ejemplo, una consulta de “*Tipo de Servicio*” Alojamiento, “*Valor Min.*” 20000, “*Valor Max.*” 40000, lugar de “*Origen*” y “*Destino*” Valparaíso retorna un total veinte ofertas disponibles. En todo el set de consultas aleatorias retorna ofertas disponibles entre los rangos de cinco a veinte servicios disponibles.

Por el lado de la infraestructura el hardware los computadores utilizados en la implementación de cuatro servicios web son unos Dell Optiplex GX270, sus especificaciones técnicas son Pentium 4 de 800 mhz de procesador, 512 mb de memoria ram y 40 GB de disco duro. La red en que se trabajó es la que ofrece la Universidad específicamente la del laboratorio de proyectos.

8.3 Escenarios de Pruebas

En los escenarios de prueba se abarcó cuatro despliegues distintos que serán descritos en detalle, en cada uno se presenta el enfoque de la prueba en función de su despliegue, además se da una descripción en términos de implementación. En cada escenario se utilizó el mismo set de datos y las bases fueron idénticas

- **Cuatro Servicios Web**

Este escenario de prueba se encargó de evaluar el tiempo de respuesta los servicios web, cuando los cuatro servicios turísticos están distribuidos en distintas estaciones de trabajo. La prueba se realizó en una serie de diez corridas con diez consultas que se componen de solicitudes para los cuatro servicios disponibles en el sistema. Cada consulta se ejecutó de forma iterativa, en donde la solicitud de los servicios corrió de forma paralela, todo esto hasta completar las diez consultas se consideró una corrida exitosa. Este proceso de corridas se ejecutó diez veces. Los resultados de tiempos en cada consulta son almacenados en un archivo de texto plano.

- **Dos Servicios Web**

Al igual que el caso anterior se ejecutaron diez corridas de diez consultas donde cada consulta consta de los cuatro servicios, pero en este caso solo existen dos estaciones de trabajo lo que significó dos servicios web por cada estación de trabajo. Se consideró una estación aparte para el contenedor de los agentes.

- **Un Servicio Web**

Una sola estación de trabajo para los cuatro servicios web donde se realizaron las diez corridas con las diez consultas, en este caso se centraliza los servicios web y sus respectivas base de datos en un solo equipo, el contenedor de agentes se mantiene en una maquina aparte.

- **LocalHost**

Solución más centralizada consta del contenedor de los agentes y los cuatro servicios web en una sola estación de trabajo donde se realizaron las diez corridas con diez consultas cada una con el fin de probar cómo se comporta versus las otras descentralizadas

8.4 Resultados Prueba de Performance

Los resultados obtenidos en los cuatro escenarios se muestran a continuación en la Tabla 8.2 donde de cada despliegue se realizó un análisis de los resultados calculando el promedio de respuesta en cada consulta.

La Tabla 8.2 muestra en su eje de Y las n consultas que se utilizaron en esta prueba, el eje de las X describe los cuatro despliegues validados, y el contenido de la matriz que se genero posee los tiempos de ejecución expresados en milisegundos (ms).

Tabla 8.2 Promedio de cada Consulta en los cuatro despliegues

	Promedio (ms)			
	4 en 4	4 en 2	4 en 1	Local Host
<i>CL:1</i>	16849,7	23854,6	34900,1	23706,1
<i>CL:2</i>	18092,0	23914,1	49086,0	23753,2
<i>CL:3</i>	17895,6	23317,2	26814,1	23303,1
<i>CL:4</i>	18104,4	23373,6	29814,1	23747,1
<i>CL:5</i>	16854,8	23990,4	26649,9	23804,4
<i>CL:6</i>	17775,2	25765,8	26429,7	25453,3
<i>CL:7</i>	17356,0	24092,2	28442,2	24193,8
<i>CL:8</i>	17684,4	23573,4	30801,4	23690,5
<i>CL:9</i>	16896,9	23860,8	29057,8	24042,1
<i>CL:10</i>	17661,1	23837,6	25518,8	23743,8

De acuerdo a lo planificado se tomaron los promedios por consulta en cada uno de los escenarios planteados, debido a que en su conjunto son los datos más relevantes. La Tabla 8.2 muestra que el valor mínimo se obtuvo en el despliegue 4 en 4 en la consulta 1 con 16.849,7 ms y el máximo en el despliegue 4 en 1 en la consulta 2 con 49.086,0 ms marcando ya una gran diferencia de performance en función del tiempo, aunque debido a que no son la misma consulta se puede ver una inconsistencia respecto a la comparación, pero al analizar la misma consulta (CL:2) en el despliegue 4 en 4 su performance es más eficiente con un tiempo de 18.092,0 ms, lo que valida la afirmación respecto a la diferencia de performance.

A continuación el gráfico de la Ilustración 8.1 muestra el despliegue de tiempo de la consulta 2 validando la afirmación anterior. Donde el mejor resultado promedio se obtiene en el despliegue de cuatro servicios web en cuatro estaciones de trabajo (18.092 ms) dejando con el peor rendimiento en el despliegue de cuatro servicios web en una estación de trabajo (49.086 ms).

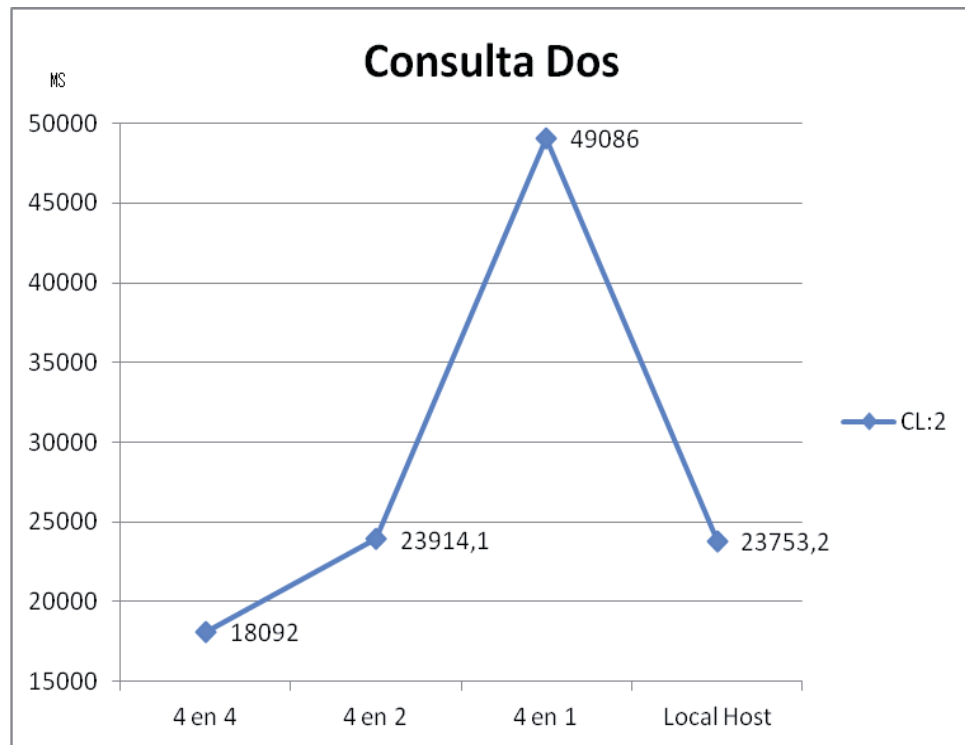


Ilustración 8.1 Performance Cuatro Escenarios Consulta dos

La Tabla 8.3 muestra la desviación estándar en función de toda la muestra por escenario, es decir se tomaron los tiempos de todas las consultas y se calculó la desviación estándar y una vez más el escenario que mostro la menor dispersión de datos en sus resultados es el de cuatro servicios web en cuatro estaciones de trabajo, esto significa que existe mayor estabilidad en el funcionamiento de este despliegue esto se suma al mejor rendimiento ya obtenido en las pruebas de performance anteriores.

Tabla 8.3 Variables de Estadística Descriptiva

Milisegundos	4 en 4	4 en 2	4 en 1	Local Host
<i>D. ESTANDAR</i>	497,63	685,07	6998,67	578,92
<i>MINIMO</i>	16849,7	23317,2	25518,8	23303,1
<i>MAXIMO</i>	18104,4	25765,8	49086,0	25453,3

Como síntesis de los cuatro casos se obtuvo que el que tiene menos posibilidades de ser la solución a implementar es la de cuatro servicios web en una sola maquina, debido a que

los resultados arrojados en este escenario demostraron ser los más inestables y de peor rendimiento, donde una eventual sobrecarga perjudicaría el funcionamiento normal del sistema y podría calificar la respuesta a una solicitud como excesivamente larga.

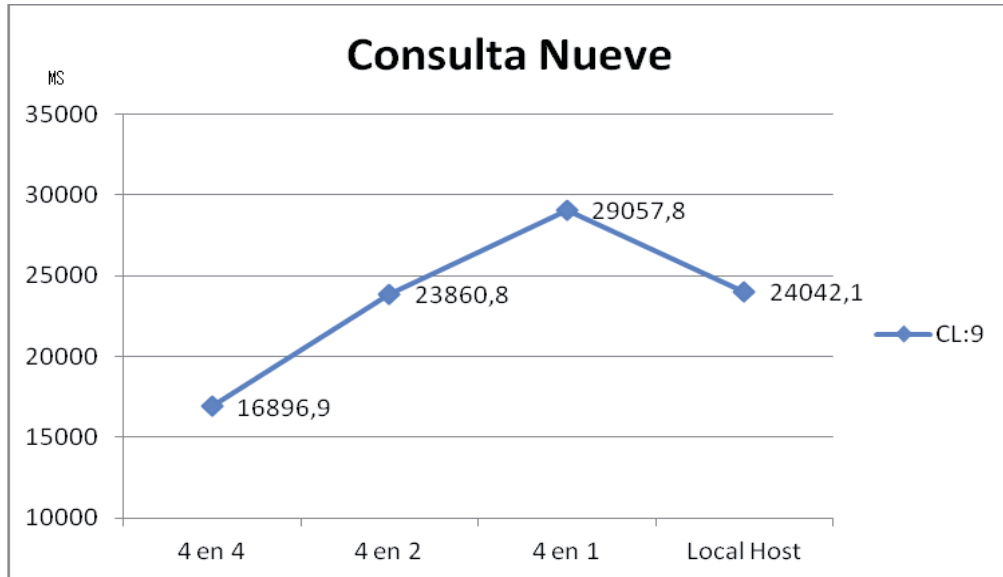


Ilustración 8.2 Performance Cuatro Escenarios Consulta nueva

Por otro parte existe una diferencia muy sutil en los resultados de la solución centralizada (Local Host) y el caso de dos servicios web en dos estaciones de trabajo. Acá al ver la Tabla 8.1 la diferencia por ejemplo de la consulta número dos (CL: 2) es de 106,15 milisegundos para la implementación del Local Host, caso opuesto la consulta número nueve (CL: 9) donde la diferencia es de 181,3 milisegundos. Con esta información se puede desprender que existe prácticamente una igualdad entre estos dos despliegues, aunque cabe mencionar que para estaciones de trabajos en igual condiciones si se aumentara la cantidad de agentes la primera en experimentar aumento de tiempo de respuesta sería la Local Host ya que los recursos de las estaciones de trabajo son limitados y al tener los cuatro servicios en una maquina y el contenedor de los agentes incorporado en la misma se podrían consumir antes los recursos versus las dos estaciones de trabajo más la que contiene los agentes.

Como solución el despliegue que presento los mejores resultados fue el caso de cuatro servicios web en cuatro estaciones de trabajo al observar los tiempos máximos 18.104,4 milisegundos y mínimos de 16.849,7 en la Tabla 8.1 estos representan las cotas del tiempo en que tardaron de responder las diez consultas.

Para terminar pensando en el sistema y llevándolo mas allá por ejemplo en un sistema de alta demanda se puede concluir que al tener distribuido el sistema de servicios web en distintas estaciones de trabajo se obtiene una mejor solución ya que en los resultados obtenidos la mejor solución se centra en el despliegue con mayor distribución de servicios.

Los resultados obtenidos en los cuatro escenarios se muestran en el Anexo B que contiene las tablas con los tiempos de cada uno de los escenarios y los gráficos por cada consulta

Capítulo 9

Conclusiones

9.1 Conclusiones del Proyecto

Para finalizar el presente proyecto se puede concluir que como tecnologías distribuidas los servicios web y los agentes inteligentes son herramientas muy poderosas y que se tiene a disposición de los desarrolladores para solucionar distintos tipos de problemas.

El trabajo realizado en el proyecto entrega los conocimientos necesarios para enfrentar el desarrollo de cualquier o nueva herramienta con arquitectura móvil distribuida, donde puede ser orientada a agentes, a servicios web, o ambas. Cabe mencionar que el proyecto no estuvo exento de dificultades, ya que el trabajo de proyectos distribuidos se torna tedioso y complejo a la hora de trabajar en la red que ofrece la Universidad específicamente el laboratorio, aunque de todos modos se pudieron superar aquellos inconvenientes.

Por otro lado se expresa que se cumple con las fechas y planes de trabajos expuestos en la planificación del mismo. Durante la ejecución del proyecto, la metodología planteada fue moldeándose de acuerdo a los requerimientos del trabajo de tal forma, que pudiera cumplir a cabalidad los objetivos propuestos al inicio de este trabajo.

Como síntesis personal del trabajo realizado, entre agentes y servicios web se demuestra que son complementarios y que en la medida que la industria los empiece a incorporar se podrían solucionar muchos problemas que presenta el uso de una sola alternativa. Como opinión, considero que los agentes son más poderosos que servicios web en el sentido de la capacidad para resolver problemas asíncronos, pero la interoperabilidad de plataformas y la sencilla implementación que ofrecen los servicios web es la piedra de tope característica que supera y pone sobre los SMA a los servicios web. En la medida que las plataformas futuras soporten a los agentes sean menos complejas y más sencillas, irá mermando esta característica que hoy por hoy tiene a los servicios web al tope del desarrollo de software y como una de las tecnologías relevantes para las soluciones distribuidas.

9.2 Trabajo Futuro

A modo de trabajo futuro, algunos aspectos que se pueden desarrollar e incluir son:

- Generar una manera de guardar las búsquedas del usuario.
- Mejorar la forma de entregar los mensajes de notificaciones al turista.
- Expandir los casos de estudio a ámbitos más complejos.

Otras mejoras que se podrían realizar.

- Mejorar la interfaz de usuario a una más amigable.
- Realizar pruebas de usabilidad en el sistema de planificación.
- Expandir la aplicación a otros dispositivos móviles que utilicen otro sistema no sólo symbian.

Capítulo 10

Referencias

- [1] Craig Larman, "UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado", 2ª edición, Ed. Prentice-Hall, 2002.
- [2] Aleida Mirian Barrientos Enríquez, "El desarrollo de sistemas de información empleando el lenguaje de modelado unificado UML", Universidad de Oriente La Habana Cuba – Universidad Autónoma Tomás Frías, Potosí-Bolivia, 2002.
- [3] ¿Por qué el Turismo? | Organización Mundial del Turismo OMT, <http://unwto.org/es/about/turismo>, visitada 12-04-2011
- [4] Entender el Turismo: Glosario Básico | Organización Mundial del Turismo OMT, <http://media.unwto.org/es/content/entender-el-turismo-glosario-basico>, visitada 12-04-2011
- [5] Russel and P. Norvig. "Artificial Intelligence: A Modern Approach". Prentice Hall, 2nd edition, 2003.
- [6] Wooldridge, M. en "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence". Ed. G. Weiss, MIT Press, Massachusetts, USA. 1999.
- [7] Ana Mas, "Agentes Software y Sistemas Multi-Agente", Pearson Educacion, 2005
- [8] FIPA Agent Management Specification, <http://www.fipa.org/specs/fipa00023/SC00023K.html>, Visitada 12-04-2011
- [9] Ernesto Ocampo Edey, "Ingeniería de Software Orientada a Agentes", Universidad Pontificia de Salamanca Campus de Madrid, España, 2005
- [10] Carmen Fernández Chamizo, Jorge Gómez Sanz, Juan Pavón Mestras, "Metodologías de desarrollo de sistemas multi-agente", Universidad Complutense de Madrid, 2003.
- [11] W3C, Guías breves Web services, <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb#masinfo>, 01 - 09 -2009.
- [12] Jennifer Pérez Benedi - Web Services, Universidad Politécnica de España, Valencia.
- [13] Web Services Architecture, <http://www.w3.org/TR/ws-arch/>, 11-02-2004.
- [14] Web Services Policy 1.5 - Framework, <http://www.w3.org/TR/ws-policy/>, 04-09-2007.
- [15] Guía Breve de Tecnologías XML, <http://w3c.es/Divulgacion/GuiasBreves/TecnologiasXML/>

- [16] Introducción al lenguaje XML, <http://geneura.ugr.es/~jmerelo/xml/>, 10-11-2004.
- [17] SOAP Version 1.2 Part 2: Adjuncts, <http://www.w3.org/TR/2003/REC-soap12-part2-20030624/>, 24-06-2004
- [18] Microsoft Corporation, Arquitectura orientada a servicios (SOA) de Microsoft aplicada al mundo real, diciembre 2006.
- [19] Getting Started with JAX-RPC, <http://java.sun.com/developer/technicalArticles/WebServices/getstartjaxrpc/> , 01-04-2002.
- [20] Building JAX-WS 2.0 Services with NetBeans 5.0 and GlassFish, <http://jax-ws.java.net/articles/jaxws-netbeans/glassfish.html>, 01-03-2006.
- [21] Ignacio García, Macario Polo, Francisco Ruiz, Mario Piattini - Servicios Web, Universidad de Castilla-La Mancha, España, 2005.
- [22] Eva Aguayo - Conceptos básicos del turismo y su impacto, Universidad Santiago de Compostela, España.
- [23] Ramón Rufin Moreno – “Las empresas turísticas en la sociedad de la información“, editorial Universitaria, 2002

Anexo A

Casos de Usos

Detalle del Caso de Uso: Administrar Empresa Turística

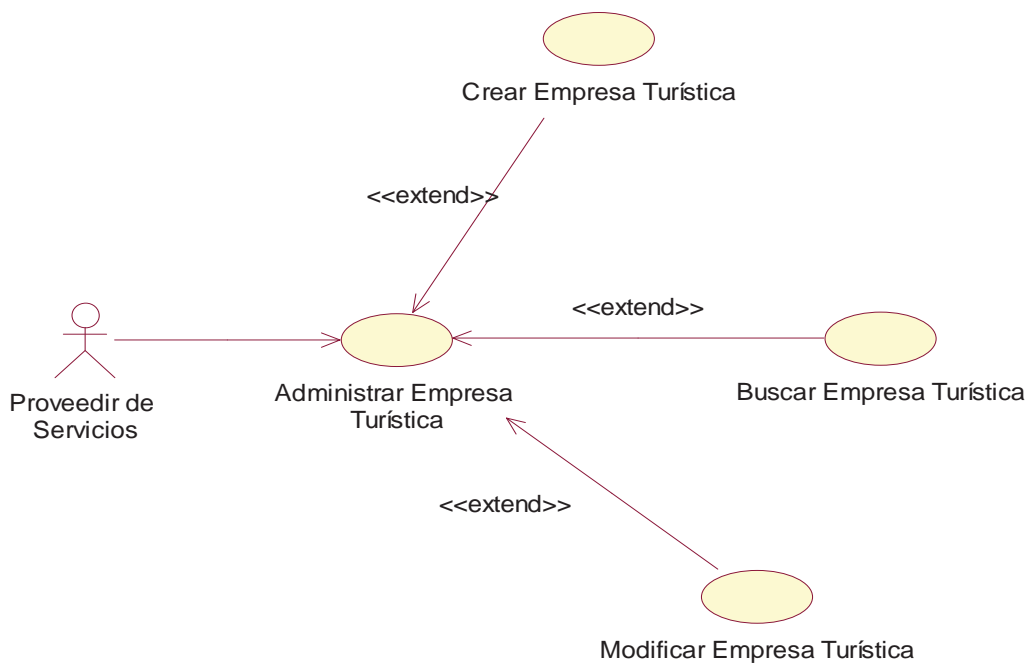


Ilustración A.1 Diagrama de Casos de Uso Administrar Empresa Turística

Nombre de Caso de Uso	CUS-MDAL04.1: Crear Empresa Turístico
Descripción de Caso de Uso	Este caso de uso posibilita la creación de Empresas Turísticas.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de creación de Empresas Turísticas. 2. El Proveedor debe completar los siguientes campos. <ul style="list-style-type: none"> ▪ Nombre Empresa ▪ Dirección ▪ Teléfono ▪ URL ▪ Email ▪ Región ▪ Ciudad 3. El Proveedor de Servicios presiona el botón “Crear”. 4. El Sistema de Proveedores valida que los datos sean

	validos. 5. El Sistema muestra un mensaje de operación realizada exitosamente.
Flujo Alternativo	4.1 El Sistema de Proveedores encuentra campos incorrectos y/o en blanco. 4.2 El Sistema de Proveedores muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL04.2: Buscar Empresa Turístico
Descripción de Caso de Uso	Este caso de uso posibilita listar las Empresas Turísticas.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de búsqueda de Empresas Turísticas. 2. El Proveedor completa los siguientes campos: <ul style="list-style-type: none"> ▪ ID ▪ Nombre Empresa 3. El Proveedor de Servicios presiona el botón “Buscar”. 4. El Sistema de Proveedores busca empresas relacionadas con los criterios. 5. El Sistema muestra la lista de empresas de la búsqueda.
Flujo Alternativo	<ol style="list-style-type: none"> 3.1 El Sistema de Proveedores encuentra campos incorrectos. 3.2 El Sistema de Proveedores muestra una alerta indicando el error. 5.1 El Sistema no encuentra resultado a los criterios de búsqueda.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL04.3: Modificar Empresa Turística
Descripción de Caso de Uso	Este caso de uso posibilita la edición de la Empresa Turística.
Actores Directos	Proveedor de Servicios
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Proveedor de Servicios ingresa a la interfaz de edición para Servicios Turísticos. 2. El Proveedor puede modificar. <ul style="list-style-type: none"> ▪ Nombre Empresa ▪ Dirección ▪ Teléfono

	<ul style="list-style-type: none"> ▪ URL ▪ Email ▪ Región ▪ Ciudad <ol style="list-style-type: none"> 3. El Proveedor de Servicios presiona el botón “Modificar”. 4. El Sistema de Proveedores valida que los datos sean validos. 5. El Sistema muestra un mensaje indicando que la modificación se realizo de manera correcta.
Flujo Alternativo	<ol style="list-style-type: none"> 4.2 El Sistema de Proveedores encuentra campos incorrectos. 4.2 El Sistema de Proveedores muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Detalle del Caso de Uso: Administrar Sistemas

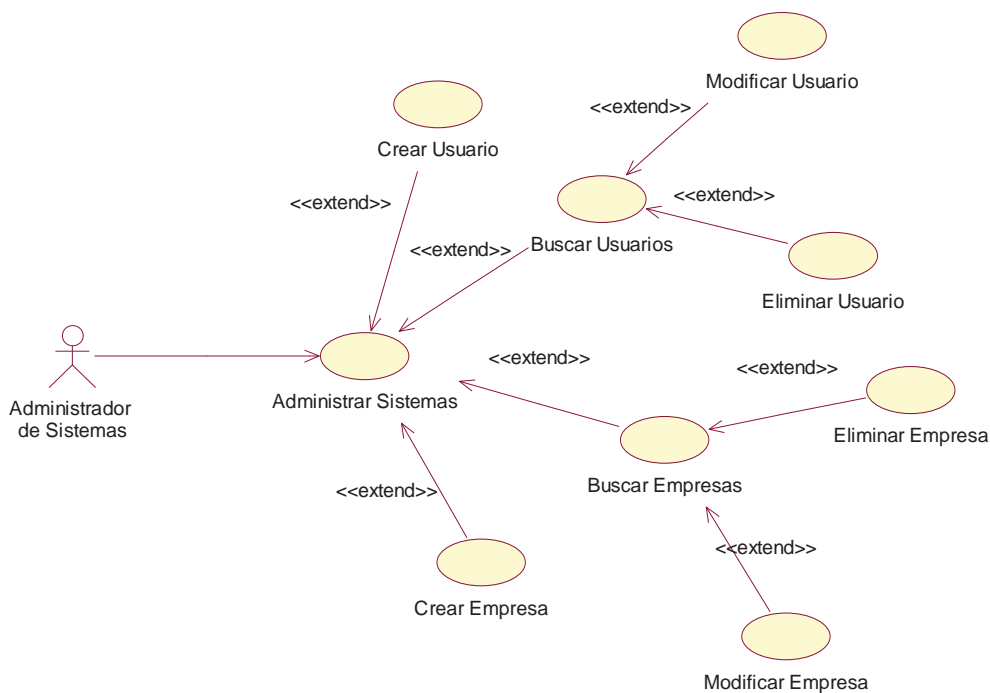


Ilustración A.2 Diagrama de Casos de Uso Registrar Proveedor

Nombre de Caso de Uso	CUS-MDAL05.1: Crear Usuario o Empresa
Descripción de Caso de Uso	Este caso de uso posibilita la creación de Usuarios o Empresas Turísticas.
Actores Directos	Administrador Sistemas

Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Administrador Sistemas ingresa a la interfaz de creación de Usuarios o Empresas Turísticas. 2. El Administrador de Sistema selecciona crear empresa. 3. El Administrador de Sistema debe completar los siguientes campos. <ul style="list-style-type: none"> ▪ Nombre Empresa ▪ Dirección ▪ Teléfono ▪ URL ▪ Email ▪ Región ▪ Ciudad 4. El Administrador de Sistema presiona el botón “Crear”. 5. El Sistema valida que los datos sean validos. 6. El Sistema muestra un mensaje de operación realizada exitosamente.
Flujo Alternativo	<ol style="list-style-type: none"> 2.1 El Administrador de Sistema selecciona crear Usuario. 3.1 El Administrador de Sistema debe completar los siguientes campos. <ul style="list-style-type: none"> ▪ Nombre ▪ Apellido ▪ Seudónimo ▪ Fecha Nacimiento ▪ Email ▪ Contraseña ▪ Duplicado Contraseña 4.1 El Administrador de Sistema presiona el botón “Crear”. 5.1 El Sistema valida que los datos sean validos. 6.1 El Sistema muestra un mensaje de operación realizada exitosamente. 5.2 El Sistema de Proveedores encuentra campos incorrectos. 6.2 El Sistema de Proveedores muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL05.2: Buscar Usuario y/o Empresa Turística
Descripción de Caso de Uso	Este caso de uso posibilita listar los Usuario y/o Empresas Turísticas.
Actores Directos	Administrador Sistemas
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Administrador Sistemas ingresa a la interfaz de búsqueda de Usuario y/o Empresas Turísticas..

	<ol style="list-style-type: none"> 2. El Administrador Sistemas completa los siguientes campos: <ul style="list-style-type: none"> ▪ ID ▪ Nombre ▪ Usuario o Empresa (opcional) 3. El Proveedor de Servicios presiona el botón “Buscar”. 4. El Sistema busca usuarios y/o empresas relacionadas con los criterios. 5. El Sistema muestra la lista de empresas de la búsqueda.
Flujo Alternativo	<ol style="list-style-type: none"> 4.1 El Sistema busca usuarios y/o empresas relacionadas con los criterios pero no encuentra información. 5.1 El Sistema muestra un mensaje indicando que no existe información para esta búsqueda.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL05.3: Modificar Usuario o Empresa Turística
Descripción de Caso de Uso	Este caso de uso posibilita la edición de un Usuario o una Empresa Turística.
Actores Directos	Administrador Sistemas
Flujo Principal de Éxito	<ul style="list-style-type: none"> • El Administrador Sistemas realiza una búsqueda de Usuarios y/o una Empresa Turística. • El Administrador Sistemas puede modificar uno solo. <ul style="list-style-type: none"> ▪ Usuario (ver: CUS MDAL02.3 Modificar Usuario) ▪ Empresa Turística (ver: CUS MDAL04.3 Modificar Empresa Turística) • El Administrador Sistemas presiona el botón “Modificar”. • El Sistema verifica que los datos sean validos. • El Sistema muestra un mensaje indicando que la modificación se realizo de manera correcta.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El Sistema encuentra campos incorrectos. 4.2 El Sistema muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado y debe existir una Búsqueda valida.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Nombre de Caso de Uso	CUS-MDAL05.4: Eliminar Usuario o Empresa Turística
Descripción de Caso de Uso	Este caso de uso posibilita la eliminación lógica de un Usuario o una Empresa Turística.

Actores Directos	Administrador Sistemas
Flujo Principal de Éxito	<ol style="list-style-type: none"> 1. El Administrador Sistemas realiza una búsqueda de Usuarios y/o una Empresa Turística. 2. El Administrador Sistemas puede Eliminar uno solo. <ul style="list-style-type: none"> ▪ Usuario ▪ Empresa Turística 3. El Administrador Sistemas selecciona un elemento de la lista. 4. El Administrador Sistemas presiona el botón “Eliminar”. 5. El Sistema verifica que no existan dependencias. 6. El Sistema muestra un mensaje indicando que la eliminación se realizo de manera correcta.
Flujo Alternativo	<ol style="list-style-type: none"> 5.1 El Sistema encuentra dependencias entre las tablas de Usuario a Alojamiento o Alojamiento a Habitación. 6.1 El Sistema muestra una alerta indicando el error.
Escenario de Extensiones Alternativas	No aplica en este nivel.
Pre-requisitos	El Usuario debe estar Logueado y debe existir una Búsqueda valida.
Post-condiciones	No aplica en este nivel.
Requisitos no Funcionales	No aplica en este nivel.

Anexo B

Resultados Pruebas Performance

En el siguiente Anexo se presentan los resultados obtenidos en los cuatro despliegues propuestos en las pruebas de performance. Las tablas a continuación describen el tiempo en milisegundos (ms) por cada consulta y al final se expresa el promedio de la consulta por las diez corridas.

- **Resultados**

Tabla B.1 Cuatro Servicios Web en Cuatro estaciones de Trabajo

Ms	CR:1	CR:2	CR:3	CR:4	CR:5	CR:6	CR:7	CR:8	CR:9	CR:10	Promedio Consulta
CL:1	18904	16610	17250	16062	17000	14891	16750	15796	16625	18609	16849.7
CL:2	27703	18062	16156	16844	16687	15968	17609	16907	17047	17937	18092
CL:3	23594	16907	16000	16906	18782	17157	18047	16765	17594	17204	17895.6
CL:4	22484	19406	16906	18078	16765	17687	17219	16000	18562	17937	18104.4
CL:5	19531	16953	16985	16797	16563	14000	16719	16625	16625	17750	16854.8
CL:6	22657	17016	15687	15438	19109	16938	17781	17078	18782	17266	17775.2
CL:7	17843	16250	19203	19093	17656	16859	17047	16313	15812	17484	17356
CL:8	19922	15812	16500	18953	16891	16781	18578	18344	18063	17000	17684.4
CL:9	15844	17688	15844	16672	18000	16875	16812	16812	16672	17750	16896.9
CL:10	19359	18078	16891	16688	16750	17016	18532	18453	17625	17219	17661.1

Tabla B.2 Cuatro Servicios Web en dos estaciones de Trabajo

Ms	CR:1	CR:2	CR:3	CR:4	CR:5	CR:6	CR:7	CR:8	CR:9	CR:10	Promedio Consulta
CL:1	24156	23359	23187	24391	22890	23187	23172	24313	25203	24688	23854.6
CL:2	25125	24860	24203	24281	23485	23125	23312	23250	24297	23203	23914.1
CL:3	23266	24156	22094	22422	24515	25297	24344	21234	24250	21594	23317.2
CL:4	22297	23422	26125	23922	22329	23125	23219	25485	22359	21453	23373.6
CL:5	22187	23265	25078	21844	22546	25078	23281	25171	26266	25188	23990.4
CL:6	22344	31360	25172	26093	22688	32047	25188	25235	24203	23328	25765.8
CL:7	23281	24375	26234	21438	23453	30047	23203	24344	24344	20203	24092.2
CL:8	24344	25328	22157	21515	24234	25141	21390	26203	24203	21219	23573.4
CL:9	23266	24187	22265	23563	27172	25062	24172	23328	23203	22390	23860.8
CL:10	26734	23204	21500	21547	24266	23266	24203	25156	24297	24203	23837.6

Tabla B.3 Cuatro Servicios Web en una estación de Trabajo

Ms	CR:1	CR:2	CR:3	CR:4	CR:5	CR:6	CR:7	CR:8	CR:9	CR:10	Promedio Consulta
CL:1	108859	33641	22766	23938	22969	24969	26234	34625	26375	24625	34900.1
CL:2	72766	23922	24921	24390	23469	23328	211594	33766	27766	24938	49086
CL:3	23891	24828	23454	24516	23687	24219	33172	31312	30344	28718	26814.1
CL:4	24890	25547	24531	23109	23688	23828	37141	33141	31531	50735	29814.1
CL:5	30157	25656	26547	27781	22000	25312	29484	28609	27156	23797	26649.9
CL:6	25921	23750	24562	24875	26953	24375	29719	31485	27157	25500	26429.7
CL:7	26704	27625	23453	23079	25453	23282	30687	51343	28390	24406	28442.2
CL:8	22953	25844	23328	23968	28531	23937	78453	23641	31328	26031	30801.4
CL:9	29218	31515	24266	23453	23344	23047	59313	22609	28610	25203	29057.8
CL:10	24641	24844	25500	24750	23500	22500	37281	24750	24812	22610	25518.8

Tabla B.4 Local Host

Ms	CR:1	CR:2	CR:3	CR:4	CR:5	CR:6	CR:7	CR:8	CR:9	CR:10	Promedio Consulta
CL:1	24156	23359	23187	24391	22890	23187	23172	24313	25203	23203	23706.1
CL:2	25125	24860	24203	24281	23485	23125	23312	23250	24297	21594	23753.2
CL:3	23266	24156	22094	22422	24515	25297	24344	21234	24250	21453	23303.1
CL:4	22297	23422	26125	23922	22329	23125	23219	25485	22359	25188	23747.1
CL:5	22187	23265	25078	21844	22546	25078	23281	25171	26266	23328	23804.4
CL:6	22344	31360	25172	26093	22688	32047	25188	25235	24203	20203	25453.3
CL:7	23281	24375	26234	21438	23453	30047	23203	24344	24344	21219	24193.8
CL:8	24344	25328	22157	21515	24234	25141	21390	26203	24203	22390	23690.5
CL:9	23266	24187	22265	23563	27172	25062	24172	23328	23203	24203	24042.1
CL:10	26734	23204	21500	21547	24266	23266	24203	25156	24297	23265	23743.8

▪ **Gráficos**

A continuación se presentan los gráficos en sus cuatros escenarios de pruebas mostrando el promedio de cada consulta en función del tiempo (milisegundos) en que demoro en completar su respuesta.

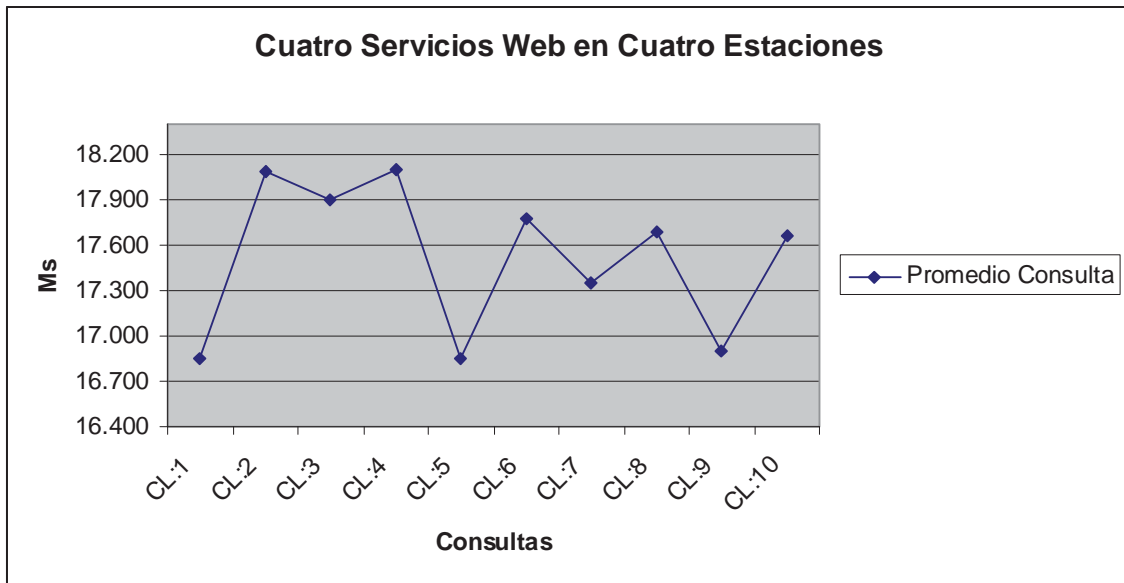


Ilustración B.1 Promedio consultas cuatro Servicios Web en cuatro estaciones

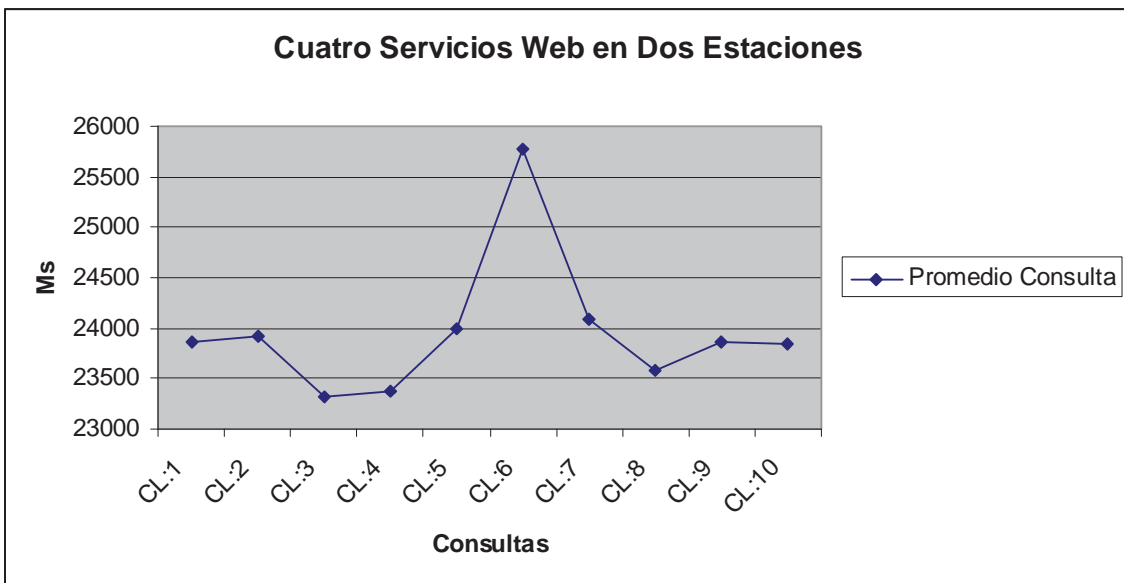


Ilustración B.2 Promedio consultas cuatro Servicios Web en dos estaciones



Ilustración B.3 Promedio consultas cuatro Servicios Web en una estación

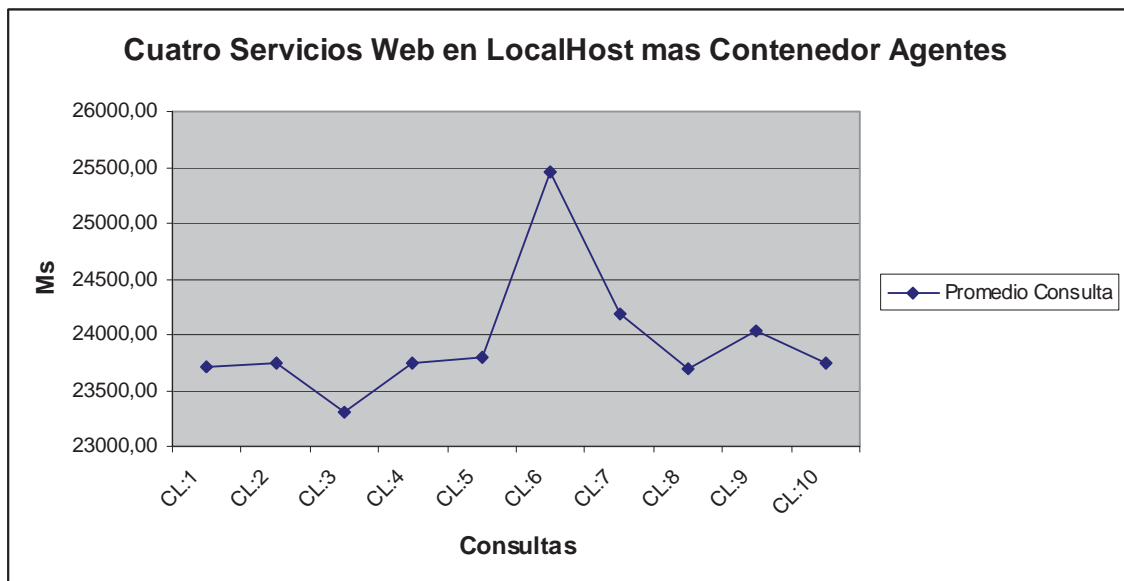


Ilustración B.4 Caso Estudio: cuatro servicios web en LocalHost

Anexo C

Manual de Usuario

Debido a la diversidad de los Usuarios se generan manuales para los distintos niveles de Usuarios que puedan existir.

1 Manual del Turista

Este manual corresponde al Usuario Final del Sistema el cual hará uso de la aplicación y tendrá acceso a todas las funcionalidades que posee.

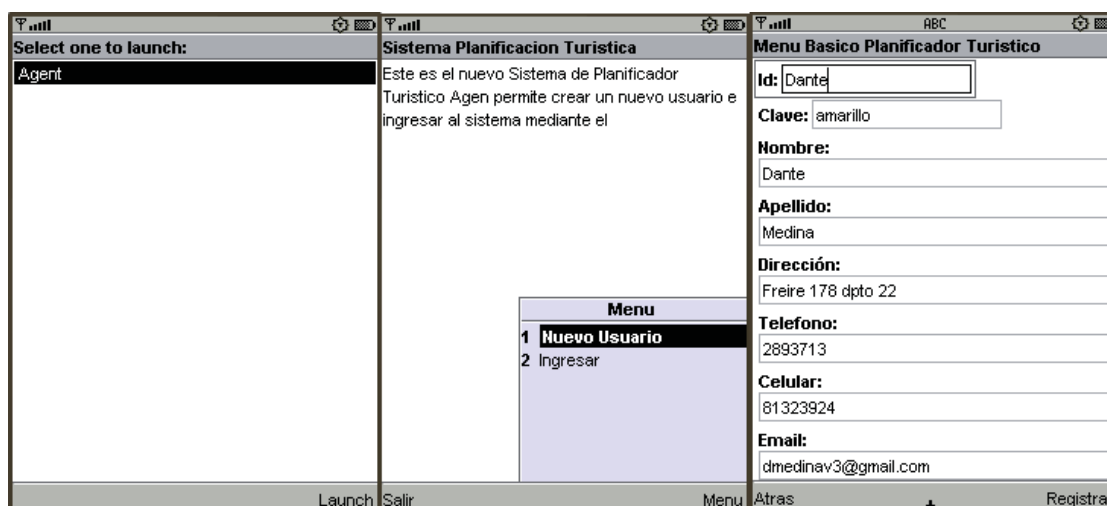
1.1 Requerimientos

Los requerimientos para el uso de este programa son contar con un teléfono con conexión wifi incluida, donde se puedan ejecutar aplicaciones J2ME específicamente CLDC 1.0 y MIP 2.0.

Con estos requerimientos cumplidos se copian los archivos *CELU1.jar* y *CELU.jad*, donde se ejecuta en el celular el archivo con la extensión *.jad* donde se encuentran los datos de la dirección del intermediario y el agente a lanzar en el celular.

1.2 Cargar la Aplicación

Al tener ya instalada la aplicación en el celular se ejecuta y se ingresa a crear nuevo usuario completando todos los campos y presionando continuar.

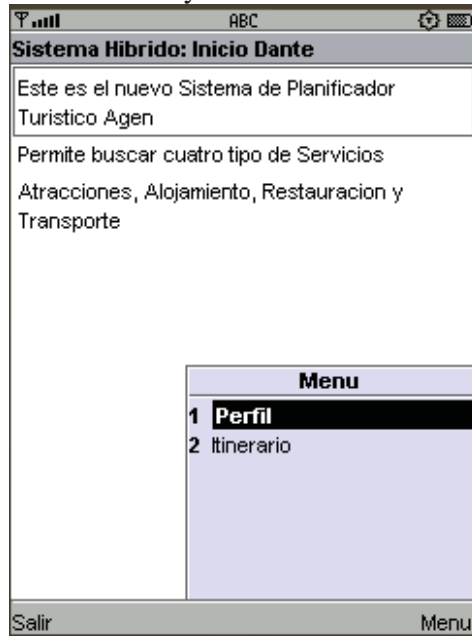


(A)

Nota: la figura A muestra la secuencia correcta de ejecutar y los pasos que hay que cumplir para poder ingresar al sistema si no se posee un id y password autorizado.

1.3 Menú Principal

Del ingreso al menú principal se puede optar a una serie de opciones que dispone el sistema las cuales son, el perfil de usuario y el itinerario.



(B)

Nota: la figura B muestra el panel principal donde se despliega las opciones principales del sistema híbrido.

1.4 Ver Perfil

Corresponde a la opción de listar los datos del perfil del usuario esto se realiza ingresando al menú de perfil y posteriormente seleccionar *ver* para de esta manera listar los datos del perfil del turista



(C)

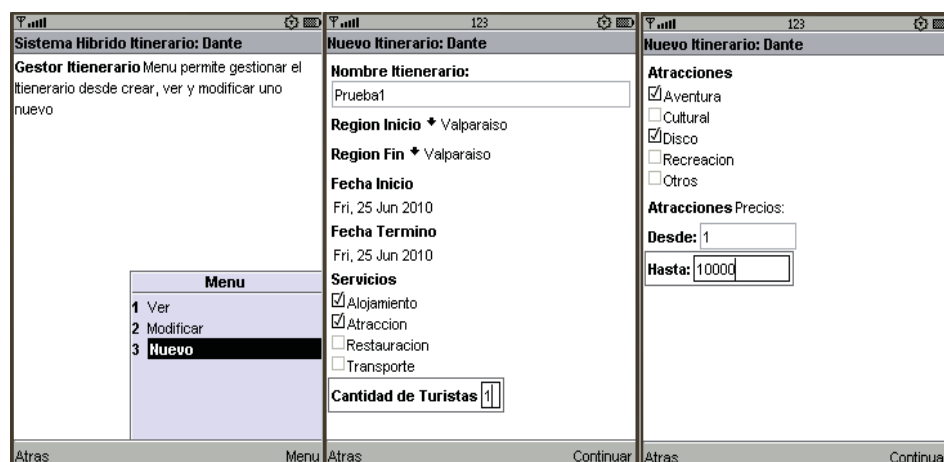
Nota: la figura C muestra el panel principal del perfil donde se despliega las opciones principales del sistema híbrido, en este caso se muestra la opción de listar los datos del usuario.

1.5 Modificar Perfil

De la misma forma que se realiza el *Ver Perfil* se lleva a cabo el *Modificar Perfil*, este caso los campos que son listados en la figura C en el lado izquierdo poseen campos de modificación para poder modificar su contenido.

1.6 Nuevo Itinerario

Del menú principal se selecciona Itinerario para luego presionar *nuevo itinerario*, donde se completan los datos que pide el sistema para realizar la búsqueda de forma correcta.

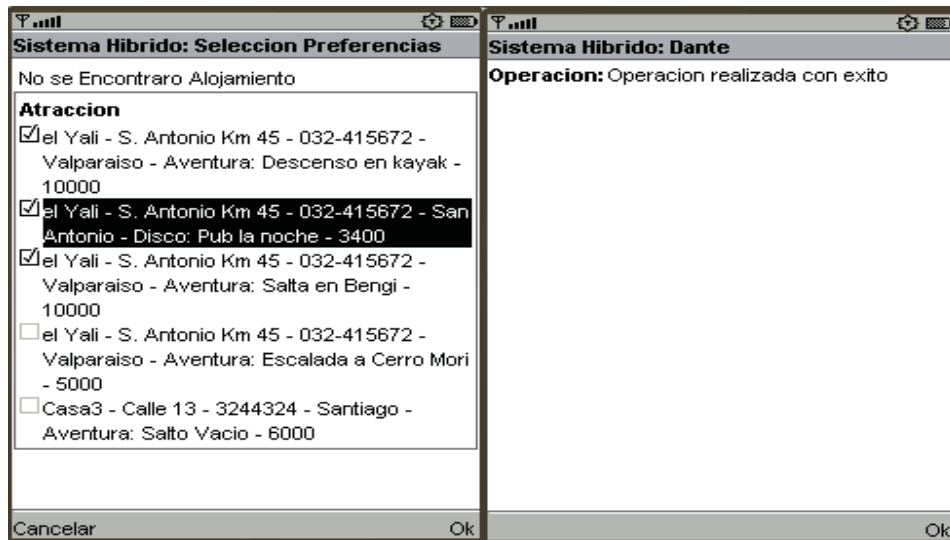


(D)

Nota: la figura D muestra la secuencia que sigue generar un nuevo itinerario donde todos los campos deben ser completados para continuar el proceso de captura de preferencias del turista para finalmente presionar continuar y esperar la respuesta de los servicios web.

1.7 Seleccionar Servicio

Corresponde a la acción de elegir los servicios que uno desea consultar para posteriormente contratar esta operación se realiza previa búsqueda de las preferencias donde puede o no encontrar servicios que se ajusten a la solicitud como lo muestra la figura E a continuación:

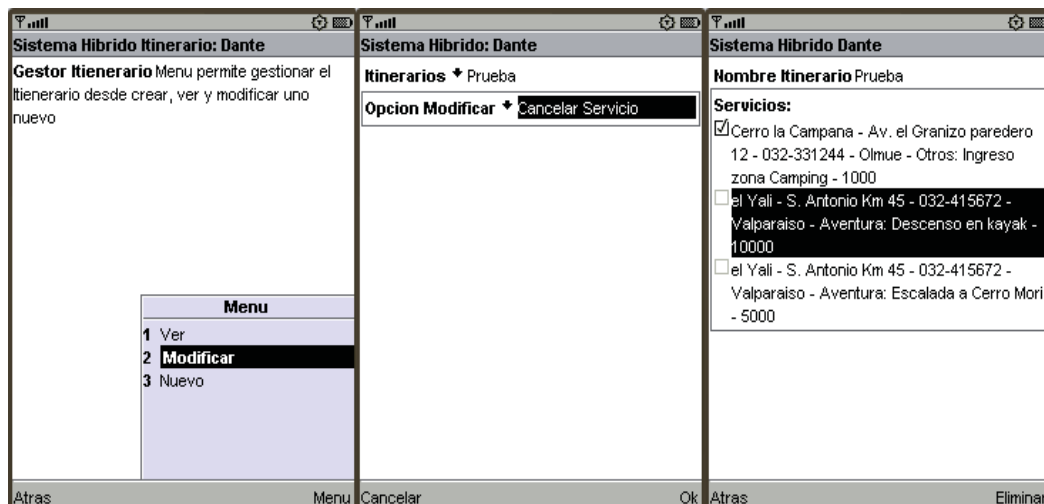


(E)

Nota: la figura E muestra un caso exitoso de servicio y uno donde no se encontró nada dada la selección inicial de turista.

1.8 Cancelar Servicio

Corresponde a eliminar servicios del itinerario en el caso que el usuario lo estime convenientes es por eso que se ingresa al menú de Itinerario y se selecciona modificar Itinerario para posteriormente seguir la secuencia de la figura F.



(F)

Nota: la figura F muestra un caso exitoso de cancelación de servicio donde se selecciona el o los servicios que se desean eliminar para posteriormente presionar *eliminar*.

1.9 Ver Itinerario

Se realiza de la misma forma que Eliminar Servicio, sólo que en vez de ingresar al menú modificar del itinerario se elige la opción ver.

1.10 Eliminar Itinerario

Igual que Cancelar Servicio solo que no se elige servicio a eliminar y se eliminar por completo el itinerario junto con sus servicios asociados.

Anexo D

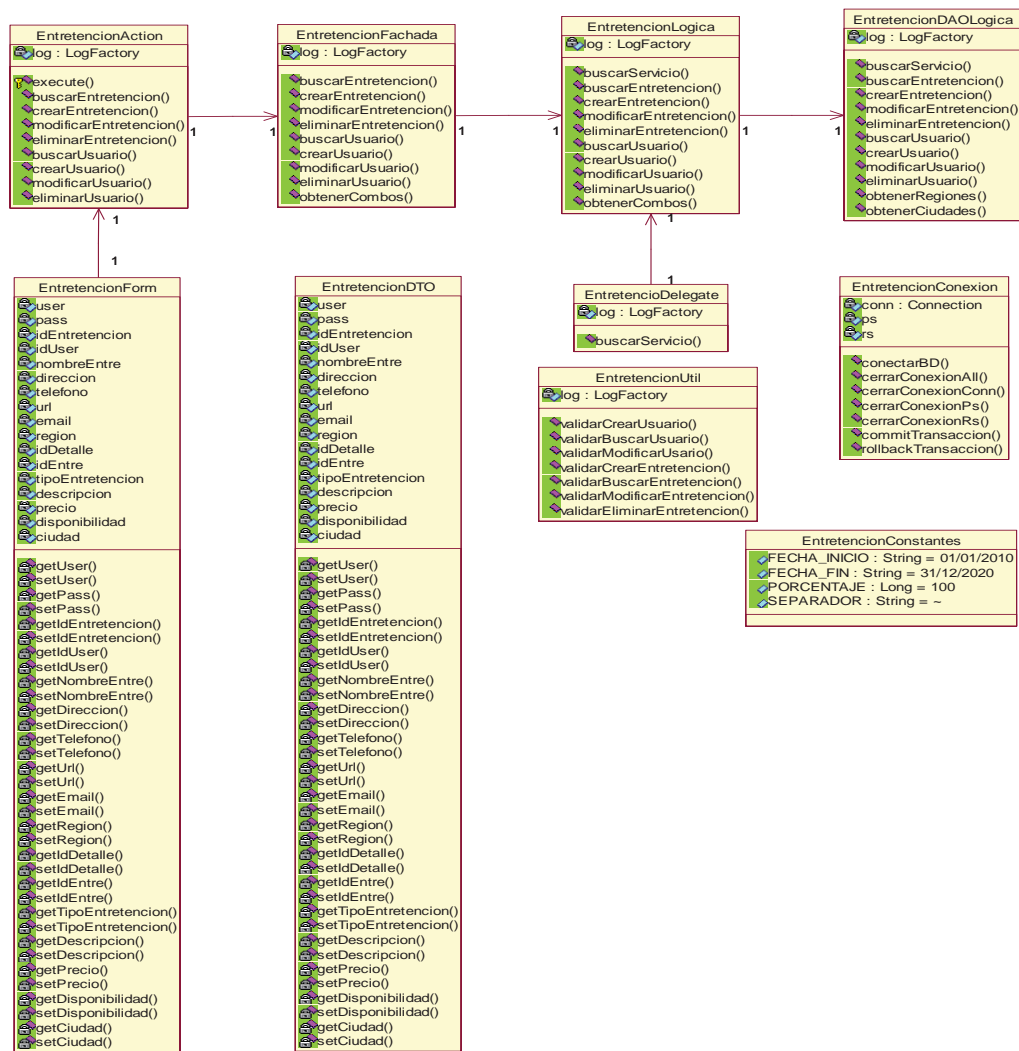
Diagramas y Diccionario de Datos

1. Introducción

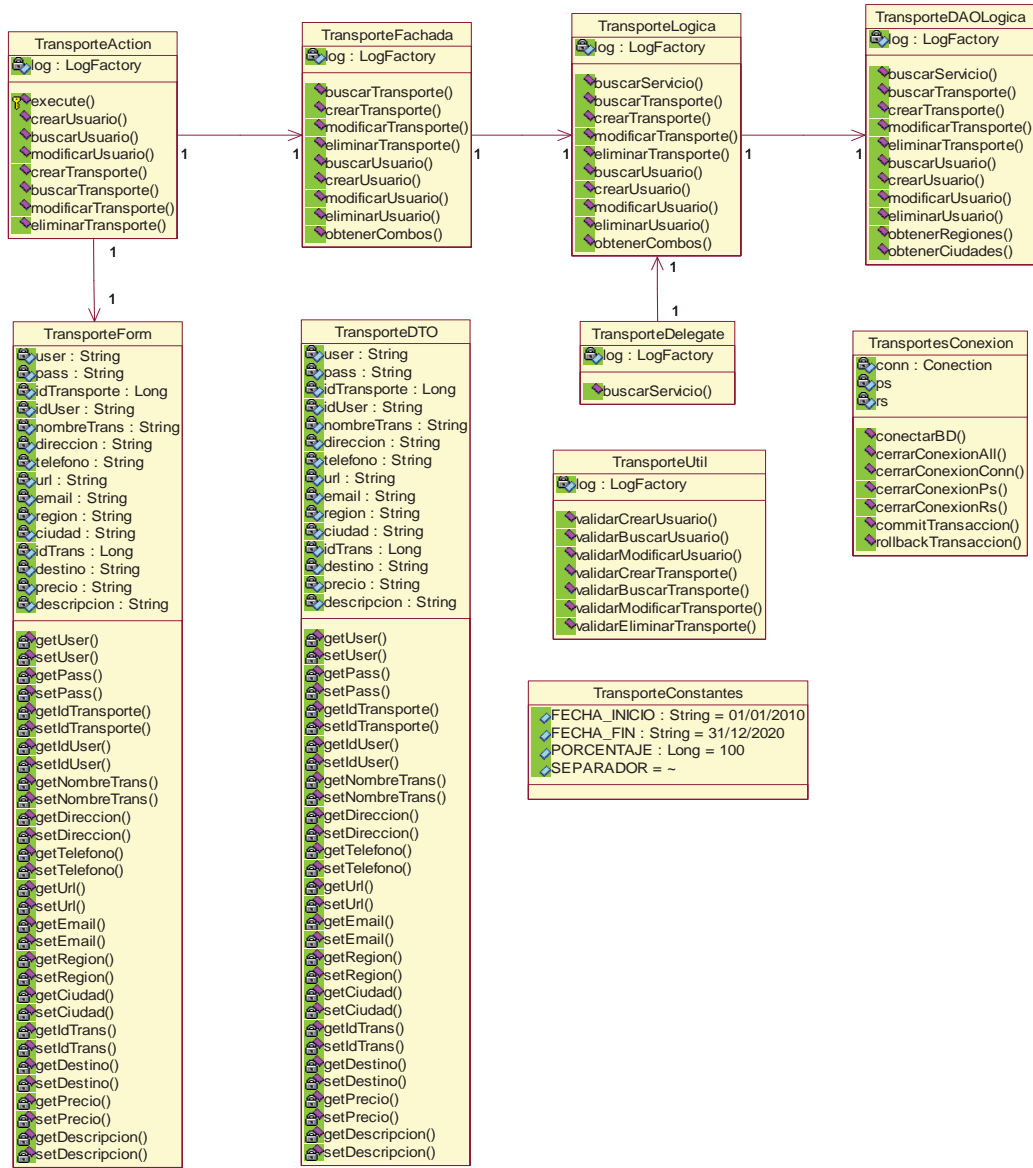
Esta sección consta de una serie de diagramas interesantes de documentar más el diccionario de datos de las bases de datos del Turista, Atracción, Alojamiento, Restauración y Transporte.

2. Diagramas

- Diagrama de Clases Entretención



▪ **Diagrama de Clases Transporte**



3. Base de Datos

- **Turista**

TABLA	Perfil	
BASE DE DATOS	Turista	
CAMPO	TIPO	DESCRIPCIÓN
User	varchar(12)	Seudónimo
Pass	varchar(12)	Contraseña Usuario
Nombre	varchar(30)	Nombre Usuario
Apellido	varchar(30)	Apellido Usuario
direccion	varchar(30)	Dirección Particular Usuario
email	varchar(30)	Dirección correo electrónico
telefono	integer	Numero teléfono fijo
movil	varchar(30)	Numero teléfono móvil
región	varchar(30)	Región de residencia
cuidad	varchar(30)	Ciudad de residencia

TABLA	Itinerario	
BASE DE DATOS	Turista	
CAMPO	TIPO	DESCRIPCIÓN
idItine	integer	Identificador Itinerario
idUser	varchar(12)	Identificador Usuario
nombreItinerio	varchar(30)	Nombre Itinerario
fechaIni	varchar(30)	Fecha Inicio Global del Itinerario
fechaFin	varchar(30)	Fecha Finalización del Itinerario

TABLA	Detalle	
BASE DE DATOS	Turista	
CAMPO	TIPO	DESCRIPCIÓN
idDetalle	integer	Identificador Detalle
idItine	varchar(12)	Identificador Itinerario
fechaInicio	varchar(30)	Fecha Inicio de la actividad del Itinerario
fechaTermino	varchar(30)	Fecha Finalización de la actividad del Itinerario
precio	varchar(30)	Precio de la actividad

TABLA	Servicio	
BASE DE DATOS	Turista	

CAMPO	TIPO	DESCRIPCIÓN
idServicio	integer	Identificador Servicio
idProducto	varchar(12)	Identificador Producto
idDetalle	integer	Identificador Detalle
nombreProducto	varchar(30)	Nombre de producto contratado
fechaInicio	varchar(30)	Fecha Inicio del Servicio
fechaTermino	varchar(30)	Fecha Fin Servicio
region	varchar(30)	Región prestación servicio
cuidad	varchar(30)	Ciudad prestación servicio

▪ Alojamiento

TABLA	Usuario	
BASE DE DATOS	Alojamiento	
CAMPO	TIPO	DESCRIPCIÓN
user	varchar(20)	Seudónimo Usuario Alojamiento
pass	varchar(20)	Contraseña Usuario

TABLA	Alojamiento	
BASE DE DATOS	Alojamiento	
CAMPO	TIPO	DESCRIPCIÓN
idAlojamiento	integer	Seudónimo Usuario Alojamiento
idUser	varchar(20)	Identificador Usuario
direccion	varchar(20)	Dirección Particular Servicio Alojamiento
telefono	varchar(20)	Teléfono Servicio Alojamiento
url	varchar(30)	Página Web Alojamiento
email	varchar(30)	Correo Electrónico Servicio Alojamiento
region	varchar(20)	Región Servicio Alojamiento
ciudad	varchar(20)	Ciudad Servicio Alojamiento

TABLA	Habitación	
BASE DE DATOS	Alojamiento	
CAMPO	TIPO	DESCRIPCIÓN
idHabitacion	integer	Identificador Habitación
idAloja	integer	Identificador Alojamiento

TABLA	Reserva	
BASE DE DATOS	Alojamiento	
CAMPO	TIPO	DESCRIPCIÓN

idReserva	integer	Identificador Reserva
idHabitacion	integer	Identificador Habitación
fechaInicio	varchar(20)	Fecha inicio reserva
fechaFin	varchar(20)	Fecha termino reserva

TABLA	Tipohabitacion	
BASE DE DATOS	Alojamiento	
CAMPO	TIPO	DESCRIPCIÓN
idTipoHab	integer	Identificador Tipo Habitación
idTipo	integer	Identificador Habitación
Precio	varchar(20)	Valor de la Habitación

- **Atracción**

TABLA	Usuario	
BASE DE DATOS	Atracción	
CAMPO	TIPO	DESCRIPCIÓN
user	varchar(20)	Seudónimo Usuario Atracción
pass	varchar(20)	Contraseña Usuario

TABLA	Atracción	
BASE DE DATOS	Atracción	
CAMPO	TIPO	DESCRIPCIÓN
idAtraccion	integer	Identificador Atracción
idUser	varchar(20)	Identificador Usuario
nombreAtracc	varchar(20)	Nombre descriptivo Atracción
direccion	varchar(20)	Dirección de Servicio Atracción
telefono	varchar(20)	Contacto telefónico Servicio Atracción
url	varchar(30)	Página Web Servicio Atracción
email	varchar(30)	Correo Electrónico Servicio Atracción
region	varchar(20)	Región Servicio Atracción

TABLA	Detalle	
BASE DE DATOS	Atracción	
CAMPO	TIPO	DESCRIPCIÓN
idDetalle	integer	Identificador Detalle
idAtraccion	integer	Identificador Atracción
Descripcion	varchar(90)	Descripción del Servicio de Atracción
Disponibilidad	integer	Valor booleano de 1 o 0

Ciudad	varchar(20)	Contacto telefónico Servicio Atracción
Precio	integer	Valor comercial del Servicio de Atracción

▪ **Restauración**

TABLA	Usuario	
BASE DE DATOS	Restauración	
CAMPO	TIPO	DESCRIPCIÓN
user	varchar(20)	Seudónimo Usuario Restauración
pass	varchar(20)	Contraseña Usuario

TABLA	Restauración	
BASE DE DATOS	Restauración	
CAMPO	TIPO	DESCRIPCIÓN
idRest	integer	Identificador Restauración
idUser	varchar(20)	Identificador Usuario
nombreRest	varchar(20)	Nombre descriptivo Restauración
direccion	varchar(20)	Dirección de Servicio Restauración
telefono	varchar(20)	Contacto telefónico Servicio Restauración
url	varchar(30)	Página Web Servicio Restauración
email	varchar(30)	Correo Electrónico Servicio Restauración
region	varchar(20)	Región Servicio Restauración
Ciudad	varchar(20)	Ciudad Servicio Restauración

TABLA	Detalle	
BASE DE DATOS	Restauración	
CAMPO	TIPO	DESCRIPCIÓN
idDetalle	integer	Identificador Detalle
idRest	integer	Identificador Restauración
Descripcion	varchar(90)	Descripción del Servicio de Restauración
Disponibilidad	integer	Valor booleano de 1 o 0
Precio	integer	Valor comercial del Servicio de Restauración

▪ **Transporte**

TABLA	Transporte	
BASE DE DATOS	Transporte	

CAMPO	TIPO	DESCRIPCIÓN
idTransporte	integer	Identificador Transporte
idUser	varchar(20)	Identificador Usuario
nombreTrans	varchar(20)	Nombre descriptivo Transporte
direccion	varchar(20)	Dirección de Servicio Transporte
telefono	varchar(20)	Contacto telefónico Servicio Transporte
url	varchar(30)	Página Web Servicio Transporte
email	varchar(30)	Correo Electrónico Servicio Transporte
region	varchar(20)	Región Servicio Transporte
Ciudad	varchar(20)	Ciudad Servicio Transporte

TABLA	Bus	
BASE DE DATOS	Transporte	
CAMPO	TIPO	DESCRIPCIÓN
idBus	integer	Identificador Tabla Bus
idTrans	integer	Identificador Tabla Transporte

TABLA	Tramo	
BASE DE DATOS	Transporte	
CAMPO	TIPO	DESCRIPCIÓN
idTramo	integer	Identificador Tabla Tramo
idBus	integer	Identificador Tabla Bus
Inicio	varchar(20)	Lugar Inicio
Destino	varchar(20)	Lugar Destino

TABLA	Precio	
BASE DE DATOS	Transporte	
CAMPO	TIPO	DESCRIPCIÓN
idPrecio	integer	Identificador Tabla Precio
idTramo	integer	Identificador Tabla Bus
Precio	varchar(20)	Valor del Viaje
temporadaDescrip	varchar(20)	Descripción de la Temporada