



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas

**(APLICACIÓN WAP PARA LA CONSULTA DE SALDOS DE LOS
CLIENTES DE COMERCIAL JAHHER)**

**Trabajo de graduación previo a la obtención del título de
Ingeniero de Sistemas**

**Autores: María Cristina Ortega Toledo
Juan Diego Montero Armijos**

Director: Ing. Marcos Orellana Cordero

Cuenca, Ecuador

2006

Índice de Contenidos

ÍNDICE DE CONTENIDOS	II
ÍNDICE DE ILUSTRACIONES	IV
ÍNDICE DE CÓDIGO	IV
RESUMEN	V
ABSTRACT	VI
 INTRODUCCIÓN.....	 1
 CAPÍTULO I:	 4
GENERALIDADES SOBRE LA TECNOLOGÍA A UTILIZAR.....	4
INTRODUCCION	4
1.1 GENERALIDADES DE TECNOLOGÍA WAP	5
1.1.1 Definición de WAP	5
1.1.2 Sintaxis general	10
1.1.3 Etiquetas del lenguaje	11
1.2 GENERALIDADES DE DESARROLLO DE PÁGINAS WEB DINÁMICAS.....	17
1.2.1 Interacción con páginas dinámicas basadas en el lenguaje PHP	17
1.2.2 Conexión de una Base de Datos Mysql con PHP.....	20
1.2.3 Funciones principales de PHP aplicadas a lenguaje WML	24
1.3 ADAPTACIÓN DE UN SERVIDOR WEB PARA EL SOPORTE DE PÁGINAS WAP	26
1.3.1 Configuración del servidor Linux para la interpretación de etiquetas WML.....	26
1.4 CONCLUSIONES.....	27
 CAPITULO II:.....	 28
ANÁLISIS DE RIESGOS.....	28
INTRODUCCION	28
2.1 ENFOQUE DE RIESGOS PRINCIPALES ORIGINADOS POR LA NATURALEZA DE LA APLICACIÓN WAP.....	29
2.2 MÉTODOS DE PREVENCIÓN DE POSIBLE DIVULGACIÓN DE INFORMACIÓN.	33
2.3 CONCLUSIONES.....	33
 CAPÍTULO III	 34
ANÁLISIS DE LOS DATOS A PRESENTAR.....	34
INTRODUCCIÓN	34
3.1 MODELO ENTIDAD – RELACIÓN DE LA BASE DE DATOS A USAR.	35
3.2 DICcionario DE DATOS.	36
3.3 DIAGRAMAS DE FLUJO DE INFORMACIÓN.....	38
3.3.1 Consulta de Comprobantes.....	38
3.3.2 Consulta de Saldo	39
3.3.3 Consulta Cuotas del Comprobante	39
3.4 CONCLUSIONES.....	40
 CAPÍTULO IV.....	 41
PROGRAMACIÓN Y HERRAMIENTAS.....	41
INTRODUCCION	41
4.1 PROGRAMACIÓN.....	42
4.1.1 Método de Obtención de información para el almacenamiento en la base de datos.	42
4.1.2 Método de Autenticación de un usuario al acceder al sitio WAP	44
4.1.3 Método de Búsqueda de la información requerida por el usuario.	45
4.1.4 Método de Presentación de la Información al Usuario.	46
4.2 HERRAMIENTAS.....	50

4.2.1	<i>Descripción de las herramientas de diseño.</i>	50
4.2.2	<i>Descripción de las herramientas de programación.</i>	51
4.2.3	<i>Descripción de las herramientas de validación y simulación.</i>	51
4.3	CONCLUSIONES.....	52
CAPÍTULO V:.....		53
PRUEBAS.		53
INTRODUCCION		53
5.1	PRUEBAS DE CONFIGURACIÓN DEL SERVIDOR DE LINUX	54
5.2	PRUEBAS DE OBTENCIÓN DE INFORMACIÓN.....	55
5.3	PRUEBAS DE LA APLICACIÓN WAP EN UN DISPOSITIVO DE TELEFONÍA MÓVIL.....	58
5.4	CONCLUSIONES.....	58
CAPÍTULO VI.....		60
CONCLUSIONES Y BIBLIOGRAFÍA		60
6.1	CONCLUSIONES.....	60
6.2	APÉNDICES.....	62
6.2.1	APENDICE 6.2.1: DICCIONARIO DE DATOS DE LA BASE DE DATOS DE LA APLICACIÓN.....	62
6.2.2	APENDICE 6.2.2: DIAGRAMA DE FLUJO DE INFORMACION DE CONSULTA DE COMPROBANTES.....	64
6.2.3	APENDICE 6.2.3: DIAGRAMA DE FLUJO DE INFORMACION DE CONSULTA DE SALDOS.....	65
6.2.4	APENDICE 6.2.4: DIAGRAMA DE FLUJO DE INFORMACION DE CONSULTA DE CUOTAS.....	66
6.3	BIBLIOGRAFÍA	67
ANEXOS		71
ANEXO 1: COSTOS DE LAS OPERADORAS A NIVEL NACIONAL POR BRINDAR EL SERVICIO WAP		71
ANEXO 2: PROGRAMAS FUENTE DE LA APLICACIÓN DESARROLLADA		73
ANEXO 3: GLOSARIO DE TERMINOS		73

Índice de Ilustraciones

Figura 1.1: Política de distribución de servicios entre diferentes agentes de usuario	7
Figura 1.2: Esquema de interconexión entre Cliente, Pasarela y Servidor Web	9
Figura 1.3: Diagrama de la solicitud de datos y ejecución del lenguaje PHP	17
Figura 3.1 Modelo Entidad – Relación	35
Figura 4.1: Splash de presentación Macromedia Dreamweaver MX	50
Figura 4.2: Splash de presentación Fireworks MX	51
Figura 5.1: Mensaje de error capturado del simulador wap http://www.wapsilon.com	54
Figura 5.2: Imagen de petición de Cédula	56
Figura 5.3: Imagen del error al validar la Cédula	56
Figura 5.6: Imagen con listado de Comprobantes	57
Figura 5.5: Imagen que muestra información del Saldo	57
Figura 5.6 Imagen que muestra información de Cuotas	57

Índice de Código

R1.1: Cabecera PHP para la interpretación de código WML	18
R1.2: Configuración de un servidor Web Linux para soportar código WML	27
R4.1: Sentencia SQL -Oracle para obtención de información nueva de la tabla Clientes	42
R4.2: Sentencia PHP para cargar los datos nuevos en la tabla Clientes	42
R4.3: Sentencia SQL-Oracle para obtener los nuevos datos de la tabla Comprobante	43
R4.4: Sentencia PHP para cargar los datos nuevos en la tabla Comprobante	43
R4.5: Sentencia SQL-Oracle para obtener los nuevos datos de la tabla Cuotas	44
R4.6: Sentencia PHP para cargar los datos nuevos en la tabla Cuotas	44
R4.7: Sentencia SQL-Mysql para obtener los códigos de los comprobantes pendientes del cliente	45
R4.8: Sentencia SQL-Mysql para obtener los datos de los comprobantes pendientes del cliente	46
R4.9: Sentencia SQL-Mysql para obtener los datos de las cuotas correspondientes al comprobante consultado por el cliente	46
R4.10: Código PHP-WML utilizado para mostrar la lista de comprobantes para el cliente	47
R4.11: Código PHP-WML utilizado para mostrar los datos en la pantalla de Saldos del cliente	48
R4.12: Código PHP-WML utilizado para mostrar los datos en la pantalla de Cuotas del comprobante pendiente de pago por el cliente	49
R5.1: Código WML utilizado para generar una página sencilla de prueba de tipo WML puro	54

Resumen

La siguiente monografía trata sobre la creación de un sitio WAP (*Wireless Application Protocol*) o Protocolo de Aplicaciones Inalámbricas, en el cual los clientes de IMPOCOMJAHHER Cia. Ltda., al navegar por Internet, puedan consultar sus saldos con la institución mediante un dispositivo móvil, como un teléfono celular, que soporte WAP, cuente con un micronavegador y tenga contratado el servicio de e-navegación cuyo costo dependerá de la operadora. (Ver Anexo 1).

Los objetivos de un software de esta naturaleza se enfocan en cubrir la necesidad de información actualizada de saldo, fecha de vencimiento y valor de las cuotas a las cuales se han diferido los pagos de los clientes. La dirección del sitio es <http://www.jaher.com.ec/saldos>. Para ingresar a la consulta el usuario debe digitar su número de identificación, accediendo a una interfaz amigable y sencilla, desarrollada en lenguajes de programación como WML, PHP e interactuando con una Base de Datos de Mysql. La aplicación se encuentra alojada en el servidor Web que utiliza la Empresa.

La Consulta WAP se utilizará como una herramienta para involucrar al Cliente con la Empresa al proporcionarle información sobre los pagos que realiza y aquellas cuotas que tiene pendientes. La implementación de este servicio WAP ha permitido a sus desarrolladores, crear un medio accesible independiente de factores como tiempo y espacio.

Abstract

The following monograph treats on the creation of a site WAP (Wireless Application Protocol) or Protocol of Wireless Applications, in which the clients of IMPOCOMJAHER Cia. Ltda., on having sailed along Internet, they can consult their balances with the institution by means of a mobile device, as a cellular telephone, which support WAP, one counts with a microbrowser and one has the service contracted of and navigation which cost will depend of their suppliers. (To see 1 Attached).

The objectives of a software of this nature are focused in to cover the balance brought up to date information need, due date and value of the quotas to which have differed the payments of the clients. The direction of the place is <http://www.jaher.com.ec/saldos>. To enter to the consultation the user should send his number of identification, agreeing to a simple and friendly interface, developed in programming languages as WML, PHP and interacting with a database of Mysql. The application is found lodged in the servant Web that utilizes the Company.

The Consultation WAP will be in use as a tool to involve the Client with the Company to provide him information about the payments that he realizes and those quotas that it has hanging to him. The implementation of this service WAP has allowed to this developers, to create an accessible way independent from factors as time and space.

Introducción

La posibilidad de interconectarse a través de redes inalámbricas, ha abierto nuevos horizontes que permiten comunicar más allá de las fronteras de espacio y de tiempo entre una organización y sus respectivos clientes. Por esto nos vemos en la obligación de crear herramientas que faciliten a los administradores de un sistema brindar un buen servicio de información, y a los clientes, darles la capacidad de acceder a estos datos, mediante un equipo de telefonía móvil. El desarrollo de un servicio de esta naturaleza es impulsado por el avance de las Telecomunicaciones, por el incremento de personas que cuentan con un dispositivo de telefonía móvil, que en nuestro país es aproximadamente del 50% de ciudadanos, cifra que va en incremento.

La siguiente monografía trata sobre el desarrollo de una aplicación basada en una tecnología relativamente nueva denominada WAP (*Wireless Application Protocol*) o Protocolo de Aplicaciones Inalámbricas. Este protocolo surge de la fusión de las Comunicaciones Inalámbricas y la Internet. Una de las posibilidades que brinda esta tecnología es la de acceder a información actualizada, veraz y confiable de un Servidor Web configurado para soportar código WML (*Wireless Markup Language*), lenguaje de programación acorde con aplicaciones WAP, y obtener los datos requeridos por la consulta rescatando información de una Base de Datos de Mysql mediante el uso del lenguaje PHP, que cual permite la ejecución de sentencias SQL.

El producto desarrollado involucra un sitio WAP, similar a un sitio Web, pero accesible mediante el uso de un dispositivo móvil, como podría ser un teléfono celular. Al visitar la siguiente dirección <http://www.jaher.com.ec/saldos> los clientes de la Empresa Comercial IMPOCOMJAHER Cía. Ltda. pueden realizar consultas online mediante la digitación de su identificador personal en la institución (ya sea su Cédula o su RUC) y acceder a información de los saldos de las facturas pendientes con la Empresa así como el valor y fecha de vencimiento de las cuotas a las cuales han sido diferidos sus pagos.

Al ser una aplicación que por su naturaleza puede ser accesible desde cualquier medio móvil que soporte la tecnología WAP, brindará a sus usuarios un medio de

información disponible independientemente del tiempo y espacio en el que se encuentre la persona. Esta fue una de las razones que nos llevó al desarrollo de una Consulta WAP de Saldos con el deseo de brindar al cliente de la institución un servicio que le proporcione datos actualizados, confiables, que cubran su necesidad de información y que de alguna manera agilicen los procesos de recuperación de pagos de los clientes a la Empresa.

Los procesos que involucra la aplicación se refieren puramente a la recuperación de información de dos tipos de Bases de Datos. En primera instancia se trata de la migración de los datos desde una Base de Oracle, fuente bajo la cual trabajan las aplicaciones del sistema que maneja la Empresa de manera diaria, hacia la fuente de datos con la que trabajan las aplicaciones WAP que es Mysql. Y, por otro lado, en el momento que el usuario realiza la petición de información mediante el uso de la Consulta WAP, se realiza una recuperación de los datos de la fuente de Mysql utilizando lenguaje SQL y ejecutando código PHP y WML para la presentación de los saldos y cuotas en la interfaz de usuario. El acceso al sitio WAP está limitado solamente a clientes de la institución. Dicha característica se logró mediante la programación de la aplicación de tal manera que valide la existencia del usuario final dentro de los registros de clientes de Comercial Jaher.

Al ser WML un lenguaje de programación relativamente nuevo, que surge aproximadamente en el año de 1997, y de igual manera WAP un protocolo en desarrollo y crecimiento, la mayoría de bibliografía disponible se encuentra en sitios Web, razón por la cual casi todas las referencias bibliográficas de esta monografía se orientan a sitios del Internet disponibles de manera gratuita para el público en general.

Este documento contiene información relevante para el desarrollo de la aplicación WAP, ya que consta de un primer capítulo donde trata de las generalidades teóricas sobre los lenguajes utilizados definiendo las características y propiedades de los mismos, así como la configuración de un servidor para el soporte de código WML. Un segundo capítulo que trata esencialmente sobre los riesgos de una aplicación de esta naturaleza y los mecanismos empleados para cubrir las vulnerabilidades del sistema. Un tercer capítulo que presenta un estudio detallado de la base de datos

utilizada, entidades y procesos necesarios por la aplicación. Estos datos están disponibles también gráficamente en la sección de anexos del mismo documento.

De igual manera siguiendo con el esquema, la monografía consta de un cuarto capítulo que describe de manera detallada los métodos utilizados para la autenticación, búsqueda, obtención y presentación de información al usuario. También consta de un quinto capítulo donde describe las pruebas realizadas para reforzar la aplicación y un sexto capítulo que define referencias bibliográficas, define anexos y conclusiones de la presente monografía.

CAPÍTULO I:
GENERALIDADES SOBRE LA TECNOLOGÍA A UTILIZAR.

INTRODUCCION

El siguiente capítulo tratará sobre generalidades de la tecnología WAP presentando al lector una pequeña reseña histórica de los orígenes de este protocolo y su arquitectura. Definiciones básicas importantes para entender el siguiente documento, nomenclatura de términos utilizados con frecuencia en el desarrollo de esta monografía. De igual manera se describe la sintaxis del lenguaje WML, definiendo las etiquetas que posee el mismo así como todas sus opciones y posibles valores con sus significados prácticos.

Fue muy importante para el desarrollo de la aplicación el conocimiento de temas tratados en este capítulo como lo son el conocimiento del lenguaje de programación PHP para la generación de páginas WAP dinámicas, es decir con recuperación de información basándose en lenguaje SQL, así como trabajar con bases de datos, en nuestro caso con Mysql. Se define de manera clara y sencilla sentencias utilizadas para la recuperación de información así como se describirá el proceso para la configuración del Servidor Web, con sistema operativo *Red Hat Linux*, para que soporte la generación de páginas WML.

1.1 Generalidades de Tecnología WAP

1.1.1 Definición de WAP

WAP (*Wireless Application Protocol*)

WAP es un protocolo basado en los estándares de Internet que ha sido desarrollado para permitir la navegación en Internet a través de dispositivos móviles. En 1995 Ericsson inicio un proyecto que se proponía un protocolo general que sumara servicios a redes móviles. El protocolo fue llamado *Intelligent Terminal Transfer Protocol* (ITTP) y manejaba la comunicación entre un nodo de servicio (donde la aplicación de servicio era implementada) y un móvil inteligente. El objetivo era lograr imponer a ITTP como un estándar en el segmento de servicios agregados a las redes móviles.

Durante 1996 y 1997, *Unwired Planet*, Nokia, y otro desarrollaron conceptos en la misma área. En el caso de *Unwired Planet* presentó en forma simultánea el *Handheld Device MarkupLanguage* (HDML) y *Handheld Device Transport Protocol* (HDTP). A semejanza de HTML, que es usado en la web, HDML proponía su uso para describir contenidos y contener interfaces visuales graficas, con la ventaja de estar optimizado para el acceso a Internet desde determinados dispositivos. De la misma forma HDTP podía ser considerado un equivalente del protocolo estándar de Internet HTTP y buscaba erigirse una vía apta para el manejo de transacciones cliente-servidor.

Por su parte en marzo de 1997, Nokia presento oficialmente el concepto “*Smart Messaging*”, una tecnología de acceso a Internet diseñada para dispositivos GSM (*Global System for Mobile Communications*). Es mismo año el protocolo vería la luz, cuando Ericsson, Nokia, Motorola y *Unwired Planet* forman el WAP Forum con la clara intención de unificar conceptos y desarrollar nuevas aplicaciones de amplia aceptación para la industria de las telecomunicaciones inalámbricas así como también contar con el esfuerzo coordinado de más de un centenar de miembros de diversos sectores de la industria. De esta forma WAP sería un estándar global que no

estaría controlado por ninguna compañía, lo que aseguraría su apertura y universalidad.¹

Actualmente WAP esta en su versión 2.0. Con esta versión se adoptan TCP y XHTML *Mobile Profile* (también conocido como XHTML MP) como estándar y soporta WAP CSS, una versión propia de hojas de estilo. Con WAP desaparece una de las capas típicas de WAP 1.1, el “*gateway WAP*”, el cual conecta Internet y la red móvil. Lo que se pretende con la tecnología WAP es que desde cualquier teléfono celular que soporte y tenga contratado el servicio de navegación por Internet, se pueda acceder a la consulta de información así como realizar operaciones de comercio electrónico.

Los elementos más importantes de esta especificación son:

- Un lenguaje de códigos basado en XML para la definición y presentación de contenidos y la interacción del usuario, denominado WML (*Wireless Markup Language*).
- Un lenguaje de *scripts* complementario, denominado WMLScript, para la realización de actividades de procesamiento en el lado del cliente (agente de usuario WAP).
- La especificación de un micronavegador en la que se define cómo se deben interpretar tanto el código WML como el WMLScript en el terminal móvil y que, en general, gestiona la interfaz de usuario final del servicio WAP.
- Un entorno para aplicaciones de telefonía (WTA, *Wireless Telephony Applications Framework*) que permite a los operadores la integración de funciones de telefonía del propio dispositivo móvil con el micronavegador incorporado.
- Una pila de protocolos estándar que hace posible el transporte real de los contenidos, y que está basada en la arquitectura de protocolos existentes en el WWW (*World Wide Web*). Estos protocolos han sido diseñados para operar sobre distintos tipos de servicios portadores de diferentes niveles de calidad de servicio (ancho de banda, latencias, tasas de error, etc.).

¹ DATTATEC.COM. Seminario sobre Historia de la tecnología WAP. 26 diciembre 2005. Disponible en World Wide Web: http://seminarios.dattatec.com/seminario_281205.doc

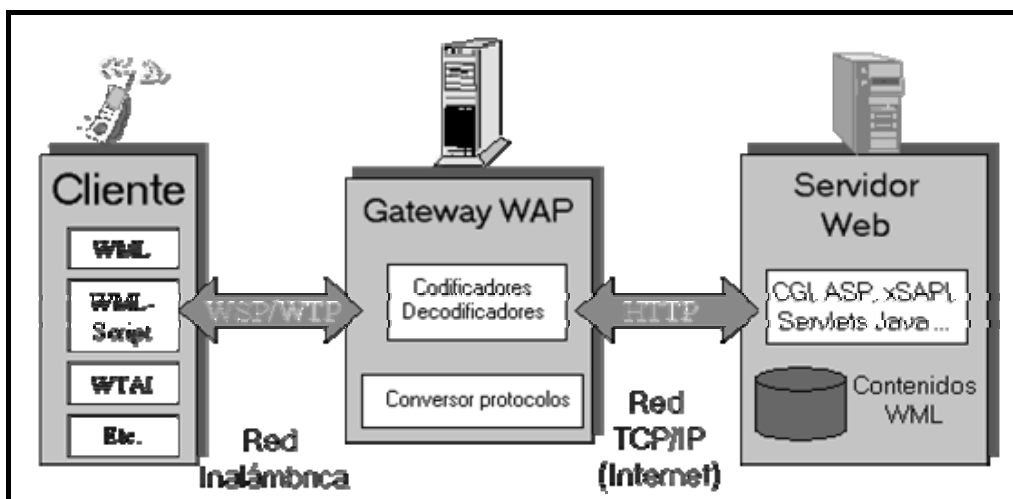
Agentes de usuario WAE

El nivel de aplicación de la arquitectura de protocolos de WAP, WAE (*Wireless Application Environment*) se considera como la base de un entorno de desarrollo de propósito general para aplicaciones WAP, en el que se combinan tecnologías similares a las usadas en el WWW y tecnologías relacionadas con Telefonía Móvil.

A nivel software, WAE se divide en dos niveles lógicos:

- Agentes de usuario: micronavegadores, agendas telefónicas, editores de mensajes, etc.
- Servicios y Formatos comunes, accesibles a los agentes de usuario, como WML, WMLScript, formatos de imágenes, de calendarios y agendas telefónicas, etc.

WAE separa los agentes de usuario de los servicios, y asume la existencia de un entorno con múltiples agentes de usuario, sin que ello suponga ninguna implicación para la implementación. En la mayoría de los casos se suelen combinar todos los servicios en un único agente de usuario, mientras que en otros se puede optar por una política de distribución de servicios entre diferentes agentes de usuario¹ (Figura 1.1).



(Disponible en Web: http://www.webexperto.com/articulos/archivos/art38/wml_ge_1.gif)

Figura 1.1: Política de distribución de servicios entre diferentes agentes de usuario

¹ RIOS, Sergio. Generación Dinámica de Contenidos WAP para Terminales Móviles [en línea]. 18/06/2002. Disponible en World Wide Web: <http://www.aui.es/biblio/libros/mi2000/Sergio%20Rios.htm>

Las características principales de WML son:

- Soporte para imágenes y texto, con posibilidad de texto con formato.
- Tarjetas agrupadas en barajas. Una pagina WML es como una página HTML en la que hay una serie de cartas, al conjunto de estas cartas se les suele llamar baraja.
- Posibilidad de navegar entre cartas y barajas de la misma forma que se navega entre páginas Web.
- Manejo de variables y formularios para el intercambio de información entre el teléfono celular y el servidor.

WML es un lenguaje de marcas similar al HTML. WML es compatible con XML 1.0. Las páginas WML son llamadas barajas ya que están compuestas por cartas, un navegador WAP, solo puede mostrar una carta al mismo tiempo.

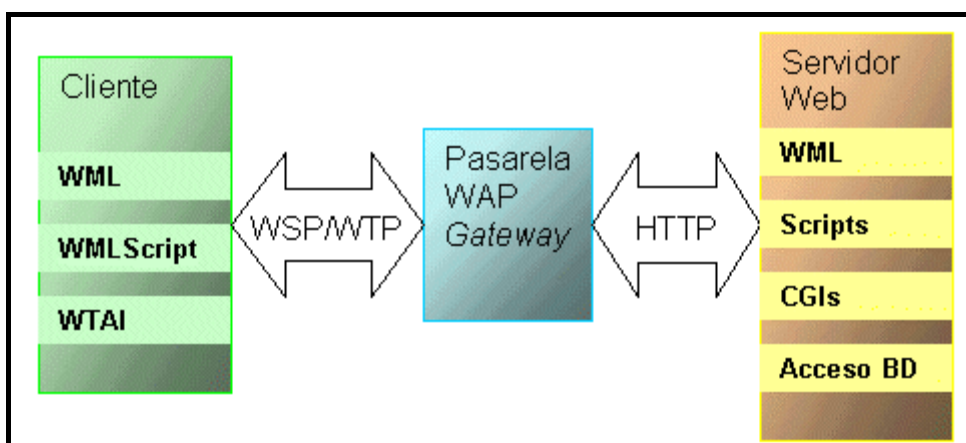
Modelo general de operación de WAP/WAE

El modelo típico de operación WAP se basa en el paradigma cliente/servidor multinivel (Figura 1.2):

- **Cliente o dispositivo WAP:** el cliente será el micro-navegador WML de un terminal móvil, es decir está provisto de un micro-navegador que constituye la interfaz de usuario para realizar las funciones de navegación y además interpreta las páginas WML. Este será el equivalente al navegador de un PC en el Internet "fijo" que todos conocemos.
- **Pasarela:** la pasarela constituye la interfaz entre la red inalámbrica y la red física. Es lo que se conoce como el WAP *Gateway*. Realiza 2 funciones básicas: conversión de protocolos (de HTTP a WSP/WTP y viceversa) y codificación / decodificación de las páginas WML. Estos procesos permiten la adaptación a la red inalámbrica del protocolo y de los contenidos.

- **Servidor Web:** en toda arquitectura cliente / servidor tenemos un servidor encargado de procesar las peticiones del cliente y enviar las páginas solicitadas.

Aquí residen las páginas, así como cualquier otra lógica basada en CGI, acceso a bases de datos o lenguajes de *script*. WAP es compatible con servidores HTTP 1.1, lo que facilita la adopción del estándar por parte de los proveedores de contenidos Web ya existentes.



(Disponible en Web: <http://www.elcodigo.com/tutoriales/wap/arquitecturawap.gif>)

Figura 1.2: Esquema de interconexión entre Cliente, Pasarela y Servidor Web

Así, cuando el usuario suministra una URL en el cliente (terminal móvil), un agente de usuario WAE usa el protocolo WSP (*Wireless Session Protocol*) para enviar solicitudes de contenidos y servicios a través de una red inalámbrica, en formato comprimido (y opcionalmente seguro), al *gateway*, el cual convierte dichas solicitudes a formato HTTP (o HTTPS si se utilizan transacciones seguras¹) no comprimido y las envía a un servidor de contenidos Web. Este servidor responde, mediante HTTP o HTTPS, enviando un documento u objeto directamente en formato WML, WMLScript, WBMP o bien en formato HTML, en cuyo caso habría que aplicar un filtro de conversión a WML.

Cuando el *gateway* WAP recibe la respuesta, compila y comprime el objeto WMLScript o WML, respectivamente, en formato binario compacto y la devuelve al agente de usuario.

¹ RIOS, S, "Intranets Corporativas: Arquitecturas y Protocolos de Seguridad para comercio electrónico seguro en Internet y migración a Extranets" Libro de Ponencias del I Forum Internet-Empresa, Mayo 1998 Instituto Catalán de Tecnología (ICT)

Algunas de las funciones del *gateway* que se pueden citar son:

- **Pasarela de protocolos:** se traducen solicitudes basadas en protocolos WAP (WSP, WTP, WTLS y WDP) a solicitudes de protocolos WWW (HTTP y TCP/IP), realizando el proceso análogo para el transporte de las respuestas al cliente originario de las solicitudes.
- **Codificación y decodificación de contenidos:** Se transforman los contenidos WAP a formatos compactos y codificados para reducir el tamaño de los datos en tránsito por la red celular, así como para minimizar la capacidad de procesamiento necesaria en el cliente para el tratamiento de los datos.

Ahora la configuración planteada anteriormente no es la única, ya que existen servidores de contenidos y servicios WTA (*Wireless Telephony Application*) que no necesitan realizar conversión de protocolos, y son capaces de responder directamente a solicitudes WAP del cliente. Los servidores WTA permiten ofrecer acceso WAP a determinadas características de la infraestructura de comunicaciones del operador de Red, y para poder implementar los servicios ofrecidos se usan interfaces propietarias de servidor o bien interfaces abiertas y bien documentadas como la API de *Servlets Java*. Por otra parte, y por completitud, mencionar que en el cliente (terminal móvil), para operaciones locales de control de llamadas (recepción, iniciación y finalización) y de acceso a listines telefónicos, se utiliza la interfaz WTAI (*Wireless Telephony Application Interface*).

1.1.2 Sintaxis general

A las páginas WML se les suele llamar barajas porque están compuestas por cartas. Una carta es la unidad de información que un navegador WAP puede mostrar. El navegador nos permite pasar de una carta a otra dentro de la baraja para así poder acceder a todas las cartas. Una baraja de cartas se marca con las etiquetas `<wml>...</wml>` dentro de estas marcas irán todas las cartas de la baraja. Puede contener las etiquetas *head*, *template* y es obligatorio que al menos tenga una

etiqueta *card*. Una carta es la unidad de información que se muestra en un navegador WAP, una carta puede contener texto, campos de datos, enlaces.¹

El conjunto de caracteres definido por defecto es el ISO/IEC-10646 que es el mismo que el Unicode 2.0 WAP soporta los siguientes subconjuntos de Unicode:

- UTF-8
- ISO-8859-1 o ISO Latin-1
- UCS-2

Se definen en la etiqueta `<?xml version="1.0" encoding="UTF-8"?>`.²

Las imágenes en WAP deben tener el formato soportable por WML que es WBMP. Estas pueden ser convertidas a dicho formato con uso del algún software como lo es el utilizado en nuestra aplicación: Macromedia FireWorks MX.

Cabe destacar que WML es sensible a mayúsculas/minúsculas, esto incluye las etiquetas, los atributos, los identificadores, las variables, etc.

1.1.3 Etiquetas del lenguaje

Todas las etiquetas en WML se escriben en minúsculas. Hay dos tipos de etiquetas, las que contienen elementos, para lo cual hay una etiqueta de inicio y otra de fin. Los atributos de las etiquetas han de ir siempre en la etiqueta de inicio.

`<etiqueta>` Inicio

`</etiqueta>` Fin

Y las etiquetas que no contienen elementos que tienen el siguiente formato:
`<etiqueta/>`

A continuación se detallan algunas etiquetas para el manejo de formato de página.

`<wml></wml>`: Inicio y fin del código de la página WAP.

¹ GRACIA, Joaquín. WAP WML Sintaxis de WML. Última modificación: 06 de Junio de 2005. España. ©1998-2004. Disponible en World Wide Web:<http://www.webestilo.com/wml/cap03.phtml>

² GRACIA, Joaquín. WAP WML Barajas y cartas. Última modificación: 06 de Junio de 2005. España. ©1998-2004. Disponible en World Wide Web:<http://www.webestilo.com/wml/cap01.phtml>

`<card [atributos]></card>`: Define el inicio y fin de una baraja. Una o varias barajas forman una aplicación wml.

Opciones:

- *id*: es común para todas las etiquetas WML y nos permite identificar un elemento dentro de un documento WML.
- *title*: permite indicar el título de la carta.
- *onenterforward*: Dirección a la que se va al ocurrir este evento (ir adelante) usando la etiqueta go.
- *onenterbackward*: Dirección a la que se va al ocurrir este evento (ir atrás) usando la etiqueta go.

`<go/>`: es una etiqueta para definir una dirección que se va a visitar.

Opciones:

- *href*: Dirección a la que se quiere ir
- *method*: Utilizado para enviar datos al servidor. Pueden ser get o post.

`<do></do>`: define un botón para el acceso a otra carta o pantalla WAP.

Opciones:

- *name*: *Nombre del objeto.*
- *label*: *Texto que aparece al pulsar el botón options en el navegador para iniciar el evento.*
- *type*: *Define el tipo de evento que se va a realizar.*

`<prev/>`: ir a la página anterior.

`<noop/>`: ninguna operación.

`<refresh/>`: resetea los valores.

`<a></a>`: se utiliza para definir un link. De igual manera en esta etiqueta es posible el paso de variables como parámetros de la siguiente manera:

```
<a href='url?nombre_variable1=$valor1&nombre_variable2=$valor2'></a>
```

Opciones:

- *title*: define el texto que se muestra en el link.
- *href*: define la dirección a la que se accederá con dicho link.

`<img/>`: utilizada para insertar imágenes.

Opciones:

- *src*: Lugar donde se encuentra la imagen
- *alt*: Texto alternativo. Se recomienda usarlo siempre
- *hspace, vspace*: Espacio en blanco entre la imagen y el resto de contenidos. Si es expresado en porcentaje, representa el espacio total de visión del browser, no las medidas de la imagen
- *localsrc*: Lugar donde se encuentra la imagen
- *height y width*: Alto y ancho de la imagen.

`<table></table>`: utilizado para crear tablas.

Opciones:

- *title*: título del elemento:
- *cols*: número de columnas.
- *align*: Alineación del texto dentro de su celda

`<tr></tr>`: definición de una línea

`<td></td>`: definición de una columna

`<input/>`: utilizada para el ingreso de datos.

Opciones:

- *name*: Nombre de la variable que va a tomar los datos que introduzca el usuario
- *value*: Valor que aparece en el campo input por defecto.
- *type*: Forma en la que se introducen los datos
- *format*: Formato en que se acepta la introducción de datos y puede definirse de las siguientes maneras:

X: únicamente en mayúsculas.

x: únicamente en minúsculas.

A: mayúsculas y signos de puntuación.

a: minúsculas y signos de puntuación.

M: mayúsculas, dígitos y signos de puntuación.

m: minúsculas, dígitos y signos de puntuación.

N: dígitos.

- *size*: Especifica el ancho en caracteres del input.
- *maxlength*: Especifica el número máximo de caracteres a introducir
- *title*: Título usado en la presentación del objeto
- *tabindex*: Especifica el orden en que se accede a los distintos objetos dentro de una *card*. Por defecto de arriba abajo.
- *emptyok*: Esto define si el campo es o no obligatorio. Para que sea obligatorio, asigne al atributo el valor *TRUE*. Para que no sea obligatorio, asigne el valor *FALSE*.

<select></select>: utilizada en formularios para escoger entre opciones.

Opciones:

- *name*: Nombre de la variable que va a tomar los datos que introduzca el usuario.
- *value*: Valor que tiene la variable por defecto.
- *multiple*: Posibilidad de seleccionar más de una opción. Si es así el valor de la variable tendrá ambos valores separados por comas (;).
- *ivalue*: Determina el index de la opción que queremos que esté seleccionada por defecto. El primer ítem es 0.
- *title*: Título usado en la presentación del objeto.

<option></option>: define las opciones que se desea asignar a un control select.

Opciones:

- *value*: Valor que toma la variable al seleccionar este ítem
- *title*: Título usado en la presentación del objeto
- *onpick*: Evento que ocurre al seleccionar o deseleccionar un ítem.

<p></p>: formateo de texto.

Opciones:

- *align*: define la alineación del texto: *left*, *right* y *center*.

: formatea el texto a enfatizado.

: formatea el texto a fuerte.

<i></i>: formatea el texto a cursiva.

: formatea el texto a negrita.

<u></u>: formatea el texto a subrayado.

<big></big>: formatea el texto a letras grandes.

<small></small>: formatea el texto a letras pequeñas.

: define el salto de línea.

<anchor></anchor>: esta etiqueta es lo que permite que un documento WML se enlace consigo mismo o con otro.

<postfield></postfield>: esta etiqueta se utiliza para transformar los datos en formato de escape para enviarlos al servidor generalmente cuando se reciben de un “input “. Hay que tener en cuenta que se utiliza dentro de una etiqueta “go”.

1.2 Generalidades de desarrollo de páginas WEB dinámicas

1.2.1 Interacción con páginas dinámicas basadas en el lenguaje PHP

El lenguaje PHP es un lenguaje de programación de estilo clásico con variables, sentencias condicionales, bucles, funciones, etc. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a *JavaScript* o al lenguaje “C”, para aquellos que conocen estos lenguajes. Pero a diferencia de *Java* o *JavaScript* que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El *script* de PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero de igual forma podría ser una pagina WML.



(Disponible en Web: <http://www.webestilo.com/graficos/php1.gif>)

Figura 1.3: Diagrama de la solicitud de datos y ejecución del lenguaje PHP.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador WAP lo soporte, pero para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP¹.

Para que una página WML se ejecute debe estar almacenada en el servidor pero nombrada con extensión PHP, para que el código pueda ser interpretado por el Servidor. Para ello existe una cabecera primordial en el desarrollo de una página WML que permitirá la interacción del servidor con la aplicación WAP:

```
<?
header ("Content-Type: text/vnd.wap.wml");
echo "<?xml version=\"1.0\"?>";
echo "<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\"
\"http://www.wapforum.org/DTD/wml_1.1.xml\">";
?>
```

R1.1: Cabecera PHP para interpretación de código WML.

Estas líneas de código permitirán que la página con extensión PHP sea interpretada por navegadores WAP.

A partir de la siguiente línea de código en la página PHP pueden intercalarse código WML y bloques de código PHP dependiendo de nuestras necesidades, pero hay que recalcar que se deben respetar las particularidades del WML como lo son sus etiquetas y el orden de las mismas. De igual forma el código PHP que se utilice es el mismo que se puede utilizar para el desarrollo de una aplicación HTML. Se puede hacer uso de las variables que son contenedores de información, en el que podemos almacenar números enteros, números decimales, caracteres, etc. El contenido de las variables se puede leer y se puede cambiar durante la ejecución de la página PHP.

Para la lectura del contenido de las variables que se envían entre páginas PHP podemos hacer uso de la siguiente sentencia si es que se utiliza el método POST para el envío de datos al servidor:

```
$nueva_variable = $_POST['variable_del_formulario'];
```

¹ GRACIA, Joaquín. Cómo aprender PHP fácilmente. Última modificación: 6 de Junio de 2005. España. ©1998-2004. Disponible en World Wide Web: <http://www.webestilo.com/php/php00.phtml>

De esta forma podemos recuperar el valor de las variables que se declaran y que se envían al servidor WEB.

Debemos conocer que en PHP todas las variables comienzan con el símbolo de *string* (\$), no es necesario definir una variable antes de usarla y tampoco tienen tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres. Existen 2 tipos de variables en PHP, las variables locales que solo pueden ser usadas dentro de funciones y las variables globales que tienen su ámbito de uso fuera de las funciones, podemos acceder a una variable global desde una función con la instrucción:

```
global nombre_variable
```

En PHP existe la posibilidad del uso de operadores de tipo Aritmético, de Comparación, Lógicos. También podemos utilizar los Condicionales, definir Bucles. En lo que se refiere a mostrar los datos en pantalla por lo general se utiliza el comando *Echo* pero este es limitado por ello también existe la posibilidad de utilizar la sentencia *Print* para el formateo especial de texto. También las funciones predefinidas del lenguaje para el manejo de Texto o la definición de Funciones propias de usuario están habilitadas para el desarrollo de páginas WML que utilizan código PHP.

Por ejemplo utilizamos funciones como:

strchr: esta función encuentra la primera aparición de un carácter en una cadena dada.

```
strchr ( string_donde_buscar, string_buscado)
```

1.2.2 Conexión de una Base de Datos Mysql con PHP

Para la conexión de PHP con una base de datos de Mysql podemos utilizar las siguientes sentencias:

mysql_connect

```
int mysql_connect(string [hostname] , string [username] , string [password] );
```

Devuelve: un identificador de conexión, o FALSE en caso de error.

sql_connect establece una conexión a un servidor de MySQL. Todos los argumentos son optativos, y si no se especifican, los valores por defecto son (' el localhost', nombre del usuario del usuario que posee el proceso del servidor, la contraseña vacía). La cadena hostname también puede incluir un número del puerto, "hostname:port".

En caso de realizar una segunda llamada a mysql_connect con los mismos argumentos, no se establecerá ninguna nueva conexión, sino se devolverá el identificador de conexión de la ya existente. La conexión al servidor se cerrará en cuanto la ejecución del *script* acabe, a menos que la cerremos antes con la función mysql_close.

mysql_affected_rows

```
int mysql_affected_rows(int [link_identifier] );
```

mysql_affected_rows devuelve el número de filas afectado en el último SELECT, UPDATE o DELETE aplicado al servidor asociado con el identificador de conexión especificado. Si no se especifica un identificador de conexión, se asume le de la última conexión abierta.

Este orden no es eficaz para las instrucciones SELECT, sólo en instrucciones que modifican archivos. Para recuperar el número de filas vuelto de un SELECT, usa mysql_num_rows.

mysql_close

```
int mysql_close(int [link_identifier] );
```

Devuelve: TRUE si se ha cerrado correctamente, FALSE en caso de error.

mysql_close cierra la conexión con la base de datos MySQL asociada al identificador de conexión especificado. Si no se especifica un identificador de conexión, se asume el de la última conexión abierta.

Note que esta función no es normalmente necesaria en conexiones no-persistentes (abiertas con mysql_connect) ya que esta se cerrará automáticamente al final de la ejecución del *script* o página. La función mysql_close no cierra una conexión persistente (abierta con mysql_connect()).

mysql_fetch_array

```
array mysql_fetch_array(int result);
```

Devuelve un array con la información correspondiente al resultado de una consulta especificado por su identificador o 'false' si ya no hay más filas. Almacena datos a través de índices numéricos del array o también lo hace a través de índices asociativos, utilizando los nombres de los campos como claves. Si dos o más columnas del resultado tienen el mismo nombre de campo, la última es la que tiene preferencia. Para acceder a las demás es necesario utilizar el índice numérico o construir un alias para la columna.

mysql_free_result

```
int mysql_free_result(int result);
```

Sólo debería ser utilizada si la cantidad de memoria utilizada para almacenar el resultado de una consulta es muy grande. Cuando se ejecuta esta función, toda la memoria asociada al resultado se libera.

mysql_num_rows

```
int mysql_num_rows(string result);
```

Devuelve el número de filas del resultado de una consulta.

mysql_query

```
int mysql_query(string query, int [link_identifier] );
```

Ejecuta una consulta a la base de datos activa en el servidor asociado al identificador de conexión. Si no se especifica, se utiliza la última conexión abierta. Si no hay conexiones abiertas la función intenta establecer una.

Esta función devuelve *TRUE* o *FALSE* para indicar si las operaciones *UPDATE*, *INSERT* o *DELETE* han tenido éxito. Para la operación *SELECT* devuelve un nuevo identificador de resultado.

mysql_select_db

```
int mysql_select_db(string database_name, int [link_identifier] );
```

Devuelve: *true* si se ejecuta con éxito, *false* si se produjo un error.

Establece la base de datos activa en el servidor. Si no se especifica identificador de conexión se utiliza la última conexión abierta. Si no hay conexiones activas, la función intenta establecer una. A partir de la llamada a `mysql_select_db` las llamadas a `mysql_query( )` actúan sobre la nueva base de datos activa.

load data

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL]
```

```
INFILE 'file_name.csv' [REPLACE | IGNORE]
```

INTO TABLE tbl_name

[

FIELDS [TERMINATED BY '\t'] [[OPTIONALLY] ENCLOSED BY '']

[ESCAPED BY '\\']

]

[LINES [STARTING BY ''] [TERMINATED BY '\n']]

[IGNORE number LINES] [(col_name,...)]

Esta función lee filas desde un fichero de texto a una tabla a gran velocidad. Si se especifica la palabra *LOCAL*, se interpreta con respecto al cliente final de la conexión. Cuando se especifica *LOCAL*, el fichero es leído por el programa del cliente en el ordenador cliente y se envía al servidor. Si no se especifica *LOCAL*, el fichero debe estar en el ordenador servidor y es leído directamente por el servidor.

CONVERT(expresión as type):

Puede usarse para tomar un valor de un tipo y obtener uno de otro tipo.

Los valores de 'type' pueden ser uno de los siguientes:

- BINARY
- CHAR
- DATE
- DATETIME
- SIGNED {INTEGER}
- TIME
- UNSIGNED {INTEGER}

1.2.3 Funciones principales de PHP aplicadas a lenguaje WML

Las funciones mas comunes del lenguaje PHP utilizadas en esta aplicación son aquellas que sirven para la interacción de la aplicación WAP con la base de datos de Mysql, para la recuperación de información así como se han utilizado funciones predefinidas del lenguaje para el formateo de los textos que se presentarán en pantalla, entre otras. Por ejemplo podemos citar.

`$_POST['variable']`: se utiliza para recuperar los parámetros que viajan de un formulario hacia el servidor. El parámetro variable corresponde al nombre de la variable del formulario de la página WML.

`echo "mensaje"` : el comando hecho se utiliza para mostrar información en la pantalla ya sea texto o variables de la aplicación.

`sprintf(%especificador,argumento)`: al igual que el comando `echo sprintf` se utiliza para dar formato a datos que se presentarán en pantalla, según el patrón o especificador que se utilice:

Lista de Especificadores

- especificador b: el argumento es interpretado como un número entero, y presentado como un número binario.
- especificador c: el argumento es interpretado como un entero, y presentado como el caracter ASCII con dicho valor.
- especificador d: el argumento es tratado como un entero y presentado con notación decimal.
- especificador e: el argumento es tratado como un entero y presentado con notación exponencial.
- especificador f: el argumento es tratado como un *double* y presentado como un número de coma flotante.
- especificador o: el argumento es tratado como un entero, y presentado como un número octal.

- especificador x: el argumento es tratado como un entero y presentado como un número hexadecimal (con minúsculas).
- especificador X: el argumento es tratado como un entero y presentado como un número hexadecimal (con mayúsculas).
- especificador s: el argumento es tratado como una cadena y es presentado como tal.

Entre el signo % y el caracter especificador podemos incluir uno o dos números, ambos opcionales. Si existen, el primer número indica el número de caracteres (anchura) que debe ocupar la cadena formateada. *sprintf* devolverá como mínimo ese número de caracteres, o más si el resultado excede. Si no indicamos este número, *sprintf* devolverá los caracteres que sean necesarios.

`substr ( string cadena, int comienzo [, int longitud] )`: devuelve la porción de cadena especificada por los parámetros comienzo y longitud.

Si comienzo es positivo o 0, la cadena devuelta comenzará en dicho carácter de cadena (los caracteres empiezan a contarse en cero).

`round(valor,numero_decimales)`: se utiliza para truncar un valor numérico a un número de decimales determinado.

`while(condición)`: sirve para ejecutar una o varias sentencias mientras se cumpla el argumento de “ condición ”.

`mysql_connect("host","usuario","clave")`: se utilizará para crear una conexión con Mysql.

`mysql_select_db("base",$conexión)`: con este comando seleccionamos la base de datos a utilizar.

`mysql_query("sentencia_sql" , $conexión)`: sirve para ejecutar una consulta a la base de datos.

`mysql_num_rows($consulta)`: con este comando obtenemos el número de filas devuelto por la consulta.

`mysql_fetch_array($consulta)`: sirve para pasar una consulta hacia un array.

1.3 Adaptación de un servidor WEB para el soporte de páginas WAP

1.3.1 Configuración del servidor Linux para la interpretación de etiquetas WML

El servidor en el que se alojará nuestra aplicación WAP es un servidor que tiene un sistema operativo *Red Hat Linux* y que utiliza el *software* Apache 4.4.1 como Servidor Web. Para que el servidor de HTTP sepa que debe servir las páginas *.WML como páginas WAP debemos indicárselo, de lo contrario el servidor responderá como que el tipo de fichero es de texto. Para ello se debe configurar los tipos MIME que indicarán al servidor como se debe comunicar con el cliente cuando le solicitan una página WML.

Pasos para la configuración:

1. Ubicamos el archivo `httpd.conf`, normalmente ubicado en el directorio `/etc/http/conf`, archivo que contiene la configuración de tipos MIMES.
2. Editamos archivo y agregamos las siguientes líneas:

MIME Types for WAP

- `AddType text/vnd.wap.wml .wml`
- `AddType text/vnd.wap.wmlscript .wmls`
- `AddType application/vnd.wap.wmlc .wmlc`
- `AddType application/vnd.wap.wmlscriptc .wmlsc`
- `AddType image/vnd.wap.wbmp .wbmp`

3. Grabamos y cerramos el archivo. De esta manera se encuentra configurado el servidor Web para soportar código WML.

R1.2: Configuración de un servidor Web Linux para soportar código WML.

1.4 Conclusiones.

Como hemos podido notar la aplicación de esta tecnología no es compleja y únicamente se requiere las herramientas necesarias y los conocimientos previos especificados en el capítulo estudiado, por lo demás corre de cuenta de las compañías operadoras de telefonía móvil el promocionar el uso de la navegación WAP y de las instituciones que empiecen a brindar estos nuevos servicios a sus clientes como medio de consulta de información. Por nuestra parte estamos seguros que el sitio WAP creado para la Empresa será de gran utilidad para mantener a los clientes informados de manera oportuna.

CAPITULO II:

ANÁLISIS DE RIESGOS.

INTRODUCCION

Es siguiente capítulo trata de enfocar los posibles riesgos que puedan surgir en el momento de ejecutar la aplicación debido a la naturaleza WAP de la misma. Se debe tener en cuenta que la seguridad brindada a la información de los clientes es una parte importante para el éxito de un sistema de tipo online. Por ello deseamos definir a continuación, la arquitectura que integra WAP para sus capas de transporte y seguridad, teniendo en cuenta que las mismas han sido configuradas para el soporte de datagramas para que exista una compatibilidad con otros protocolos de iguales condiciones.

Los protocolos de seguridad en WAP aseguran la confidencialidad e integridad de los datos, pero se debe recalcar que existe una zona en donde la información puede quedar vulnerable a filtrados debido a la interconexión con el protocolo TCP/IP, este riesgo se da a nivel del *gateway*. De igual manera en esta misma área se puede evitar la expansión de virus en las redes. A continuación se describirá como los proveedores del servicio de telefonía celular brindan tres modalidades en el servicio de navegación que de alguna manera garantizan la seguridad de la información a sus usuarios.

Además de los riesgos implícitos en el uso de la tecnología utilizada, se ha citado algunos puntos propios de la aplicación en el momento de acceder a la consulta de saldos, pero de igual manera se describe la metodología utilizada en el producto para cubrir estas vulnerabilidades y prevenir la posible divulgación de la información de los clientes de la Empresa IMPOCOMJAHER.

2.1 Enfoque de riesgos principales originados por la naturaleza de la aplicación WAP.

La tecnología inalámbrica ha permitido que el día de hoy sea posible navegar en Internet mediante un teléfono celular a través de lo que se conoce como WAP (*Wireless Application Protocol*) o Protocolo para Aplicaciones Inalámbricas. WAP es el estándar internacional para la entrega y presentación de información inalámbrica y servicios en dispositivos móviles o terminales de esta naturaleza, ya que integra la funcionalidad de redes móviles con el poder de las aplicaciones basadas en navegadores.

WAP se articula como una arquitectura en capas en la que la capa de transporte se denomina WDP (*Wireless Datagram Protocol*). Sobre esa capa de transporte se sitúa una capa opcional de seguridad, denominada WTLS (*Wireless Transport Layer Security*). Al igual que en TCP/IP se ha consolidado un estándar de facto, el SSL (*Secure Sockets Layer*) o su evolución, TLS (*Transport Layer Security*), como capa de seguridad entre los protocolos de aplicación (HTTP, FTP, SMTP) y la capa de transporte, la especificación WAP ha definido WTLS. Este protocolo se ha diseñado siguiendo una serie de criterios:

- WTLS debe soportar datagramas.
- Debe soportar portadoras de ancho de banda variado.
- Debe soportar periodos de latencia potencialmente largos.
- La capacidad de memoria y procesamiento de los terminales puede ser pequeña.

Así, TLS y WTLS son protocolos equivalentes siendo patente que la intención de los autores del protocolo fue coger TLS y añadir soporte a datagramas, optimizar el tamaño de los paquetes transmitidos y seleccionar algoritmos rápidos entre los permitidos. Tanto SSL-TLS como WTLS incluyen una fase de *handshake* en la que se negocian los algoritmos utilizados, se intercambian claves y se verifican certificados, seguida de una fase de registro (en la que garantiza también la integridad de los mensajes intercambiados). Del mismo modo que con SSL-TLS, se han definido las siguientes características:

- Integridad de los datos. Se asegura que los datos intercambiados entre el terminal y la pasarela WAP no han sido modificados.
- Confidencialidad de los datos. Se asegura que la información intercambiada entre el terminal y la pasarela WAP no puede ser entendida por terceras partes que puedan interceptar el flujo de datos.
- Autenticación. El protocolo contiene servicios para autenticar el terminal y la pasarela WAP.

Una debilidad del modelo de seguridad en WAP es la presencia de dos zonas de seguridad, fruto de la existencia de un par de dominios tecnológicos diferentes (la zona TCP/IP y la zona WAP), conectados mediante la pasarela WAP (*WAP Gateway*). Así, mientras que en el dominio WAP existe seguridad en el transporte mediante WTLS, en el dominio TCP/IP esta protección en el transporte se consigue utilizando SSL. La pasarela debe ser capaz de hablar WTLS con el dispositivo móvil y SSL con el proveedor de contenidos.

De esta manera la conversión de protocolos se realiza en un nivel más alto. Por ello es necesario un proceso de descifrado y recifrado en la pasarela WAP (descifrado WTLS y recifrado SSL cuando la comunicación se realiza desde el dispositivo móvil al servidor de contenido y viceversa si el camino es inverso). Este proceso causará pérdida de las cualidades de SSL en el dominio TCP/IP porque el cifrado ya no es de extremo a extremo y en la pasarela WAP los datos de la transacción se encontrarán en claro. Este es el fenómeno que se denomina *white spot*.

El proveedor de contenidos o de servicios puede brindar tres modelos de prestación de servicios WAP:

- El proveedor de contenidos integra su servicio en un portal móvil de un operador. En este caso, el propietario de la pasarela, el operador móvil, firma un contrato con el proveedor de contenidos por el que garantiza la seguridad de los datos que transitan por la pasarela (evidentemente sólo integridad y confidencialidad) y, opcionalmente, estableciendo la comunicación con el proveedor de contenidos mediante una VPN, consiguiendo así garantizar la seguridad de los datos en tránsito.

La seguridad del tramo WAP se haya protegida explícitamente con WTLS y la del tramo Internet con SSL.

- El proveedor de contenidos utiliza la infraestructura de cualquier operador móvil para garantizar el acceso a su servicio o contenido, pero sin haber llegado a ningún acuerdo con el operador (el usuario accede al servicio introduciendo una URL en tras haber accedido a la conectividad WAP del operador). En este caso, no se garantiza ningún servicio de seguridad, puesto que depende de la configuración de la pasarela del operador ofrecer SSL en el tramo Internet (incluso aunque se solicite una URL utilizando el protocolo HTTPS). Si la presencia de WTLS también depende de la configuración de la pasarela, puesto que aunque sea requerida por el usuario, generalmente no será implementada.
- El proveedor de contenidos decide garantizar él mismo la seguridad, ofreciendo los servicios de acceso al usuario. Para ello, el usuario tendrá que crear un nuevo perfil dentro de su teléfono WAP para asegurar la conexión con su servidor (número de teléfono, pasarela WAP, etc.) o indicar al proveedor de contenidos cual es su número de teléfono para que éste configure vía OTA el teléfono del usuario. En este caso, el proveedor de contenidos puede adquirir y configurar su propia pasarela WAP garantizando la existencia de WTLS en el tramo WAP de la comunicación. La existencia de SSL no es necesaria puesto que, o bien el servidor de origen se halla dentro de la Intranet del proveedor o bien utiliza un servidor WAP en vez de una pasarela (como el servidor WAP de Nokia), integrando los contenidos con la infraestructura WAP.

Existen varias especificaciones para solucionar este problema. La más significativa es la que permitirá la "tunelización" de conexiones a través de una pasarela, de forma que una conexión WAP sea atendida por la pasarela del operador y traspasada a la del proveedor de servicios, estableciéndose WTLS entre el terminal WAP del usuario y la pasarela WAP del proveedor de servicios. Esta especificación se denomina *Wireless Port Proxy*.

¿Donde está el riesgo?

Debido a que los clientes de WAP manejan información personal y de negocios son vulnerables a los riesgos de seguridad, por ejemplo: los códigos maliciosos se pueden

extender entre las redes móviles a la velocidad de la luz muy parecido a lo que se ha experimentado en Internet con los gusanos y virus. Sería necesario filtrar los paquetes que viajan a través de esas redes, lo que se lograría a través del análisis a nivel *gateway*.

Algunos de los riesgos considerados para nuestra aplicación WAP son:

Ya que es una aplicación donde los datos son accesibles de una manera pública lo cual es la idea misma de la aplicación, que los usuarios tengan disponibilidad de la información independientemente del tiempo y espacio, por ello existe la posibilidad que usuarios no autorizados puedan acceder a la revisión de datos confidenciales de los clientes de la empresa como lo son facturas y cuotas pendientes. Para ello se ha creado un mecanismo de autenticación de los clientes que ingresan a la página WAP a revisar la información concerniente a sus cuotas y pagos. El mecanismo consiste en la validación del Identificador personal del Cliente en el momento de la consulta WAP para verificación de que solo usuarios de “Comercial Jaher” accedan a la aplicación y no saturen al servidor de la empresa.

Otro riesgo que se corre es que como los datos a los que accede la aplicación WAP se obtienen de una base de datos Mysql con información migrada oportunamente a dicha fuente desde la base de datos Oracle que es la propia manejada por el sistema de la Empresa, puede existir momentos en que talvez la información no este actualizada lo cual pueda causar confusión en los clientes. Para solucionar esto se debe ejecutar la migración de los datos varias veces al día para mantener la información actualizada en la base de datos de Mysql y al mismo tiempo no correr el riesgo de que la información quede clara en Internet sino solo los datos que se necesitan.

De igual forma debido a la naturaleza de la información puede que existan tiempos de respuesta relativamente altos para las consultas. El servidor de aplicaciones proveerá la capacidad para responder a las consultas de los clientes las mismas que serán minimizadas con el proceso de autenticación de cédula que permitirá solo a los clientes de la Empresa acceder a nuestra aplicación.

2.2 Métodos de prevención de posible divulgación de información.

Para evitar la posible divulgación de la información que se presenta a los usuarios se manejará códigos para la generación de páginas lo cual limitará la generación de las mismas solamente a clientes de la Empresa así como accederán solamente a datos concernientes a sus transacciones personales. Los mecanismos a utilizar serán claves generadas por PHP para las páginas únicamente invocadas por un cliente de Comercial Jaher una vez verificada la cédula del mismo. Dicha clave la manejará el cliente durante toda su navegación por nuestro portal WAP de manera que no le permitirá consultar datos de otro cliente.

2.3 Conclusiones.

Como hemos analizado en este capítulo tenemos las herramientas suficientes para dar seguridad tanto a nivel de protocolo como a nivel de aplicación en el momento de acceder a los datos sin descuidar el hecho que lo que el cliente busca es obtener su información y que esta esté a su alcance de una manera rápida rompiendo barreras de tiempo y espacio. Por todas estas razones se implementó una base de datos diferente a la que tiene la Empresa para sus transacciones diarias, dicha base de datos se encuentra aislada de la original y esta accesible desde aplicaciones tipo Web o WAP, pero no por ello debería creerse que contiene información falsa o desactualizada.

También se tiene controlado que la aplicación desarrollada sea accesible solamente por clientes de la Institución, evitando la saturación del servidor por usuarios que no podrían recuperar información alguna de nuestra base de datos. Cada cliente tiene garantizada su navegación segura ya que se crearán sesiones por cada usuario que ingrese al sitio para alguna consulta de saldos, pudiendo el mismo solamente acceder a datos personales y no de algún otro individuo existente.

CAPÍTULO III

ANÁLISIS DE LOS DATOS A PRESENTAR.

INTRODUCCIÓN

Una de las partes esenciales del desarrollo de una aplicación software es el Análisis de los procesos que trata básicamente de determinar los objetivos y límites del sistema tratado, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar sus consecuencias. En el presente capítulo se desea presentar al lector una muestra gráfica y sintética de la estructura de los datos utilizados para la aplicación, de igual forma se desea describir las relaciones existentes entre las diferentes entidades que integran la Base de Datos a la cual accederán los usuarios en el momento de realizar su respectiva consulta de saldos vía online.

Para mayor entendimiento del lector se ha incluido un diccionario de datos de las entidades utilizadas. También se presentará un análisis de los procesos que realiza la aplicación para la búsqueda de la información que posteriormente será entregada al usuario final. Se citan los métodos de acceso a datos, validación de datos enviados al servidor, entrega de información, selección de datos según el usuario que realice la petición del servicio, extracción correcta desde los almacenes de información.

3.1 Modelo entidad – relación de la base de datos a usar.

Para la implementación de la aplicación WAP de consulta de saldos de los clientes de IMPOCOMHAHER Cía. Ltda., se concluyó necesario utilizar las siguientes entidades que se incluyen en la base de datos denominada “jaher”:

- Tabla Clientes: es la entidad en la que se almacenarán los datos de los clientes de la Empresa y a la cual se accederá para la validación de la existencia del identificador enviado por el usuario en el momento de acceder al sitio WAP de la Empresa.
- Tabla Comprobante: es la entidad en la que se almacenarán los datos de los comprobantes o facturas generados por las compras de los clientes de la empresa a nivel nacional.
- Tabla Cuotas: es la entidad en la que se almacenarán los datos de las cuotas a las que ha sido diferidos los pagos de cada una de las facturas de los clientes, estas pueden variar desde 1 hasta 24 cuotas por cada comprobante.

De igual manera las entidades citadas anteriormente han sido relacionadas mediante llaves principales para evitar la duplicación de datos y llaves foráneas para garantizar búsquedas y acceso a los datos de una manera más rápida y efectiva.

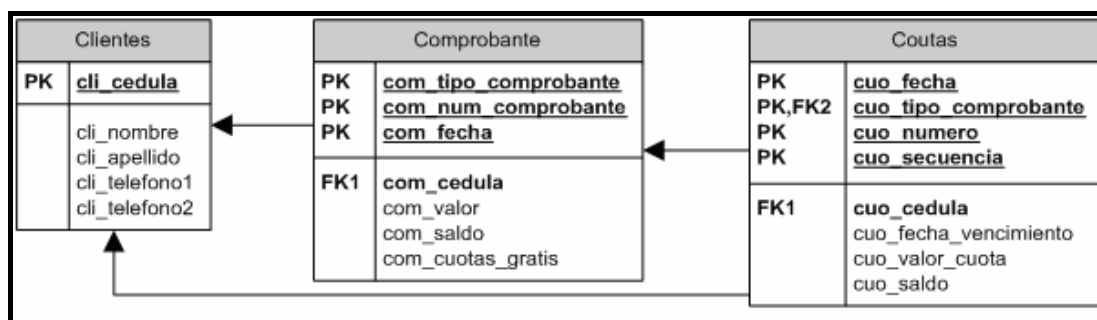


Figura 3.1: Modelo Entidad – Relación

3.2 Diccionario de Datos.

Una de las herramientas necesarias en el análisis de los datos en el momento del desarrollo de una aplicación sea de la naturaleza que fuere es el Diccionario de Datos de las entidades que forman parte de la base de datos utilizada. Este elemento trata de una enumeración de los campos que contiene cada una de las tablas a las que se accede en la aplicación definiendo características complementarias como lo son el tipo de dato, la longitud del campo, el formato que lleva, una descripción y se indica si el atributo es llave o no, definiendo el tipo de llave en caso de que lo aplique.

A continuación se enumeraran los campos de las entidades utilizadas en la aplicación:

Tabla Clientes: consta de los siguientes campos:

- cli_cedula: almacena el identificador del cliente.
- cli_nombre: almacena el nombre del cliente.
- cli_apellido: almacena el apellido del cliente.
- cli_telefono1: almacena el teléfono del cliente.
- cli_telefono2: almacena el teléfono alternativo del cliente.

Tabla Comprobante: consta de los siguientes campos:

- com_cedula: almacena el identificador del cliente dueño de ese comprobante.
- com_tipo_comprobante: almacena el tipo de comprobante que puede tomar tres valores NV, DB, A0.
- com_num_comprobante: almacena el número del comprobante, el mismo que junto con el tipo de comprobante forman la llave principal de la entidad Comprobante.
- com_fecha: almacena la fecha en la que se generó el comprobante.
- com_valor: almacena el valor total del comprobante.

- com_saldo: almacena el valor adeudado por el cliente para completar el pago total del comprobante.
- com_cuotas_gratis: almacena el número de cuotas gratis a las que se hace acreedor el clientes en caso de que cumpla puntualmente con el pago de las cuotas a las que se ha diferido el comprobante.

Tabla Cuotas: consta de los siguientes campos:

- cuo_cedula: almacena el identificador del cliente dueños del comprobante al que hace referencia la cuota.
- cuo_fecha: almacena la fecha en la que se generó el comprobante al que hace referencia dicha cuota.
- cuo_tipo_comprobante: almacena el tipo de comprobante de la factura a la que hace referencia dicha cuota, puede tener unos de los tres valores NV, DB, A0.
- cuo_numero: almacena el número de comprobante al que hace referencia dicha cuota.
- cuo_secuencia: almacena un número secuencial que puede ir desde 1 hasta 24 e identifica el número de cuota del total de cuotas a las que se ha diferido el pago del comprobante al que hace referencia dicha cuota.
- cuo_fecha_vencimiento: almacena la fecha en la que se vence el pago de la cuota.
- cuo_valor_cuota: almacena el valor total adeudado por la cuota en cuestión.
- cuo_saldo: almacena el valor adeudado por el cliente para completar el pago total de la cuota.

Como se ha podido observar hemos escogido nombre nemotécnicos para cada uno de los campos de manera que identifiquen a que entidad pertenecen y evitar la redundancia de los nombres de los campos en el momento de utilizarlos ya sea en

sentencias SQL o para referencias de consultas de datos. Para mayor detalle de las entidades se puede recurrir al Apéndice 6.2.1.1 en el que se detallan las características más descriptivas de cada uno de los campos de las tablas citadas anteriormente.

3.3 Diagramas de Flujo de Información.

Los diagramas de flujo de información describen de manera gráfica los procesos que se ha utilizado en la aplicación ya sea para realizar verificación de datos, búsquedas, extracción de información de los almacenes de datos, recuperación de datos o valores necesarios para otros procesos que son dependientes, etc. A continuación se describen los principales procesos realizados en la consulta de saldos de los clientes de la Empresa.

3.3.1 Consulta de Comprobantes

Este proceso es el primero que se realiza al momento de navegar por el sitio WAP de la aplicación. Lo que realiza el proceso es recibir el identificador digitado por el usuario y tomarlo para realizar una búsqueda en la entidad de Clientes, en caso de no encontrarlo resultaría que el cliente no está registrado o que el identificador fue digitado de manera incorrecta. Esto se realiza mediante un primer proceso denominado “Verifica” (Ver Apéndice 6.2.1.2).

En caso de que el cliente exista se proceden a mostrar sus datos personales y a realizar una nueva consulta de extracción de los comprobantes desde la entidad Comprobantes, facturas que pertenezcan al cliente en sesión y que tengan un saldo mayor a 0. Esta operación se realiza en un segundo proceso denominado “Busco Comprobantes” (Ver Apéndice 6.2.1.2), en caso de encontrarlos se muestra al usuario para que escoja uno caso contrario se notificará al usuario que no tiene cuentas pendientes con la Empresa.

Si existieron comprobantes el cliente tiene la opción de escoger uno de la lista desplegada y seleccionarlo para su consulta, esto se realiza a través de un tercer

proceso denominado “Selecciono Comprobante” (Ver Apéndice 6.2.1.2). Todos estos procesos se ejecutan para mostrar una primera pantalla con los datos correspondientes al usuario que esta consultando sus saldos por el sitio WAP de la Institución.

3.3.2 Consulta de Saldo

Este proceso se ejecuta para desplegar una segunda pantalla de la aplicación WAP al cliente que se encuentra solicitando el servicio de consulta WAP. Lo que trata esta operación es recuperar los datos concernientes al Saldo, Valor Total, Número, Total Pagado, Número de Cuotas Gratuitas para el cliente del comprobante seleccionado en el proceso anterior. Para una descripción más exacta de cómo se efectúa el proceso se puede acceder al Apéndice 6.2.1.3, donde se grafica el siguiente proceso:

El comprobante seleccionado por el cliente es enviado para una consulta y recuperación de datos desde la entidad Comprobantes, proceso denominado “Extraer Información Saldo”. Si el comprobante se encuentra se devuelven los datos necesarios caso contrario se mostrará un mensaje notificando al usuario que debe escoger un comprobante de la lista que se le mostró en la página anterior a la visitada actualmente.

3.3.3 Consulta Cuotas del Comprobante

La aplicación tiene una opción que involucra la consulta de las cuotas pendientes de pago de cada uno de los comprobantes que pueda tener el cliente. En esta pantalla se mostraran datos como el número de cuota, la fecha de vencimiento de la misma y el valor adeudado. Para realizar dicha extracción de información se ejecuta un proceso denominado “Extraer Información Cuotas” (Ver Apéndice 6.2.1.4), donde se utiliza el número, tipo de comprobante e identificador del cliente en una consulta SQL realizada a la entidad Cuotas de la que se obtendrá una lista de las Cuotas con saldo mayor a 0 que tenga el cliente sin importa si se encuentran vencidas sin pago o tengan fecha de cancelación futura.

3.4 Conclusiones.

Después de desarrollado este capítulo definimos las entidades que utilizaremos en nuestra aplicación Wap, de igual forma los procesos que utilizaremos para la obtención de información según los requerimientos de los clientes de Comercial Jaher. Como se puede observar la estructura de las diferentes tablas que almacenan información están diseñadas de tal manera que sea de fácil acceso y rápida búsqueda los datos de cada cliente, mediante el uso de llaves Principales por cada entidad y de llaves Foráneas. Estas llaves en algunos casos fueron formadas por un solo campo pero en el caso de la tabla Comprobante y Cuotas fue necesario el uso de llaves compuestas debido a la naturaleza de los datos.

CAPÍTULO IV

PROGRAMACIÓN Y HERRAMIENTAS.

INTRODUCCION

En el presente capítulo presentamos las técnicas de programación y obtención de datos, y de igual manera describiremos brevemente las herramientas de software usado para estos fines.

Dos elementos esenciales del desarrollo de una aplicación son el tipo de herramientas y el lenguaje de programación utilizado, ya que permitirán definir el alcance del software desarrollado y también de que manera se pueda cumplir las expectativas planteadas en los objetivos del producto. Para esta aplicación hemos utilizado exclusivamente el lenguaje de programación PHP junto con el WML para el desarrollo de las páginas WAP, pero además se utilizó lenguaje SQL para la interacción con la base de datos llevada en MYSQL.

Debido a que intentamos mostrar datos al público y queremos precautelar la integridad y la privacidad de la base de datos principal de “IMPOCOMJAHHER CIA. LTDA” lo que nos hemos planteado es migrar el contenido que será de interés para la consulta de los clientes a una base de datos que estará en el Web, de esta forma se mantiene aislado el contenido que no es de interés para el usuario del portal WAP.

Entre las herramientas a utilizar están Dreamweaver en su versión MX y Fireworks en la misma versión MX que forman parte del paquete Macromedia, y que su uso está destinado tanto al diseño de páginas Web como a programación de páginas dinámicas en cualquiera de sus lenguajes siendo PHP con WML el que utilizaremos.

Teniendo este enfoque de lo que buscamos realizar nos hemos planteado como objetivo el desarrollar un portal WAP que sea visible en la mayoría de los micronavegadores de los teléfonos móviles y que sea de fácil acceso para los usuarios.

4.1 Programación.

4.1.1 Método de Obtención de información para el almacenamiento en la base de datos.

Los datos existentes en la base de datos de Mysql son actualizados diariamente obteniendo la información respectiva de la base de Oracle, fuente en la cual se almacenan los datos de las aplicaciones de la empresa.

El proceso de actualización de la base de Mysql es el siguiente:

- Se obtienen los datos de los clientes ingresados el día anterior mediante la siguiente consulta SQL, la misma que será exportada en un archivo tipo csv:

```
select c.cod_cliente,c.nombre,c.apellido1,d.telefono1h,d.telefono2h,'\r\n'
from   cliente c,
cliente_hor d
where c.empresa='CODIGO_EMPRESA'
and   d.empresa=c.empresa
and   d.cod_clienteh=c.cod_cliente
and   (to_char( d.fecha_creacionh,'YYYY-MM-DD') = 'FECHA_ANTERIOR')
```

R4.1: Sentencia SQL-Oracle para obtención de información nueva de la tabla Clientes.

- Luego los datos de este archivo csv son cargados a la tabla de Mysql mediante la siguiente función:

```
LOAD DATA LOCAL INFILE 'archivo.csv' INTO TABLE `clientes` FIELDS
TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES
TERMINATED BY '\r\n'
```

R4.2: Sentencia PHP para cargar los datos nuevos en la tabla Clientes.

De esta forma se van ingresando los datos de los clientes en la base de datos para su posterior consulta.

- Se obtienen los datos de los comprobantes o facturas registradas el día anterior mediante la siguiente consulta SQL, la misma que será exportada en un archivo tipo csv:

```
Select c.cod_cliente, a.tipo_comprobante, a.cod_comprobante,
to_char(a.fecha,'YYYY-MM-DD'), a.valor,a.saldo.cuotas_gratis,'\r\n'
from   comprobante a, cliente c, cliente_hor d
where a.empresa = COD_EMPRESA
and   a.tipo_comprobante in ('A0','NV','DB')
and   a.saldo>0
and   (to_char( a.fecha,'YYYY-MM-DD') = 'FECHA_ANTERIOR')
and   a.anulado='N'
and   c.empresa=2
and   c.cod_cliente=a.cod_persona
and   d.empresah=c.empresa
and   d.cod_clienteh=c.cod_cliente
```

R4.3: Sentencia SQL-Oracle para obtener los nuevos datos de la tabla Comprobante.

- Luego los datos de este archivo csv son cargados a la tabla de Mysql mediante la siguiente función:

```
LOAD DATA LOCAL INFILE 'archivo.csv' INTO TABLE `comprobante`
FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES
TERMINATED BY '\r\n'
```

R4.4: Sentencia PHP para cargar los datos nuevos en la tabla Comprobante.

De esta forma se van ingresando los datos de los comprobantes en la base de datos para su posterior consulta.

- Se obtienen los datos de las cuotas de las facturas registradas el día anterior mediante la siguiente consulta SQL, la misma que será exportada en un archivo tipo csv:

```
Select a.cod_persona, to_char(a.fecha,'YYYY-MM-DD'), a.tipo_comprobante,
a.cod_comprobante, b.secuencia, to_char( b.vencimiento,'YYYY-MM-DD'),
b.valor,b.saldo,'\r\n'
from comprobante a,
forma_pago_comprobante b
where a.empresa=CODIGO_EMPRESA
and a.tipo_comprobante in ('A0','NV','DB')
and a.saldo>0
and (to_char( a.fecha,'YYYY-MM-DD') = 'FECHA_ANTERIOR')
and a.anulado='N'
and b.empresa=a.empresa
and b.tipo_comprobante=a.tipo_comprobante
and b.cod_comprobante=a.cod_comprobante
and b.cod_forma_pago in ('CRE','NDB')
```

R4.5: Sentencia SQL-Oracle para obtener los nuevos datos de la tabla Cuotas.

- Luego los datos de este archivo csv son cargados a la tabla de Mysql mediante la siguiente función:

```
LOAD DATA LOCAL INFILE 'archivo.csv' INTO TABLE `cuotas` FIELDS
TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES
TERMINATED BY '\r\n'
```

R4.6: Sentencia PHP para cargar los datos nuevos en la tabla Cuotas.

De esta forma se van ingresando los datos de los detalles de los comprobantes o facturas en la base de datos para su posterior consulta.

4.1.2 Método de Autenticación de un usuario al acceder al sitio WAP

El sitio WAP creado es exclusivo para clientes de Comercial Jaher por lo cual esta restringido a la validación de su existencia en la base de datos de la aplicación.

Para el proceso de autenticación se pedirá al usuario que introduzca su número de Cédula con guión o su número de RUC en caso de tenerlo. Entonces se procederá a

acceder a una búsqueda del dato digitado en la base de datos, al no encontrarlo se presentará un mensaje al usuario de que no aparece como Usuario Registrado o que la Cédula es errónea. En caso de existir la Cédula o RUC se permitirá el acceso del cliente a la consulta de los datos de sus facturas pendientes con la empresa.

4.1.3 Método de Búsqueda de la información requerida por el usuario.

Una vez que el usuario a podido acceder al sitio Wap de la empresa se le mostrará una pantalla con una lista de las facturas que tiene pendientes de pago, es decir que tengan un saldo mayor a 0.

Para ello se ejecutará la siguiente consulta SQL:

```
SELECT com_tipo_comprobante,com_num_comprobante FROM comprobante  
WHERE com_cedula='$cedula_cliente' and com_saldo>0
```

R4.7: Sentencia SQL-Mysql para obtener los códigos de los comprobantes pendientes del cliente.

De esta consulta se obtendrán uno o más registros los cuales se mostrarán en un control <select> del lenguaje WML. Si es que el cliente no posee facturas con saldo mayor a 0 se mostrará un mensaje de que no tiene facturas pendientes con la Empresa.

El cliente podrá acceder a la consulta de una de las facturas que se le presenten en el listado en una pantalla posterior donde se mostrará datos como el Saldo, Valor total de la factura, Valor cancelado de la Factura, el número de Cuotas Gratis que tiene el Cliente, etc.

Para ello se enviarán a la siguiente página WAP los datos de la cédula del cliente, el tipo de comprobante y el número de factura seleccionados por el mismo los cuales se utilizarán para la siguiente consulta:

```
SELECT * FROM comprobante WHERE com_cedula='$cedula_cliente' and  
com_tipo_comprobante='$tipo_escogido' and  
com_num_comprobante='$numero_escogido'
```

R4.8: Sentencia SQL-Mysql para obtener los datos de los comprobantes pendientes del cliente.

En esta pantalla existe una opción que le permite acceder al detalle de las cuotas de la factura del cliente en caso de tenerlo, para lo cual se envían datos de la cédula del cliente, el tipo de comprobante y el número de factura seleccionados por el mismo los cuales se utilizarán para una nueva consulta que es la siguiente:

```
SELECT * FROM cuotas WHERE cuo_cedula='$cedula_cliente' and  
cuo_tipo_comprobante='$tipo_comprobante' and  
cuo_numero='$numero_comprobante' and cuo_saldo>0 order by  
CONVERT(cuo_secuencia,UNSIGNED INTEGER) asc
```

R4.9: Sentencia SQL-Mysql para obtener los datos de las cuotas correspondientes al comprobante consultado por el cliente.

De esta consulta se obtendrán datos como el número de cuota, su valor y la fecha de vencimiento.

4.1.4 Método de Presentación de la Información al Usuario.

Los datos al usuario se presentan con el uso del comando:

```
echo "<small>Texto</small>";
```

- echo: función propia de PHP que permite la visualización de los datos en la página WAP.
- <small></small>: función propia del lenguaje WML que permite el formato de texto con fuente pequeña.
- Texto: en texto puede ir una variable o un texto lo cual se visualizará en el navegador.

La presentación de los datos en la pantalla de Comprobantes se realiza mediante los siguientes comandos:

```
echo "<small>Cliente: $nombre_cliente</small>";
echo "<small>Escoja la factura:</small>";
echo "<select name='valor' ivalue='1'>";
while ($registro = mysql_fetch_array($consulta))
{
    $tipo_comprobante=trim($registro [subindice]);
    $numero_comprobante=trim($registro [subindice]);
    $dato=$tipo_comprobante.$numero_comprobante;
    echo "<option value='$dato' >$numero_comprobante</option>";
}
echo "</select>";
```

R4.10: Código PHP-WML utilizado para mostrar la lista de comprobantes para el cliente.

- `<select></select>`: se utiliza este comando para crear una lista desplegable en la cual se mostrará los comprobantes que tenga el usuario para que escoja uno para la posterior consulta.
- `while ($registro = mysql_fetch_array($consulta))`: pasa los resultados de una consulta SQL contenida en la variable `$consulta` y la descarga registro por registro a la variable `$registro` a la cual se accede mediante un subíndice que empieza en 0.

El subíndice indica cada una de las columnas que se obtuvieron en el SQL aplicado.

De igual forma se mostrarán tantos registros como líneas de la consulta se hayan obtenido.

- `<option value='$dato' >$numero_comprobante</option>`: con este comando se mostrará cada uno de los comprobantes resultados de la consulta. “Value” es el valor del control. “\$numero_comprobante” tiene el valor del texto que se visualiza en el navegador WAP.

La presentación de los datos en la pantalla de Saldos se realiza de la siguiente manera:

```
while ($registro = mysql_fetch_array($consulta))
{
$val=$ registro [subindice]-$ registro [subindice];
    echo "<small><b>Cliente:</b> $nombre</small>";
    echo "<small><b>Factura:</b> $numero_comprobante</small>";
    echo "<small><b>Total:</b> $ registro [subindice]</small>";
    echo "<small><b>Pagado:</b> $val</small>";
    echo "<small><b>Saldo:</b> $ registro [subindice]</small>";
    echo "<small><b>Val.Prox.Cuo:</b> $ registro [subindice]</small><br/>";
    echo "<small><b>Vence:</b> $ registro [subindice]</small><br/>";
    echo "<small><b>Cuotas Gratis:</b> $ registro [subindice]</small>";
}
```

R4.11: Código PHP-WML utilizado para mostrar los datos en la pantalla de Saldos del cliente.

- `while ($registro = mysql_fetch_array($consulta))`: de igual forma pasa los resultados de una consulta SQL contenida en la variable \$consulta, que en este caso será una sola línea y la descarga a la variable \$registro a la cual se accede mediante un subíndice que empieza en 0.

El subíndice indica cada una de las columnas que se obtuvieron en el SQL aplicado.

- `<b></b>`: formatea el texto con énfasis.

La presentación de los datos en la pantalla de Cuotas se realiza de la siguiente manera:

```
echo "<table columns='3' align='C'>
<tr><td><small><b>No.</b></small></td>
<td><small><b>Vence</b></small></td>
<td><small><b>Valor</b></small></td></tr>";
while ($registro = mysql_fetch_array($consulta))
{
echo "<tr><td><small><b>$c[4]</b></small></td>
<td><small><b>$ registro [subindice]</b></small></td>
<td><small><b>$ registro [subindice]</b></small></td></tr>";
}
echo "</table>";
```

R4.12: Código PHP-WML utilizado para mostrar los datos en la pantalla de Cuotas del comprobante pendiente de pago por el cliente.

Para la presentación de los datos al usuario en esta pantalla se utiliza la etiqueta `<table>` para mostrar los datos de manera mas ordenada.

A continuación se explica rápidamente lo que hace cada uno de los comandos del código expuesto anteriormente:

- `<table columns='3' align='C'></table>`: define una tabla con 3 columnas y centrada en la pantalla.
- `<tr><td></td><td></td><td></td></tr>`: define una línea con 3 columnas.
- `while ($registro = mysql_fetch_array($consulta))`: pasa los resultados de una consulta SQL contenida en la variable `$consulta`, pasándola línea por línea y la descarga a la variable `$registro` a la cual se accede mediante un subíndice que empieza en 0.

El subíndice indica cada una de las columnas que se obtuvieron en el SQL aplicado.

- `<td><small><b>$registro[subindice]</b></small></td>`: muestra el texto que irá en cada columna de la tabla.

4.2 Herramientas.

4.2.1 Descripción de las herramientas de diseño.

Las herramientas utilizadas para el diseño de las páginas WML son las siguientes:

Macromedia Dreamweaver MX: este software utilizado en el diseño de páginas Html, PHP así como de páginas Wml, permite la inserción de etiquetas propias del lenguaje escogido presentándonos las opciones disponibles de las mismas facilitando el diseño de la presentación de las aplicaciones ya que tiene la opción de una vista preliminar mientras se desarrolla el trabajo.



Figura 4.1: Imagen de presentación Macromedia Dreamweaver MX

Firewoks MX: esta herramienta fue utilizada para la edición del logo de la Empresa, el cual se presenta en la página principal de la aplicación. La mayor utilidad del mismo es que permite exportar las imágenes desde cualquier formato, como por ejemplo imágenes de tipo gif, jpeg, bmp, etc. convirtiéndolas a imágenes de tipo wbmp que es el formato soportado por el lenguaje WML.



Figura 4.2: Imagen de presentación Macromedia Dreamweaver MX

4.2.2 Descripción de las herramientas de programación.

Macromedia Dreamweaver MX: este software utilizado al igual que en el diseño fue requerido para la programación de las aplicaciones ya sea en lenguaje de páginas Html, PHP o Wml. La opción de presentación de las etiquetas predefinidas por el lenguaje es de gran ayuda y guía para el usuario, exhibiendo las opciones disponibles de las mismas y facilitando el desarrollo de las aplicaciones.

4.2.3 Descripción de las herramientas de validación y simulación.

Debido a la naturaleza de la aplicación el realizar todas las pruebas necesarias en su elaboración resultaría costoso por el uso de un dispositivo móvil por ello hemos optado en utilizar algunas aplicaciones disponibles en el Internet como lo son:

- <http://www.wapsilon.com>
- <http://www.inetis.com/ttemulator.asp>

Estas páginas dan acceso gratuito al uso de emuladores de páginas WAP con solo poner la dirección o url de la misma.

Ambas aplicaciones permiten observar la emulación en un dispositivo móvil estándar en la pantalla del ordenador lo cual es muy útil ya que nos podrá dar una idea del tamaño de los textos y del orden que convendría que lleven los mismos.

De igual forma permite la validación de todas las páginas y las etiquetas utilizadas en el desarrollo de la aplicación.

4.3 Conclusiones.

Luego del desarrollo de este capítulo pudimos conocer a cabalidad las capacidades del lenguaje PHP al trabajar con el lenguaje de páginas WAP, como lo es el WML. Para los desarrolladores fue esencial la ayuda proporcionada por el software Dreamweaver MX, ya que permite el uso de etiquetas prediseñada del lenguaje usado, mostrando todas las opciones que tiene la misma y de igual forma da a elegir los valores que ellas pudieran tomar variando el estilo de presentación de la aplicación.

Además se ha comprendido la importancia de mantener aislado el contenido de la base de datos principal de la empresa de la base de datos Web para que así estén ubicados en la base de datos de Mysql únicamente la información a la que el cliente accederá. Entre este aspecto de aislamiento debemos tener en cuenta que la migración de datos desde la base de datos principal de la empresa a la Base de datos Web debe realizarse de una manera constante y periódica para que así los datos que se vayan a mostrar al cliente mediante la aplicación Wap sean actualizados y reales.

CAPÍTULO V:

PRUEBAS.

INTRODUCCION

Toda aplicación previa entrega a los usuarios finales debe ser evaluada y pulida de tal manera que sean validados todas las posibles acciones del usuario como prevención tomada para el buen funcionamiento del producto. Las pruebas se basaron en la configuración del Servidor donde se encuentra alojada la aplicación para constatar que el mismo soporte el lenguaje WML, bajo el cual se ha desarrollado el sitio de consulta. Posteriormente, según se fue desarrollando la aplicación se realizaron pruebas de obtención de información, de concordancia de datos, de validación de datos de ingreso, existencia de los mismos en la base de datos del sistema, entre otras pruebas. Y finalmente se realizaron las respectivas pruebas en algunos dispositivos móviles para ver la funcionalidad de la aplicación según la compatibilidad con el lenguaje de desarrollo.

Todas estas pruebas nos asegurarán que ante diferentes circunstancias tanto de ingreso de datos erróneos como otros imprevistos la aplicación nos siga entregando datos confiables y mantenga la privacidad de los datos.

Hemos iniciado las pruebas inicialmente con la configuración del servidor Web que contendrá la aplicación WAP y la base de datos de la que realizaremos las consultas, todos estos parámetros iniciales deberán estar puestos a punto para las posteriores pruebas ya con la aplicación WAP en los diferentes emuladores o en los micronavegadores de los teléfonos móviles en los que se realizarán las pruebas.

5.1 Pruebas de Configuración del Servidor de Linux

Para que el lenguaje WML sea soportado por un servidor Linux debe añadirse algunos MIME Types en el archivo httpd.conf, como ya se explicó en el Capítulo 1, Sección 1.3.1.

Para comprobar que el servidor se encontraba pre configurado con lo necesario para nuestra aplicación se hizo una prueba con una página sencilla con el siguiente código básico:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml><card id="card1" title=" Inicio">
    <p>Prueba</p>
</card></wml>
```

R5.1: Código WML utilizado para generar una página sencilla de prueba de tipo WML puro.

Al ejecutar esta página nos dio el siguiente error en el servidor:

The webserver told me that 406 No se han encontrado objetos aceptables



Figura 5.1: Mensaje de error capturado del simulador wap <http://www.wapsilon.com>

Posteriormente pedimos al Administrador del Servidor Web que habilite los MIME Types para que nuestra aplicación pueda ejecutarse.

5.2 Pruebas de Obtención de Información

De igual forma se hicieron las respectivas pruebas de obtención de información en los siguientes aspectos:

- Conexiones con la Base de Datos de la Empresa: para ello se crea la variable de conexión la cual devolverá un valor “True o Verdadero” el momento que se estableció la conexión exitosamente caso contrario se enviará un mensaje de “No se ha podido establecer la conexión con el servidor”.
- Obtención de datos a partir de consultas SQL: para la obtención exitosa de los datos es necesario que la conexión con la Base de Datos este abierta y además que los parámetros requeridos en la consulta sean válidos. Por ello al principio de la aplicación se validará que el usuario que desee acceder a navegar por el sitio Wap de Comercial Jaher este registrado como Cliente. Esta información se obtendrá a partir de una consulta sql con el parámetro del código de la persona que solicite la aplicación sea su Cédula o su Ruc. La consulta devuelve los datos si se encuentran bajo el parámetro de búsqueda caso contrario la aplicación devolverá un mensaje de “Estimado usuario su cédula no se encuentra registrada”.



Figura 5.2: Imagen de petición de Cédula



Figura 5.3: Imagen del error al validar la cédula

- Llaves principales y foráneas definidas correctamente: para poder acceder a los datos de manera correcta fue necesario crear llaves principales y foráneas en las tablas de la base de datos de Mysql de la Empresa. De esta manera los datos que existen no se duplicarán y no se dará la posibilidad de acceder a datos erróneos o de otros clientes no requeridos.

La llave Principal de la tabla Comprobantes compuesta por el número del comprobante más el tipo del mismo crearán unicidad en la información almacenada permitiendo a los clientes acceder a todos los comprobantes que tengan pendiente con la empresa de forma única y rápida.

Se realizaron las pruebas con varios clientes mandando como parámetro su número de identificador y se compararon los resultados con un estado de cuenta propio del sistema desarrollado en Oracle que utiliza la empresa diariamente observando que los resultados eran los mismos datos para el cliente utilizado como prueba.



Figura 5.4: Imagen con listado de Comprobantes.



Figura 5.5: Imagen que muestra información del Saldo.



Figura 5.6: Imagen que muestra información de Cuotas.

5.3 Pruebas de la Aplicación WAP en un dispositivo de telefonía móvil

Las pruebas realizadas de la aplicación sobre un dispositivo móvil nos dieron a conocer que los emuladores disponibles en el Internet son eficaces ya que valida casi en su totalidad la aplicación WML salvo el tipo de datos de entrada, es decir, mientras el usuario se encuentre navegando por la aplicación Wap desde un emulador los datos que puede digitar en la pantalla del ordenador pueden ser todos los presentados en el teclado a pesar que la aplicación Wap exija la introducción de datos en base a una máscara predefinida que restringe a datos numéricos o de tipo caracter.

En cambio cuando el usuario se encuentre navegando en el sitio Wap desde un dispositivo móvil la aplicación validará el ingreso correcto de datos, es decir, no permitirá el ingreso de caracteres no válidos para la aplicación, respetando la máscara definida para el ingreso de información.

5.4 Conclusiones.

Luego de haber pasado por el proceso de pruebas hemos concluido nuestra aplicación tomando muy en cuenta que para que la aplicación funcione correctamente previamente el servidor Web que contendrá la aplicación deberá estar a punto en sus configuraciones. De igual forma hemos determinado que nuestra aplicación valida cualquier falla del usuario ya sea en la digitación en el momento de entrada de su código personal, o en el momento de elegir entre los comprobantes que el cliente tenga pendientes con la Empresa.

Para dar por concluido el proceso de pruebas se procedió a realizar las correspondientes en diferentes modelos de teléfonos móviles en las diferentes compañías que soportan el servicio de navegación WAP por lo que hemos podido constatar que las características visuales con las que cuenta la aplicación WAP no están disponibles en todos los modelos y variarán según la marca del teléfono móvil y el micronavegador que tenga instalado.

Por lo demás hemos determinado que la aplicación está lista para ser usada por los clientes de la Empresa “IMPOCOMJAHHER CÍA. LTDA.” y que la actualización de los datos desde la base de datos central hasta la base de datos Web debe ser realizada constantemente para que se obtengan así los resultados esperados.

CAPÍTULO VI

CONCLUSIONES Y BIBLIOGRAFÍA

6.1 Conclusiones.

Al final del desarrollo de esta aplicación hemos llegado a las siguientes conclusiones:

- Las tecnologías de difusión de información inalámbrica están en constante desarrollo y en nuestro medio no debe quedar relegada, a un segundo plano, la comodidad de tener la información que requerimos al alcance de nuestras manos, lo cual es primordial y más aún cuando los dispositivos de telefonía móvil se han convertido en medios ya no solamente para transmitir voz sino imágenes y datos, y cada vez incorporan nuevos servicios como micronavegadores que nos permiten tener acceso al Internet. Con todas estas ventajas del medio tecnológico las compañías proveedoras de servicio telefónico móvil han abierto la posibilidad de obtener información vía WAP, lo cual nos permiten acceder a sitios de consulta que serán de gran ayuda tanto para las empresas que brindan crédito como para los clientes que ya no tendrán que recurrir a las oficinas de las empresas a consultar sus saldos.
- Los beneficios que obtendrá la empresa “IMPOCOMJAHHER CIA. LTDA.” con respecto a sus competidores será que mantendrá a sus clientes informados sobre la fecha de vencimiento de sus cuotas y así mantendrá un nivel de cartera bajo y mejorará sus tiempos de cobros.
- La importancia de mantener aislada la base de datos central de la empresa de su base de datos Web es primordial debido a que no se debería cargar al Web todos los datos sino únicamente los que al cliente que va a consultar le interesan. Con esto protegemos los datos centrales tanto de ataques externos como la privacidad de los mismos.
- La programación de la aplicación en general no es compleja ya que se puede desarrollar enlazando conocimientos de programación Web, configuración de

servidores, lenguaje de consultas SQL y migración de datos y conocimientos básicos de WML.

- Esta aplicación al ser de un gran utilidad entre las empresas que brindan créditos puede ser comercializada fácilmente obteniendo así un valor comercial, y de igual manera se puede ir agregando a la misma otros servicios como el envío de información mediante mensajes de texto o SMS.
- En conclusión hemos obtenido un valor agregado para la empresa “Comercial Jaher” que ahora tiene una nueva manera de dar información a sus clientes. Así mismo los desarrolladores han obtenido conocimientos más profundos sobre la difusión de información mediante el WAP.

Algunas recomendaciones que damos como desarrolladores de la aplicación son:

- Dentro del plano tecnológico debemos acostumbrar a la gente a usar las nuevas técnicas para divulgación de información mediante estos dispositivos de telefonía móvil que en nuestro medio se puede decir que un 90% de la población económicamente activa dispone al menos de un teléfono móvil
- A las empresas que desean expandir sus mercados y dar facilidades a sus clientes deberían establecer medios que aprovechen la existencia de estas nuevas tecnologías.
- Los programadores no deben relegarse al viejo estilo de establecer soluciones puntuales sino más bien dejar puertas abiertas a la escalabilidad de un servicio, en vía de brindar mejoras conforme vayan avanzando las tecnologías y de igual manera ir enlazando conocimientos para así dar soluciones integrales.
- En el aspecto de seguridad mantener siempre las protecciones debidas a la información de la empresa para prevenir ataques externos que puedan comprometer la integridad de la información que es tan celosamente guardada por las empresas.

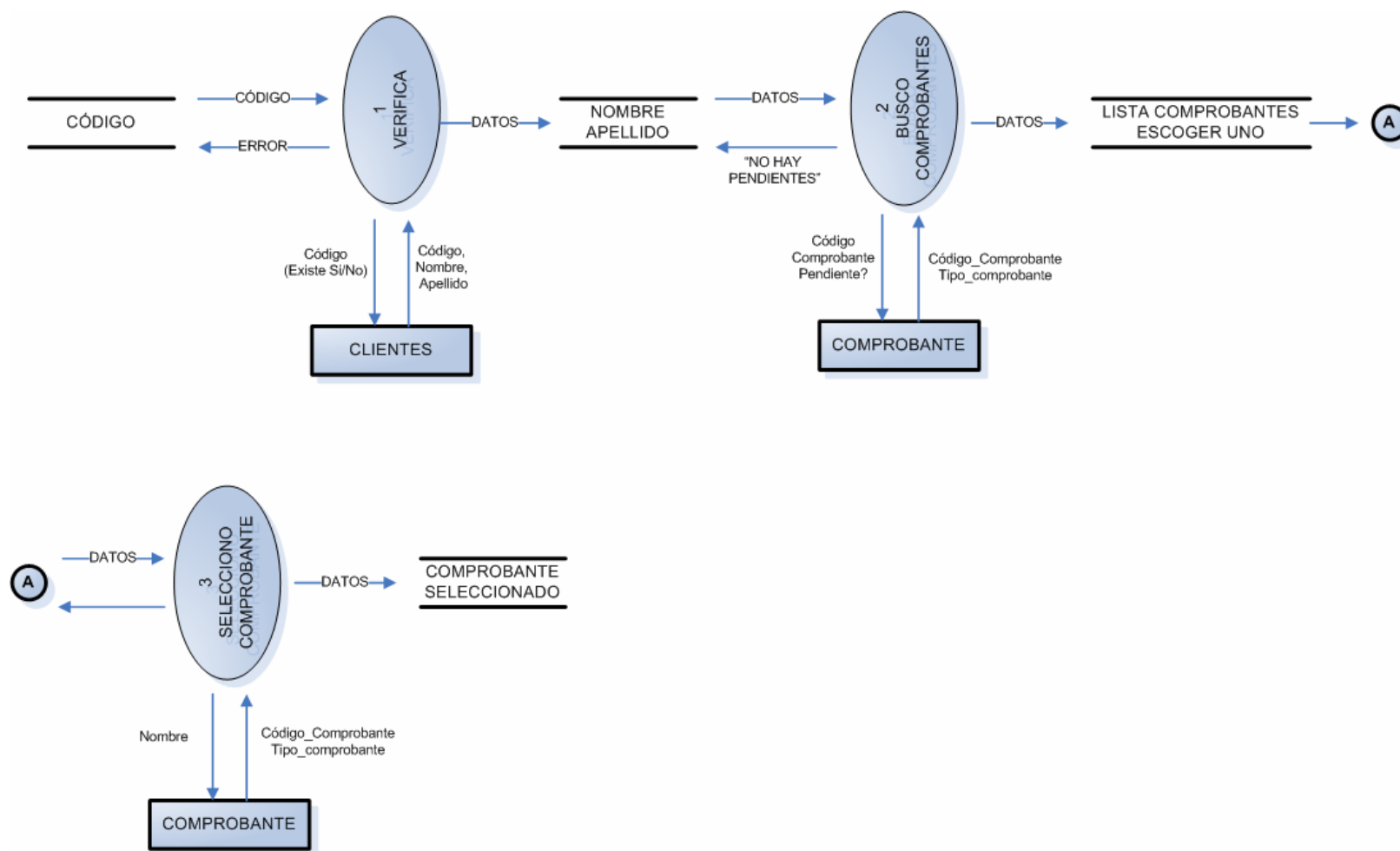
6.2 Apéndices

6.2.1 APENDICE 6.2.1: DICCIONARIO DE DATOS DE LA BASE DE DATOS DE LA APLICACIÓN.

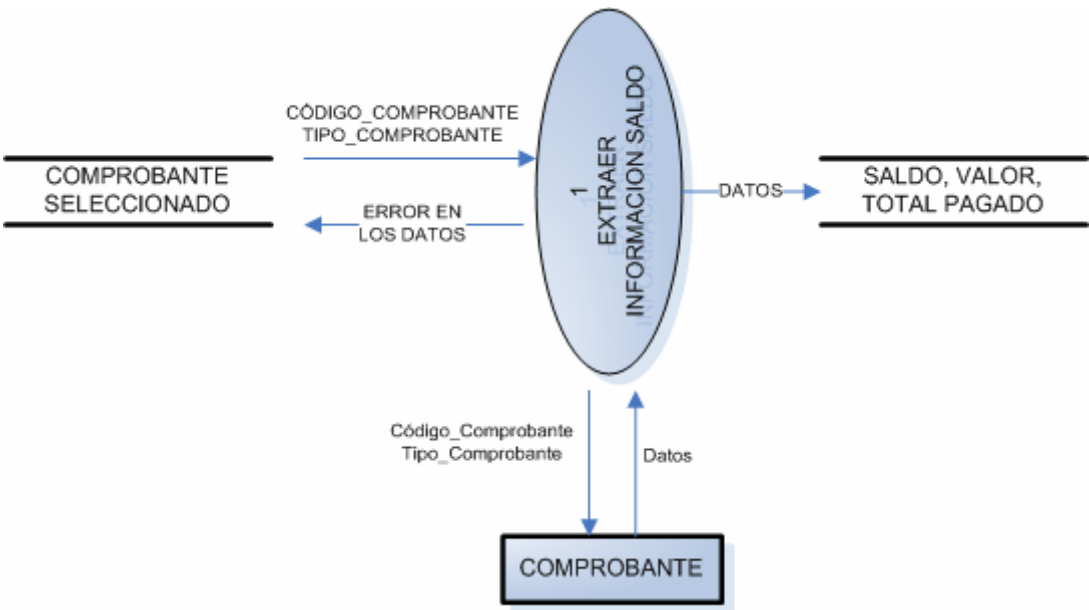
A continuación se presenta el diccionario de datos de la aplicación desarrollada para la consulta de datos de saldos de los clientes de Comercial Jaher.

TABLA	CAMPOS	TIPO	LONGITUD	FORMATO	LLAVE	DESCRIPCIÓN
<i>Cientes</i>						
	cli_cedula	varchar	14		 LP	Cédula del Cliente
	cli_nombre	varchar	30			Nombre del Cliente
	cli_apellido	varchar	30			Apellido del Cliente
	cli_telefono1	varchar	9			Número Telefónico del Cliente
	cli_telefono2	varchar	9			Número Telefónico Opcional del Cliente
<i>Comprobante</i>						
	com_cedula	varchar	14		 LF	Cédula del Cliente
	com_tipo_comprobante	char	3		LP	Tipo de Comprobante
	com_num_comprobante	varchar	10			Número del Comprobante
	com_fecha	date		aaaa-mm-dd		Fecha del Comprobante
	com_valor	double	14,2			Valor del Comprobante
	com_saldo	double	14,2			Saldo del Comprobante
	com_cuotas_gratis	integer	2			Número de Cuotas Gratis del Comprobante
<i>Cuotas</i>						
	cuo_cedula	varchar	14		 LF	Cédula del Cliente
	cuo_fecha	date		aaaa-mm-dd	 LP	Fecha de Comprobante
	cuo_tipo_comprobante	char	3			Tipo de Comprobante
	cuo_numero	varchar	10			Número del Comprobante
	cuo_secuencia	char	2			Número de Secuencia de la Cuota
	cuo_fecha_vencimiento	date		aaaa-mm-dd		Fecha de Vencimiento de la Cuota
	cuo_valor_cuota	double	14,2			Valor de la Cuota
	cuo_saldo	double	14,2			Saldo de la Cuota

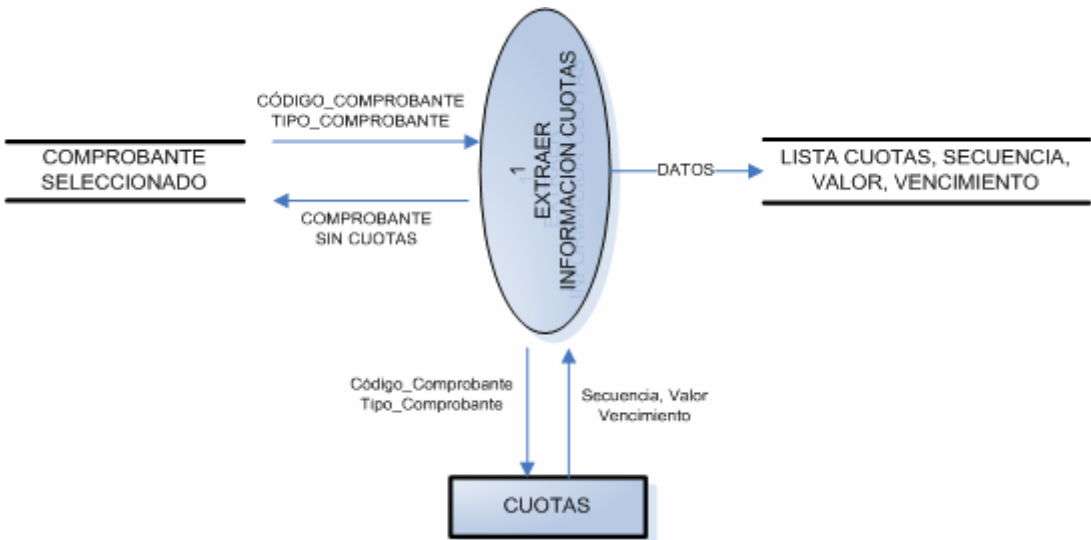
APENDICE 6.2.2: DIAGRAMA DE FLUJO DE INFORMACION DE CONSULTA DE COMPROBANTES.



APENDICE 6.2.3: DIAGRAMA DE FLUJO DE INFORMACION DE CONSULTA DE SALDOS.



APENDICE 6.2.4: DIAGRAMA DE FLUJO DE INFORMACION DE CONSULTA DE CUOTAS.



6.3 Bibliografía

- GRUPO EDITORIAL MEGABYTE. Programación Básica para dispositivos Móviles con tecnología WAP (WML-WMLScript), Grupo Editorial Megabyte SAC, Primera Edición, Agosto 2004.

Páginas de Internet consultadas:

- Jupitermedia Corporation. Building dynamic WML sites. Writing the query [en línea]. Ed. 1. [s.l.], Jupitermedia Corporation, 2006, 2006, [citado: 2005-12-12] Disponible en World Wide Web: <http://www.phpbuilder.com/columns/baravalle20010221.php3?page=6>

- GRACIA M, Joaquín. Configuración del servidor. Tipos MIMES [en línea]. Ed. 1. [España]: Joaquín Gracia, 2004, Abril 2002, [citado: 2005-12-12]. Disponible en World Wide Web: <http://www.webestilo.com/wml/serv01.phtml>

- GRACIA M, Joaquín. Configuración del servidor. Tipos MIMES [en línea]. Ed. 1. [España]: Joaquín Gracia, 2004, Abril 2002, [citado: 2006-01-09]. Disponible en World Wide Web:

<http://www.webestilo.com/wml/cap01.phtml>

<http://www.webestilo.com/wml/cap04.phtml>

<http://www.webestilo.com/php/php00.phtml>

<http://www.webestilo.com/wml/serv01.phtml>

- GRACIA M, Joaquín. Configuración del servidor. Tipos MIMES [en línea]. Ed. 1. [España]: Joaquín Gracia, 2004, Abril 2002, [citado: 2006-01-10]. Disponible en World Wide Web: <http://www.webestilo.com/wml/cap05.phtml>
- INÍCIAte PIO9.COM . WAP Base de Datos, Archivos, Correo [en línea]. Ed. 1. [s.l.]: Juan A. Villalpando, 2002, 2006, [citado: 2005-12-12]. Disponible en World Wide Web: <http://www.pio9.com/222wap.htm>
- RÍOS, Sergio. Manual WML [en línea]. Ed. 1. [s.l.]: Sergio Ríos, 1999, 2002, [citado: 2006-01-16]. Disponible en World Wide Web:
<http://www.terra.es/personal/wapfacil/manwml.htm#formateo>
<http://www.terra.es/personal/wapfacil/manwml2.html>
<http://www.terra.es/personal/wapfacil/manwml4.htm>
<http://www.terra.es/personal/wapfacil/manwml5.htm>
<http://www.webexperto.com/articulos/articulo.php?cod=47>
- ROMO Z. Fabián. WAP, la unión del Internet y el celular [en línea]. Sin ed.. [s.l.]: ROMO Z. Fabián, [citado: 2006-01-16]. Disponible en World Wide Web: <http://www.enterate.unam.mx/Articulos/2004/Febrero/wap.htm>
- NIETO P, Iván. Introducción a la Tecnología WAP [en línea]. Ed. 1. [España]: Joaquín Gracia, 2004, Abril 2002, [citado: 2006-01-10]. Disponible en World Wide Web: <http://www.webestilo.com/wml/cap05.phtml>

- HOUSTE, François. Listas de selección de opciones múltiples [en línea]. Sin ed. [s.l.]: Francois Houste, 2003, 2003, [citado: 2006-01-17]. Disponible en World Wide Web: <http://webmaster.lycos.es/topics/technic/wap/wap-workshop5/3/>
- RODRÍGUEZ, José. Funciones PHP de acceso a Mysql [en línea]. Sin ed. [s.l.]: José Antonio Rodríguez, 2000, 2000, [citado: 2006-01-17]. Disponible en World Wide Web: http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/mysql/mysql_func.htm
- WEBTALLER INC. Print() y Sprintf() en PHP [en línea]. Sin ed. [s.l.]: Webtaller, 2003, 2005, [citado: 2006-01-18]. Disponible en World Wide Web: <http://www.webtaller.com/construccion/lenguajes/php/lecciones/printf-sprintf-php.php>
- CASTELLANO G, J. Instalación del servidor de páginas WML y de un emulador [en línea]. Sin ed. [s.l.]: G. J. Catellano, 2000, 2000, [citado: 2006-01-19]. Disponible en World Wide Web: <http://flanagan.ugr.es/wap/curso/apache.htm>
- PAGOAGA F, Juan. Seguridad en WAP [en línea]. Sin ed. [s.l.]: Juan F. Pagoaga, [citado: 2006-01-23]. Disponible en World Wide Web: http://www.wmlclub.com/articulos/seguridad.htm#_Toc489251031
- PHP BUILDER. Strchr [en línea]. Sin ed. [s.l.]: PHP BUILDER, 2001, 2002-10-31, [citado: 2006-01-31]. Disponible en World Wide Web: <http://www.phpbuilder.com/manual2/manual/es/function.strchr.php>

- POZO, S. Instalación del servidor de páginas WML y de un emulador [en línea]. Sin ed. [s.l.]: Salvador Pozo, Diciembre 2003, Diciembre 2003, [citado: 2006-02-08]. Disponible en World Wide Web:

<http://mysql.conclase.net/curso/index.php?fun=CAST>

<http://www.phpes.net/manual/index.php?ver=function.substr.php>

http://mysql.conclase.net/curso/index.php?sen=LOAD_DATA

- DATTATEC.COM. Seminario sobre Historia de la tecnología WAP. Sin ed. [s.l.] 2005-12-26, 2005-12-26, [citado: 2006-02-22] Disponible en World Wide Web: http://seminarios.dattatec.com/seminario_281205.doc

ANEXOS

ANEXO 1: COSTOS DE LAS OPERADORAS A NIVEL NACIONAL POR BRINDAR EL SERVICIO WAP

El servicio Internet Móvil ofrece a los clientes la posibilidad de acceder a Internet empleando el teléfono celular como un MODEM que se conecta a la computadora portátil, utilizando para ello la tecnología GPRS (*General Packet Radio Service*) que brinda a los clientes mayor velocidad en transmisión de datos. Con dicho servicio se pueden alternar sesiones de datos y voz ofreciendo mayor capacidad de acceso inalámbrico a información y más opciones para el consumidor, mediante portafolios de aplicaciones personales. Junto con ello hay una disminución significativa de los tiempos de conexión.

Los proveedores de servicio WAP en nuestro país son Operadora de Telefonía Movistar que trabaja mediante la tecnología CDMA 1x la cual permite contar con velocidades de transmisión de hasta 2.4 mbps y tiene un costo de tarifado por minuto de navegación que es:

Plan Prepago = \$ 0.04 / minuto

Plan Pospago = \$ 19.99 / 25 horas = \$ 0.01 / minuto

Otro de los proveedores es Porta que trabaja con la tecnología *General Packet Radio Service* (GPRS) que permite una nueva manera de enviar y recibir información usando un teléfono celular. Con GPRS se puede navegar en portales WAP directamente en la pantalla de tu celular o Internet desde tu computador portátil o tu PDA utilizando como MODEM tu teléfono 3GSM. La Operadora Porta tarifa por la cantidad de información transmitida o recibida en Kilobytes aun si esta conectado todo el tiempo a Internet. Los costos concernientes son:

Plan	MB Incluidos	Precio	Kb Adicional
INTERNET+WAP0.5MB*	1,5	\$3	0,0030
INTERNET+WAP 1MB*	3	\$5	0,0030
INTERNET+WAP 20MB*	20	\$29	0,0030
GSM TURBO 400M**	400	\$49	0,0005
GSM TURBO 800M**	800	\$79	0,0005

ANEXO 2: PROGRAMAS FUENTE DE LA APLICACIÓN DESARROLLADA

Este anexo se encuentra disponible en el CD adjunto de la aplicación en un directorio denominado “WML”. Dentro del mismo se encuentran los archivos *.PHP que se han desarrollado para la implementación de la aplicación.

ANEXO 3: GLOSARIO DE TERMINOS

API: (*Application Programming Interface*) Interfaz de Programación de Aplicaciones es un conjunto de especificaciones de comunicación entre componentes *software*.

CGI: (*Common Gateway Interface*) Pasarela de Interfaz Común, es una importante tecnología de la *World Wide Web* que permite a un cliente (explorador Web) solicitar datos de un programa ejecutado en un servidor Web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor Web y una aplicación externa.

FTP: (*File Transfer Protocol*) Protocolo de Transferencia de Archivos es uno de los diversos protocolos de la red Internet, y es el ideal para transferir grandes bloques de datos por la red.

GSM: (*Global System for Mobile Communications*) Sistema Global para las Comunicaciones Móviles, formalmente conocida como "*Group Special Mobile*" (GSM, Grupo Especial Móvil) es un estándar mundial para teléfonos móviles digitales.

HDML: (*Handheld Device MarkupLanguage*) Lenguaje de marcado para aparatos de mano. Modificación del lenguaje html desarrollada por la empresa *Unwired Planet* que permite usar el protocolo de transferencia de hipertexto http para que se puedan ver versiones en modo texto de páginas Web en los teléfonos móviles, las agendas electrónicas (PDA's) y los beepers.

HDTP: (*Handheld Device Transport Protocol*) Protocolo de transporte para dispositivos portátiles Utilizado para construir aplicaciones Web para dispositivos portátiles o "de bolsillo" como teléfonos inteligentes.

HTML: (*Hypertext Markup Language*) Lenguaje de formato de documentos de hipertexto, es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

HTTPS: Versión segura del protocolo HTTP. El sistema HTTPS utiliza un cifrado basado en las *Secure Socket Layers* (SSL) para crear un canal cifrado cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente más apropiado para el tráfico de información sensible que el protocolo HTTP.

ITTP: (*Intelligent Terminal Transfer Protocol*) Protocolo de Transferencia para Terminales Inteligentes, definía la comunicación entre el nodo de servicio donde reside la aplicación y el teléfono móvil inteligente.

MIME: (*Multi-Purpose Internet Mail Extensions*) Extensiones de correo Internet multipropósito, son una serie de convenciones o especificaciones dirigidas a que se puedan intercambiar a través de Internet todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario.

OTA: (*Over-the-Air*) ajuste mediante SMS, permite configurar un teléfono móvil con un requerimiento a un servidor mediante un mensaje de texto originado en el teléfono móvil. Este tipo de configuración a distancia es aún específico de cada teléfono móvil, cada servidor OTA no tiene capacidad para realizar los ajustes de cualquier teléfono móvil. Se pueden enviar y recibir, por ejemplo, favoritos u opciones WAP *Over-the-Air* como *Smart Messaging* (con SMS como medio de transferencia)

PHP: Lenguaje de programación usado generalmente para la creación de contenido para sitios Web. PHP es el acrónimo recursivo de "PHP: *Hypertext Preprocessor*", inicialmente PHP Tools, o, *Personal Home Page Tools*) es un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web

SMTP: (*Simple Mail Transfer Protocol*) protocolo simple de transferencia de correo electrónico. Protocolo de red basado en texto utilizado para el intercambio de

mensajes de correo electrónico entre computadoras o distintos dispositivos (PDA's, Celulares, etc).

SSL: (*Secure Sockets Layer*) proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, solo el servidor es autenticado mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes.

TCP: (*Transmission Control Protocol*) Protocolo de Control de Transmisión, es uno de los protocolos fundamentales en Internet. Muchos programas dentro de una red de datos compuesta por ordenadores pueden usar TCP para crear conexiones entre ellos a través de las cuales enviarse datos.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP) La familia de protocolos de Internet es un conjunto de protocolos de red que implementa la pila de protocolos en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

TLS: (*Transport Layer Security*) Capa de Transporte Segura es una versión estandarizada por el IETF del protocolo SSL que pretende abarcar toda la capa de transporte de la pila OSI.

URL: (*Uniform Resource Locator*) Localizador Uniforme de Recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

VPN: (*Virtual Private Network*) Red Privada Virtual, es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet.

WAE: (*Wireless Application Environment*) Ambiente de Aplicación Inalámbrica, contiene las herramientas que usan los desarrolladores de contenido inalámbrico. Ahí se incluye WML y WMLScript, donde este último opera de forma similar a como lo hace JavaScript

WAP: (*Wireless Application Protocol*) Protocolo de Aplicaciones Inalámbricas, es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, por ejemplo: acceso a Internet desde un teléfono móvil.

WBMP: (*Wireless Bitmap*) utilizado para gráficos monocromáticos, permitiendo que los desarrolladores diseñen aplicaciones de interconexión para dispositivos portátiles.

WDP: (*Wireless Datagram Protocol*) Protocolo de Datagrama Inalámbrico, trabaja en conjunto con la capa de transporte de la red, esto facilita la adaptación de WAP a una variedad de ambientes, ya que todas las necesidades de cambio de información se mantienen en este nivel. Los transportes de red son las diversas tecnologías disponibles para el intercambio de datos digitales.

WML: (*Wireless Markup Language*). Lenguaje de Marcas Inalámbrico, describe la estructura de datos que permite presentar los datos de páginas Web en dispositivos móviles a través del acceso inalámbrico.

WSP: (*Wireless Session Protocol*) Protocolo de Sesión inalámbrica, determina si la sesión entre el dispositivo y la red será orientada a la conexión o sin conexión, o sea, si el dispositivo necesitará o no de establecer comunicación una y otra vez hacia la red, después de iniciada la sesión

WTA: (*Wireless Telephony Applications*) un entorno para aplicaciones de telefonía que permite a los operadores la integración de funciones de telefonía del propio dispositivo móvil con el micronavegador incorporado.

WTAI (*Wireless Telephony Application Interface*) Es una interfaz utilizada en los terminales móviles para operaciones locales de control de llamadas (recepción, iniciación y terminación) y de acceso a listines telefónicos.

WTLS (*Wireless Transport Layer Security*) Capa de Seguridad de Transporte Inalámbrico, se encarga de las capacidades de seguridad, revisa la integridad de los datos, así como el encriptamiento, y realiza autenticación de clientes y servidores

WTP: (*Wireless Transaction Protocol*) Protocolo de Transacción Inalámbrico, actúa como un controlador de tráfico, manteniendo los datos en flujo constante. También define cómo clasificar cada petición de transacción, si es posible hacerla en dos sentidos o en uno solo.

WWW: (*World Wide Web*) Telaraña Mundial, la Web o WWW, es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador Web para extraer elementos de información (llamados

"documentos" o "páginas Web") de los servidores Web (o "sitios") y mostrarlos en la pantalla del usuario.

XHTML: (*eXtensible Hyper Text Markup Language*) Lenguaje extensible de marcado de hipertexto, es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML

XML: (*eXtensible Markup Language*) Lenguaje de marcado ampliable o extensible, desarrollado por el *World Wide Web Consortium* (W3C).