

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

**IMPLEMENTACIÓN DE UN APLICATIVO PARA TELÉFONOS
MÓVILES QUE INDIQUE LAS RUTAS DE TRANSPORTE
PÚBLICO DE LA CIUDAD DE LIMA A PARTIR DE LA
UBICACIÓN DEL USUARIO**

Tesis para optar el Título de Ingeniera de las Telecomunicaciones, que
presenta la bachillera:

Angie Monique Añazgo La Rosa

ASESOR: Ing. Enrique Larios Vargas

Lima, mayo de 2010

Resumen

El presente proyecto de tesis consiste en la implementación un aplicativo para teléfonos móviles que pueda listar las rutas de transporte público urbano de la ciudad de Lima Metropolitana tomando como datos de entrada la información de la ubicación del usuario.

Para ello la tesis se ha dividido en cinco capítulos que abarcan los aspectos teóricos y prácticos de la implementación del sistema el cual está compuesto por un servidor de aplicaciones, un servidor de base de datos y la aplicación móvil.

El primer capítulo se enfoca en el estudio de las tecnologías relacionadas al desarrollo de las aplicaciones móviles en general, asimismo se detallan las características de las herramientas disponibles para el desarrollo del proyecto.

El segundo capítulo busca presentar el análisis de la solución realizado a partir de los conceptos y herramientas presentadas en el capítulo anterior, además de ello se explica las características de la propuesta de solución.

El tercer capítulo describe el diseño de la solución el cual incluye la información de la arquitectura del sistema, el flujo de datos, la descripción de la base de datos a utilizar y finalmente las interfaces de la aplicación.

El cuarto capítulo, por su parte, explica la implementación realizada para ello se detalla tanto la estructura de la aplicación como las configuraciones realizadas en el sistema.

El quinto capítulo expone las pruebas del prototipo, las herramientas utilizadas y los resultados obtenidos durante ese proceso.

Por último, se presentan las conclusiones y recomendaciones del trabajo, además de proponer algunos trabajos futuros que permitan la integración de este sistema con nuevas herramientas de telecomunicaciones que complementen el trabajo presentado y le añadan nuevas características de acuerdo a los nuevos requerimientos de los usuarios.

Dedicatoria

A mi papá Víctor Añazgo,

A mi madre Betty La Rosa,

A mi hermana Kristel Añazgo,

A mi hermanita Antonella Añazgo,

A mis maestros por su apoyo constante,

A mis amigos por sus consejos y correcciones,

Y a usted lector por brindarme parte de su tiempo.

Agradecimientos

Agradezco profundamente a Dios por darme la oportunidad de vivir, aprender cada día algo nuevo, conocer a personas maravillosas y disfrutar de su compañía.

A mis padres por apoyarme en los proyectos que emprendo y orientarme con su experiencia, a mis hermanas por llenar de sincera alegría mi vida desde su nacimiento.

Al ingeniero Enrique Larios, mi asesor, por creer en este proyecto, apoyarme con sus conocimientos y enseñarme con sus correcciones.

Al doctor Carlos Silva por su amable ayuda con la gestión de la información de la Municipalidad Metropolitana de Lima.

A Oscar Yépez por confiar en mis ideas, motivarme en los momentos difíciles y sobre todo por compartir conmigo su cariño.

A mis profesores por hacer llegar sus enseñanzas y conocimientos dentro de un ambiente cálido y amical.

A mis compañeros de promoción, en ellos pude encontrar ideas muy valiosas que guardo en mi memoria. Deseo agradecer especialmente a Diego Narváez por darme luces y orientación en los aspectos técnicos, a Juan Carlos Llap por su amistad y sus consejos y a Elga Esteves por su apoyo con la revisión de los últimos detalles de mi trabajo.

Índice

Índice	v
Lista de Figuras	vii
Lista de Tablas	viii
Introducción	9
Capítulo 1 Estudio de las tecnologías involucradas para el desarrollo de aplicaciones vía internet móvil	10
1.1 Redes de acceso para internet móvil	10
1.1.1 WLAN	10
1.1.2 GPRS.....	11
1.1.3 Comparación	13
1.2 Herramientas de desarrollo	13
1.2.1 Lenguaje XML.....	13
1.2.2 Arquitectura orientada a servicios (SOA).....	14
1.2.2.1 Simple Object Access Protocol (SOAP)	14
1.2.2.2 Web Service Description (WSDL).....	15
1.2.2.3 Universal Description, Discovery and Integration (UDDI).....	15
1.2.3 Lenguajes de Programación	16
1.2.3.1 Java.....	16
1.2.3.2 C++.....	16
1.2.4 Servidores de Aplicaciones	16
1.2.4.1 GlassFish.....	16
1.2.4.2 Tomcat.....	17
Capítulo 2 Análisis de la Solución	19
2.1 Situación Actual	19
2.2 Propuesta de solución	20
2.3 Análisis de Requerimientos	20
2.4 Alcances y Limitaciones del Proyecto	21
2.5 Evaluación de Herramientas	22
2.5.1 Red de Servicios de Internet.....	22
2.5.2 Lenguaje de programación	23
2.5.2.1 JSR 172.....	23
Capítulo 3 Diseño de la Solución	25
3.1 Arquitectura del Sistema	25
3.2 Flujo del sistema	26
3.3 Base de Datos del Sistema.....	27
3.3.1 Empresa.....	29
3.3.2 Bus	29
3.3.3 Viaje	30
3.3.4 Ruta.....	31
3.3.5 Trayecto	32
3.3.6 Tramo	32
3.3.7 Distrito	33
3.4 Estándares del Sistema	34
3.5 Interfaces de la Aplicación Móvil	34
3.5.1 Inicio de la aplicación	34
3.5.2 Ingreso de datos para la Búsqueda de Rutas	36
3.5.3 Muestra de resultados de la Búsqueda de Rutas	37

3.5.4	Ingreso de datos para obtener Información de Empresas ...	39
3.5.5	Muestra de resultados de los Datos de Empresas	40
3.5.6	Información del Modo de Uso - Ayuda.....	41
Capítulo 4 Implementación de la Solución		43
4.1	Estructura de la Aplicación	43
4.1.1	Sistema Servidor	43
4.1.1.1	Diccionario de Clases del Sistema.....	44
4.1.1.2	Diccionario de Operaciones del Web Service	53
4.1.2	Servidor de Base de Datos	54
4.1.3	Aplicación web móvil.....	54
4.1.3.1	Diccionario de clases de la aplicación web móvil	55
4.1.4	Equipo Terminal	56
4.2	Configuraciones del Sistema	56
4.2.1	Base de Datos	56
4.2.2	Servidor de Aplicaciones web	57
4.2.3	Servicio Web.....	57
Capítulo 5 Pruebas del Prototipo		59
5.1	Introducción.....	59
5.2	Emulación del Sistema en la Herramienta J2ME Wireless Toolkit 2.5.2.	60
5.3	Ejecución del Aplicativo en el Terminal Móvil	62
5.4	Uso del CPU y Memoria	64
5.4.1	Uso del CPU y Memoria para el cliente Emulador de J2ME Wireless Toolkit 2.5.2.....	65
5.4.3	Análisis de Tramas	67
Conclusiones, Recomendaciones y Trabajos Futuros.....		69
6.1	Recomendaciones.....	69
6.2	Conclusiones.....	70
6.3	Trabajos Futuros	71
Bibliografía		72
Anexos.....		74

Lista de Figuras

FIGURA 1-1: RED GPRS	12
FIGURA 1-2: ESQUEMA DE CAPAS GPRS	12
FIGURA 1-3: ESTRUCTURA DE EMPAQUETADO SOAP	15
FIGURA 1-4: PILA DE LOS SERVICIOS WEB.....	18
FIGURA 3-1: ESQUEMA GENERAL DE LA ARQUITECTURA DE LA SOLUCIÓN	26
FIGURA 3-2: DIAGRAMA DE FLUJO DEL SERVICIO.....	27
FIGURA 3-3: DIAGRAMA DE MODELO ENTIDAD RELACIÓN.....	28
FIGURA 3-4: INTERFAZ DE INICIO DE LA APLICACIÓN.....	35
FIGURA 3-5: MENÚ DE INICIO DE LA APLICACIÓN.....	36
FIGURA 3-6: INTERFAZ DE INGRESO DE DATOS AL SISTEMA – CONSULTA DE RUTAS	37
FIGURA 3-7: SALIDA DE LOS DATOS – RESPUESTA A LA CONSULTA DE TRAMOS Y VIAJES.....	38
FIGURA 3-8: SALIDA DE LOS DATOS – RESPUESTA A LA CONSULTA DE TRAYECTOS.....	39
FIGURA 3-9: INTERFAZ DE INGRESO DE DATOS AL SISTEMA – DATOS DE EMPRESAS	40
FIGURA 3-10: SALIDA DE LOS DATOS – RESPUESTA A LA CONSULTA DE DATOS DE EMPRESAS.....	41
FIGURA 3-11: MENÚ MODO DE USO - AYUDA	42
FIGURA 5-1. FLUJO DE EJECUCIÓN DE LA APLICACIÓN EN J2ME WIRELESS TOOLKIT 2.5.2	61
FIGURA 5-2. FLUJO DE EJECUCIÓN DE LA APLICACIÓN EN EL TERMINAL MÓVIL NOKIA 5800 XPRESSMUSIC.....	63
FIGURA 5-3: PARÁMETROS DE USO DE RECURSOS SIN SERVIDOR GLASSFISH.....	65
FIGURA 5-4: USO DEL CPU Y MEMORIA RAM SIN SERVIDOR GLASSFISH.....	65
FIGURA 5-5: PARÁMETROS DE USO DE RECURSOS CON SERVIDOR GLASSFISH Y EMULADOR	66
FIGURA 5-6: USO DEL CPU Y MEMORIA RAM CON SERVIDOR GLASSFISH Y EMULADOR	66
FIGURA 5-7: PARÁMETROS DE USO DE RECURSOS CON SERVIDOR GLASSFISH Y EL TERMINAL.....	67
FIGURA 5-8: USO DEL CPU Y MEMORIA RAM CON SERVIDOR GLASSFISH Y EL TERMINAL.....	67
FIGURA 5-9: TRAMAS CAPTURADAS DURANTE LA EJECUCIÓN DE LA APLICACIÓN EN EL TERMINAL.....	68
FIGURA 5-10: EMPAQUETADO SOAP CAPTURADO DURANTE LA EJECUCIÓN DE LA APLICACIÓN EN EL TERMINAL.....	68

Lista de Tablas

TABLA 1-1: ACCESO A REDES DE INTERNET MÓVIL.....	13
TABLA 2-1: COMPARACIÓN DE LAS TECNOLOGÍAS DE ACCESO A RED INALÁMBRICAS	22
TABLA 2-2: COMPARACIÓN DEL DESEMPEÑO ENTRE GLASSFISH Y TOMCAT.....	24
TABLA 3-1: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA EMPRESA	29
TABLA 3-2: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA BUS	29
TABLA 3-3: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA VIAJE	30
TABLA 3-4: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA RUTA	31
TABLA 3-5: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA TRAYECTO	32
TABLA 3-6: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA TRAMO...33	
TABLA 3-7: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA DISTRITO	33
TABLA 4-1: CLASE BBUS.....	44
TABLA 4-2: CLASE BDISTRITO.....	45
TABLA 4-3: CLASE BEMPRESA.....	45
TABLA 4-4: CLASE BRUTA.....	45
TABLA 4-5: CLASE BTRAMO	47
TABLA 4-6: CLASE BTRAYECTO.....	49
TABLA 4-7: CLASE BVIAJE	50
TABLA 4-8: CLASE DPRIMERO	52
TABLA 4-9: DICCIONARIO DE OPERACIONES	53
TABLA 4-10: DICCIONARIO DE CLASES DE LA APLICACIÓN WEB MÓVIL	55
TABLA 4-11: INFORMACIÓN DEL EQUIPO TERMINAL.....	56
TABLA 4-12: INFORMACIÓN DE CONFIGURACIÓN DE LA BASE DE DATOS	56
TABLA 4-13: INFORMACIÓN DE CONFIGURACIÓN DEL SERVIDOR DE APLICACIONES	57

Introducción

Desde la aparición masiva de los autobuses a mediados del siglo XX se presentaron problemas en el ordenamiento de las unidades que brindaban ese servicio los cuales se incrementaron con el pasar del tiempo y el aumento desmesurado de los usuarios y la planta de buses.

Actualmente grandes países como Estados Unidos y la mayor parte de Europa tienen a su disposición herramientas tecnológicas para los usuarios quienes pueden conocer los detalles de los servicios de transporte de sus ciudades, muchos de ellos les ofrecen ubicación de paraderos, horarios, rutas, personalización de sus viajes vía web, etc. Estas herramientas van de la mano con la inversión económica del Estado quien es el promotor y fiscalizador de los proyectos que tienen como finalidad brindar un servicio integral y de la mejor calidad a sus pasajeros.

La otra cara de la moneda se observa en países en vía de desarrollo como en el caso del Perú en donde no se ha planificado adecuadamente el crecimiento del servicio de transporte urbano. Como consecuencia de la poca organización y apoyo a este rubro se presentan problemas en la calidad del servicio tanto a nivel físico: mal estado y antigüedad de sus unidades, semaforización deficiente, deterioro de las pistas y paraderos; como a nivel de soporte: pasajeros desinformados de las rutas, horarios y paraderos autorizados, informalidad en el desarrollo de la actividad, falta supervisión y capacitaciones a usuarios y conductores, entre otros.

Capítulo 1

Estudio de las tecnologías involucradas para el desarrollo de aplicaciones vía internet móvil

En el presente capítulo se describirán los conceptos que serán utilizados en el diseño de la solución, considerando las herramientas de desarrollo y los protocolos utilizados en el manejo de los datos.

1.1 Redes de acceso para internet móvil

El ingreso al sistema se realiza a través de Internet para ello se utiliza una tecnología de acceso al medio inalámbrico la cual no es relevante para la aplicación, la red que se utilice determinará las características de velocidad de transmisión de datos, cobertura y rendimiento.

El acceso a internet puede darse a través de las redes móviles que se describen a continuación:

1.1.1 WLAN

Es un sistema de comunicación de datos que puede concebirse como una extensión de la red LAN cableada dentro de una determinada zona de cobertura.

Para su funcionamiento hace uso de ondas electromagnéticas de alta frecuencia que viajan a través del aire. La popularidad de estas redes se ha visto incrementada con el tiempo a tal punto que se les reconoce como alternativas de conectividad para múltiples aplicaciones.

Los usuarios finales o nodos móviles son aquellos que están provistos de adaptadores WLAN con los cuales pueden acceder a la red.

El acceso a la red cableada utiliza un dispositivo transmisor-receptor conocido como Access Point (AP) el cual almacena, envía y transmite información a través de los nodos. El Access Point puede trabajar adecuadamente solo para un número pequeño de usuarios, asimismo el área en el cual se utiliza está limitada por algunos cuantos metros.

Por otro lado, existe un tipo de red conocido como Ad-hoc que permite la comunicación directa entre usuarios finales sin usar un AP como intermedio lo cual conlleva a problemas con la performance frente a la presencia de grandes cargas de datos, usualmente se emplea para aplicaciones donde la infraestructura no está disponible.

1.1.2 GPRS

Es una red que ofrece conectividad a nivel IP, adicionalmente a las características de una red IP convencional. Como se puede observar en la Figura 1-1, mediante el uso de recursos de radio, los usuarios móviles pueden acceder a un Packet Data Network (PDN) mediante un router remoto llamado Gateway GPRS Support Node (GGSN) [SAL2004 , ZUI2002].

GPRS ofrece acceso transparente y no transparente al PDN, en el primer caso el usuario no requiere autenticación y se le asigna la IP desde un espacio de direcciones de la red GPRS, en el segundo caso, el usuario envía credenciales a la PDN de manera que se le permite al usuario acceder a la PDN solo si se cumple adecuadamente la autenticación. Cabe resaltar, que el usuario siempre se autentica a la red GPRS antes de ingresar a ella sea cual sea el modo de acceso [SAL2004].

En la Figura 1-2 se muestra los detalles de la red GPRS, se puede ubicar el Mobile Station (MS), Gateway GPRS Serving Node (GGSN). Es importante conocer que GGSN brinda la funcionalidad del enrutamiento el cual se utiliza como interfaz hacia redes externas que están relacionadas a la PDN.

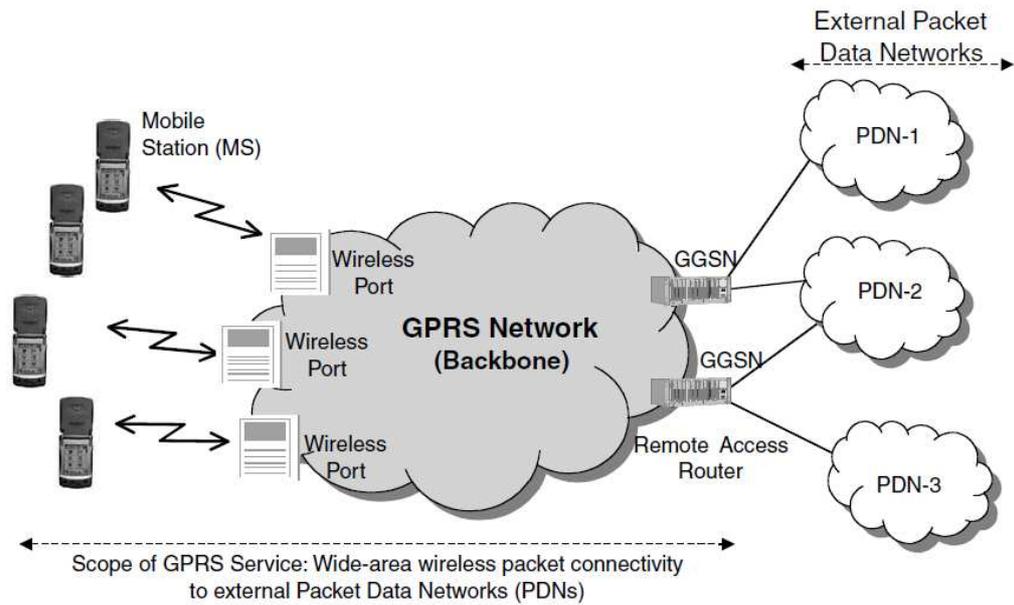


FIGURA 1-1: RED GPRS

Fuente: "High-level conceptual view of a GPRS network" [SAL2004]

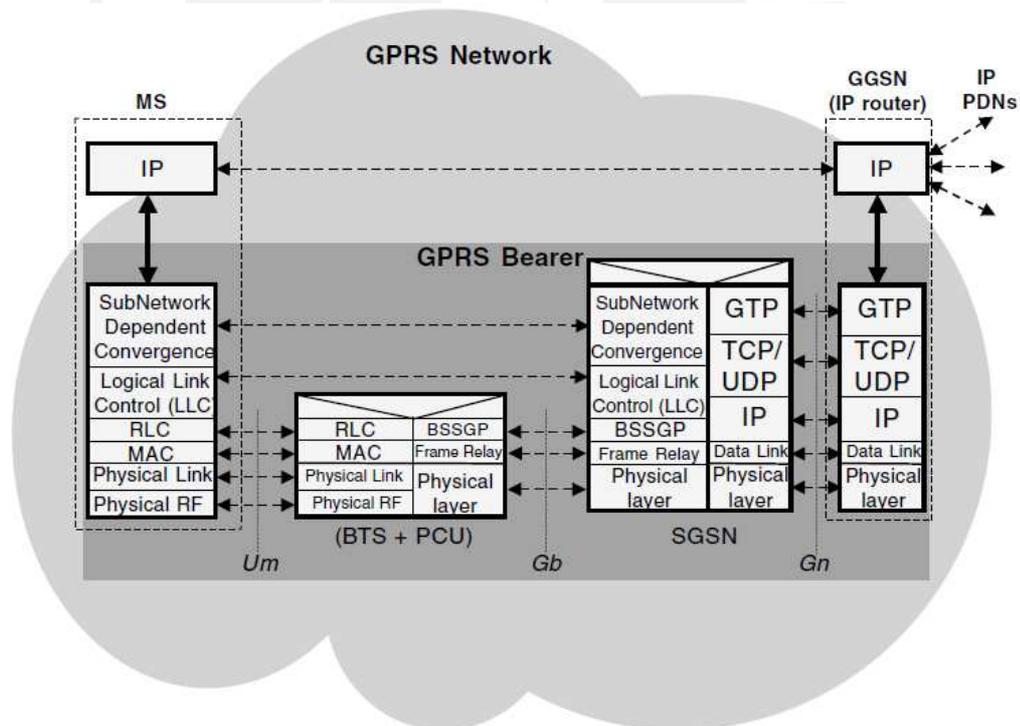


FIGURA 1-2: ESQUEMA DE CAPAS GPRS

Fuente: "The GPRS Bearer Service" [SAL2004]

1.1.3 Comparación

En la Tabla 1-1 se hace una comparación entre los tres tipos de acceso explicados anteriormente

TABLA 1-1: ACCESO A REDES DE INTERNET MÓVIL

Modo	Funcionamiento	Características
WLAN	Trabaja en base a nodos con tarjetas inalámbricas.	Requiere una distancia mínima para que los nodos puedan estar en funcionamiento.
GPRS	Utiliza la red de GSM a la cual le añade las características de conmutación de paquetes.	Requiere que los equipos de la red actualicen su software para poder soportar GPRS.

1.2 Herramientas de desarrollo

En la presente sección se describirán los protocolos que se encargan del manejo de datos enviados y de las estructuras de la información.

1.2.1 Lenguaje XML

Es un lenguaje de marcas ofrece especificaciones que definen las reglas y sintaxis para crear marcas. Los lenguajes de marcas indican el tratamiento que se le debe dar a su contenido, un ejemplo de ello es HTML.

XML consta de dos unidades básicas: elementos y atributos. Un elemento está dado por una marca de entrada, la información y la marca final.

Adicionalmente, los documentos XML pueden guardar datos en atributos los cuales están dados por variables asignadas a un valor.

El esquema XML puede manejar los tipos de datos de cada elemento y atributo, asimismo XML hace uso de etiquetas para categorizar y organizar la información de una forma específica [HIR2006, CHA2002].

1.2.2 Arquitectura orientada a servicios (SOA)

Describe un entorno donde las aplicaciones de software muestran sus funcionalidades mediante una interface de mensajes del proveedor de servicios. Por su parte, otros agentes de software pueden actuar como consumidores usando los servicios del proveedor de servicios. SOA cuenta con dos entidades principales la aplicación cliente, que solicita el servicio, y la aplicación servidora que lo provee. Asimismo, SOA permite la creación de procesos complejos basados en otros más simples.

SOA está compuesto por:

1. Descubrimiento: Antes de buscar contactarse con el proveedor para obtener servicios debe localizar el proveedor apropiado. Existen diversas técnicas de descubrimiento que serán estudiadas más adelante.
2. Descripción: Luego de ubicar al proveedor, el cliente necesita aprender las reglas necesarias para contactar el proveedor como por ejemplo los protocolos que van a usarse, la estructura de los mensajes, el modo de seguridad entre otros. La información de descripción puede ser retornada como una parte del proceso de descubrimiento, normalmente incluye información estructurada en forma de Web Service Description Language (WSDL). WSDL es un lenguaje XML usado para describir mensajes finales e indica con quien deben ser comunicados. Asimismo, para este aspecto es también de utilidad el Universal Description Discovery and Integration (UDDI). La mayor ventaja de WSDL es el habilitar la automatización en el establecimiento de la comunicación entre dos puntos finales de la red.
3. Mensajes: Permite comunicarse con el proveedor por medio de mensajes. Se suele utilizar el protocolo SOAP en conjunto con HTTP [HIR2006].

1.2.2.1 Simple Object Access Protocol (SOAP)

Este protocolo proporciona una forma de empaquetar la información en sobres para que puedan cambiar de plataforma entre distintos servicios.

Una vez que la información viene de la capa de aplicación es transferida a los demás dispositivos. Los mensajes entre la capa de presentación y la capa de aplicación serán intercambiados usando XML.

Es un protocolo destinado al intercambio de información estructurada, transporta carga útil en el cuerpo del mensaje y el control de la información en la cabecera. Se

encarga de definir la manera cómo se enviarán los datos, las reglas de codificación y representación de peticiones y respuestas.

En la Figura 1-3 se observa la estructura general de un empaquetado SOAP, las cabeceras y el cuerpo.



FIGURA 1-3: ESTRUCTURA DE EMPAQUETADO SOAP

Fuente: "SOAP Envelope Structure" [HIR2006]

Con respecto a las cabeceras, los componentes de la cabecera SOAP pueden ser añadidos o removidos de esa posición lo cual implica una gran ventaja.

La aplicación debe estructurar la carga del mensaje, normalmente se utiliza XML para este proceso.

1.2.2.2 Web Service Description (WSDL)

WSDL es un protocolo que le permite al cliente del servicio web ser configurado para interactuar con un servicio sin la necesidad de un software especializado para dicha configuración, para ello coordina con el servicio web las características del transporte y los datos que serán transmitidos [HIR2006].

1.2.2.3 Universal Description, Discovery and Integration (UDDI)

Es un conjunto de especificaciones que se encarga de definir los métodos de almacenamiento, localización y gestión del servicio de información relacionada con los servicios web.

Sus características de informante permiten realizar el descubrimiento de la información puesto que maneja los datos de la información técnica y del proveedor de servicios. Proporciona la estructura para representar empresas, relaciones comerciales, servicios web y puntos de acceso. Es un servicio de directorio para la ubicación de servicios por criterios [HIR2006].

1.2.3 Lenguajes de Programación

A continuación se describirán los principales lenguajes de programación que permiten el desarrollo de aplicaciones web:

1.2.3.1 Java

Es un lenguaje de programación que se caracteriza fundamentalmente por estar basado en clases y ser orientado a objetos, características que en adición a su simplicidad ha extendido su uso a nivel mundial.

El ser un lenguaje multiplataforma le hace posible el desarrollo de aplicaciones en diversos entornos entre ellos los dispositivos con recursos limitados para los que se utiliza la plataforma J2ME – Java Micro Edition [GOS2005].

1.2.3.2 C++

C++ es un lenguaje de programación que trabaja con clases y objetos al igual que el lenguaje Java, pero se diferencia de éste en:

- El control de plataforma, en C++ el controlador es manejado por el programador.
- El manejo de la memoria: la liberación de memoria en C++ recae en el programador.
- Uso de clases y objetos: El uso de objetos es facultativo.

1.2.4 Servidores de Aplicaciones

El servidor de aplicaciones cumplirá la tarea del manejo de las operaciones del servicio, se describen los siguientes:

1.2.4.1 GlassFish

Es un servidor de aplicaciones de código abierto creado por la Comunidad GlassFish lanzada por Sun Microsystems en el año 2005. Se caracteriza por una arquitectura modular que le brinda rapidez al desarrollo web así como por soportar lenguajes dinámicos. Su modularidad le proporciona ligereza y rapidez al actualizar

los módulos de la aplicación lo que conlleva al ahorro de memoria y disco [SUN2008].

1.2.4.2 Tomcat

Es una aplicación de código abierto de las tecnologías Java Servlet y JavaServer Pages desarrolladas según el Java Community Process por la Fundación Apache. Sus características lo han convertido en el servidor web más popular a nivel mundial [APA2009].

Tomcat hace posible que el código sea ejecutado con el servidor web que está comprendido en él. Su operación es independiente de la plataforma operativa en la que esté trabajando lo que implica una ventaja pues solo es necesaria la máquina virtual [GOM2006].

Las herramientas explicadas previamente serán utilizadas en el análisis y modelado del diseño el que se describirá con mayor detalle en el siguiente capítulo.

1.2.5 Servicio Web

Según el W3C, un Servicio Web es un software diseñado para soportar la interoperabilidad máquina-a-máquina en una red. Su interfaz está descrita en un lenguaje procesable por el computador y se comunica mediante mensajes lo cual significa una ventaja con respecto a otras Remote Procedure Call (RPCs) para Java.

Las ventajas de los Servicios Web radican en que facilita la interoperabilidad entre aplicaciones en plataformas diferentes independientemente del lenguaje de programación, utilizan estándares y protocolos abiertos, permiten reusar distintos componentes, al usar HTTP se puede emplear a través de firewalls sin cambiar sus reglas de filtrado [GOM2006].

La pila de los Servicios Web se compone de tres pilares Seguridad, Gestión y Calidad de Servicio tal como se muestra en la Figura 1-4

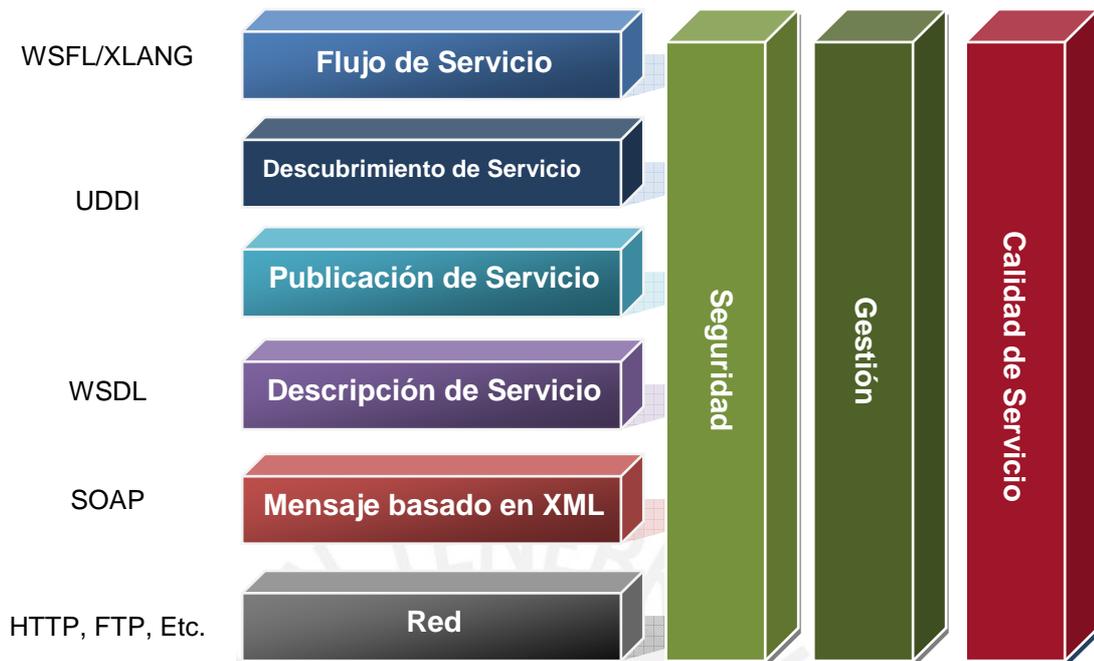


FIGURA 1-4: PILA DE LOS SERVICIOS WEB

Fuente: "The Web Service Stack" [CAS2008]

Existen tres tipos de web services XML-RPC, SOAP y REST.

XML Remote Procedure Call (XML-RPC) provee un método para llamar a los métodos/procedimientos en un servidor remoto y hacer cambios o recuperar datos, tiene como sucesor a SOAP.

Por su parte, la especificación SOAP es mantenida en W3 y provee un mecanismo para varios patrones de mensajería, los mensajes son enviados en una envoltura SOAP que es un contenedor XML para lectura de datos y es generado por el servidor SOAP, el patrón de mensaje más común es el de Llamada a Procedimiento Remoto (RPC). SOAP proporciona un medio para interactuar con un sistema remoto mediante el envío de comandos y la obtención de una respuesta.

Representational State Transfer (REST) se centra en la diversidad de los recursos, no en las acciones, considera que cada recurso es direccionable únicamente, por otro lado, todos los recursos comparten la misma interfaz limitada para la transferencia del estado (acciones), provee un único identificador de recurso para toda la información en el sistema [RAM2006].

Capítulo 2

Análisis de la Solución

En este capítulo se expondrá la situación actual, los alcances del proyecto, las herramientas a utilizar y la propuesta de solución que dará origen al diseño de la misma.

2.1 Situación Actual

Desde hace más de dos décadas la ciudad de Lima experimenta grandes volúmenes de tránsito de vehículos de transporte público diariamente, las 25 mil unidades agrupadas en 444 líneas de transporte no cuentan hasta el momento con una sistematización de fácil acceso al público lo cual dificulta a los usuarios el poder conocer los detalles exactos de la trayectoria de las unidades.

El usuario del sistema de transporte que desea conocer las posibilidades que tiene para movilizarse tiene escasas herramientas de búsqueda de información confiable y actualizada es por ello que suele consultar a las personas que transitan en la vía pública o a los trabajadores de las unidades en circulación.

Como se explicó anteriormente, la poca difusión de éstos recorridos ocasiona lentitud en el flujo y la gestión del transporte, es por ello que los trabajadores de las

unidades se ven en la necesidad de pronunciar en voz alta el recorrido de su ruta causando ruidos excesivos en las avenidas con gran afluencia.

2.2 Propuesta de solución

La solución se enfoca en resolver el problema ocasionado por la congestión de personas en los paraderos de buses, caos vehicular y uso disfuncional del transporte urbano.

Esta solución propone informar a los usuarios de manera rápida y sencilla a través de un teléfono móvil de modo que tanto el tránsito de peatones y la circulación de los buses en los paraderos sea más eficiente.

El modelo de solución tiene precedentes en ciudades con un nivel de congestión vehicular equiparable a la de Lima como es el caso de Barcelona con su solución TMB - Transports Metropolitans de Barcelona o el caso de varias ciudades de Estados Unidos con la solución Google Transit, ambos servicios brindan valores agregados como la posibilidad de planificar sus viajes, ver la ubicación actual, ingresar al listado de cines y teatros, realizar búsqueda de paraderos, entre otros.

La idea principal de la propuesta es unificar el acceso a la información sobre los servicios de transporte urbano y en un futuro permitir que el usuario pueda tener en su teléfono móvil las herramientas necesarias para planificar ordenadamente sus viajes ahorrando tiempo y dinero.

2.3 Análisis de Requerimientos

Como paso previo a la elección de las herramientas y tecnologías más adecuadas se definen los requerimientos y características que el sistema debe cumplir para satisfacer las necesidades, luego de ello y en adición a los conceptos de las tecnologías y herramientas de desarrollo mostrados previamente se decidirá cuales son las opciones a utilizar.

En aplicaciones móviles destinadas a muchos usuarios es de vital importancia que la solución sea compatible con todos los modelos y marcas de teléfonos móviles, del mismo modo que su funcionamiento ocupe poca memoria, sea escalable y sencilla de implementar.

Tomando las consideraciones que muestra Pashtan [PAS2005] sobre las limitaciones de las estaciones móviles y desarrollo de sistemas se considera:

- La interface entre el usuario y el sistema debe ser sencilla y de manejo intuitivo para lo cual se tiene presente las limitaciones de pantalla y de teclado del móvil.
- El prototipo que se desarrollará debe tener la capacidad de realizar la conexión con el servidor, transmitir la petición y recibir la respuesta por parte del mismo.
- La base de datos debe incluir tablas que manejen los campos: identificador de la ruta, nombre de la empresa de transportes, número de unidades, distrito inicial y distrito final.
- La actualización de la base de datos debe estar en coordinación con la Municipalidad Metropolitana de Lima – Gerencia de Transporte Urbano, entidad que maneja los datos de las rutas de transporte público, número de unidades de cada empresa de transportes, entre otros datos.
- El lenguaje de programación debe ser flexible y robusto en el manejo de aplicaciones orientadas a móviles.
- La capacidad de procesamiento de las consultas, de almacenamiento de datos y capacidad de manejo de conexiones que debe cumplir el servidor deberán satisfacer las consultas realizadas por el prototipo.

2.4 Alcances y Limitaciones del Proyecto

- El proyecto será un prototipo de una aplicación que indicará en forma de texto las rutas de transporte público de la ciudad de Lima a partir de la ubicación del usuario el cual se conectará al servidor para hacer la búsqueda correspondiente.
- Las búsquedas se realizarán a partir de la búsqueda de Tramos que coincida con los datos de origen y destino, el número de máximo de tramos que satisfagan esta condición será igual a dos, luego de la búsqueda de Tramos se realiza la búsqueda por Viajes cuyo número máximo a incluir será tres, finalmente se realiza la búsqueda por Trayecto cuyo número máximo será igual a cinco.
- Las rutas de transporte público urbano cambian constantemente debido a las modificaciones en las vías por lo cual se requiere una constante

actualización de la información obtenida por parte de la Gerencia de Transporte Urbano de la Municipalidad Metropolitana de Lima.

2.5 Evaluación de Herramientas

En la presente sección se elegirá las tecnologías y herramientas a utilizar que permitan el desarrollo del prototipo tomando como base los requerimientos listados anteriormente.

2.5.1 Red de Servicios de Internet

Como se explicó en el Capítulo 1, la Red de Servicios de Internet no es un aspecto relevante para la aplicación, la elección de una red recomendada está basada en las características de velocidad y accesibilidad que una red pueda proveer mas no en restricciones técnicas de funcionamiento de modo que el uso de esa red será el recomendable pero no exclusivo.

En la Tabla 2-1 se muestran las principales ventajas y desventajas de las tecnologías descritas en la Sección 1.1.

TABLA 2-1: COMPARACIÓN DE LAS TECNOLOGÍAS DE ACCESO A RED INALÁMBRICAS

Tecnología	Ventajas	Desventajas
WLAN	<ul style="list-style-type: none"> - Es independiente del proveedor del servicio de comunicación telefónica móvil. - Ofrece altas velocidades de transmisión. 	<ul style="list-style-type: none"> - Requiere un dispositivo especial que incluya las tecnologías de Wireless LAN que usa el proveedor. - Los dispositivos que incluyen tales funcionalidades tienen un costo elevado a comparación de los equipos tradicionales que no las incluyen. - Está limitada por el área de cobertura del Access Point. - No implica el pago de tráfico de datos al operador.

<p>GPRS</p>	<ul style="list-style-type: none"> - Tecnología incluida en la mayoría de dispositivos móviles del mercado. - Capacidad de transmisión suficiente para el envío de consultas de datos. - El área de cobertura es amplia pues está basada en la red GSM cuyo grado de expansión es alto. 	<ul style="list-style-type: none"> - El servicio es brindado por el operador telefónico, la transmisión y recepción de datos puede incluir un costo adicional por el tráfico de datos. - Depende de la disponibilidad de la red GSM.
-------------	--	--

Luego de haber estudiado las dos opciones se recomienda el uso de la GPRS para el escenario de uso masivo debido a que las redes de los operadores de telefonía móvil celular, sobre las cuales trabaja GPRS, están desplegadas a nivel nacional. En el caso del Departamento de Lima la penetración asciende a 128,8% [OSI2009].

Por otro lado, el uso de la red WLAN aporta grandes facilidades para entornos de estudio y desarrollo de laboratorio puesto que no se requiere el pago del tráfico utilizado desde el terminal móvil para el acceso a la aplicación, bajo esta idea se recomienda la red WLAN para el escenario de uso de laboratorio e investigación.

2.5.2 Lenguaje de programación

El lenguaje de programación elegido es Java debido a su performance a nivel de aplicaciones, el ser multiplataforma y extender las funcionalidades de un servidor web como se mencionó en el capítulo anterior lo hace el más apropiado para nuestra aplicación la cual utilizará la plataforma J2ME y sus extensiones.

2.5.2.1 JSR 172

Java Specification Request 172 indica las extensiones del J2ME Web Services para el soporte de servicios web. Los dos paquetes opcionales del API estandarizan dos áreas de funcionalidades que son cruciales para los clientes de un servicio web: invocación de servicio remoto y el XML-parsing.

El API está diseñado para trabajar con los perfiles de J2ME basados en CDC (Connected Device Configuration) o CLDC (Connected Limited Device Configuration).

La invocación remota está basada en un subconjunto de J2SE's Java API for XML-Based RPC (JAX-RPC 1.1) con algunas clases de RMI (Remote Method Invocation) que se incluyen para complementar JAX-RPC. Por otro lado, el API XML-parsing está basado en un subconjunto de SAX2 (Simple API for XML versión 2) [W3C2007, ORT2004].

2.5.3 Soporte de servicios web

Se eligió como servidor de aplicaciones a GlassFish debido a la modularidad y mejor performance que ofrece frente a Tomcat tal como se muestra en las pruebas de escalabilidad realizadas por Sun para las que se utilizó un servlet simple con el fin de reducir al mínimo el tiempo en el contenedor y medir el número de operaciones por segundo. Para el caso de 16 000 usuarios los resultados se muestran en la Tabla 2-2 [SUN2009a].

TABLA 2-2: COMPARACIÓN DEL DESEMPEÑO ENTRE GLASSFISH Y TOMCAT

Fuente: [SUN2009a]

Resultados	GlassFish	Tomcat
ops/segundo	6988.9	6615.3
Tiempo medio de rpta (segundos)	0.242	0.358
Tiempo máximo de rpta (segundos)	1.519	3.693
90% tiempo de rpta (segundos)	0.6	0.75

Es importante resaltar que GlassFish es una colección de contenedores de JavaEE en tanto que Tomcat es un contenedor web lo que abre un abanico de posibilidades mayores para GlassFish.



Capítulo 3

Diseño de la Solución

La solución ofrece proveer al usuario de un teléfono móvil una interfaz en la que él ingrese los datos de las ubicaciones origen y destino a través de las cuales desea encontrar una ruta, luego de ello se procede a enviar la solicitud al servidor el cual responderá con un listado de rutas disponibles para esos dos puntos.

3.1 Arquitectura del Sistema

La arquitectura de comunicación está compuesta en tres bloques: el cliente, el proveedor de servicios de datos y del servidor, cada uno de ellos está compuesto por los equipos que se describen a continuación y tal como se muestran en la Figura 3-1:

1. Cliente: Compuesto por el Teléfono móvil o mobile station, será quien ejecute la aplicación, ingrese los datos de origen y destino, luego de la consulta recibirá la respuesta.
2. Proveedor de servicios: Compuesto por el medio inalámbrico cuya red puede ser GSM/GPRS que ha contratado el cliente para su servicio de

transmisión y recepción de datos o una red WLAN a la que el cliente se conectó mediante un terminal con la capacidad de enviar y recibir datos por este tipo de redes.

3. Servidor: Compuesto por un computador en el cual se tiene instalado el servicio y está conectado a Internet mediante una conexión pública a la que el cliente puede conectarse sin necesidad de encontrarse en la misma red.

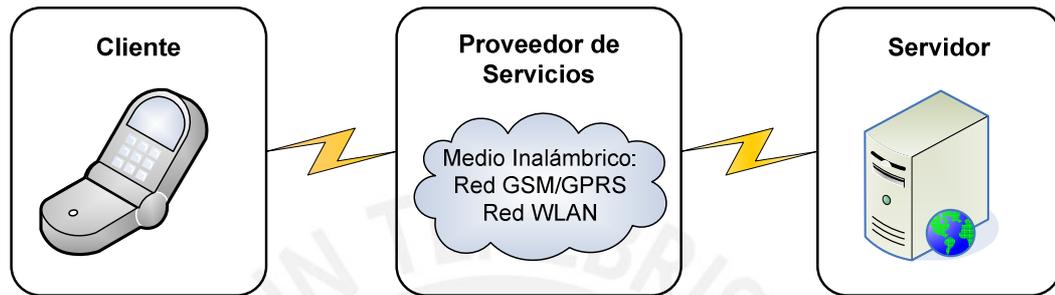


FIGURA 3-1: ESQUEMA GENERAL DE LA ARQUITECTURA DE LA SOLUCIÓN

3.2 Flujo del sistema

En la Figura 3-2 se muestra el Diagrama de Flujo del Servicio.

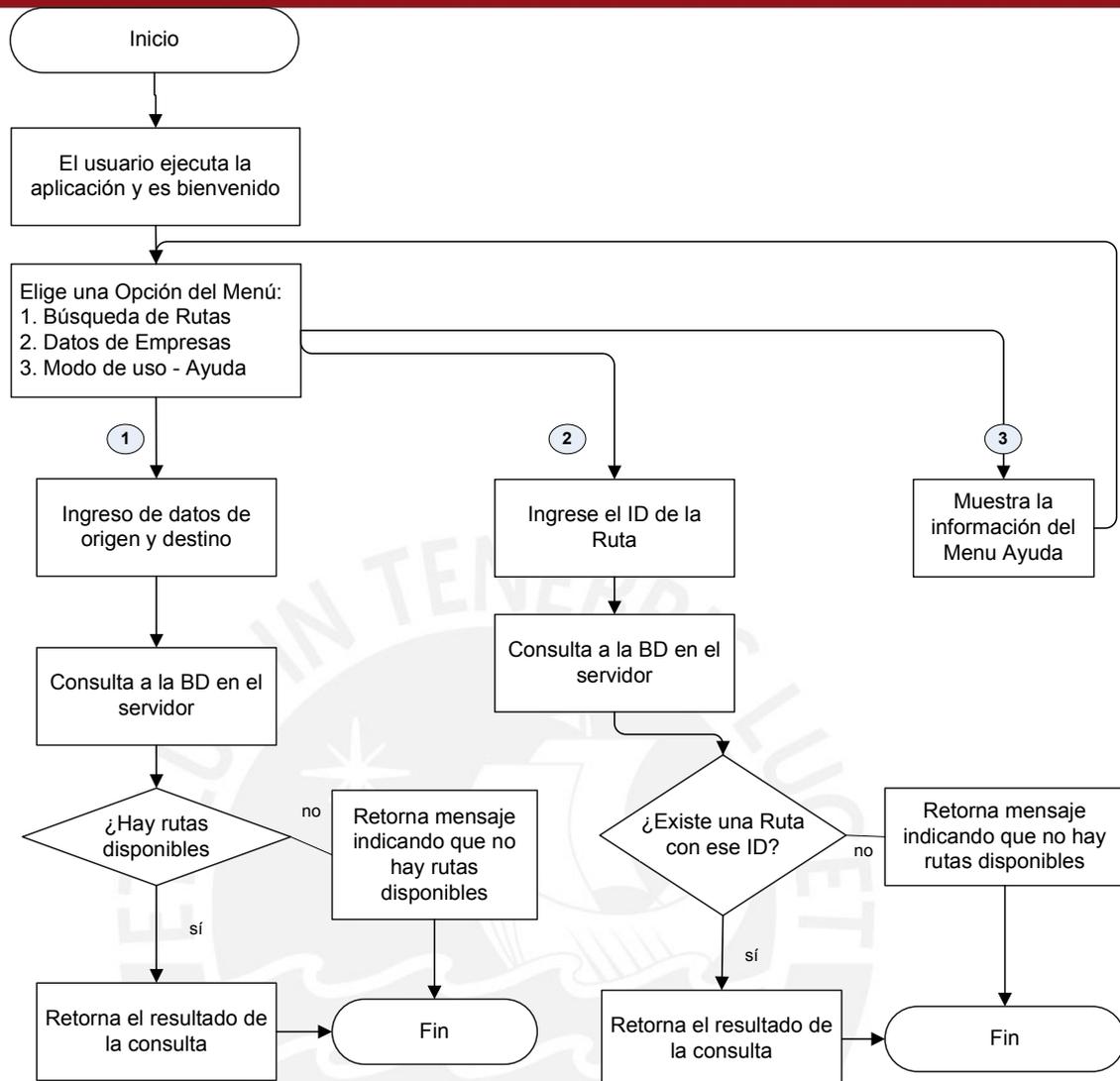


FIGURA 3-2: DIAGRAMA DE FLUJO DEL SERVICIO

3.3 Base de Datos del Sistema

La Base de Datos elegida para el sistema es MySQL, este sistema de base de datos relacional se orienta a aplicaciones cuyas funcionalidades están estrechamente relacionadas a la web. Su condición de código abierto y a su escalabilidad y performance han hecho de MySQL la base de datos más popular en el mundo.

Para la comunicación con la base de datos desde los comandos creados en Java para la ejecución de la aplicación y su ejecución con el sistema de base de datos se utilizará el archivo mysql-connector-java-5.1.10-bin.jar compatible con JDK 1.6.

En el diseño de la Base de Datos se tomaron los requerimientos definidos en la sección 2.3 con respecto a las tablas que debe manejar la Base de Datos.

El modelo de datos definido se explica gráficamente en la Figura 3-3:

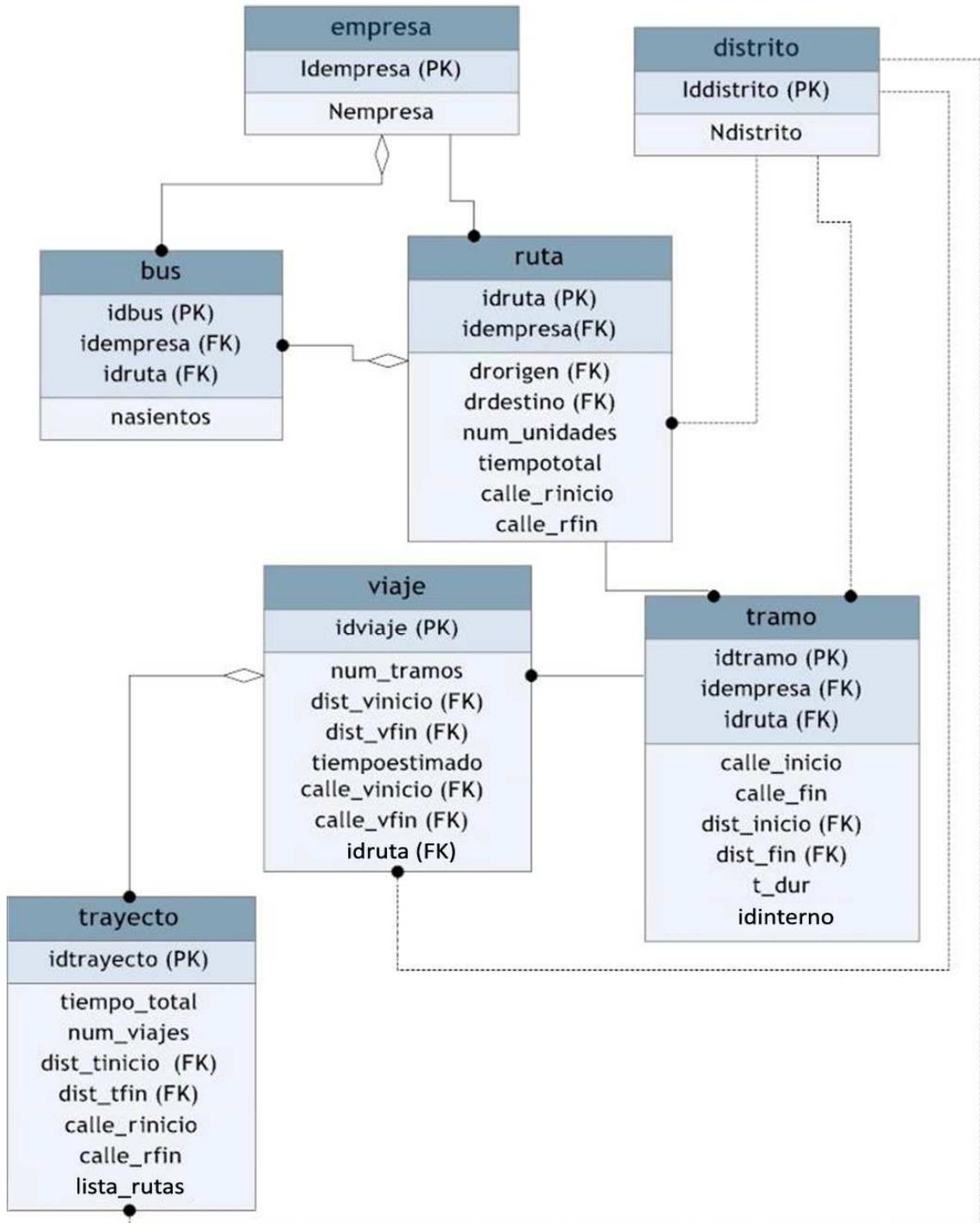


FIGURA 3-3: DIAGRAMA DE MODELO ENTIDAD RELACIÓN

Luego de haber definido las tablas de la base de datos se explicará las relaciones y atributos de cada una de ellas:

3.3.1 Empresa

Esta tabla muestra las características de la empresa de transportes que brinda los servicios de transporte de pasajeros a nivel metropolitano. En la Tabla 3-1 se especifica el tipo de datos y la descripción de cada campo.

TABLA 3-1: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA EMPRESA

Nombre	Tipo de Dato	Descripción
IDEMPRESA	INTEGER	Llave primaria, identifica cada empresa de transporte.
NEMPRESA	VARCHAR (60)	Nombre de la Empresa de Transporte.

3.3.2 Bus

Esta tabla incluye los datos del bus, la empresa a la que pertenece y la ruta a la que corresponde. En la Tabla 3-2 se especifica el tipo de datos y la descripción de cada campo.

TABLA 3-2: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA BUS

Nombre	Tipo de Dato	Descripción
IDBUS	INTEGER	Llave primaria, identifica cada empresa de transporte.
IDEMPRESA	INTEGER	Identificador de la Empresa de Transporte.
IDRUTA	VARCHAR (10)	Identificador de la Ruta.

NASIENTOS	INTEGER	Número de asientos del Bus.
-----------	---------	-----------------------------

3.3.3 Viaje

En esta tabla se incluyen los campos de los viajes, cada viaje es un conjunto de tramos que tiene una ruta en común cuyo origen y destino están incluidos en la trayectoria deseada para culminar un trayecto. En la Tabla 3-3 se especifica el tipo de datos y la descripción de cada campo.

TABLA 3-3: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA VIAJE

Nombre	Tipo de Dato	Descripción
IDVIAJE	INTEGER	Llave primaria, identifica el viaje.
NUM_TRAMOS	INTEGER	Indica el número de tramos que componen el viaje.
DIST_VINICIO	VARCHAR(55)	Distrito inicial del viaje.
DIST_VFIN	VARCHAR(55)	Distrito final del viaje.
TIEMPOESTIMADO	INTEGER	Duración estimada dada en minutos.
CALLE_VINICIO	VARCHAR(65)	Calle de inicio del viaje.
CALLE_VFIN	VARCHAR(65)	Calle de fin del viaje.
IDRUTA	VARCHAR(10)	Identificador de la Ruta.

3.3.4 Ruta

Una ruta comprende los campos de origen y destino así como la empresa que la programa. Pese a que en los alcances de la solución no se considera el tiempo total que demanda la ruta, esta tabla permite tener el manejo de este tipo de datos. En la Tabla 3-4 se especifica el tipo de datos y la descripción de cada campo.

TABLA 3-4: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA RUTA

Nombre	Tipo de Dato	Descripción
IDRUTA	VARCHAR(10)	Llave Primaria. Identificador de la ruta.
IDEMPRESA	INTEGER	Identificador de la empresa de transportes.
DRORIGEN	VARCHAR(55)	Distrito de origen de la ruta.
DRDESTINO	VARCHAR(55)	Distrito de destino de la ruta.
TIEMPOTOTAL	INTEGER	Tiempo total estimado de la ruta dado en minutos.
NUM_UNIDADES	INTEGER	Número de unidades que conforman la ruta.
CALLE_RINICIO	VARCHAR(65)	Calle de inicio de la ruta.
CALLE_RFIN	VARCHAR(65)	Calle de fin de la ruta.

3.3.5 Trayecto

La tabla Trayecto presenta los datos correspondientes a la consulta del usuario. Un trayecto está compuesto por un conjunto de viajes que corresponden a una o más rutas, incluye dos campos en los que se especifica el distrito de inicio y el final, adicionalmente tiene un campo que indica el número de viajes que lo componen. En la Tabla 3-5 se especifica el tipo de datos y la descripción de cada campo.

TABLA 3-5: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA TRAYECTO

Nombre	Tipo de Dato	Descripción
IDTRAYECTO	INTEGER	Llave Primaria. Identificador del trayecto.
NUM_VIAJES	INTEGER	Indica el número de viajes que componen el trayecto.
DIST_TINICIO	VARCHAR(55)	Distrito de inicio del trayecto.
DIST_TFIN	VARCHAR(55)	Distrito final del trayecto.
TIEMPO_TOTAL	INTEGER	Tiempo total estimado del trayecto dado en minutos.
CALLE_TINICIO	VARCHAR(65)	Calle de inicio del trayecto.
CALLE_TFIN	VARCHAR(65)	Calle de fin del trayecto.
LISTA_RUTAS	VARCHAR(95)	Listado de las rutas que conforman el trayecto.

3.3.6 Tramo

Se conforma de los campos que identifican a la empresa y a la ruta a la que pertenecen al igual que la calle y distritos en los que inicia y finaliza. Un tramo es la unidad básica de un viaje, trayecto y ruta pues a partir de él se compone la información de las tres tablas mencionadas. En la Tabla 3-6 se especifica el tipo de datos y la descripción de cada campo.

TABLA 3-6: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA TRAMO

Nombre	Tipo de Dato	Descripción
IDTRAMO	INTEGER	Llave Primaria. Identificador del tramo.
IDEMPRESA	INTEGER	Identificador de la Empresa de Transportes.
IDRUTA	VARCHAR(10)	Identificador de la Ruta.
CALLE_INICIO	VARCHAR(65)	Calle en la que se inicia el tramo.
CALLE_FIN	VARCHAR(65)	Calle en la que termina el tramo.
DIST_INICIO	VARCHAR(55)	Distrito en el que se inicia el tramo.
DIST_FIN	VARCHAR(55)	Distrito en el que finaliza el tramo.
T_DUR	INTEGER	Tiempo estimado de duración del tramo.
IDINTERNO	INTEGER	Identificador del orden del tramo dentro de una Ruta

3.3.7 Distrito

Un distrito se caracteriza por el nombre que posee y un identificador.

En la Tabla 3-7 se especifica el tipo de datos y la descripción de cada campo.

TABLA 3-7: DESCRIPCIÓN DE LOS ELEMENTOS DE LA TABLA DISTRITO

Nombre	Tipo de Dato	Descripción
IDDISTRITO	INTEGER	Llave primaria, identifica cada distrito.
NDISTRITO	VARCHAR (55)	Nombre del Distrito.

3.4 Estándares del Sistema

La aplicación tiene los siguientes estándares:

- Los nombres de los distritos y de las calles deben empezar con letra mayúscula.
- Para realizar la Búsqueda de Rutas deben completarse obligatoriamente los campos de Distrito Origen y Distrito Destino, sin embargo, los campos Calle Origen, Calle Destino y Número Máximo de Buses a tomar pueden completarse de forma facultativa.

3.5 Interfaces de la Aplicación Móvil

Las interfaces son las pantallas que se observarán durante la ejecución de la aplicación. En las siguientes secciones se detallarán las características de cada una de ellas.

3.5.1 Inicio de la aplicación

Al ingresar a la aplicación el usuario recibe un mensaje de bienvenida y luego es direccionado a una pantalla donde se ingresan los datos, en la Figura 3-4 se observa la Interfaz de Inicio de la aplicación en tanto que en la Figura 3-5 se tiene el Menú de inicio.



FIGURA 3-4: INTERFAZ DE INICIO DE LA APLICACIÓN

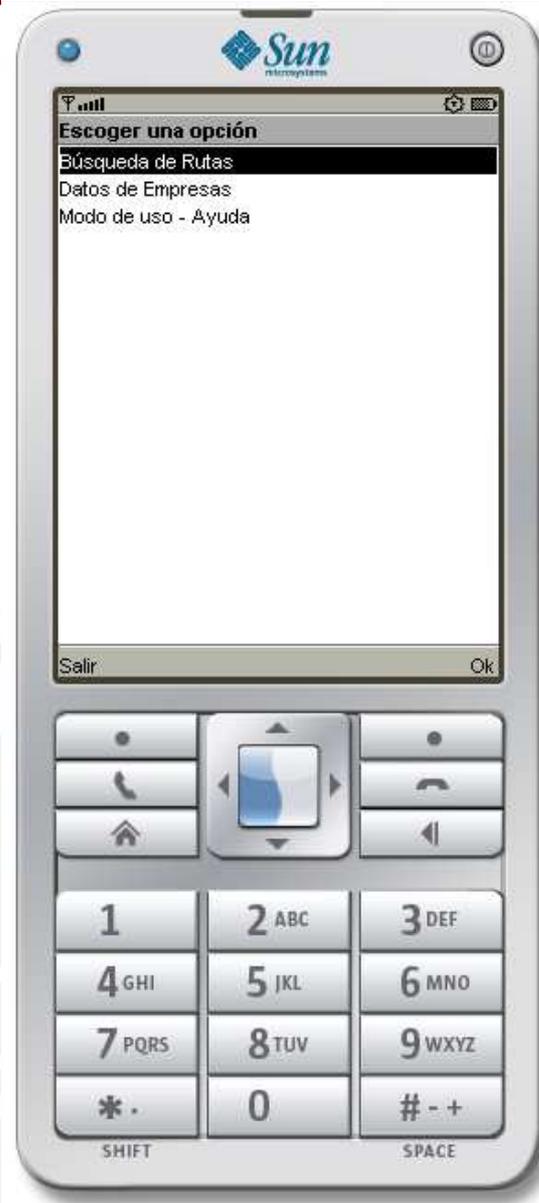


FIGURA 3-5: MENÚ DE INICIO DE LA APLICACIÓN

3.5.2 Ingreso de datos para la Búsqueda de Rutas

En la interfaz de ingreso de datos el usuario debe rellenar los campos de origen y destino y número máximo de buses a tomar. En la Figura 3-6 se toma como ejemplo los datos que tienen como distrito origen Jesús María y como fin en Pueblo Libre.



FIGURA 3-6: INTERFAZ DE INGRESO DE DATOS AL SISTEMA – CONSULTA DE RUTAS

3.5.3 Muestra de resultados de la Búsqueda de Rutas

Primero se realiza la consulta para los Tramos y Viajes, es decir aquellos en los que se utilice un solo bus para ir desde el origen hasta el destino. Luego de haberse realizado la consulta, los datos son enviados al cliente y mostrados en una tabla que muestre las Rutas y la Empresa de Transportes que cumple con los requisitos de la solicitud tal como se muestra en la Figura 3-7.

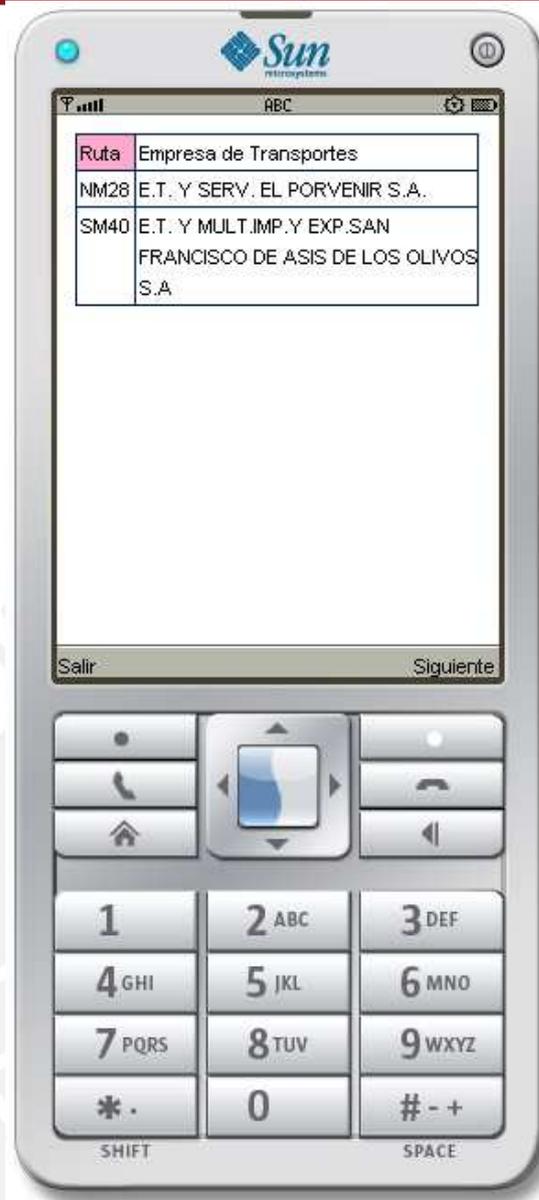


FIGURA 3-7: SALIDA DE LOS DATOS – RESPUESTA A LA CONSULTA DE TRAMOS Y VIAJES

Después de mostrar los resultados que incluyen Tramos y Viajes se realiza la consulta para conocer los Trayectos que implica tomar más de un bus y tener más de una ruta para llegar al destino deseado, para obtener este resultado se presiona el botón Siguiente y se observa el listado de respuestas tal como se muestra en la Figura 3-8.



FIGURA 3-8: SALIDA DE LOS DATOS – RESPUESTA A LA CONSULTA DE TRAYECTOS

3.5.4 Ingreso de datos para obtener Información de Empresas

En la interfaz de ingreso de Datos de Empresas se debe ingresar el identificador (ID) de la Ruta el cual para el ejemplo mostrado en la Figura 3-9 es “NM29”.

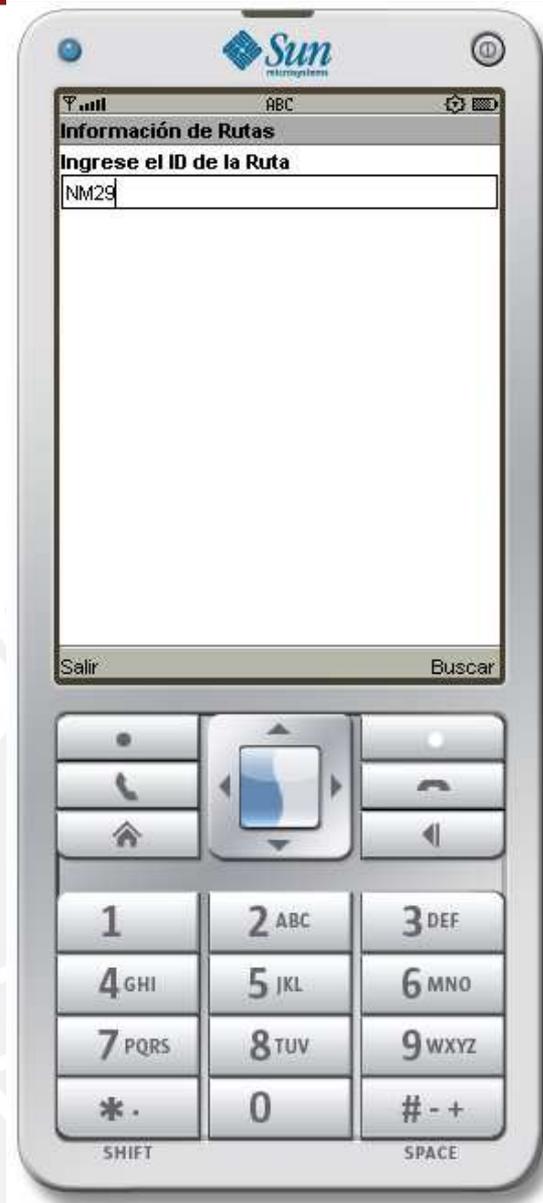


FIGURA 3-9: INTERFAZ DE INGRESO DE DATOS AL SISTEMA – DATOS DE EMPRESAS

3.5.5 Muestra de resultados de los Datos de Empresas

Se muestran los datos de la Empresa de Transportes que presta servicios en la Ruta conocida, el resultado de la búsqueda de ejemplo se observa en la Figura 3-10.

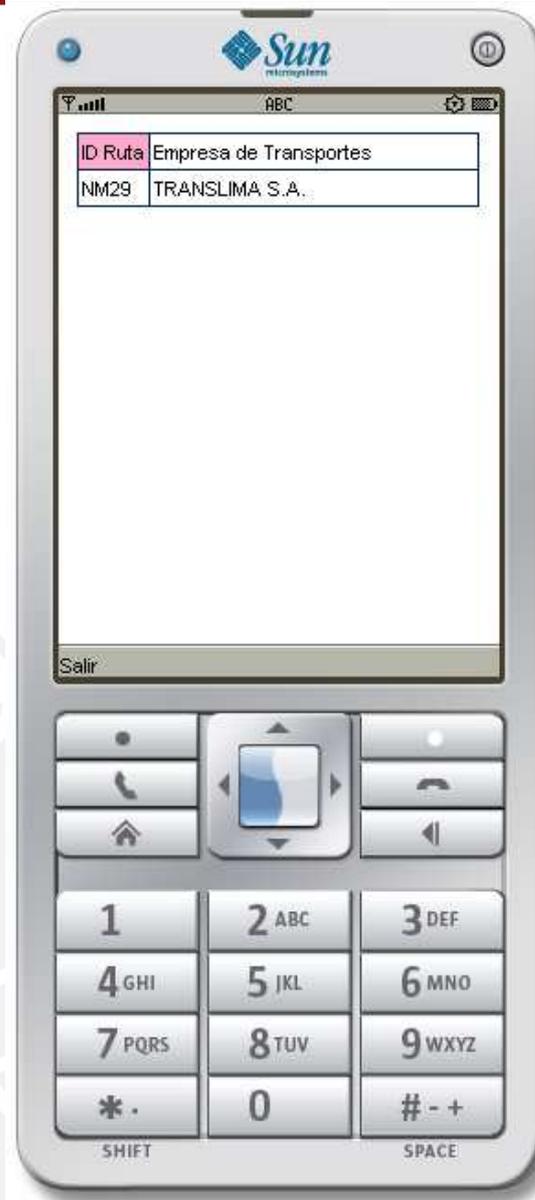


FIGURA 3-10: SALIDA DE LOS DATOS – RESPUESTA A LA CONSULTA DE DATOS DE EMPRESAS

3.5.6 Información del Modo de Uso - Ayuda

Para que el usuario pueda conocer el manejo del Menú de la Aplicación, la información requerida, los resultados de las búsquedas entre otros datos puede ingresar al Menú Modo de Uso – Ayuda como se muestra en la Figura 3-11.

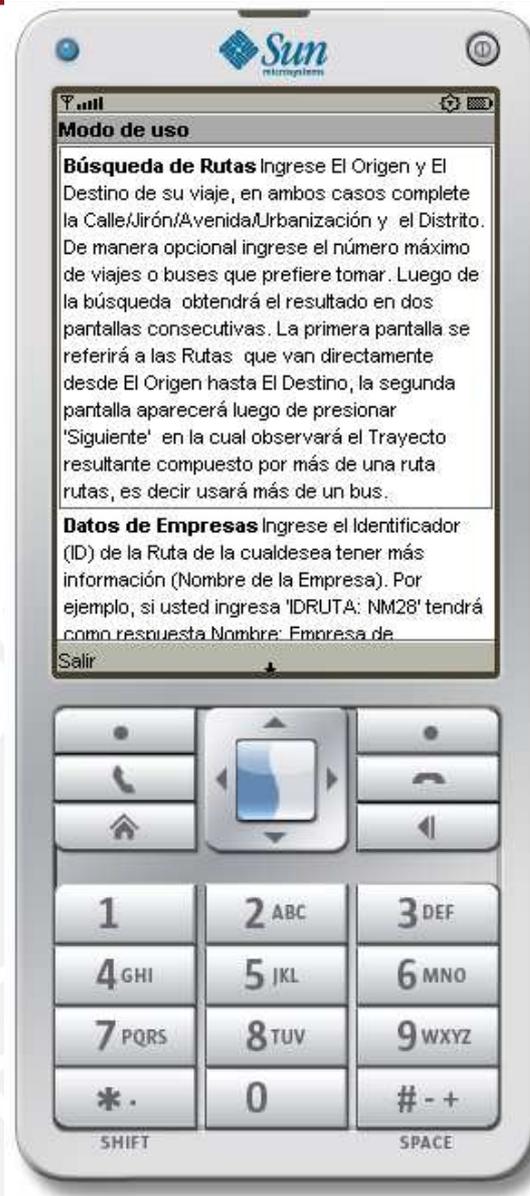


FIGURA 3-11: MENÚ MODO DE USO - AYUDA



Capítulo 4

Implementación de la Solución

En el presente capítulo se detalla las características de la implementación realizada en el equipo servidor y el terminal móvil. El sistema está dividido en los tres principales aspectos: la estructura de las aplicaciones, los requerimientos técnicos de los equipos y sus respectivas configuraciones los que en conjunto permiten el funcionamiento del mismo.

4.1 Estructura de la Aplicación

Como se explicó anteriormente, la puesta en marcha del prototipo necesita un equipo servidor y un equipo terminal, las características de los equipos utilizados se describen a continuación:

4.1.1 Sistema Servidor

Está compuesto por una aplicación web que contiene las clases que contempladas en el Diccionario de Clases y el Servicio de Aplicaciones cuyas operaciones se describen en el Diccionario de Operaciones del Web Service.

El servidor de aplicaciones está montado sobre un servidor Glassfish V2.1 y se encuentra implementado en una PC con procesador AMD Athlon Dual-Core QL-60 de 1.90GHz con 2GB de memoria RAM y 160 GB de disco duro corriendo el Sistema Operativo Windows Vista.

El servidor de aplicaciones recibirá los datos del cliente web service móvil para luego ejecutar la aplicación web del servidor que incluye las consultas a la base de datos y finalmente se encargará de la preparación y envío de los datos de retorno al cliente para ser mostrados en la pantalla del terminal.

4.1.1.1 Diccionario de Clases del Sistema

Las clases que componen la aplicación web del servidor se mostrarán en el rango Tabla 4-1 – Tabla 4-8.

TABLA 4-1: CLASE BBUS

Clase	Atributos	Métodos	Descripción
Bbus	int ldbus int ldempresa String ldruta Int Nasientos	getldbus()	Obtiene el ID del Bus
		getldempresa()	Obtiene el ID de la Empresa
		getldruta()	Obtiene el ID de la Ruta
		getNasientos()	Obtiene el número de asientos
		setldbus()	Establece o cambia el ID del Bus
		setldempresa()	Establece o cambia el ID de la Empresa
		setldruta()	Establece o cambia el ID de la Ruta
		setNasientos()	Establece o cambia el número de asientos

TABLA 4-2: CLASE BDISTRITO

Clase	Atributos	Métodos	Descripción
Bdistrito	int Iddistrito String Ndistrito	getIddistrito()	Obtiene el ID del Distrito
		getNdistrito()	Obtiene el nombre del Distrito
		setIddistrito()	Establece o cambia el ID del Distrito
		setNdistrito()	Establece o cambia el nombre del Distrito

TABLA 4-3: CLASE BEMPRESA

Clase	Atributos	Métodos	Descripción
Bempresa	int Idempresa String Nempresa	getIdempresa	Obtiene el ID de la Empresa
		getNempresa()	Obtiene el nombre de la Empresa
		setIdempresa()	Establece o cambia el ID de la Empresa
		setNempresa()	Establece o cambia el nombre de la Empresa

TABLA 4-4: CLASE BRUTA

Clase	Atributos	Métodos	Descripción
Bruta	String Idruta	getIdruta	Obtiene el ID de la Ruta
	Int Idempresa	getIdempresa()	Obtiene el ID de la Empresa
	String Drorigen String	getDrorigen()	Obtiene el nombre del Distrito origen de la Ruta

	Drdestino	getDrdestino()	Obtiene el nombre del Distrito destino de la Ruta
	Int Tiempototal		
	Int Num_unidades	getTiempototal()	Obtiene el Tiempo total promedio que toma desplazarse desde el origen de la ruta hasta el final de la misma.
	String calle_inicio		
	String calle_fin	getNum_unidades()	Obtiene el número de unidades que circulan en esa ruta.
		getCalle_inicio()	Obtiene el nombre de la calle en la que inicia la ruta
		getCalle_fin()	Obtiene el nombre de la calle en la que finaliza la ruta
		setIdruta	Establece o cambia el ID de la Ruta
		setIdempresa()	Establece o cambia el ID de la Empresa
		setDrorigen()	Establece o cambia el Distrito origen de la Ruta
		setDrdestino()	Establece o cambia el Distrito destino de la Ruta
		setTiempototal()	Establece o cambia el Tiempo total promedio que toma desplazarse desde el origen de la ruta hasta el final de la misma.
		setNum_unidades()	Establece o cambia el número de unidades que circulan en esa ruta.

		setCalle_inicio()	Establece o cambia el nombre de la calle en la que inicia la ruta
		setCalle_fin()	Establece o cambia el nombre de la calle en la que finaliza la ruta

TABLA 4-5: CLASE BTRAMO

Clase	Atributos	Métodos	Descripción
Btramo		getldtramo	Obtiene el ID del tramo
	Int Idtramo	getldempresa()	Obtiene el ID de la Empresa
	Int Idempresa	getldruta()	Obtiene el ID de la Ruta
	String ldruta	getCalle_inicio()	Obtiene el nombre de la Calle inicio del tramo
	String Calle_inicio	getCalle_fin()	Obtiene el nombre de la Calle fin del tramo
	String Calle_fin	getDist_inicio()	Obtiene el nombre del Distrito inicio del tramo
	String Dist_inicio	getDist_fin()	Obtiene el nombre del Distrito fin del tramo
	String Dist_fin	getT_dur()	Obtiene el tiempo aproximado que dura el viaje
	Int T_dur	getldinterno	Obtiene el ID interno del tramo para identificar el orden de los tramos en una misma ruta
	Int Idinterno	getNempresa()	Obtiene el nombre de la Empresa
	String Nempresa		

		setIdtramo	Establece o cambia el ID del Tramo
		setIdempresa()	Establece o cambia el ID de la Empresa
		setldruta()	Establece o cambia el ID de la Ruta
		setCalle_inicio()	Establece o cambia el nombre de la Calle inicio del tramo
		setCalle_fin()	Establece o cambia el nombre de la Calle fin del tramo
		setDist_inicio()	Establece o cambia el nombre del Distrito inicio del tramo
		setDist_fin()	Establece o cambia el nombre del Distrito fin del tramo
		setT_dur()	Establece o cambia el tiempo aproximado que dura el viaje en ese tramo
		setIdinterno	Establece o cambia el ID interno del tramo para identificar el orden de los tramos en una misma ruta
		setNempresa()	Establece o cambia el nombre de la Empresa

TABLA 4-6: CLASE BTRAYECTO

Clase	Atributos	Métodos	Descripción
Btrayecto	Int Idtrayecto Int Num_viajes String Dist_inicio String Dist_tfin Int Tiempototal String Calle_inicio String Calle_tfin String Listarutas	getIdtrayecto()	Obtiene el ID del trayecto
		getNum_viajes()	Obtiene el número de viajes de ese trayecto
		getDist_inicio()	Obtiene el nombre del Distrito inicio del trayecto
		getDist_tfin()	Obtiene el nombre del Distrito fin del trayecto
		getTiempototal()	Obtiene el Tiempo total aproximado que dura el trayecto
		getCalle_inicio()	Obtiene el nombre de la Calle inicio del trayecto
		getCalle_tfin()	Obtiene el nombre de la Calle fin del trayecto
		getListarutas()	Obtiene una lista con los IDs de las rutas que componen el trayecto
		setIdtrayecto()	Establece o cambia el ID del trayecto
		setNum_viajes()	Establece o cambia el número de viajes de ese trayecto
setDist_inicio()	Establece o cambia el nombre del Distrito inicio del trayecto		
setDist_tfin()	Establece o cambia el nombre del Distrito fin del		

			trayecto
		setTiempototal()	Establece o cambia el Tiempo total aproximado que dura el trayecto
		setCalle_tinicio()	Establece o cambia el nombre de la Calle inicio del trayecto
		setCalle_tfin()	Establece o cambia el nombre de la Calle fin del trayecto
		setListarutas()	Establece o cambia una lista con los IDs de las rutas que componen el trayecto

TABLA 4-7: CLASE BVI AJE

Clase	Atributos	Métodos	Descripción
Btrayecto	Int Idviaje	getIdviaje()	Obtiene el ID del viaje
	Int Num_tramos	getNum_tramos()	Obtiene el número de tramos de ese viaje
	String Dist_vinicio	getDist_vinicio()	Obtiene el nombre del Distrito inicio del viaje
	String Dist_vfin	getDist_vfin()	Obtiene el nombre del Distrito fin del viaje
	Int Tiempoestima do	getTiempoestimado()	Obtiene el Tiempo estimado que dura el viaje
	String Calle_vinicio	getCalle_vinicio()	Obtiene el nombre de la Calle inicio del viaje
	String	getCalle_vfin()	Obtiene el nombre de la

	Calle_vfin		Calle fin del viaje
	String Idruta	getIdruta()	Obtiene el ID de la Ruta de la cual forma parte el viaje
	String		
	Nempresa	getNempresa()	Obtiene el nombre de la Empresa asociada a la Ruta del Viaje
		getIdviaje()	Establece o cambia el ID del viaje
		getNum_tramos()	Establece o cambia el número de tramos de ese viaje
		getDist_inicio()	Establece o cambia el nombre del Distrito inicio del viaje
		getDist_vfin()	Establece o cambia el nombre del Distrito fin del viaje
		getTiempoestimado()	Establece o cambia el Tiempo estimado que dura el viaje
		getCalle_inicio()	Establece o cambia el nombre de la Calle inicio del viaje
		getCalle_vfin()	Establece o cambia el nombre de la Calle fin del viaje
		getIdruta()	Establece o cambia el ID de la Ruta de la cual forma parte el viaje
		getNempresa()	Establece o cambia el

			nombre de la Empresa asociada a la Ruta del Viaje
--	--	--	---

TABLA 4-8: CLASE DPRIMERO

Clase	Métodos	Tipo	Descripción
DPrimero	Listatramos	ArrayList	Obtiene o crea el listado de tramos que corresponden a la búsqueda de rutas a partir de los datos de origen y destino
	setDatostramo	Void	Establece o cambia los valores del bean Btramo que recibe como parámetro.
	getultimoidviaje	Int	Obtiene el valor del ID de viaje que se usará para un nuevo registro.
	getultimoidtrayecto	Int	Obtiene el valor del ID de trayecto que se usará para un nuevo registro.
	listaviajes	ArrayList	Obtiene o crea el listado de viajes que corresponden a la búsqueda de rutas a partir de los datos de origen y destino
	setDatosviaje	Void	Establece o cambia los valores del bean Bviaje que recibe como parámetro.
	Listatrayectos	ArrayList	Obtiene o crea el listado de trayectos que corresponden a la búsqueda de rutas a partir de los datos de origen y destino

	setDatostrayecto	Void	Establece o cambia los valores del bean Btrayecto que recibe como parámetro.
	datosruta	Bruta	Obtiene o crea el bean Bruta que corresponde a la búsqueda de información de rutas a partir del ID de la ruta
	setDatosruta	Void	Establece o cambia los valores del bean Bruta que recibe como parámetro.
	datosempresa	Bempresa	Obtiene o crea el vean Bempresa que corresponde a la búsqueda de información de rutas a partir del ID de la ruta
	setDatosempresa	Void	Establece o cambia los valores del bean Bempresa que recibe como parámetro.

4.1.1.2 Diccionario de Operaciones del Web Service

Se detallan en la Tabla 4-9.

TABLA 4-9: DICCIONARIO DE OPERACIONES

Operación	Parámetro	Salida	Descripción
Rxparametros	Ocalle Odistrito Dcalle Ddistrito	ArrayList<Btram >	Usa los parámetros ingresados para realizar la consulta mediante el método Listatramos y obtiene el listado de los tramos que cumplen con la búsqueda.
Rxviajes	Ocalle	ArrayList<Bviaje	Usa los parámetros ingresados para realizar la

	Odistrito Dcalle Ddistrito	>	consulta mediante el método Listaviajes y obtiene el listado de los viajes que cumplen con la búsqueda.
Rxtrayecto	Ocalle Odistrito Dcalle Ddistrito Max_viajes	ArrayList<Btrayecto>	Usa los parámetros ingresados para realizar la consulta mediante el método Listatrayectos y obtiene el listado de los trayectos que cumplen con la búsqueda.
Rxdatosruta	Inputidruta	Bruta	Usa el parámetro ingresado para realizar la consulta mediante el método Datosruta y obtiene la ruta que cumple con la búsqueda.
Rxdatosempresa	Inputidruta	Bempresa	Usa el parámetro ingresado para realizar la consulta mediante el método Datoempresa y obtiene la ruta que cumple con la búsqueda.

4.1.2 Servidor de Base de Datos

Se implementó el servidor de base de datos en una PC con procesador AMD Athlon Dual-Core QL-60 de 1.90GHz con 2GB de memoria RAM y 160 GB de disco duro corriendo el Sistema Operativo Windows Vista con el Sistema MySQL 5.1 instalado.

La implementación consistió en la creación de la base de datos y la programación de las consultas utilizadas en la aplicación web incluidas en la clase Dprimero.

4.1.3 Aplicación web móvil

En los siguientes subcapítulos se describen las características de la aplicación web móvil.

4.1.3.1 Diccionario de clases de la aplicación web móvil

Las clases que componen la aplicación web móvil se explican en la Tabla 4-10:

TABLA 4-10: DICCIONARIO DE CLASES DE LA APLICACIÓN WEB MÓVIL

Clase	Descripción
RutasTransporte	Es el MIDlet que maneja la lógica de la aplicación.
Tabla_Canvas	Se encarga de completar los valores de la Tabla que incluye la respuesta de Tramos y Viajes.
Tabla_dempresa	Se encarga de completar los valores de la Tabla que incluye la respuesta de la Información de Empresas.
Tabla_trayectos	Se encarga de completar los valores de la Tabla que incluye la respuesta de Trayectos.
Tabla_scroll	Dibuja una tabla a partir de los datos de Tramos, Viajes e Información de Empresas.
Trayecto_scroll	Dibuja una tabla a partir de los datos de Trayectos.
ServicioWebService	Llama a cada operación del webservice según corresponda e incluyendo los parámetros requeridos.
ServicioWebService_Stub	Permite que el cliente pueda invocar directamente los métodos del Servicio Web

4.1.4 Equipo Terminal

La aplicación web móvil se implementó en un equipo terminal móvil que incluye los API JSR 172 J2ME™ Web Services Specification, JSR 118 MIDP 2.1, con memoria SDRAM 128 MB con CPU tipo ARM 11 con clock rate 434 MHz y corriendo el sistema operativo Symbian v9.4.

El terminal utilizado para la implementación se describe en la Tabla 4-11:

TABLA 4-11: INFORMACIÓN DEL EQUIPO TERMINAL

Característica	Descripción
Marca	Nokia
Modelo	5800 XpressMusic
Versión de software	V 20.2.014
Tipo	RM-428

4.2 Configuraciones del Sistema

En los siguientes subcapítulos se describe la configuración del Sistema

4.2.1 Base de Datos

La información de la configuración realizada a la base de datos se observa en la Tabla 4-12.

TABLA 4-12: INFORMACIÓN DE CONFIGURACIÓN DE LA BASE DE DATOS

Característica	Descripción
Versión	5.1.38-community MySQL Community Server (GPL)
Puerto TCP	3306
Almacenamiento por defecto	INNODB
Key buffer	22M
Máximo número de conexiones concurrentes	100

4.2.2 Servidor de Aplicaciones web

El servidor usado como se mencionó anteriormente es Glassfish y funciona de la siguiente manera: La aplicación web móvil cliente envía la petición al servidor web, para ello adjunta los parámetros necesarios para la operación que se eligió en el Menú del usuario, luego de ello, el servidor hace la consulta en la base de datos y prepara la información en un paquete SOAP y la envía al cliente del servicio web. En la Tabla 4-13 se describe la información de la configuración realizada. Los pasos utilizados para la implementación del servidor de aplicaciones fueron tomados de [UNI2009].

TABLA 4-13: INFORMACIÓN DE CONFIGURACIÓN DEL SERVIDOR DE APLICACIONES

Característica	Descripción
Puerto HTTP	4848,8181,8080
Puerto IIOP	3820,3700,3920
Versión	Sun GlassFish Enterprise Server v2.1 (9.1.1) (build b60e-fcs)
Dominio	domain1
Versión de Java	1.6.0_13

4.2.3 Servicio Web

El servicio web utilizado es el del tipo SOAP debido a su flexibilidad, compatibilidad y la utilidad de la envoltura SOAP como se explicó en los capítulos anteriores.

En referente a la seguridad, esta concierne tanto al suscriptor como el proveedor de servicio de información web. El usuario desea verificar si en efecto se está comunicando con el servidor correcto, esto es conocido como autenticación del servidor; asimismo, el usuario desea asegurarse que la información transmitida en el aire hacia el servicio web sea confidencial, es decir, que no sea interceptada y comprendida por extraños. El usuario también desea conocer que la información no sea manipulada en el camino lo que significa que la integridad de los datos sea garantizada [PAS2005].

La seguridad puede ser aplicada a los servicios web a nivel de transporte y de mensaje. En la seguridad de mensaje, la información de seguridad viaja junto con el mensaje del servicio web. Web Service Security (WSS) en la capa SOAP es el uso de la encriptación XML y la firma digital XML para garantizar los mensajes SOAP. Los perfiles WSS usan varios tokens incluidos en certificados X.509, afirmaciones SAML y tokens usuario/contraseña para lograrlo.

La seguridad de la capa de mensaje difiere de la de transporte puesto que en la seguridad capa de mensaje se puede utilizar para separar la protección de mensajes desde el transporte de mensajes para que los mensajes permanezcan protegidos luego de su transmisión, independientemente del número de saltos por los que viaje.

La seguridad de la capa de mensaje difiere de la de transporte puesto que en la seguridad capa de mensaje se puede utilizar para separar la protección de mensajes desde el transporte de mensajes para que los mensajes permanezcan protegidos luego de su transmisión, independientemente del número de saltos por los que viaje.

Con respecto a la seguridad a nivel de transporte, la autenticación verifica la identidad del usuario, dispositivo u otra entidad de un sistema informático, generalmente como requisito previo a permitir el acceso a los recursos de un sistema.



Capítulo 5

Pruebas del Prototipo

En este capítulo se describirá el proceso de pruebas realizado con el prototipo, se explicarán las herramientas utilizadas y se listarán los resultados encontrados.

5.1 Introducción

Como se explicó anteriormente, el aplicativo está orientado a redes inalámbricas de transmisión de datos como GPRS, HSDPA, LTE, entre otros, sin embargo, puede ser utilizada en redes inalámbricas de área local WLAN para uso en laboratorio puesto que como se explicó en el Capítulo 2 el aplicativo es independiente del medio inalámbrico; tomando como base este precepto se realizaron las pruebas en una Red Inalámbrica de Área Local (WLAN) dentro de la cual estaban conectados el equipo servidor y el terminal móvil.

El presente proceso del sistema tiene como finalidad comprobar el funcionamiento de la implementación realizada y analizar los resultados obtenidos a partir de ella. Para este propósito se realizarán las pruebas en dos escenarios, el primero utilizando la herramienta J2ME Wireless Toolkit 2.5.2, la cual provee medios para la

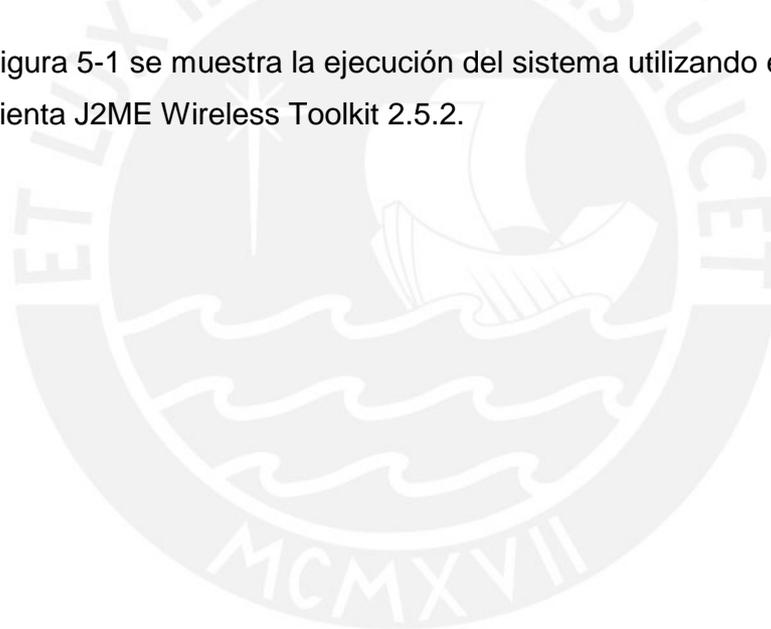
creación y ejecución de aplicaciones móviles mediante de un emulador incorporado, para el segundo escenario se utilizó el terminal móvil utilizado en la implementación.

Las pruebas se realizaron el día jueves 17 de diciembre de 2009 a las 15:05:05 y finalizaron el día jueves 17 de diciembre de 2009 a las 17:05:05.

5.2 Emulación del Sistema en la Herramienta J2ME Wireless Toolkit 2.5.2.

J2ME Wireless Toolkit es una herramienta utilizada para la creación y ejecución de aplicaciones MIDP, incluye la especificación J2ME Web Services Specification (JSR 172), asimismo, luego de tener las clases ya creadas, esta herramienta puede compilar, pre verificar las clases y empaquetar la aplicación en un MIDlet el cual puede verse en ejecución [SUN2009b].

En la Figura 5-1 se muestra la ejecución del sistema utilizando el emulador de la herramienta J2ME Wireless Toolkit 2.5.2.



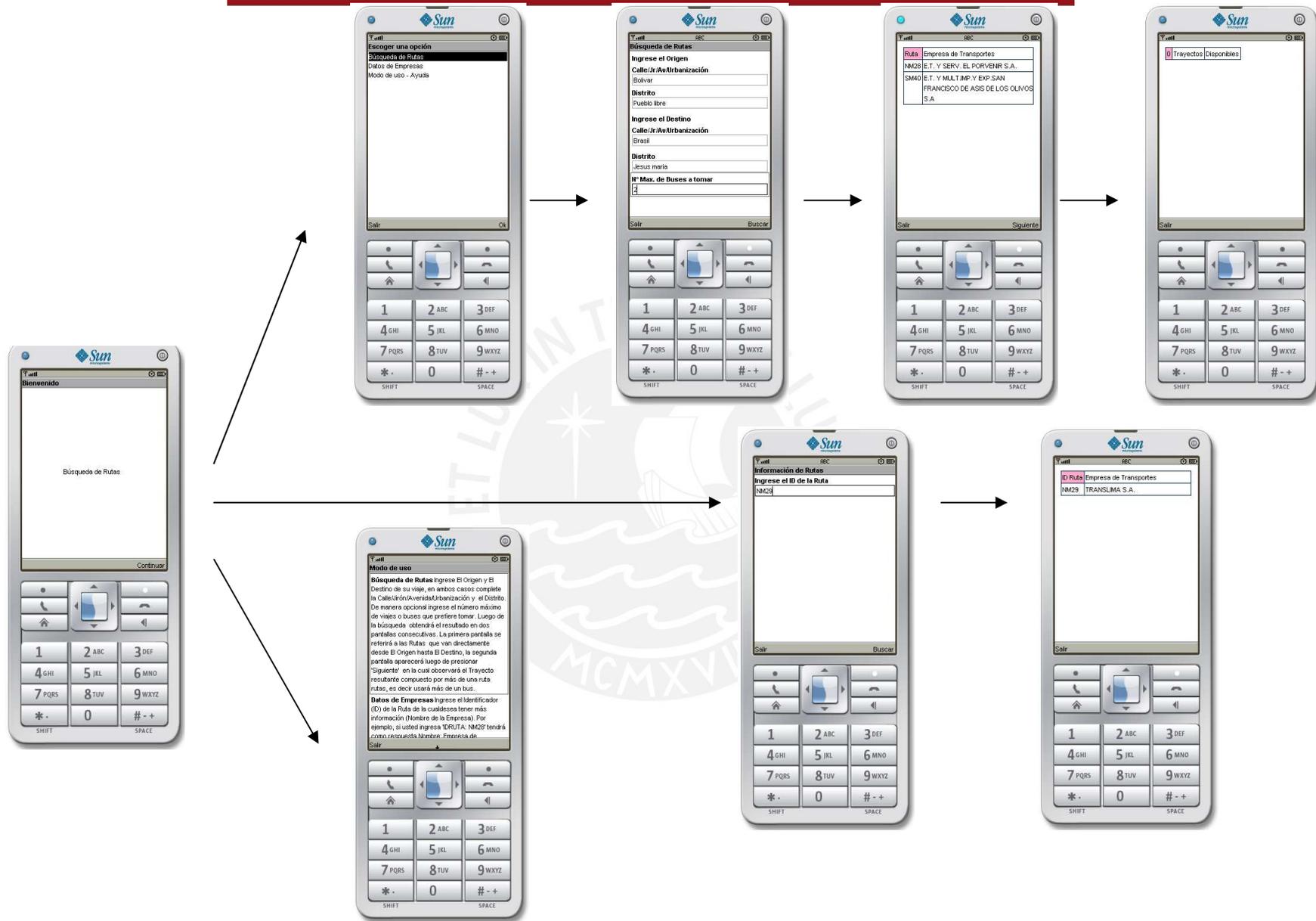


FIGURA 5-1. FLUJO DE EJECUCIÓN DE LA APLICACIÓN EN J2ME WIRELESS TOOLKIT 2.5.2

5.3 Ejecución del Aplicativo en el Terminal Móvil

Como se explicó anteriormente, se realizó la implementación en el equipo terminal Nokia 5800 XpressMusic con acceso a WLAN según el estándar IEEE 802.11 b/g. En la Figura 5.2 se muestra la ejecución del sistema utilizando el equipo terminal descrito.





FIGURA 5-2. FLUJO DE EJECUCIÓN DE LA APLICACIÓN EN EL TERMINAL MÓVIL NOKIA 5800 XPRESSMUSIC

5.4 Uso del CPU y Memoria

Se hizo la medición el uso del CPU y de la Memoria RAM durante el periodo de pruebas con el fin de encontrar el impacto del uso del servidor Glassfish durante una consulta en los dos escenarios mencionados, de manera complementaria y adicional se muestra el valor del uso de la red y del disco duro.

En la Figura 5-3, Figura 5-5 y la Figura 5-7 se analizan los cuatro principales parámetros: el CPU, Disco Duro, Uso de la Red y Memoria RAM, para el CPU el porcentaje total de la capacidad del CPU que está en uso se grafica en color verde y en azul la frecuencia máxima del mismo, para el disco se muestra la Entrada/Salida total en color verde y el porcentaje máximo de tiempo activo en color azul, en la información de red se detalla en verde el tráfico de la red total actual (en Kbps) y en azul el porcentaje de la capacidad de la red en uso, finalmente en la memoria se especifica en verde los errores severos por segundo actuales y en azul el porcentaje de la memoria física actualmente en uso.

En la Figura 5-4, Figura 5-6 y la Figura 5-8 se muestra la información del Uso del CPU, Historial de uso del CPU, el estado de la memoria y el historial de uso de la memoria física.

Los dos gráficos superiores indican cuánta CPU se está usando, tanto en ese momento como en los últimos minutos, el gráfico Historial de uso de CPU aparece dividido debido a la presencia de una CPU de doble núcleo.

El porcentaje que aparece refleja el uso de los recursos de CPU para los procesos que se están ejecutando.

Los resultados de la medición del uso del CPU sin la presencia del servidor Glassfish se observan en las Figuras 5-3 y 5-4.

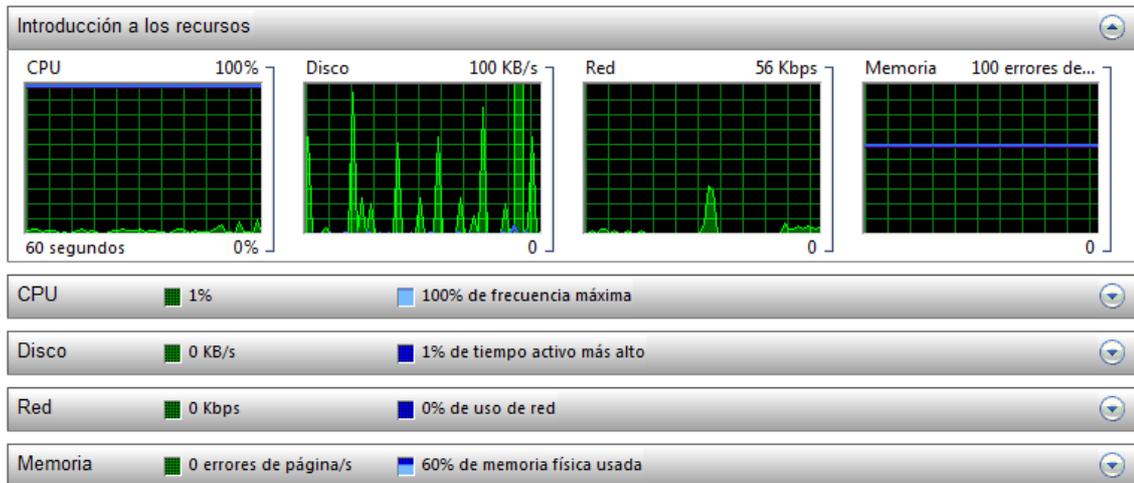


FIGURA 5-3: PARÁMETROS DE USO DE RECURSOS SIN SERVIDOR GLASSFISH

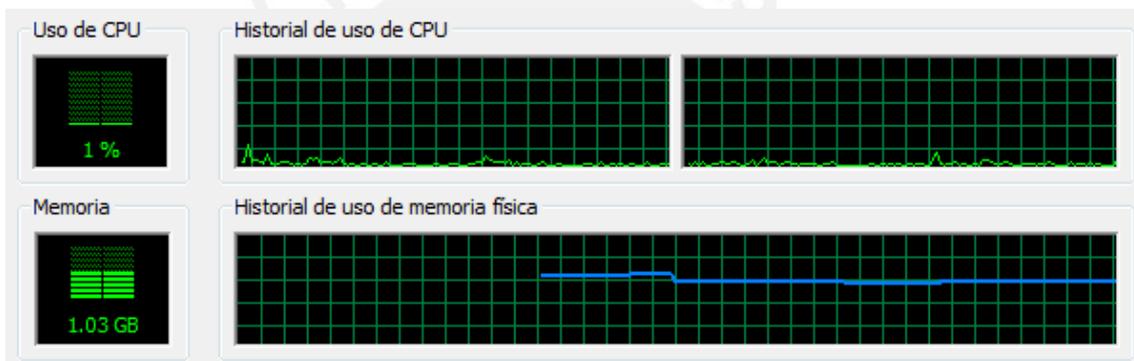


FIGURA 5-4: USO DEL CPU Y MEMORIA RAM SIN SERVIDOR GLASSFISH

5.4.1 Uso del CPU y Memoria para el cliente Emulador de J2ME Wireless Toolkit 2.5.2

Al comparar los datos tomados de la Figura 5-3 y la Figura 5-5 se observa que usando el cliente Emulador se tiene un incremento de 3% en el uso del CPU, porcentaje que se ve corroborado en las Figuras 5-4 y 5-6.

Asimismo, el uso de la Memoria RAM aumentó en 0.22GB tal como se muestra en las Figuras 5-4 y 5-6.

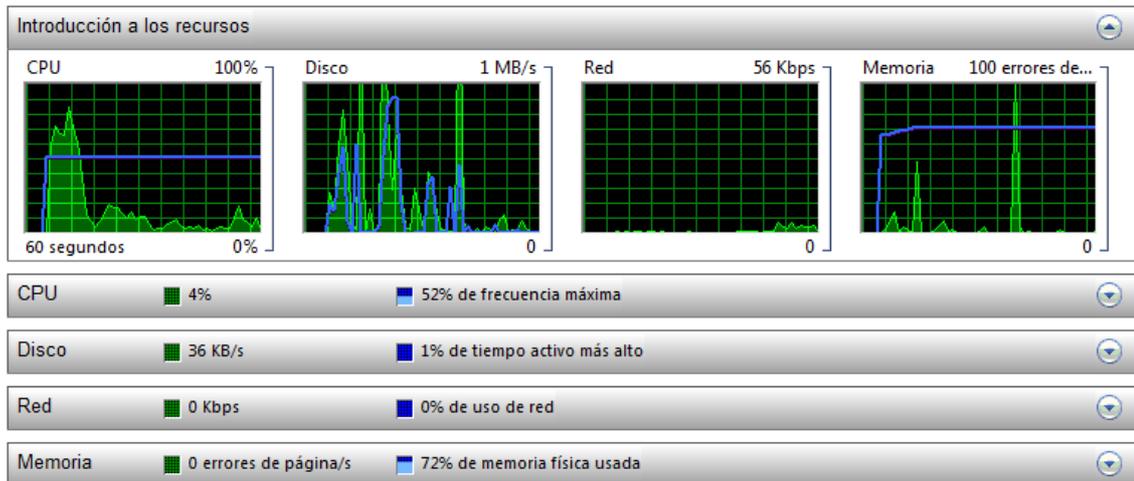


FIGURA 5-5: PARÁMETROS DE USO DE RECURSOS CON SERVIDOR GLASSFISH Y EMULADOR

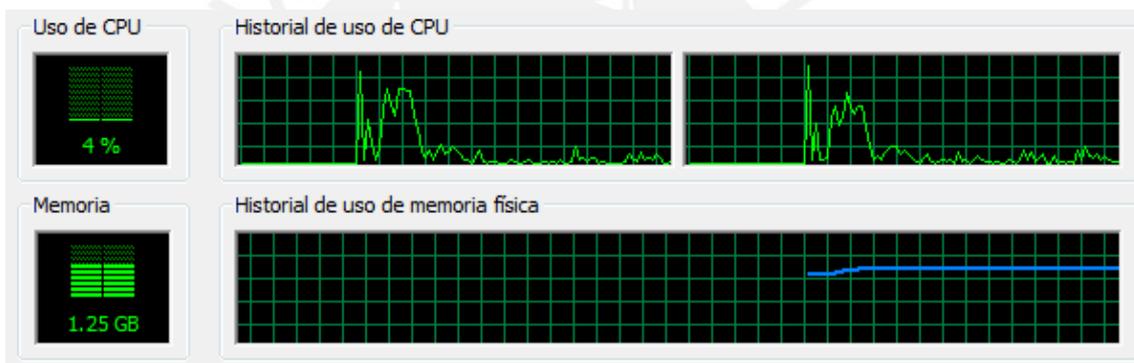


FIGURA 5-6: USO DEL CPU Y MEMORIA RAM CON SERVIDOR GLASSFISH Y EMULADOR

5.4.2 Uso del CPU y Memoria para el cliente Terminal Nokia 5800 XpressMusic

Luego de hacer una comparación entre los datos tomados de la Figura 5-3 y la Figura 5-7 se observa un incremento de 3% en el uso del CPU, porcentaje que se ve corroborado en las Figuras 5-4 y 5-8.

Asimismo, el uso de la Memoria RAM aumentó en 0.22GB tal como se muestra en las Figuras 5-4 y 5-6.

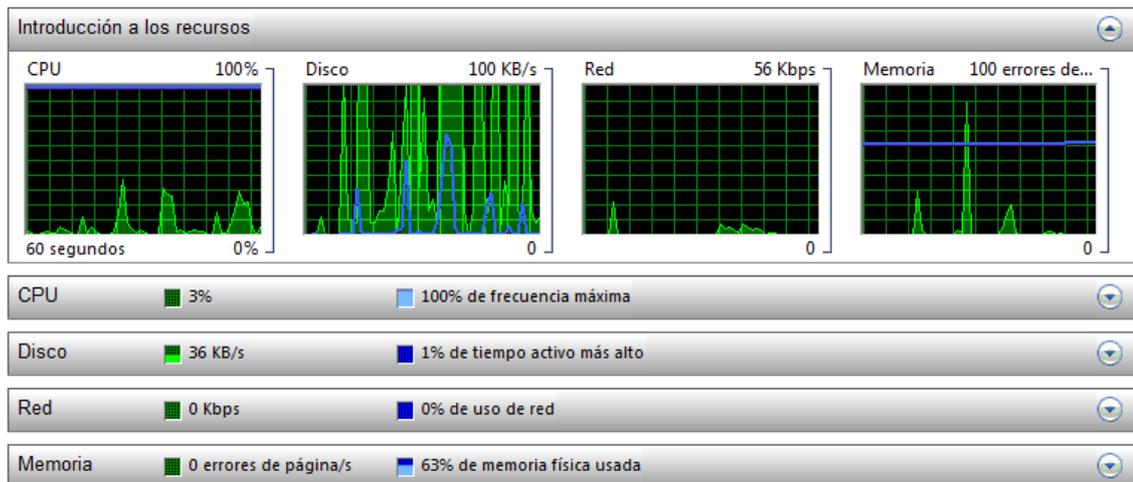


FIGURA 5-7: PARÁMETROS DE USO DE RECURSOS CON SERVIDOR GLASSFISH Y EL TERMINAL

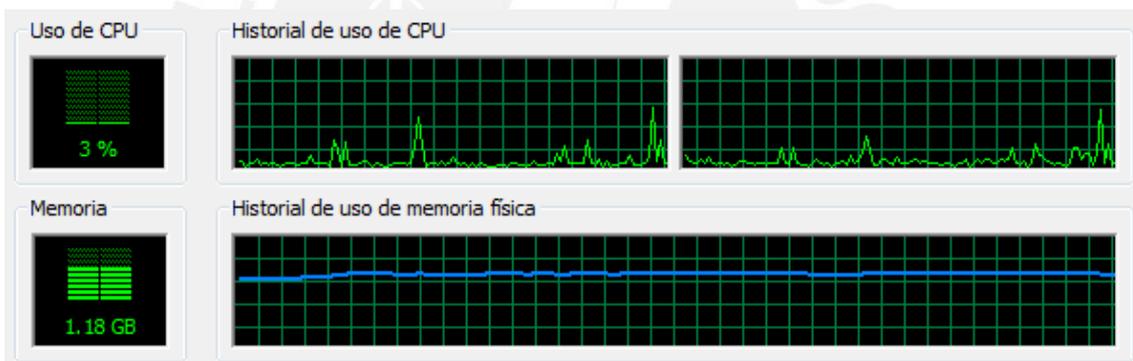


FIGURA 5-8: USO DEL CPU Y MEMORIA RAM CON SERVIDOR GLASSFISH Y EL TERMINAL

Después de comparar ambos resultados se observa que el uso del CPU es mayor cuando se ejecuta el emulador en el mismo equipo que cuando la aplicación se ejecuta directamente en el equipo terminal, el resultado que está acorde a lo esperado puesto que en el segundo caso el uso de los recursos se ve compartido en el equipo servidor y terminal a diferencia del primer caso en el cual un mismo equipo cumple las dos funciones (de servidor y cliente).

5.4.3 Análisis de Tramas

Durante la ejecución de las pruebas se realizó la captura de las tramas enviadas y recibidas las que permiten conocer el contenido de la información contenida en ellas, en la Figura 5-9 se observan los mensajes entre los hosts 192.168.1.101 que

corresponde al servidor y el host 192.168.1.104 que corresponde al terminal, los mensajes de color verde está encapsulados en XML, dentro de ellos se puede examinar la estructura del empaquetado SOAP como se muestra en la Figura 5-10.

No. .	Time	Source	Destination	Protocol	Info
38	29.038631	00:26:cc:1d:58:dc	Broadcast	ARP	who has 192.168.1.101? Tell 192.168.1.104
39	29.038664	HonHaiPr_12:04:a5	00:26:cc:1d:58:dc	ARP	192.168.1.101 is at 00:22:69:12:04:a5
40	29.041334	192.168.1.104	192.168.1.101	TCP	47203 > http-alt [SYN] Seq=0 win=64240 Len=0 MSS=1460 TSV=3817572639 TSER=
41	29.041467	HonHaiPr_12:04:a5	Broadcast	ARP	who has 192.168.1.104? Tell 192.168.1.101
42	29.045252	00:26:cc:1d:58:dc	HonHaiPr_12:04:a5	ARP	192.168.1.104 is at 00:26:cc:1d:58:dc
43	29.045267	192.168.1.101	192.168.1.104	TCP	http-alt > 47203 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 MSS=1460 WS=8 TSV=
44	29.048860	192.168.1.104	192.168.1.101	TCP	47203 > http-alt [ACK] Seq=1 Ack=1 win=64240 Len=0 TSV=3817589639 TSER=34
45	29.051248	192.168.1.104	192.168.1.101	TCP	[TCP segment of a reassembled PDU]
46	29.248617	192.168.1.101	192.168.1.104	TCP	http-alt > 47203 [ACK] Seq=1 Ack=295 win=17152 Len=0 TSV=3457973 TSER=381
47	29.366201	192.168.1.104	192.168.1.101	HTTP/XML	POST /ptesis02/ServiciowebsService HTTP/1.1
48	29.558586	192.168.1.101	192.168.1.104	TCP	http-alt > 47203 [ACK] Seq=1 Ack=771 win=16896 Len=0 TSV=3458004 TSER=381
49	35.067127	192.168.1.101	192.168.1.104	TCP	[TCP segment of a reassembled PDU]
50	35.068095	192.168.1.101	192.168.1.104	HTTP/XML	HTTP/1.1 200 OK
51	35.193539	192.168.1.104	192.168.1.101	TCP	[TCP segment of a reassembled PDU]
52	35.388617	192.168.1.101	192.168.1.104	TCP	http-alt > 47203 [ACK] Seq=1060 Ack=1065 win=16384 Len=0 TSV=3458587 TSER=
53	35.511508	192.168.1.104	192.168.1.101	HTTP/XML	POST /ptesis02/ServiciowebsService HTTP/1.1
54	35.639992	192.168.1.101	192.168.1.104	TCP	[TCP segment of a reassembled PDU]
55	35.640227	192.168.1.101	192.168.1.104	HTTP/XML	HTTP/1.1 200 OK

FIGURA 5-9: TRAMAS CAPTURADAS DURANTE LA EJECUCIÓN DE LA APLICACIÓN EN EL TERMINAL

No. .	Time	Source	Destination	Protocol	Info
52	35.388617	192.168.1.101	192.168.1.104	TCP	http-alt > 47203 [ACK]
53	35.511508	192.168.1.104	192.168.1.101	HTTP/XML	POST /ptesis02/ServiciowebsService HTTP/1.1

```

<?xml
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tns="http://ws.paq/">
  <soap:Body>
    <tns:rxviajes>
      <ocalle
        xmlns="">
        Bolivar
      </ocalle>
      <odistrito
        xmlns="">
        Pueblo libre
      </odistrito>
      <dcalle
        xmlns="">
        Brasil
      </dcalle>
      <ddistrito
        xmlns="">
        Jesus maria
      </ddistrito>
    </tns:rxviajes>
  </soap:Body>
</soap:Envelope>
    
```

02b0	3d 22 22 3e 4a 65 73 75 73 20 6d 61 72 69 61	3c	="">Jesu s maria
02c0	2f 64 64 69 73 74 72 69 74 6f 3e 0a 3c 2f 74 6e		/ddistri to>.</tn
02d0	73 3a 72 78 76 69 61 6a 65 73 3e 0a 3c 2f 73 6f		s:rxviaj es>.</so

Frame (534 bytes) Reassembled TCP (762 bytes)

FIGURA 5-10: EMPAQUETADO SOAP CAPTURADO DURANTE LA EJECUCIÓN DE LA APLICACIÓN EN EL TERMINAL



Conclusiones, Recomendaciones y Trabajos Futuros

A continuación se explicarán las conclusiones, recomendaciones de la tesis y los trabajos futuros que puedan darse a partir de ella.

6.1 Recomendaciones

Se sugieren, en lo subsiguiente, algunas recomendaciones a tener en cuenta sobre el presente proyecto.

En el equipo Servidor:

- **Hardware:** Se recomienda que el equipo sobre el cual se encuentre montado el servidor de aplicaciones y el servidor de base de datos sea de alto desempeño puesto que como se sabe la performance de las conexiones realizadas es proporcional a la cantidad de memoria RAM disponible para este fin.

- El servidor de aplicaciones recomendado es Glassfish el cual provee ligereza, eficiencia y un entorno de trabajo agradable al programador.
- Para la base de datos, el servidor recomendado es MySQL el cual cuenta con documentación bastante detallada y cuya performance es altamente reconocida a nivel mundial.

En el equipo Cliente:

- Hardware: Se requiere el uso de equipos con funcionalidades de datos que permitan el ingreso a internet y en consecuencia acceso al servicio web, para ello se recomienda tener una cuenta de datos habilitada.

6.2 Conclusiones

Luego de culminar esta tesis se concluye:

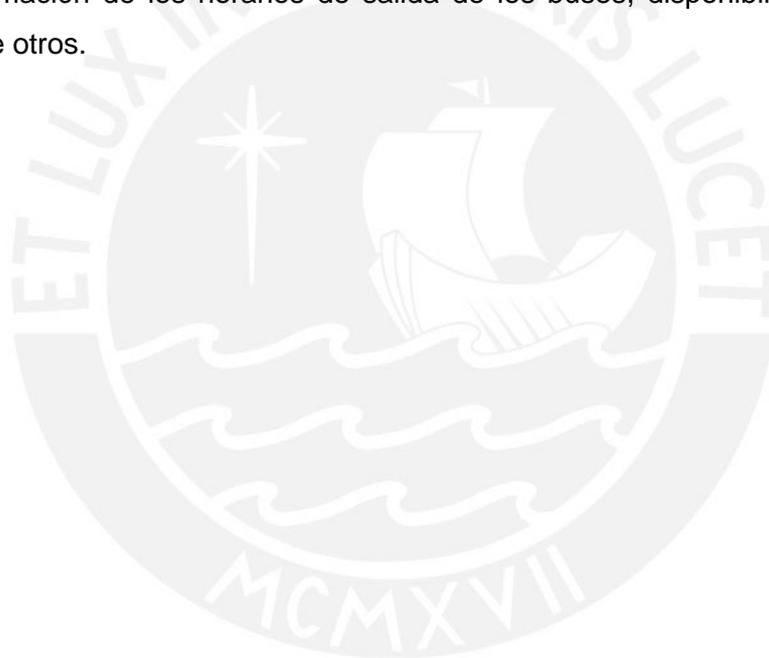
1. Se realizó la descripción de los conceptos utilizados en el diseño de la solución, para ello se tomó en cuenta las herramientas de desarrollo que ofrece el mercado así como los protocolos utilizados en el manejo de datos en las aplicaciones de servicios web y se concluye que estos elementos permiten que la información se comunique en forma rápida y sencilla siendo esta una funcionalidad elemental para los dispositivos móviles.
2. Luego de explicar la situación actual se estudió y comparó las herramientas, así también se analizó los requerimientos del sistema y a partir de ellos se expuso la propuesta de solución, se concluye que la mejor alternativa incluye el uso de lenguajes XML y Java, protocolos SOAP, WSDL, especificaciones UDDI, servidor de aplicaciones Glassfish y servidor de base de datos MySQL.
3. Se diseñó la solución y a partir de ello se concluye que la solución ofrece una interfaz web mediante la cual se ingresa los datos de ubicación del usuario y luego se envía la solicitud al servidor el cual envía de retorno la lista de rutas que cumplen con tal consulta.
4. Se implementó con éxito el sistema en el equipo servidor y el terminal móvil, además de ello se hizo una explicación detallada de los tres aspectos que comprende el sistema: la estructura de aplicaciones, requerimientos técnicos del equipamiento y las configuraciones para después concluir que el la implementación funciona adecuadamente.

5. Se hizo pruebas en el servidor implementado y el equipo terminal, se comprobó la ejecución exitosa en el prototipo, se hicieron mediciones del uso de recursos del sistema y finalmente se listó los resultados encontrados en el proceso.

6.3 Trabajos Futuros

Tomando como base este proyecto se pueden derivar distintos trabajos, entre ellos se tiene:

- Implementar un servicio más avanzado para la información del transporte urbano para los nuevos proyectos que se materializarán en los próximos dos años: El Metropolitano y El Tren Eléctrico Urbano de Lima.
- Implementar funcionalidades adicionales a la aplicación como por ejemplo la información de los horarios de salida de los buses, disponibilidad de asientos, entre otros.



Bibliografía

- [APA2009] Apache – Tomcat URL: <http://tomcat.apache.org/>
- [CAS2008] Castro, L. (2008) Sistema de Explotación de Instalaciones Deportivas basado en Servicios Web.
- [CHA2002] Chase, N. (2002) XML primer plus. Indiana: Sams.
- [GOM2006] Gómez, I. & Pérez, I., & Torres, A. (2006). Herramientas de Simulación Bursátil Sobre Agentes de Software y Web Services
- [GOS2005] Gosling, J. (2005). The Java™ Language Specification Third Edition. California: ADDISON-WESLEY
- [HIR2006] Hirsch, F. & Kemp, J. & Ilkka, J. (2006). Mobile Web Services Architecture and Implementation. England: John Wiley & Sons
- [INE2009] Instituto Nacional de Estadística e Informática. Situación del Mercado Laboral en Lima Metropolitana.
URL: <http://www1.inei.gob.pe>
- [MUN2009] Municipalidad Metropolitana de Lima – Gerencia de Transporte Urbano.
URL: <http://www.gtu.munlima.gob.pe/>
- [ORT2004] Ortiz, E. (2004). Introduction to J2ME Web Services. Extraído el 22 de junio del 2009 desde <http://developers.sun.com/mobility/apis/articles/wsa/>
- [OSI2009] Organismo Supervisor de Inversión Privada en Telecomunicaciones OSIPTEL – Estadísticas, Investigaciones y Publicaciones. Indicadores de Servicio Móvil. Densidad por Departamento.
URL: <http://www.osiptel.gob.pe>
- [PAS2005] Pashtan, A. (2005). Mobile Web Services. Cambridge: Cambridge University Press.
- [RAM2006] Ramsey, B.(2006). XML&Web Services with PHP.
- [SAL2004] Salkintzis, A. (2004). Mobile internet Enabling Technologies and Services. Florida: CRC
- [SAN2003] Sanders, G. & Thorens, L. (2003). GPRS Networks. England: John Wiley & Sons
- [SUN2005] SUN Technologies (2005).The J2EETM 1.4 Tutorial. Extraído el 20 de junio de 2009 desde <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>

- [SUN2008] SunGlassFish Enterprise Server v3 Prelude Release Notes [Versión Electrónica]. Extraído el 5 de Julio de 2009 . Disponible en:
<http://docs.sun.com/app/docs/doc/820-4494/abppa?a=browse>
- [SUN2009a] SUN Technologies (2009). Learning GlassFish for Tomcat Users. Extraído el 20 de junio de 2009 desde
www.sun.com/offers/details/GlassFish_Tomcat.xml
- [SUN2009b] Sun Developer Network (2009). Sun Java Wireless Toolkit for CLDC Download. Extraído el 30 de noviembre de 2009 desde
<http://java.sun.com/products/sjwtoolkit/download.html>
- [UNI20009] Universidad Politécnica de Madrid, Instalar GlassFish V2.1 en Windows Xp y Vista. Extraído el 30 de julio de 2009 desde
http://blogs.sun.com/upm/entry/instalar_glassfish_v2_1_en
- [VAN2007] Van de Kar, E. (2007) Designing Mobile Service Systems
- [W3C2007] W3C. (2007) Version 1.2 Part 1: Messaging Framework. Extraído el 22 de junio de 2009 desde <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/#soapfeature>
- [ZUI2002] Zuidweg, J. (2002). Next Generation Intelligent Networks. Boston: Artech House.

Anexos

Anexo 1: Documentación Java de las Clases del Cliente Prototipo

Se presenta en este anexo el código fuente del cliente prototipo.

Anexo 2: Documentación Java de las Clases de la Aplicación Web del Servidor

Se presenta en este anexo el código fuente de la aplicación web del servidor.

Anexo 3: Scripts del modelo de Base de Datos

Se presenta en este anexo los scripts necesarios para la implementación de la Base de Datos.

Anexo 4: Análisis Económico

Durante el período agosto – octubre de 2009 la Población Económicamente Activa de Lima Metropolitana asciende a 4 millones 488 mil 300 personas, asimismo, la densidad de líneas de uso móvil para el Departamento de Lima es 123,8% para el mes de setiembre de 2009. [INE2009], [OSI2009]

Se considerará un factor de uso del 15.3% anual, este valor fue obtenido de la media geométrica del crecimiento durante el período 2003 – 2009. [INE2009], [OSI2009]

De los datos previamente mostrados se obtiene el número de usuarios nuevos para el proyecto.

$$[PEA] \times \text{Densidad} \times \text{Factor de Uso}$$

$$4\,488\,300 \text{ usuarios} \times 100\% \times 15.3\%$$

Luego de resolver se obtiene usuarios es igual a 686 710 que serán cubiertos en los cuatro años de duración del proyecto.