UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

DESCRIPCIÓN DE LOS SERVICIOS, ARQUITECTURA Y TENDENCIAS DEL PROTOCOLO WAP PARA EL DESARROLLO DE APLICACIONES PARA REDES INALÁMBRICAS

RENÉ ESTUARDO ALVARADO GONZÁLEZ

ASESORADO POR ING. RENÉ FRANCISCO CONTRERAS QUEMÉ

GUATEMALA, OCTUBRE DE 2003

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DESCRIPCIÓN DE LOS SERVICIOS, ARQUITECTURA Y TENDENCIAS DEL PROTOCOLO WAP PARA EL DESARROLLO DE APLICACIONES PARA REDES INALÁMBRICAS

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

RENÉ ESTUARDO ALVARADO GONZÁLEZ
ASESORADO POR: ING. RENÉ FRANCISCO CONTRERAS QUEMÉ

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2003

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Sydney Alexander Samuels Milson
VOCAL I	Ing. Murphy Olympo Paiz Recinos
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Keneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Ricardo Alfredo Girón Solórzano
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Luis Alberto Vettorazzi España
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DESCRIPCIÓN DE LOS SERVICIOS, ARQUITECTURA Y TENDENCIAS DEL PROTOCOLO WAP PARA EL DESARROLLO DE APLICACIONES PARA REDES INALÁMBRICAS

Tema que me fuera asignado por la Coordinación de la Carrera de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería con fecha 11 de Febrero de 2003.

René Estuardo Alvarado González

ÍNDICE GENERAL

INDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

AMPS *Advance Mobile Phone System*
Especificación estándar original para los sistemas analógicos de la primera generación de la telefonía móvil.

API *Application Programming Interface*
Interfaz de Programación de Aplicación

Autenticación proceso de identificar y confirmar la identidad de cualquier teléfono desde el que se llame o que reciba una llamada.

CDMA *Code Division Multiple Access*
Acceso Múltiple por División en el Código; tecnología celular que no asigna una frecuencia específica a cada usuario, sino que cada canal utiliza un espectro completamente disponible

Celda Area geográfica que contiene una antena y los dispositivos necesarios para recibir señales desde otra celda. Abarcan diámetros desde unos pocos kilómetros hasta unos 32

Conmutación de Paquetes Método de conmutar una red por la cual se aceptan y entregan a destino paquetes individuales de tamaño y formato fijo.

Cookies Pequeños archivos de texto que un sitio de internet puede usar para reconocer usuarios que repiten el acceso al lugar, facilitar el acceso continuado de estos, y permite al sitio el seguimiento del comportamiento del usuario y la recolección de datos acumulados que permitirán la mejora del contenido y la publicidad dirigida.

Emulador Herramienta utilizada por el desarrollador de páginas WML para que pueda tener una idea aproximada del comportamiento del código en tiempo de ejecución.

GPRS *General Packet Radio Service*
Servicio de Radio de Paquetes Generales; tecnología vinculada a

paquetes que posibilita una Internet móvil de alta velocidad (115 Kb por segundo) y otras comunicaciones de datos.

GSM *Global System for Mobile Communications*
Sistema Global para Comunicaciones Móviles, protocolo móvil que opera en la banda de frecuencia entre 900 y 1800 MHz.

IP *Internet Protocol*
Protocolo de Internet

MAC *Medium Access Control*
Control de Acceso al Medio

MDG *Mobile Data Gateway*
Pasarela de Datos Móviles

Navegador Aplicación que permite "navegar" por distintas zonas de Internet y visitar un portales.

PPP *Point-to-Point Protocol*
Protocolo Punto-a-Punto.

PDA *Personal Digital Assistant*
Asistente personal digital, nombre ordinario que se le da a las agendas electrónicas de bolsillo.

Roaming Sistema mediante el cual el teléfono se va conectando de manera automática a las células que le permitan una mejor comunicación.

SMS *Short Message Services*
Servicio disponible en redes digitales que permite enviar y recibir hasta 160 caracteres en un teléfono móvil, a través del centro de mensajes del operador de la red.

TDMA *Time Division Multiple Access*

Tecnología que permite llevar servicios inalámbricos digitales utilizando y manteniendo diversos flujos de información de manera independiente a través de un mismo canal de comunicación.

Terminal Estación de trabajo dentro de una red de computadoras, trabaja en base a los datos que le provee uno ó mas servidor(es) de información.

UMTS *Universal Mobile Telephone System*.

Es el protocolo de la tercera generación de teléfonos celulares.

WAE *Wireless Application Environment*

Entorno Inalámbrico de Aplicación, Parte integrada del Protocolo de Aplicaciones Inalámbricas cuyo fin, basado en tecnología para WWW, es especificar todas aquellas materias que permitan a los operadores y proveedores crear aplicaciones y servicios accesibles desde la mayor gama posible de plataformas.

WAP Browser *Un micro navegador incluido en el teléfono móvil en Wireless Markup Language (WML)*

WAP Gateway Dispositivo de dos direcciones a través del cual se convierte el contenido que hay en el WAP Server al formato WML que pueda entender la terminal WAP.

WAP Server Básicamente es un servidor HTTP, es decir un servidor de la red Internet como otro cualquiera. En cambio, Nokia llama WAP Server a un conjunto de dos productos: un gateway WAP y un servidor HTTP (servidor de contenidos).

WAP *Wireless Application Protocol*

Protocolo Inalámbrico de Aplicación, tecnología que permite enviar y recibir información de Internet desde un teléfono móvil,

WDP *Wireless Datagram Protocol*

Protocolo Inalámbrico de Datagramas.

WSP *Wireless Session Protocol*

Protocolo Inalámbrico de Sesión; Hace referencia a la aplicación de más alto nivel que ofrece WAP a través de un interfaz para servicios de dos sesiones. La primera consistiría en un servicio con conexión que operaría sobre el protocolo del nivel de transacción y el segundo sería sin conexión y operaría sobre el servicio de transporte de información.

WTA *Wireless Telephony Applications*

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.

WTP *Wireless Transaction Protocol*

Protocolo Inalámbrico de Transacciones.

RESUMEN

WAP es un protocolo para aplicaciones inalámbricas, que nace de la idea de prestar acceso a Internet a través de un dispositivo móvil, este consiste en un conjunto de especificaciones, que permite la utilización de un lenguaje de marcas inalámbrico permitiendo que los desarrolladores diseñen aplicaciones de interconexión para dispositivos portátiles.

El protocolo WAP pretende dar una solución unificada para los servicios de valor añadido existentes y futuros. Incluye especificaciones para las capas de la torre OSI de sesión y de transporte, así como funcionalidades de seguridad. WAP también define un entorno de aplicaciones. Como cualquier estándar, las ventajas son múltiples a la hora de desarrollar aplicaciones, fabricar terminales o estructurar la red.

El diseño de WAP fue creado para trabajar bajo restricciones de memoria y procesadores está convirtiendo a WAP en el estándar, permitiendo que cualquier usuario con un dispositivo de mano pueda tener acceso a la información, lo que ha traído por consiguiente que cada vez más información se transmita por redes inalámbricas, tema que también es tratado en este trabajo de graduación.

WAP ha heredado gran parte del diseño de red de Internet, por lo que proporciona grandes ventajas tanto para el desarrollo de aplicaciones como para los operadores que quieran ofrecer WAP, por lo que en la arquitectura básica del protocolo, se definen entonces dos partes bien diferenciadas, la

primera entre el el dispositivo *cliente* y una pasarela de información (*gateway*) y la segunda entre la pasarela y el servidor de información.

Por último, se hace hincapié en la generación dinámica de contenidos para WAP, donde se tratarán los temas de análisis de modelos y tecnologías de procesamiento en servidor existentes y la negociación del formato de presentación de los contenidos para su correcta visualización en terminales móviles concretos.

OBJETIVOS

- **General**

Presentar una visión amplia del protocolo WAP, desde sus inicios hasta una introducción a la nueva versión del protocolo, WAP 2.0, pasando por su arquitectura, servicios y mostrar una aplicación funcional.

- **Específicos**

1. Dar a entender al lector que la tecnología es un mundo en constante cambio, y su finalidad no es complicar al usuario, si no todo lo contrario, es decir, proveer soluciones que de adapten a las necesidades de la sociedad en que vivimos, en este caso, el acceso a servicios en línea desde un teléfono celular.

2. Mostrar de forma concisa los conceptos clave para entender qué es el protocolo WAP, el por qué del protocolo y cuales son sus tendencias.

3. Mostrar e interpretar claramente la arquitectura del protocolo mediante la explicación de la estructura y funcionamiento de las redes inalámbricas con tecnología 3G.

4. Hacer de este documento un manual de términos de referencia, donde se muestren las tendencias de la tecnología en el campo de las redes inalámbricas y acceso a servicios en línea.

5. Introducir al lector en conceptos que se derivan de WAP, tales como MMS y M-*Commerce*, que en un futuro a mediano plazo revolucionarán nuestra forma de concebir negocios y de comunicarnos, teniendo la sensación de “ir a un lugar sin estar precisamente en el”

6. Describir los protocolos para manejo de seguridad en las redes inalámbricas, específicamente con el protocolo WAP, para mostrar la confiabilidad del protocolo al momento de realizar transacciones.

7. Desarrollar una aplicación que sirva como ejemplo de cómo el protocolo WAP puede ser utilizado para brindar servicios que contribuyan a facilitar la vida de los usuarios, teniendo acceso a servicios sin necesidad de contar con un ordenador estático para acceder a Internet.

8. Hacer de este documento un manual técnico para referencias sobre el lenguaje WML y WMLScript utilizando ASP para la elaboración de páginas dinámicas que puedan ejecutarse en un ambiente WAP.

INTRODUCCIÓN

No se puede negar que la unión de Internet y telefonía móvil ha sido un campo que ha alcanzado un elevadísimo grado de desarrollo tecnológico, y podría decirse que es uno de los tópicos que más investigación genera en el ámbito mundial actualmente.

El principal objetivo de estas investigaciones es llegar al punto de permitir acceso a cantidades de información tal, y un punto de ubicuidad tal, que nuestra terminal móvil se convierta en un objeto cotidiano, indispensable y absolutamente popular, que permitirá al usuario beneficiarse de aplicaciones multimedia en tiempo real sin importar distancias ni volúmenes de información, sin importar tipos de aplicaciones ni horarios de acceso.

Es en este ambiente cuando nace el Wireless Application Protocol o Protocolo de Aplicaciones Inalámbricas (WAP) el cual es un intento de estandarizar el acceso inalámbrico a Internet desde cualquier sitio a cualquier momento

Sobre esta premisa se basaron en 1997 pesos pesados en el mundo de las telecomunicaciones tales como Nokia, Motorola, Ericsson y Phone.com para unirse y dar origen al *Wapforum*, organización que estandariza y regula las especificaciones en el mundo WAP a diversos dispositivos inalámbricos tales como teléfonos móviles, buscapersonas y asistentes digitales personales. El *Wapforum* ha querido beneficiarse de la arquitectura *World Wide Web* y de tecnologías de Internet ya existentes tales como IP, HTTP, XML, SSL, URLs, *Scripting* y otros formatos de contenido, para desarrollar a partir de ellas nuevas especificaciones.

Debido a su naturaleza nómada, WAP esta diseñado para operar sobres dispositivos móviles con grandes limitaciones respecto al lo que se podría encontrar en una PC de escritorio. CPUs menos potentes, menos memoria ROM y RAM, pequeñas pantallas y bajo consumo son las constantes en estos dispositivos.

Por supuesto, el lenguaje HTML no es él mas indicado para llevar información a dispositivos de tales características, es por eso que se adopto el *Wireless Markup Language* (WML) y el *Wireless Markup Language Script* (WML Script) los cuales son una versión reducida del *Extensible Markup Language* (XML) y del *ECMAScript*, respectivamente, permitiendo la escalabilidad desde dos líneas de texto hasta gráficos de pantalla completa.

Otro aspecto a considerar es la tecnología *iMode* desarrollada por la japonesa NTT DoCoMo que en relativamente poco tiempo ha popularizado el acceso a contenido a través de terminales móviles llegando a millones de usuarios, recolectándolos a un ritmo de varios miles de personas por semana. Aunque *iMode* se desarrollo con esta rapidez debido a que tiene completo control de las redes y terminales que soportan su servicio, no se piensa que *iMode* sea un rival de WAP, de hecho se a anunciado su cooperación y se piensa que en el futuro puedan coexistir.

El autor de este trabajo de grado tiene muy claro que WAP no será la tecnología que dominara a largo plazo el servicio de Internet inalámbrico, pero si que es un paso necesario e indispensable para adecuar el ambiente a la llegada de los servicios de Tercera Generación, y en las siguientes paginas esto tratará de ser demostrado con claridad.

1. VISTA GLOBAL AL PROTOCOLO Y ESTÁNDAR WAP

No hay duda que los nuevos dispositivos tecnológicos son más potentes y livianos cada vez, permitiendo que nuestra comunicación sea cada vez más eficaz.

El Internet es una forma de comunicación universal que captura la atención de miles de personas diariamente y gracias a la variedad de formas de conectarse posibles, deben surgir nuevas tecnologías que se adapten a los dispositivos en el mercado, dándoles nuevas opciones.

Los protagonistas tecnológicos de los últimos años han sido principalmente dos: Internet y las telecomunicaciones.

- Internet como medio de obtención de servicios e información de manera rápida, amena y global. Se estiman cerca de 150 millones de usuarios en el mundo, pero la cifra aumenta rápidamente.

- El crecimiento imparable de las telecomunicaciones y especialmente la telefonía móvil motivado por una parte por los adelantos tecnológicos que han permitido la fabricación de dispositivos portátiles más pequeños, ligeros y manejables.

El estándar de acceso a la Red desde dispositivos móviles nace sobre GSM *(Global System for Mobile Communications)*. Sin embargo, se auguran grandes avances en el sector en los próximos años.

Aunque la introducción en el mercado de los móviles WAP está siendo relativamente lenta, las principales marcas ya han iniciado la comercialización de al menos un modelo. Hasta tal punto se prevé el éxito de su implantación, que muchos estudios de mercado ya aseguran que en cuestión de tan sólo un par de años habrá más móviles conectados a Internet que PCs. A pesar de ello, muchos se han aventurado a decir - y aún lo sostienen -, que en tan pequeña pantalla la navegación por Internet podría resultar lenta e incómoda.

1.1 El estándar *WAP*

1.1.1 El foro *WAP*

El Foro *WAP* (*Wireless Application Protocol Forum*) es un grupo industrial dedicado a habilitar servicios de información y telefonía en dispositivos inalámbricos móviles. Estos dispositivos incluyen teléfonos móviles, *pagers*, PDAs (*Personal Digital Assistant*) y otros terminales inalámbricos. Basándose en la arquitectura del World Wide Web, el Foro WAP alinea su tecnología de manera muy próxima a Internet y la Web. La especificación WAP se extiende e influye en las tecnologías existentes (como los estándares de tratamiento digital de señal) y tecnologías de Internet (como IP, HTTP, XML, SSL, URL, etc.).

El Foro WAP se fundó en junio de 1997 por Nokia, Ericsson, Motorola y Phone.com (formalmente *Unwired Planet*). Desde entonces ha experimentado un importante crecimiento en cuanto al número de miembros, que representan el 95% de las empresas de telecomunicaciones. Estas incluyen fabricantes de dispositivos móviles, proveedores de servicios, desarrolladores de software, y otras organizaciones que aportan soluciones inalámbricas. Actualmente lo forman más de 200 miembros.

El objetivo del Foro WAP es crear una especificación global de un protocolo inalámbrico para todas las redes inalámbricas y para asegurar la interoperabilidad del producto y el crecimiento del mercado inalámbrico. Esta especificación WAP habilita a fabricantes, operadores de red, proveedores de contenidos y desarrolladores de software para ofrecer productos compatibles y servicios seguros en todos los dispositivos y redes. Algunas claves de su éxito son:

- Proporcionar contenidos de Internet y servicios avanzados de datos a teléfonos móviles y otros terminales inalámbricos.

- Crear una especificación global de un protocolo inalámbrico independiente de las tecnologías de redes inalámbricas.

- Fomentar la creación de contenidos y aplicaciones para todo tipo de redes inalámbricas y tipos de dispositivos.

- Agrupar y extender los estándares existentes y tecnologías.

El Foro WAP no desarrolla productos, pero crea estándares libres de licencia para que toda la industria pueda usarlos en sus productos. Cada compañía puede introducir sus rasgos específicos y característicos, siempre que se respete el estándar. De esta manera ninguna compañía compite con WAP ya que no favorece ningún producto en particular. En cambio sí asesora a todas las compañías que desarrollan productos basados en las especificaciones WAP.

1.1.2 ¿Qué es el protocolo de aplicaciones inalámbricas?

WAP (*Wireles Application Protocol*, por sus siglas en inglés) es el protocolo para aplicaciones inalámbricas.

Consiste en un conjunto de especificaciones, que se han desarrollado por medio del WAP Forum y que permite la utilización del WML[1] que es el lenguaje de marcas inalámbrico, así como de WBMP[2] utilizado para gráficos monocromáticos, permitiendo que los desarrolladores diseñen aplicaciones de interconexión para dispositivos portátiles.

> Pretende dar una solución unificada para los servicios de valor añadido existentes y futuros. El protocolo incluye especificaciones para las capas de la torre OSI de sesión y de transporte, así como funcionalidades de seguridad. WAP también define un entorno de aplicaciones. Como cualquier estándar, las ventajas son múltiples a la hora de desarrollar aplicaciones, fabricar terminales o estructurar la red.

Y es que los nuevos teléfonos celulares, *pagers*, *palmtops*, etc. están cambiando la forma de comunicación personal portátil. Gracias a esta tecnología estos pequeños aparatos pueden conectarse al Web. El diseño de WAP fue creado para trabajar bajo restricciones de memoria y procesadores, pequeñas pantallas monocromáticas capaces de desplegar muy pocas líneas de texto y conexiones irregulares debido al ancho de banda reducido. Gracias al apoyo de varios cientos de vendedores de estos dispositivos, el WAP Forum, está convirtiendo a WAP en el estándar, permitiendo que cualquier usuario con un dispositivo de mano pueda tener acceso a la información, lo que ha traído por consiguiente que cada vez más información se transmita por redes inalámbricas.

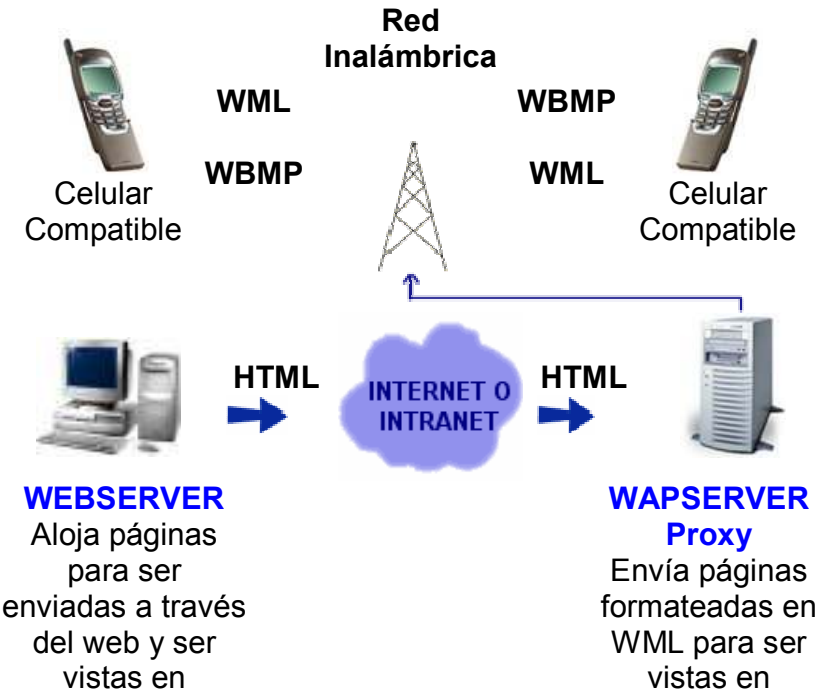
[1] Wireless Markup Language (Ver Capítulo 6, dedicado exclusivamente a WML)

[2] Wireless Bitmap (Mapa de bits con un formato especial para ser soportado por aplicaciones WAP)

El Protocolo de Aplicaciones Inalámbricas surge como la combinación de dos tecnologías de amplio crecimiento y difusión durante los últimos años: Las Comunicaciones Inalámbricas e Internet. Mas allá de la posibilidad de acceder a los servicios de información contenidos en Internet, el protocolo pretende proveer de servicios avanzados adicionales como, por ejemplo, el desvío de llamadas inteligente, en el cual se proporcione una interfaz al usuario en el cual se le pregunte la acción que desea realizar: aceptar la llamada, desviarla a otra persona, desviarla a un buzón vocal, etc.

Para ello, se parte de una arquitectura basada en la arquitectura definida para el WWW (World Wide Web), pero adaptada a los nuevos requisitos del sistema. En la Figura 1 se muestra el esquema de la arquitectura WAP.

Figura 1: Modelo de funcionamiento del WAP



navegadores dispositivos portátiles.

De esta forma, en el terminal inalámbrico existiría un "Micro Navegador" [3] encargado de la coordinación con la pasarela, a la cual la realiza peticiones de información que son adecuadamente tratadas y redirigidas al servidor de información adecuado. Una vez procesada la petición de información en el servidor, se envía esta información a la pasarela que de nuevo procesa adecuadamente para enviarlo al terminal inalámbrico. Para conseguir consistencia en la comunicación entre el terminal móvil y los servidores de red que proporcionan la información, WAP define un conjunto de componentes estándar:

- Un modelo de nombres estándar. Se utilizan las URIs[4] definidas en WWW para identificar los recursos locales del dispositivo (tales como funciones de control de llamada) y las URLs[5] (también definidas en el WWW) para identificar el contenido WAP en los servidores de información.

- Un formato de contenido estándar, basado en la tecnología WWW.

- Unos protocolos de comunicación estándares, que permitan la comunicación del micro navegador del terminal móvil con el servidor Web en red.

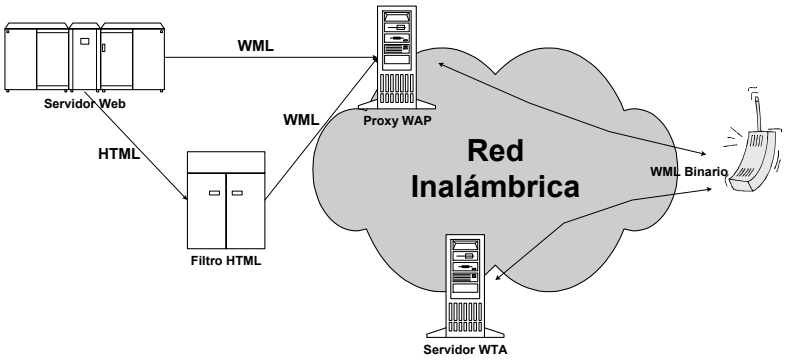
El modelo global de funcionamiento de este sistema se puede observar en la Figura 2.

[3] Se pretende que este *micro navegador* actúe de interfaz con el usuario de la misma forma que lo hacen los navegadores estándar.

[4] *Universal/Uniform Resource Identifier* ó Identificador Uniforme/Universal de Recurso

[5] *Universal/Uniform Resource Location* ó Localización Universal/Uniforme de Recurso

Figura 2: Ejemplo de una red WAP



En el ejemplo de la figura, nuestro terminal móvil tiene dos posibilidades de conexión: a un Proxy WAP, o a un servidor WTA. El primero de ellos, el *Proxy* WAP traduce las peticiones WAP a peticiones Web, de forma que el cliente WAP (el terminal inalámbrico) pueda realizar peticiones de información al servidor Web. Adicionalmente, este Proxy codifica las respuestas del servidor Web en un formato binario compacto, que es interpretable por el cliente. Por otra parte, el segundo de ellos, el servidor WTA[6] está pensado para proporcionar acceso WAP a las facilidades proporcionadas por la infraestructura de telecomunicaciones del proveedor de conexiones de red.

1.1.2.1 Características principales del WAP

Las características principales se describen a continuación:

- El propósito de WAP es habilitar la entrega rápida y fácil de información y servicios a los usuarios de móviles.

[6] *Wireless Telephony Application* ó Aplicación de Telefonía Inalámbrica

- WAP es escalable, permitiendo así a las aplicaciones disponer de las capacidades de pantalla y recursos de red según su necesidad y en un gran número de tipos de terminales. Los servicios podrán ser aplicables a pantallas de una sola línea o a terminales mucho más complejos.

- WAP trabaja con dispositivos inalámbricos digitales como teléfonos móviles, PDAs (*Personal Digital Assistant*), *Handhelds*, *pagers*, radios bidireccionales, *smartphones* y comunicadores.

- WAP está diseñado para trabajar con la mayoría de las redes inalámbricas actuales como CDPD, CDMA, GSM, PDC, PHS, TDMA, FLEX, ReFLEX, iDEN, TETRA, DECT, DataTAC, Mobitex, así como para las redes futuras como UMTS.

- WAP es un protocolo de comunicaciones y un entorno de la aplicación. Puede implantarse en cualquier sistema operativo incluso PalmOS, EPOC, Windows CE, FLEXOS, OS/9, JavaOS etc. Proporciona interoperabilidad de servicio incluso entre diferentes familias de dispositivos.

WAP es el estándar "de-facto", es decir, es de hecho el estándar más usado aunque no haya sido creado o regulado por una organización internacional de estandarización.

A continuación se muestran con más detalle cuatro de las principales características de la tecnología WAP: Base en estándares existentes, independencia de la red, independencia del dispositivo, e interoperabilidad.

1.1.2.1.1 WAP basado en estándares existentes

WAP busca usar en lo máximo posible los estándares existentes como base para su propia arquitectura y diseño. Por ejemplo, la especificación WAP no especifica como se transmiten los datos a través del interfaz aéreo. En vez de eso, la especificación WAP está pensada para situarse por encima de los estándares de las redes portadoras, de tal manera que cada estándar se puede usar con los protocolos WAP para implementar soluciones de producto completas.

1.1.2.1.2 Mantener la independencia de la red

El objetivo es minimizar las restricciones que impone la red inalámbrica como puede ser el bajo ancho de banda, alta latencia, menor estabilidad en la conexión y menor disponibilidad predecible.

Para lograr la mayor aceptación y generalización hacia el mayor número de usuarios, WAP está diseñado para trabajar de manera óptima en todo tipo de redes inalámbricas.

Este principio permite a fabricantes, programadores y servidores beneficiarse de un estándar unificado. Los proveedores de servicios pueden implementar una solución común para todas las diferentes redes, para que cada usuario pueda hacer mejor uso de su experiencia en cada red. Las aplicaciones pueden desarrollarse usando un estándar que trabaje en todas las variedades de redes. Los fabricantes de terminales pueden usar el mismo software en todas sus líneas de producción, reduciendo el tiempo de desarrollo y disminuyendo los problemas de soporte.

Requiriendo mínimas demandas a los portadores, WAP puede trabajar en casi todas las interfaces. Define un protocolo que puede trabajar en redes con pequeño ancho de banda como SMS (*Short Message Service*) o GSM USSD (*Unstructured Supplementary Service Data*).

El protocolo WAP se creo inicialmente sobre la base de GSM (segunda generación de telefonía móvil) pero se adapta perfectamente UMTS (denominada la tercera generación).

Al ser independiente del interfaz aéreo, sólo es necesario que distintos futuros interfaces cumplan las especificaciones, en vez de buscar un interfaz estándar único.

1.1.2.1.3 Mantener la independencia del dispositivo

La especificación WAP es también independiente de cualquier tipo de dispositivo. En cambio, especifica los requisitos mínimos de funcionalidad que un dispositivo debe tener, y está diseñado para adaptarse a cualquier funcionalidad extra que se quiera añadir por encima de ese mínimo.

La independencia del dispositivo otorga beneficios similares a la independencia de la red: aplicaciones desarrolladas para un estándar pueden trabajar en una amplia gama de dispositivos que implementen la especificación; los operadores de red ganan un interfaz de usuario consistente para sus servicios utilizando modelos de diferentes fabricantes; los programadores no tienen que diseñar versiones diferentes de sus programas para cada dispositivo; y los proveedores de servicios pueden elegir aquel dispositivo que se adecue a sus características de mercado

específicas. Los fabricantes de terminales se aseguran que tendrán más aplicaciones diseñadas para sus dispositivos con sólo implementar la especificación, y además pueden añadir sus propios rasgos característicos por encima de los mínimos que exige el estándar para que su modelo sea único en el mercado y que siga habiendo competitividad.

1.1.2.1.4 Asegurar interoperabilidad

Los proveedores de servicios deben sentir de manera segura de que sacarán rendimiento de sus inversiones en el futuro. No serán capaces de hacerlo hasta que el equipamiento y el software ofrecido por diferentes proveedores estén diseñados de tal forma que puedan trabajar juntos. La especificación WAP está diseñada para animar la interoperabilidad fácil y abierta entre sus componentes. Cualquier solución construida para adaptarse a WAP puede interoperar con cualquier otro componente que también esté adaptado a WAP. Los proveedores de servicios pueden elegir equipamiento y software de múltiples vendedores, seleccionando cada pieza de su solución que es apropiada para las necesidades particulares de cada proveedor de servicios.

La independencia de red y dispositivo ayudan mucho a la interoperabilidad. Pero este concepto va más lejos que estos dos principios requiriendo que cada componente compatible con WAP se pueda comunicar con otros componentes usando los métodos estándar y protocolos definidos en las especificaciones.

La interoperabilidad proporciona claros beneficios a los fabricantes de terminales y proveedores de infraestructuras. Los fabricantes de terminales se

aseguran que si sus dispositivos cumplen la especificación WAP, son capaces de comunicarse con cualquier servidor WAP, independiente del fabricante. Igualmente los creadores de servidores WAP se aseguran que cualquier terminal es capaz de comunicarse con ellos.

1.1.3 Especificación WAP

La especificación WAP define a un mismo tiempo una arquitectura estándar abierta y una serie de protocolos pensados para implementar el acceso inalámbrico a Internet. También proporciona soluciones para problemas no resueltos por otros estándares (W3C, ETSI, TIA, IETF, etc.).

Los principales elementos de la especificación WAP son:

- Una definición de un Modelo de Programación WAP, que está basado fuertemente en el existente Modelo de Programación WWW (aunque también se basa en otros estándares existentes). Esto proporciona una serie de beneficios a la comunidad de desarrolladores de aplicaciones, incluyendo un modelo de programación familiar, una arquitectura probada y la influencia de herramientas existentes (por ejemplo: servidores Web, XML, herramientas, etc.). Se han realizado optimizaciones y extensiones con el fin de adecuarlo a las características propias del entorno inalámbrico.

- Un lenguaje de marcas proveniente de los estándares XML que fueron diseñados para realizar aplicaciones potentes sin las restricciones de los dispositivos móviles. El WML y WMLScript asumen que los datos de entrada no provienen de un teclado QWERTY o ratón, y que están

diseñados para pantallas pequeñas. En vez de la estructura plana de los documentos HTML, los documentos WML están divididos en un grupo de unidades de interacción con el usuario definidas. Una unidad de interacción se denomina *card* (carta), y los servicios se crean mediante la navegación entre *cards* (hacia delante y hacia atrás) de uno o más documentos WML. WML proporciona una pequeña (consecuente con las limitaciones de los dispositivos) grupo de marcas (*tags*) más apropiados que el HTML para implementar en terminales. Desde el *Gateway* WAP, todos los contenidos WAP se acceden desde Internet usando el estándar HTTP 1.1, por lo que los tradicionales servidores Web, herramientas y técnicas se pueden usar en este nuevo mercado (migración de contenidos).

- Una especificación para el micro-navegador en el terminal inalámbrico que controla el interfaz de usuario y es análogo al navegador Web estándar. Esta especificación define cómo WML y WMLScript debe ser interpretado en el dispositivo móvil y presentado al usuario. La especificación del micro-navegador ha sido diseñada para las características especificas de los terminales inalámbricos por lo que el código resultante es más compacto y eficiente. Un protocolo diseñado para minimizar los requerimientos de ancho de banda, garantizando que una amplia variedad de redes inalámbricas puedan usar y transmitir aplicaciones WAP.

- Un marco de trabajo para WTA (*Wireless Telephony Application*) que permite acceder a funcionalidades telefónicas como llamada de control, acceso a guía telefónica y mensajería mediante *applets* WMLScript. Esto permite a los operadores desarrollar aplicaciones de telefonía seguras integradas en los servicios WML y WMLScript. Por ejemplo,

servicios como *Call Forwardin*g proveen usuario de la capacidad de elegir entre aceptar una llamada, enviársela a otra persona y reenviarla mediante *voicemail*.

1.1.4 Objetivos del WAP

Podríamos enumerar los siguientes objetivos que persigue el estándar WAP:

- Entregar contenidos de Internet a cualquier tipo de terminales móviles, ya sea teléfonos, PDAs o cualquier otro dispositivo similar.

- Ofrecer servicios de valor añadido de todo tipo, desde banca móvil (*banking*) a información en tiempo real de los horarios de trenes. (ver sección soluciones profesionales).

- Conseguir una especificación de protocolos global que permita acceder y trabajar desde cualquier tipología de red, como se hace en Internet, donde da igual que el terminal esté conectado a través de RDSI, ATM, vía módem, etc.

- Permitir la escalabilidad de las distintas aplicaciones para las diversas opciones de transporte y tipos de dispositivos.

- Definir los protocolos, no los productos que los implementan, abriendo un amplio abanico de posibles soluciones.

- Conseguir que algunas de las funcionalidades de un ordenador puedan hacerse mediante un terminal móvil, añadiendo las ventajas que este presenta frente al ordenador.

1.1.5 Versiones del estándar WAP

- **v 1.0** (Noviembre de 1998).

  Sin implementación comercial. Los terminales correspondientes a ese periodo se conocen como terminales Pre-WAP.

- **v 1.1.** (Junio de 1999)
  - Algunos problemas de seguridad
  - Interoperabilidad y conformidad de los terminales
  - Plataformas mal definida
  - Modificaciones del WML para compatibilizarlo con XHTML

- **v 1.2.** (Diciembre de 1999)
  - Disponible comercialmente mediados 2.000.
  - Seguridad mejorada con el incremento de potencia de WTLS.
  - Mejoras WTA.
  - Servicios Push.
  - Test de interoperabilidad móviles/plataformas.
  - Navegación Proxy - Corporate Proxy
  - Incluye soporte para redes con nuevas tecnologías de transporte

1.2 Antecedentes del protocolo WAP

Aunque existe una patente sobre tecnología inalámbrica de *Geoworks Corp*. de Alameda, California en 1994, se considera que la primera compañía

en iniciar un proyecto con el propósito de crear un protocolo genérico fue Ericsson en 1995. Este protocolo, denominado ITTP (*Intelligent Terminal Transfer Protocol*), pretendía ofrecer nuevos servicios mediante telefonía móvil estableciendo comunicaciones entre un nodo (donde resida la aplicación) y un teléfono móvil inteligente. Este protocolo no tuvo éxito.

Posteriormente Nokia y Unwired Planet (que luego paso a ser Phone.com) empezaron a crear sus propios productos con sus propios protocolos con la intención de que alguno de ellos se estandarizase y obtener ventajas competitivas. La propuesta de *Unwired Planet* fue el HDML (*Handheld Device Markup Language*) y el HDTP (*Handheld Device Transport Protocol*), con una estructura bastante similar al HTML adaptado a pequeñas pantallas y capacidad de comunicación mediante radiofrecuencia. Por su parte, Nokia presentó el *Smart Messaging* que procuraba una conexión entre Internet y dispositivos GSM mediante SMS (*Short Message Service*) y el lenguaje TTML (*Tagged Text Markup Language*).

La cantidad de diferentes protocolos en el mercado hacía peligrar la estandarización de uno de ellos, por lo que finalmente algunas de las principales compañías se unieron para crear una organización que se denominó Foro WAP (*Wireless Application Protocol Forum*), el objetivo de este Foro era proponer unas especificaciones comunes que se convirtieran en un estándar "de facto" para asegurar el crecimiento de la telefonía móvil y compatibilidad de todos los componentes WAP independientemente de la compañía que lo haya desarrollado. En abril de 1998 apareció la primera versión (WAP 1.0). Este protocolo ya ha tenido dos actualizaciones (WAP 1.1 y WAP 1.2).

1.3 Servicios ofrecidos

Los servicios que ofrece la tecnología WAP son similares a los que ofrece Internet, pero actualmente con serias limitaciones. Estos servicios pueden ser: navegar, realizar transacciones bancarias, comprar o reservar entradas para espectáculos, reservar billetes de avión o tren, consultar el tiempo y las noticias o mandar un *eMail*.

Además WAP habilita un nuevo tipo de servicios que no se encuentran en Internet, los servicios telefónicos. Algunos de estos ya están incorporados desde hace tiempo en los teléfonos móviles convencionales. Estos servicios dependen de las operadoras que pueden ofrecerlos gratuitamente como estrategia de marketing o bajo determinadas condiciones de contrato.

Entre los principales servicios cabe destacar la administración de llamadas (selección de llamadas, llamada en espera, etc.), buzón de voz y mensajería unificada (e-mail, faxes, mensajes de voz, etc.), servicios SMS, agenda y servicios de atención al cliente.

El diseño de avanzados micro-navegadores, menús lógicos y pantallas de mayor tamaño, seguramente permitirán un manejo ágil de la Web a través del móvil. De momento surgen de manera constante páginas y sitios Web con contenidos y servicios específicos para los usuarios de estos pequeños teléfonos. También cabe destacar el aumento de las alianzas entre compañías de distintos ámbitos (bancos, cadenas de televisión, agencias de viajes, etc.) con empresas de telecomunicaciones para mostrar sus contenidos en este nuevo mercado que está surgiendo. Esto permite augurar que en poco tiempo la limitación del uso de estos servicios no será debida a la falta de contenidos, sino a carencias técnicas, de manejabilidad o de precio.

Los grandes beneficiarios de esta nueva tecnología en principio no van a ser los usuarios particulares, que pueden encontrar actualmente poco práctico en comparación con la versatilidad un PC, sino las empresas dado que se podrá acceder a información de *stocks*, realizar pedidos o acceder a la Intranet desde cualquier sitio del mundo y en cualquier circunstancia.

1.4 Ventajas e inconvenientes del WAP

1.4.1 Ventajas de la tecnología WAP

A continuación se enumeran algunas de las ventajas que propone esta tecnología. En siguientes apartados se analizaran algunas de ellas con más detalle.

- Portabilidad: acceso a Internet desde cualquier lugar
- Arquitectura cliente/servidor: dispositivos con micro-navegador y almacenamiento de servicios/aplicaciones temporalmente.
- Soporte HTTP 1.1 basado en WML (migración desde HTML)
- Mayor nivel de seguridad en las transacciones.
- Dispositivo como "*mobile wallet*" o cartera móvil en comercio electrónico
- Unificación de mensajería y contenidos en servicios Internet.
- Soporte para la tercera generación de telefonía celular (aplicaciones multimedia y acceso a alta velocidad)

1.4.2 Carencias actuales en el mercado WAP

Los principales inconvenientes que surgen de las características en las que se encuentra actualmente el mercado. Aunque también hay quien pone

en duda la propia base donde se sustenta el WAP. Cabe destacar los siguientes:

- Disponibilidad y coste de dispositivos adaptados en el mercado.

- Limitaciones físicas de los terminales presentados (manejabilidad y usabilidad principalmente). Esto es debido a que las pantallas son pequeñas, monocromas y las posibilidades de navegación escasas. También son importantes otros factores como la CPU, la memoria y las baterías.

- Disponibilidad de contenidos Internet con versión WAP: portales, proveedores y empresas de servicios aún muy limitadas en este protocolo.

- Especificaciones incompletas: soporte a tecnología *Push* o aplicaciones de telefonía inalámbricas no definidas totalmente.

- El coste de las llamadas podría resultar un problema, dependiendo de la compañía que preste el servicio de telefonía celular y el tipo de suscripción a la misma.

- Velocidad de transferencia baja (pico máximo de 9 Kbit/s) y no garantizada.

- Alta latencia. La latencia consiste en largos períodos de tiempo (comparando con el tiempo útil de transmisión de datos) en la que la sesión abierta está en estado latente, es decir, sin recibir datos.

1.5 El entorno de aplicaciones

1.5.1 El entorno de aplicaciones WAE

1.5.1.1 El Micro-*Browser*

Se trata de un elemento residente en el terminal cuya función principal es la de traducir aquello que recibe y adaptarlo a las características del aparato

- Al igual que un navegador HTML, es quien se encargar de transmitir las peticiones y recibir las respuestas en el terminal móvil

- Incorpora además interpretes de WML y WMLScript. Se encarga de extraer la información que le llega código binario y presentarlo lo mejor posible en la pantalla del terminal
- Conoce el uso de las URL ya que es este mecanismo el que permite realizar las peticiones. Para ello, se comunica con las capas inferiores de la torre de protocolos. Puede entonces disparar peticiones, establecer comunicaciones seguras.

- Contiene una Caché, del mismo modo que la tienen los navegadores Web convencionales. Esta caché le permitirá asegurar una navegación más rápida, almacenando el contenido de cartas en memoria para no tener que volver a bajarlas así como almacenar variables, que en WML pueden tener un tiempo de vida relativamente largo.

- Una pila. Es una porción de memoria en la que almacena por ejemplo las URLs más recientemente visitadas (el Histórico), el contexto de cada página que permiirá volver a publicarla en el display con los datos de la *caché*.

- Necesita conocer algo el protocolo HTTP 1.1 con el fin de incorporar a sus peticiones los encabezados adecuados, si bien es papel del *Gateway* realizar la conversión de protocolos.

- Tiene que ser capaz de realizar las funciones que le son atribuidas con las limitaciones de RAM, ROM, pantalla, capacidad de entrada de datos, procesado.

1.5.2 WML

Este lenguaje se basa en Etiquetas (Tags) y se diseñó para trabajar sobre un Hardware con serias limitaciones en sus prestaciones, un ancho de banda pequeño y terminales móviles con escasa capacidad de entrada y salida.

Por otra parte se estructura los datos en cartas, que corresponden en principio a lo que se presenta en pantalla, agrupadas en barajas. Una carta contiene contenidos visibles y a lo sumo una lista de opciones, un formulario o la navegación a otra carta. La idea es que el terminal en sí sea transparente a WML, para darles más posibilidades de desarrollo a los fabricantes.

WML debe asegurar las siguientes prestaciones:

- Texto e imagines

- Entrada de datos, como texto, listas de opciones, códigos, comandos del tipo Get, Submit.

- Navegación, para poder moverse dentro de las cartas de una baraja oentre barajas. Cada terminal incorporará un Histórico que permita una navegación suplementaria mediante un botón Atrás por ejemplo

- Uso de varios idiomas

- Gestión de Estado y Contexto, que permita pasar de una baraja a otra las variables más importantes, sustituir variables y gestión de la Caché

1.5.3 *WMLScript*

Es un lenguaje de *scripts* ligero pero extensible, que podría realizar funciones similares a *JavaScript* en HTML. WML maneja entradas y salidas, entregas de contenidos y eventos pero, todo ello, sin capacidad seria de procesado. *WMLScript* pretende compensar esa deficiencia.

WMLScript aporta la potencia del uso de expresión de lenguajes de alto nivel, como If...Then, For..., asignaciones, llamadas a funciones, el uso del tipo Booleano y enteros además de operadores lógicos, aritméticos, etc. Por otra parte también puede recurrir a librerías *WMLScript* 1.1 pudiendo entonces utilizar puntos flotantes, *strings*, *URLs*, librerías de diálogo con el *microbrowser*, librerías de lenguajes, etc.

1.5.4 La interfaz de aplicación telefónica móvil (WTAI):

Los terminales que soportan WAP son en la mayoría de las ocasiones teléfonos móviles. Puede pues resultar interesante añadirles características y funciones. Este es el papel de la WTAI que define las API adecuadas. Se accede a estas API a través de WML y WMLScript.

Es, a efectos prácticos, mucho más cómodo, para el operador, crear las API y dejarlas residentes en el terminal antes que escribir aplicaciones específicas para terminal y meterlas en la ROM. Este sistema permite actualizar las versiones de las aplicaciones residentes prestando así los mejores servicios posibles. Además estas versiones pueden ser genéricas para todos los terminales. Se basa la programación de dichas aplicaciones WTAI en modelos dirigidos por eventos asíncronos (señales), lo que se asemeja mucho más al entorno de telefonía. Un ejemplo de evento asíncrono puede ser una sencilla llamada recibida por el terminal. Para que u terminal soporte esta mejora debe tener la posibilidad, mediante el operador, de acceder al servidor WTA con una conexión directa sobre la red GSM, es decir, sin pasar por la Web, sin pasar por el *proxy*.

Descripción de los servicios, arquitectura y tendencias del protocolo WAP para el desarrollo de aplicaciones para redes inalámbricas.

2. INTRODUCCIÓN AL ESTUDIO DE REDES INALAMBRICAS (IEEE 802.11) Y TECNOLOGÍAS DE TELEFONÍA MÓVIL

Los dispositivos móviles, tales como PDA (*Personal Digital Assistants*) y agendas, han tenido un crecimiento muy rápido dentro de la industria de los dispositivos. Su uso es amplio. El más común es poder tener una oficina portátil: poder recibir y enviar llamadas telefónicas, fax, correo electrónico, leer ficheros remotos y acceder a máquinas remotas; y todo ello desde tierra, mar o aire.

Aunque las redes inalámbricas y dispositivos portátiles van muchas veces relacionados, no son lo mismo. Los dispositivos portátiles a veces son cableados y hay redes inalámbricas que no son portátiles. Un ejemplo de dispositivo portátil y no inalámbrico es aquel que se puede llevar fácilmente en un viaje pero debe ser conectado a una red para poder utilizarlo. Por otra parte, una red inalámbrica no móvil es aquella que comunica dos LANs de dos edificios mediante emisores/receptores de láser situados en sus tejados.

Pero por supuesto hay aplicaciones móviles e inalámbricas, por ejemplo equipos portátiles para gente que trabaja vendiendo en la calle en torno a un almacén con un PDA utilizando inventarios.

Las redes inalámbricas tienen distintas formas. Por ejemplo, una universidad puede tener una antena que permita a los estudiantes acceder a los catálogos de libros mientras están bajo un árbol del campus. Estos dispositivos se comunican directamente con las LANs cableadas en forma

digital. Otra posibilidad es tener un servicio celular digital llamado CDPD (*Celular Digital Packet Data*) que está disponible en muchas ciudades.

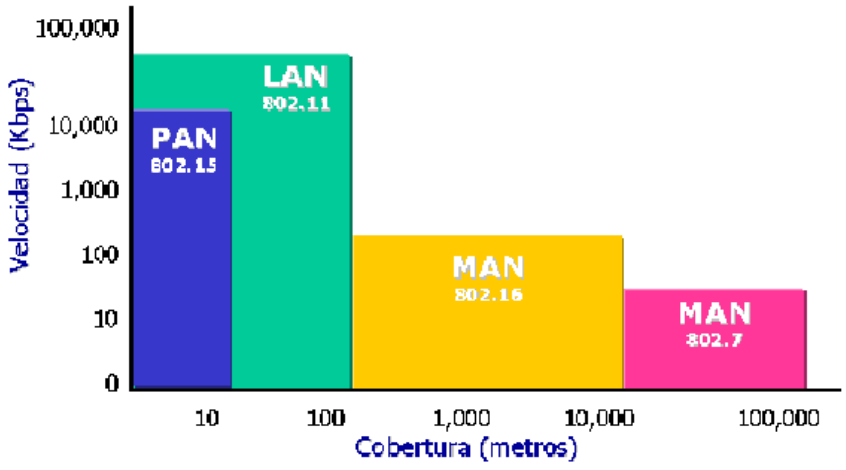
Por último mencionar que es posible tener combinadas redes inalámbricas con redes cableadas, imaginemos una LAN cableada en un avión que se comunique inalámbricamente con tierra.

2.1 Clasificación de las redes inalámbricas

Al igual que las redes tradicionales cableadas vamos a clasificar a las redes inalámbricas en tres categorías:

- WAN/MAN (*Wide Area Network/Metropolitan Area Network*)
- LAN (*Local Area Network*)
- PAN[7] (*Personal Area Network*)

Figura 4: Comparativa Distancia/Velocidad de tipos de redes



[7] El concepto de red inalámbrica de área personal o WPAN (Wireless Personal Area Network) se refiere a una red sin cables que se extiende a un espacio de funcionamiento personal o POS (Personal Operating Space) con un radio de 10 metros.

Como podemos ver en la Figura 4, en la primer categoría WAN/MAN, pondremos a las redes que cubren desde decenas hasta miles de kilómetros. En la segunda categoría LAN, pondremos las redes que comprenden de varios metros hasta decenas de metros. Y en la última y nueva categoría PAN, pondremos a las redes que comprenden desde 1 metro hasta 30 metros.

La norma IEEE 802.11 estableció el estándar para redes inalámbricas. Una red de área local inalámbrica puede definirse como a una red de alcance local[8] que tiene como medio de transmisión el aire. Siendo su finalización definitiva para la introducción y desarrollo de los sistemas WLAN en el mercado. El estándar 802.11 es muy similar al 802.3 (*Ethernet*) con la diferencia que tiene que adaptar todos sus métodos al medio *no guiado* de transmisión. En este estándar se encuentran las especificaciones tanto físicas como a nivel MAC.

2.1.1 Conceptos generales de redes inalámbricas

Una red de área local o WLAN (*Wireless* LAN) utiliza ondas electromagnéticas (radio e infrarrojo) para enlazar mediante un adaptador los equipos conectados a la red, en lugar de los cables coaxiales o de fibra óptica que se utilizan en las LAN convencionales cableadas. En la figura 5 se puede ver un ejemplo de una red inalámbrica sencilla.

[8] Red que cubre un entorno geográfico limitado, con velocidad de transferencia mayor o igual a 1 Mbps

Figura 5: Ejemplo de una red inalámbrica sencilla

Las redes locales inalámbricas mas que una sustitución de las LANs convencionales son una extensión de las mismas, ya que permite el intercambio de información entre los distintos medios en una forma transparente al usuario. En este sentido el objetivo fundamental de las redes WLAN es el de proporcionar las facilidades no disponibles en los sistemas cableados y formar una red total donde coexistan los dos tipos de sistemas. Enlazando los diferentes equipos ó terminales móviles asociados a la red.

Este hecho proporciona al usuario una gran movilidad sin perder conectividad. El atractivo fundamental de este tipo de redes es la facilidad de instalación y el ahorro que supone la supresión del medio de transmisión cableado. Aún así sus prestaciones son menores en lo referente a la velocidad de transmisión que se sitúa entre los 2 y los 10 Mbps. frente a los 10 y hasta los 100 Mbps. ofrecidos por una red convencional.

Las redes inalámbricas son la alternativa ideal para hacer llegar una red tradicional a lugares donde el cableado no lo permite. En general las WLAN son se utilizarán como complemento de las redes fijas.

2.1.2 Ámbito de aplicación

Las aplicaciones más típicas de las redes de área local que podemos encontrar actualmente son las siguientes:

- Implementación de redes de área local en edificios históricos, de difícil acceso y en general en entornos donde la solución cableada es inviable.

- Posibilidad de reconfiguración de la topología de la red sin añadir costes adicionales. Esta solución es muy típica en entornos cambiantes que necesitan una estructura de red flexible que se adapte a estos cambios.

- Redes locales para situaciones de emergencia o congestión de la red cableada.

- Estas redes permiten el acceso a la información mientras el usuario se encuentra en movimiento. Habitualmente esta solución es requerida en hospitales, fábricas, almacenes.

- Generación de grupos de trabajo eventuales y reuniones ad-hoc. En estos casos no valdría la pena instalar una red cableada. Con la solución inalámbrica es viable implementar una red de área local aunque sea para un plazo corto de tiempo.

- En ambientes industriales con severas condiciones ambientales este tipo de redes sirve para interconectar diferentes dispositivos y máquinas.
- Interconexión de redes de área local que se encuentran en lugares físicos distintos. Por ejemplo, se puede utilizar una red de área local inalámbrica para interconectar dos o más redes de área local cableadas situadas en dos edificios distintos.

2.2 Topologías y configuraciones

La versatilidad y flexibilidad de las redes inalámbricas es el motivo por el cual la complejidad de una LAN implementada con esta tecnología sea tremendamente variable. Esta gran variedad de configuraciones ayuda a que este tipo de redes se adapte a casi cualquier necesidad.

Estas configuraciones se pueden dividir en dos grandes grupos, las *redes peer to peer* y las que utilizan *Puntos de Acceso*.

2.2.1 *Peer to Peer*

También conocidas como redes *ad-hoc*, es la configuración más sencilla, ya que en ella los únicos elementos necesarios son terminales móviles equipados con los correspondientes adaptadores para comunicaciones inalámbricas.

En este tipo de redes, el único requisito deriva del rango de cobertura de la señal, ya que es necesario que los terminales móviles estén dentro de este rango para que la comunicación sea posible. Por otro lado, estas

configuraciones son muy sencillas de implementar y no es necesario ningún tipo de gestión administrativa de la red. Un ejemplo sencillo de esta configuración se muestra en la figura 6.

Figura 6: Conexión *Peer to Peer*

2.2.2 Punto de acceso

Estas configuraciones utilizan el concepto de *celda*, ya utilizado en otras comunicaciones inalámbricas, como la telefonía móvil. Una *celda* podría entenderse como el área en el que una señal radioeléctrica es efectiva. A pesar de que en el caso de las redes inalámbricas esta celda suele tener un tamaño reducido, mediante el uso de varias fuentes de emisión es posible combinar las celdas de estas señales para cubrir de forma casi total un área más extensa.

La estrategia empleada para aumentar el número de celdas, y por lo tanto el área cubierta por la red, es la utilización de los llamados *Puntos de acceso*, que funcionan como repetidores, y por tanto son capaces de doblar el alcance de una red inalámbrica, ya que ahora la distancia máxima permitida no es entre estaciones, sino entre una estación y un punto de acceso.

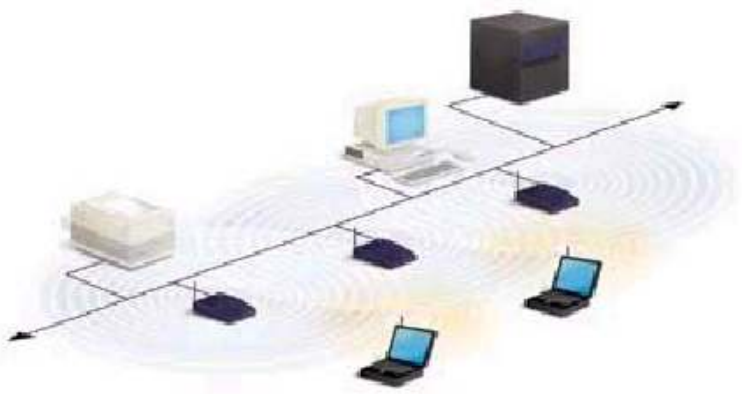
Los *Puntos de acceso* son colocados normalmente en alto, pero solo es necesario que estén situados estratégicamente para que dispongan de la cobertura necesaria para dar servicio a los terminales que soportan.

Un único punto de acceso (como en la figura 7) puede soportar un pequeño grupo de usuarios y puede funcionar en un rango de al menos treinta metros y hasta varios cientos de metros.

Figura 7: Utilización de un *Punto de Acceso*

La técnica de *Punto de acceso* es capaz de dotar a una red inalámbrica de muchas más posibilidades. Además del evidente aumento del alcance de la red, ya que la utilización de varios puntos de acceso, y por lo tanto del empleo de varias celdas que colapsen el lugar donde se encuentre la red, permite lo que se conoce como *roaming*, es decir que los terminales puedan moverse sin perder la cobertura y sin sufrir cortes en la comunicación. Esto representa una de las características más interesantes de las redes inalámbricas. En la figura 8 se puede ver la un ejemplo de una red con varios Puntos de *Acceo y terminales* con capacidades de *roaming*.

Figura 8: Utilización de varios Punto de Acceso. Terminales con capacidades de *Roaming*



2.2.3 Otras configuraciones: Interconexión de redes

Las posibilidades de las redes inalámbricas pueden verse ampliadas gracias a la interconexión con otras redes, sobre todo con redes no inalámbricas. De esta forma los recursos disponibles en ambas redes se amplían.

Mediante el uso de antenas (direccionales o omnidireccionales) es posible conectar dos redes separadas por varios cientos de metros, como por ejemplo dos redes locales situadas en dos edificios distintos. De esta forma, una LAN no inalámbrica se beneficia de la tecnología inalámbrica para realizar interconexiones con otras redes, que de otra forma serian más

costosas, o simplemente imposibles. En la figura 9 se puede ver un ejemplo de interconexión de redes mediante antenas direccionales.

Figura 9: Interconexión de LAN mediante antenas direccionales

2.3 Sistema de Radiotelefonía celular pública

2.3.1 Fundamentos del servicio

La finalidad de este servicio, conocido como TMA (Telefonía Móvil Automática) celular o simplemente TMA, es la de proporcionar al usuario un servicio telefónico público móvil. La telefonía móvil permite mantener comunicación telefónica desde equipos móviles de la misma forma que si utilizaran un teléfono fijo convencional. Un usuario móvil puede efectuar y recibir llamadas telefónicas automáticas con cualquier otro abonado fijo o móvil de la red telefónica.

El usuario de Telefonía móvil puede realizar llamadas nacionales e internacionales en sus desplazamientos, manteniendo en su zona de cobertura la disponibilidad telefónica de su domicilio.

TMA maneja un gran número de abonados móviles dispersos por una amplia zona con explotación automática. Esto supone resolver una serie de aspectos:

- Conmutación automática de la comunicación y su continuidad.
- Radiobúsqueda de un móvil, que debe preceder a toda comunicación.
- Consecución de un nivel de calidad de la conmutación con la selección automática de estaciones para mantener esa calidad en el curso de la conversación.

En los sistemas TMA se necesita conseguir una amplia cobertura con gran capacidad de tráfico y con un número limitado de frecuencias. Esto se consigue gracias a la reutilización sistemática de las frecuencias, lo que se logra mediante estructuras celulares.

2.3.2 Elementos del Servicio de TMA

Los elementos básicos del sistema TMA son las Estaciones Base, las Centrales de Conmutación, zona de cobertura y las estaciones móviles.

2.3.2.1 Estaciones base:

Las estaciones base son los equipos que establecen el contacto con los teléfonos móviles del cliente y por tanto determina la cobertura del servicio. Consiste en un ordenador y un transmisor/receptor conectado a una antena.

Existe una amplia red de Estaciones Base las cuales están conectadas a centrales de conmutación específicas para la telefónica móvil (MTSO, *Mobil Telephonic Switch Office* o MSC, *Mobil Swith Center*).

2.3.2.2 Centrales de conmutación para telefonía móvil

Dan servicio a las estaciones Base y a su vez se conectan con las Centrales de la Red Telefónica Fija, para poder establecer conversaciones tanto entre teléfonos móviles como entre teléfonos móviles y fijos. En grandes sistemas son necesarios múltiples MTSO conmutándose éstos en un segundo nivel de MTSO y así sucesivamente.

2.3.2.3 Zona de cobertura

La zona de cobertura del servicio contempla la totalidad del territorio nacional, especialmente las áreas urbanas y vías de comunicación más importantes. La superficie total a la que se extiende el servicio es dividida en subáreas o celdas atendidas por una estación base.

2.3.2.4 Estación móvil

Es el terminal telefónico móvil. El propio teléfono móvil indicará al usuario cuando se encuentra dentro de la zona de cobertura.

Las personas que quieran establecer una comunicación con un usuario del servicio de telefonía móvil no necesitan saber dónde se encuentra éste ya que el propio sistema se encarga automáticamente de localizarle para establecer la comunicación.

2.3.3 Estructura celular de las horas TDMA

La comunicación base-móvil o móvil-móvil en una frecuencia específica sólo es posible si no se supera una distancia entre ellos denominada radio de cobertura, cuyo valor es proporcional a la altura de las antenas de la estación móvil y la estación base. Superada esta distancia la atenuación es tan elevada que no es posible la comunicación.

Los sistemas celulares se basan en subdividir la superficie total a cubrir en zonas más pequeñas llamadas celdas o células**,** a las que se asigna una estación base con un cierto número de frecuencias o canales.

Como el espectro radioeléctrico y el número de canales o comunicaciones posibles al mismo tiempo son limitados, se puede dividir la superficie total a cubrir en celdas de modo que las frecuencias que se usan en una celda puedan ser reutilizadas en otra celda lejana. Separando adecuadamente las celdas a una distancia (llamada distancia co-canal o de reutilización) determinada por la relación de protección de RF, puede reutilizarse el mismo juego o conjunto de frecuencias en diversas celdas. Reutilizando frecuencias, un sistema celular puede cursar un tráfico superior al número de frecuencias asignadas a la banda.

En la práctica, el número total de canales disponibles “C” se divide entre las celdas de una configuración unitaria básica denominada grupo básico. Un tamaño típico para el grupo es siete si las estaciones base utilizan antenas omnidireccionales, nombrándose las celdas, de la A a la G.

2.3.4 Forma geométrica de las celdas

La forma geométrica más conveniente para las celdas a de estar de acuerdo con los siguientes criterios:

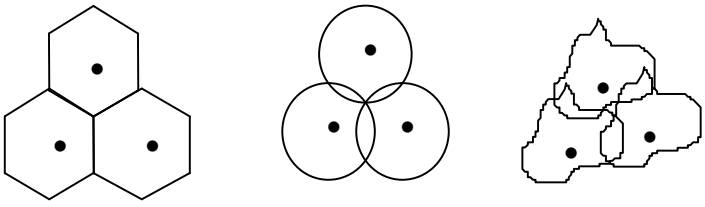
- Se debe procurar que no existan huecos o solapamientos en los bordes.

- Se debe buscar la forma que para un radio dado R se obtenga la mayor superficie posible. De esta forma se utilizarán un menor número de celdas para servir la misma zona de cobertura y, por tanto, utilizarán menor número de frecuencias.

Tomando en cuenta los criterios anteriores y analizando la figura 10, se encuentra que el primer criterio impide elegir el círculo. Las alternativas posibles son: el triángulo, el cuadrado y el hexágono. el hexágono es mejor al aplicar el segundo criterio.

En realidad las celdas no son hexagonales, sino que tienen una forma irregular determinada por parámetros como la propagación de las ondas de radio en el terreno, obstáculos y las restricciones de la estación base debidas a factores geográficos.

Figura 10: Relación entre las coberturas ideales y reales



FICTICIA IDEAL REAL

2.3.5 *Handover* entre las celdas

En todo momento, un teléfono móvil está situado en una celda determinada y bajo el control de la estación base de dicha celda. Cuando un móvil deja una celda, su estación base detecta que la señal del teléfono se apaga y pregunta a todas las estaciones adyacentes qué potencia tienen de ella.

La estación base entonces transfiere su propiedad a la celda que obtuviera la mayor señal, esto es, a la nueva celda donde se localice el teléfono. El teléfono es informado y si hubiera una llamada a medio, se cuestionaría el cambiar al nuevo canal (pues el viejo puede estar usándose por celdas adyacentes). Este proceso es llamado *handoff o handover* y suele hacerse en 300 msg. La asignación de canales se lleva a cabo por el MTSO.

Cada sistema tiene una solución para llevar a cabo este proceso, generalmente mediante mensajes de control (señalización) que se intercambian los terminales móviles y la estación de control. Pero lo que sí es importante señalar aquí, es el hecho de que una de las medidas de calidad de un sistema de TMA celular sea la probabilidad de pérdida de una llamada cuando se cruza una celda.

2.3.6 Técnicas para aumentar la capacidad de los sistemas celulares

Cuando un sistema celular da servicio a un área urbana pueden llegar a darse situaciones de saturación.

Existen básicamente dos técnicas:

- La subdivisión de las celdas en otras más pequeñas.
- la sectorización.

La subdivisión de una celda suele hacerse reduciendo a la mitad el radio de la celda. Esto implica:

- Reducir por cuatro la superficie.
- Aumentar la capacidad de tráfico por un factor aproximadamente igual a cuatro.
- Aumentar el número de estaciones base y emplazarlas de forma más precisa.
- Aumento en el tráfico de señalización al aumentar el número de *handovers*.

El proceso de subdivisión tiene un límite fijado por las tolerancias de los emplazamientos y la complejidad y la carga del procesamiento de llamadas, que suele corresponder a un radio de 1,5 km. No obstante los

nuevos sistemas de TMA celular digital contemplan celdas de unos 0,3 km de radio.

Se puede proceder a una subdivisión adicional, sin necesidad de emplear más estaciones base, sectorizando la cobertura. Para ello se subdivide una celda en tres sectores a los que se da servicio desde vértices alternos del hexágono, mediante tres estaciones base con haces de antena de 120º. Con ello se pueden cubrir sectores de celdas vecinas, lo cual supone un ahorro de estaciones base. En la práctica, para realizar una sectorización no es necesario crear nuevos emplazamientos sino sólo transformar los ya existentes .

2.3.7 Asignación de frecuencias entre celdas

Podríamos suponer que a cada celda se le asigna de forma fija un número de canales. Pero si en una celda existe congestión (todos sus canales están ocupados) y en otra contigua hay canales libres, podríamos pensar en tomar prestados algunos de estos canales sólo durante el período de congestión.

Así pues, el principio general de la asignación dinámica es que cualquier canal puede ser utilizado en cualquier celda. El análisis del tráfico es por ello bastante complejo, por lo que se suele recurrir a la simulación por ordenador para el estudio y dimensionamiento de estos sistemas.

2.4.8 Funcionamiento de un sistema celular típico

Las estaciones base (*Base Station*, BS) están conectadas a los centros de Conmutación del Servicio Móvil (*Mobile Switching Centre,* MSC), que son centrales de conmutación especializadas para ejecutar las funciones necesarias para el funcionamiento del sistema. La conexión BS-MSC se realiza mediante enlaces dedicados.

Las comunicaciones en los SRTM son *full-duplex* por lo que se requieren dos frecuencias diferentes para cada conexión, una en el sentido móvil-base y otra en el sentido contrario. Además a cada BS se le asigna un canal de señalización y control para tareas tales como el establecimiento de la conexión.

Los abonados deben estar localizados en todo momento para poder dirigirles las llamadas que se produzcan. Para ello, todo MSC dispone de dos tipos de bases de datos: el HLR (*Home Location Register*) donde se inscriben los abonados locales y el VLR (*Visitors Location Register*) donde se inscriben los abonados que están de paso.

Cuando el abonado conecta su equipo, éste explora los canales de control de la BS y se sintoniza en aquél en el que reciba mayor señal, retornando su identificación. Si está en su MSC local se inscribe en la HLR, de lo contrario se inscribe en la VLR y se notifica a la HLR de su MSC. De esta forma, cuando llegue una llamada a su MSC, éste, tras consultar el HLR, podrá redirigirla al MSC en cuyo VLR esté inscrito el abonado. A esta facilidad de conexión del móvil donde quiera que esté se denomina roaming (vagabundeo). El MSC convierte el número del abonado destino en el código de identificación del abonado y difunde un mensaje de búsqueda (*paging message*) en las BSs que dependen de la MSC de paso en la que se encuentre el abonado.

2.4.9 Señalización

Cuando un usuario móvil realiza una llamada, el contacto inicial se realiza a través del canal de control. La señalización se realiza intercambiando paquetes de datos (modulados en FSK) que ocupan la totalidad del canal de control. La BS asignará un canal de voz (una pareja de frecuencias) a la nueva conversación y ambos, BS y terminal móvil, conmutarán al canal de voz mientras dure la conversación.

Para realizar tareas de supervisión, se envían dos tonos dentro del canal vocal pero fuera de banda. El primero es el SAT (*Supervisory Audio Tone*), un tono que es enviado por la BS y que debe ser devuelto por el terminal móvil mientras la conversación está en curso. Su pérdida le indica a la BS que la señal es muy débil, bien por acercarse a la frontera de la celda (por tanto se deberá proceder a un *handover*), o bien por otras razones (*fading*, desconexión, abrupta, etc.). El segundo es el ST (*Signaling Tone*) que se usa, por ejemplo, al final de una conversación para indicar que el terminal ha sido colgado.

Cuando se pierde el SAT puede desencadenarse el proceso de *handover*. Para ello la MSC pide a las BS adyacentes que monitoricen el nivel de la señal del canal de voz correspondiente, asumiendo que el móvil ha entrado en la zona de cobertura de aquella BS cuya señal se reciba con mayor amplitud. Si la conexión es posible en la nueva BS (quedan canales sin usar) la MSC, a través de la BS, le indicará al móvil la nueva frecuencia a utilizar. Para ello, se interrumpe la señal de voz por un momento (unos 400 ms) y se le envía un mensaje de señalización al móvil. Esta interrupción es apenas distinguible durante una conversación, pero si se están transmitiendo datos se producirán una pérdida de información. Esta es una de las razones que ha llevado a la introducción de sistemas más avanzados como pueda ser GSM.

2.4 Sistemas de TMA celular digital

Las ventajas de un sistema digital son:

- Se posibilita una interconexión con la RDSI debido a la naturaleza digital de la información que maneja el sistema (voz o datos).

- Un sistema digital permite la implantación de protocolos señalización rápidos, potentes seguros y flexibles que permiten amplia gama de servicios suplementarios y protección contra el fraudulencias en la red.

- Pueden usarse técnicas de cifrado digitales aseguran la confidencialidad.

- Mejora de la calidad de las comunicaciones al incorporarse modulaciones digitales que permiten trabajar con mayores relaciones señal-ruido, códigos de detección y corrección de errores, técnicas de ecualización potentes, etc. Todo ello permite reducir la distancia de reutilización y por tanto conseguir una mayor densidad de tráfico con el mismo ancho de banda.

- La tecnología digital permite el uso de técnicas de acceso múltiple división en el tiempo (TDMA). Es decir, que un mismo radiocanal es utilizado por distintos usuarios en intervalos de tiempo distinto y solapados. Esto permite aumentar todavía más la eficiencia espectral.

2.5 Acceso múltiple TDMA

En cualquier sistema de comunicación, la técnica de acceso múltiple es la metodología que regula la disponibilidad por parte de los usuarios de los recursos y facilidades de la red y establece las directrices para el establecimiento de protocolos de acceso específicos.

En las radiocomunicaciones móviles, el recurso primario es el espectro radioeléctrico. De entre las técnicas de multiacceso, en los sistemas móviles digitales se utilizan el acceso múltiple por división en el tiempo, TDMA, y el acceso múltiple por división de código, CDMA.

En los sistemas TDMA normalizados hasta ahora, la banda total disponible se divide en sub-bandas. A cada sub-banda se le asigna una portadora a la cual se aplica TDMA, por lo que, de hecho, tales sistemas son TDMA/FDMA, ya que los TDMA parciales están multiplexados en frecuencia. Se les llama TDMA de banda estrecha.

En los sistemas TDMA se divide el tiempo en intervalos sustentados por una portadora. Cada usuario del TDMA sintonizado a esa portadora transmite y recibe la información en forma de ráfagas en los intervalos de tiempo que tiene asignados.

La técnica TDMA requiere una memoria intermedia en la que se van depositando los bits de la señal digital durante el tiempo que transcurre entre dos intervalos consecutivos de un mismo canal de usuario. Llegado el momento del acceso de este canal, la memoria se lee con rapidez para vaciarla en la duración del intervalo. Esta técnica de lectura/escritura a velocidades diferentes se le conoce también como “compresión temporal”.

En los sistemas TDMA se denomina trama a un ciclo completo de intervención de los diferentes canales, es decir, la trama es la sucesión ordenada y secuencial de intervalos.

En el estudio del TDMA deben distinguirse los dos sentidos de funcionamiento de Terminal (T) y Estación Base (BS). En el sentido ascendente T➔BS, la contienda entre los terminales para el acceso a la BS se resuelve mediante la asignación a cada uno del intervalo de tiempo en el que se efectuarán sus transmisiones. Hay una correspondencia uno a uno entre cada terminal y el intervalo de tiempo de la trama.

Sin embargo, en el sentido descendente, BS➔T, la estación base transmite los intervalos sucesivos destinados a los diferentes terminales durante toda la trama. Cada terminal recibe toda la trama, pero entresaca de la misma el intervalo asignado con la información que le está destinada.

Pueden utilizarse dos portadoras diferentes para los sentidos ascendente y descendente, aunque no es absolutamente necesario, dada la separación temporal de la transmisión y recepción.

2.6 Sistema de telefonía móvil digital GSM

En 1982, cuando aparecieron los primeros servicios celulares comerciales, la CEPT (*Conference Européen des Postes et Télécommunications*) tomó la iniciativa de poner en marcha un grupo de trabajo (llamado *Groupe Spécial Mobile*) encargada de especificar un sistema de comunicaciones móviles común para Europa en la banda de 900 Mhz.

Hoy en día el estándar GSM está funcionando con éxito tanto en países europeos como del resto del mundo. En 1993 habría más de 36 redes GSM en servicio en 22 países. Además más de 25 países no europeos o habían adoptado el estándar o estaban considerando su adopción.

Los servicios móviles pueden asociarse más fácilmente a un abonado que a un equipo o a una terminación de línea, proporcionando lo que se conoce como servicios de comunicación personal. Se espera que el desarrollo de redes de comunicación personal con acceso radio a la red fija afecte a una proporción significativa de abonados en los próximos diez años. De entre las posibles tecnologías que podrían ser utilizadas para soportar tales servicios se ha escogido la GSM, adaptada a la banda recientemente reservada de 1800 Mhz (1850-2200 Mhz), constituyendo el llamado DCS 1800 (*Digital Cellular System)***.** Se espera que el estándar GSM/DCS dé servicio a un número de abonados entre 20 y 40 millones en el año 2000.

2.6.1 Servicios y facilidades del Sistema GSM

En principio, todos los servicios disponibles en la Red Digital de Servicios Integrados (RDSI) han sido incluidos en el desarrollo del GSM. Pero debido a las restricciones de velocidad de transmisión de datos y tasa de errores, algunos de los servicios han sido desarrollados con restricciones.

2.6.1.1 Teleservicios

- La telefonía es el teleservicio más importante del sistema GSM. Permite llamadas entre la red pública (RTPC/RDSI) y la red móvil. Además existe el sistema GSM de llamadas de emergencia que permite una conexión directa y automática con el servicio de emergencia más próximo, marcando el 112.

- Soporta el servicio de fax del Grupo 3 si se dispone de los adaptadores de interfaz correspondientes.

- Ofrece un tipo de correo electrónico (E-mail) de mensajes cortos (140 bytes) que puede considerarse como un servicio de búsqueda (*paging*) alfanumérico y bidireccional. Se confirma la entrega de los mensajes, lo que constituye una ventaja importante sobre los sistemas de búsqueda. Y está disponible en modo punto a punto y difusión.

2.6.1.2 Servicios portadores

Para servicios de datos, soportan velocidades de transmisión que van de los 300 bits/s a los 9.6 Kbits/s.

2.6.1.3 Servicios suplementarios

Muchos de estos servicios son equivalentes a los disponibles en la RDSI. Los principales son:

- Llamada restringida.
- Desvío de llamadas.
- Identificación del abonado llamante.

2.6.1.4 Módulo de identidad de abonado

Un terminal GSM no tiene acceso a la red salvo si dispone de todos los datos específicos del abonado. Estos datos están incluidos en una tarjeta inteligente llamada SIM (*Subscriber Identy Module*) que debe introducirse en el terminal. La tarjeta SIM, cuyo acceso se protege con un número de identificación personal, contiene, no sólo los datos del abonado (número en la

RDSI, clave personal, etc.) sino también determinada información personal, como marcación abreviada de números, lista de redes preferentes e información de tarificación. En la tarjeta SIM también se almacenan los mensajes cortos.

2.6.1.5 Funciones de seguridad

En el sistema GSM la protección de la información se realiza a tres niveles:

- Autentificación por el sistema de las tarjetas SIM, para impedir el acceso a usuarios registrados.
- Cifrado de la transmisión para impedir escuchas no autorizadas (voz y datos).
- Protección de la identidad del abonado.

2.7 Asuntos de seguridad

Los teléfonos celulares son totalmente inseguros. Cualquiera que tenga un receptor de radio a toda banda puede sintonizar y oír cualquier cosa emitida en una celda. como la mayoría de los usuarios no se da cuenta de lo inseguro que es este sistema, a menudo dan números de tarjetas de crédito y otras informaciones confidenciales.

Otro problema es el robo del "tiempo de aire". Con un receptor a toda banda añadida al ordenador, un ladrón puede monitorizar el canal de control y registrar los 32 bits serie de un número y los 34 de un número de teléfono de todos los teléfonos móviles que oiga.

Solamente con conducir alrededor de un par de horas por la ciudad puede construirse una gran base de datos. El ladrón puede entonces escoger un número y usarlo para sus llamadas. Esta trampa podrá usarse hasta que la víctima reciba un recibo semanas más tarde.

Algunos ladrones ofrecen un servicio de telefonía a bajo costo al hacer llamadas por sus clientes usando los números robados. Otros reprograman teléfonos con los números robados y los venden como teléfonos en los que puede llamarse libremente.

Algunos de estos problemas podrían resolverse por encriptación, pero entonces la policía no podría realizar "*wiretaps*" con criminales inalámbricos.

Otro asunto en el área general de la seguridad es el vandalismo y daño a antenas y estaciones base. Todos estos problemas son graves y cuestan cientos de millones de dólares al año en pérdidas a la industria celular.

3. ARQUITECTURA DE UNA RED WAP

3.1 El modelo WWW

El modelo que sigue la Red WAP es el de la *World Wide Web*. Resulta ser una arquitectura muy flexible y potente. Se basa en el principio de Cliente–Servidor.

Las aplicaciones y contenidos se presentan en formatos de datos estándar (HTML, JavaScript...) y se almacenan en unidades llamadas servidores. Los usuarios visitan dichos contenidos y aplicaciones mediante una herramienta denominada "Navegador". Al requerir un documento especifican la dirección en la que se encuentra dando su URL (todos los servidores y contenidos en la Web tienen una URL asociada), la máquina del usuario realiza una petición al servidor que responde con los datos tan pronto como los ha encontrado. Estos intercambios entre cliente y servidor se pueden regir por diversos protocolos (HTTP, FTP...).

3.2 El modelo WAP

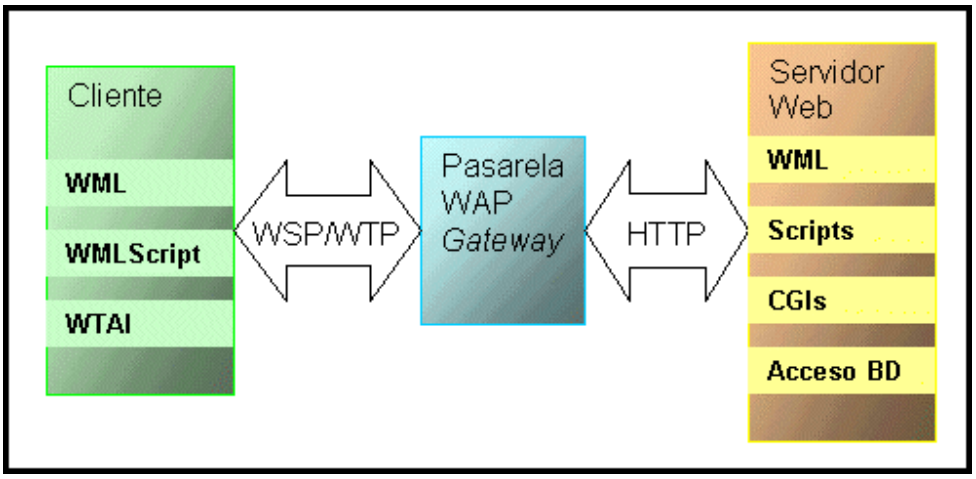
WAP nace de la idea de prestar acceso a Internet a través de un terminal móvil, luego es lógico que herede gran parte de su diseño de red. Esto proporciona grandes ventajas tanto para el desarrollo de aplicaciones como para los operadores que quieran ofrecer WAP así como para los proveedores de contenidos.

Obviamente, resulta necesario adaptar la red a un acceso inalámbrico, optimizándola todo lo posible debido a las pérdidas generadas en la

transmisión de radio con la consiguiente limitación de velocidad binaria. Para ello se contará con una interfaz que adapte los protocolos de GSM a WWW.

Como lo muestra la figura 11, se definen entonces dos partes bien diferenciadas, la primera entre el dispositivo móvil y el *Gateway*, donde se seguirá el protocolo WAP, y el lenguaje será WML, y la segunda, entre el *Gateway* y la dirección deseada de datos por visitar, que seguirá el protocolo HTTP.

Figura 11: Relación entre las 3 partes principales del modelo de aplicaciones WAP



Los pasos que sigue una petición desde la terminal son los siguientes:

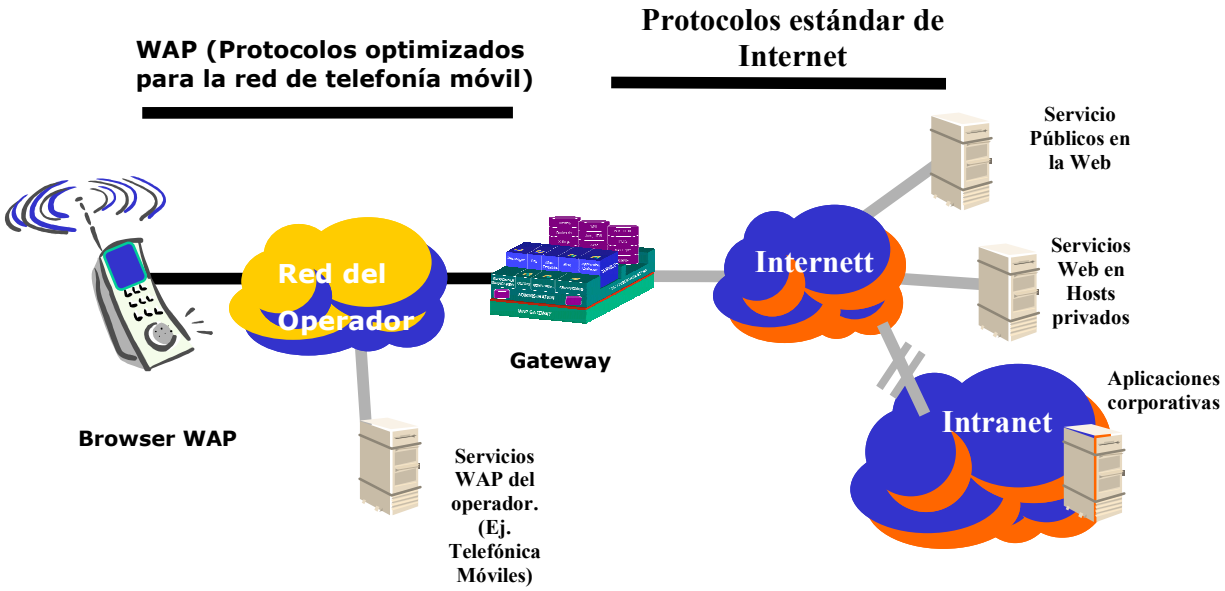
- La petición viaja vía radio hasta la estación base la que se encuentra conectada la terminal.

- La red GSM la encamina al *Gateway* que da acceso a WAP. Hasta éste momento la información estaba codificada en formato compacto en WML binario, para reducir el volumen de datos con el fin optimizar el rendimiento de la conexión con el móvil.

- El Gateway al que también se denomina *Proxy*, descodifica estos datos mediante sus codificadores/decodificadores. La función de *Proxy* del *Gateway* consiste en que recibe las peticiones del terminal, comportándose de cara a este último como un servidor, y realiza él las peticiones al servidor deseado, comportándose así como un cliente. Es pues un intermediario que no deja que el terminal vea el servidor final y viceversa. Por otra parte, en ocasiones, los contenidos solicitados se encuentran en el servidor Web final en formato WML por lo que se mandan directamente al terminal a través del *Gateway*, que los codifica. Pero puede ocurrir que los contenidos requeridos se encuentren en formato HTML (porque el proveedor de contenidos no haya pensado en su acceso por WAP), y sea necesario el empleo de un filtro, que suele encontrarse en el Gateway, que traduzca de HTML a WML los contenidos.

- El servidor recibe la petición, la procesa y responde.

Figura 12: Servidor WTA e Intranet en una red WAP



En la figura 12 puden identificarse dos elementos más. En primer lugar el servidor WTA[9], cuyo papel es el de dar acceso WAP a ciertas funciones de la red de móviles que ofrece el operador. El acceso a este servidor es directo, no se pasa por un *Gateway*/*Proxy*.

En segundo lugar también se abre la posibilidad de prestar acceso a la Intranet de una empresa a travez de móviles, si está conectada a la Web.

WAP define unos componentes estándares que permiten la comunicación entre los terminales y los servidores:

- URL- se usa el mismo estándar empleado para la www.
- El contenido especificado para aplicaciones WAP está basado en la tecnología empleada para la WWW, incluyen imágenes o lenguaje '*Script*'.
- Protocolos de comunicación: permiten la comunicación de los terminales con los servidores.

WAP emplea tecnología *proxy* para conectar el mundo de las ondas hertzianas con la el de la *World Wide Web*. Un WAP proxy normalmente cumple con las siguientes funciones:

- Protocolo *Gateway*: Se encarga de traducir la pila de protocolos WAP (WSP, WTP, WTLS y WDP) a la de protocolos WWW (HTTP y TCP/IP). El WAP *Gateway* es un *software* capaz de estar conectado a la red de telefonía móvil y a internet, actuando como intermediario

[9] WTA: *Wireless Telephony Application* (Servicios WAP del operador).

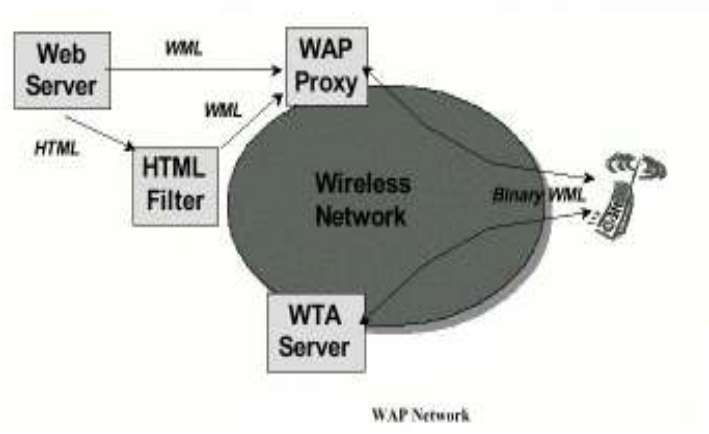
entre el dispositivo WAP y los servidores donde se encuentra la información requerida.

- Codificadores y decodificadores: Comprimen la información con el objetivo de permitir una transmisión más fluida.

Esta infraestructura asegura a los usuarios el acceso a una amplia variedad de servicios WAP y permite desarrollar aplicaciones que funcionen en todos los dispositivos que la acepten. El WAP *Proxy* permite que las aplicaciones y la información creadas para este fin, sean alojadas en los servidores de internet y desarrollas con las herramientas ya conocidas, como los CGI o los Servlets de Java.

La arquitectura WAP permite la creación de diferentes configuraciones a la hora de establecer los servicios. Un buen ejemplo sería la configuración: *web server-WAP proxy- cliente WAP,* aunque también es posible que un servidor contenga la función de WAP *proxy*, configuración que sirve para realizar un mejor control acceso. Como ejemplo podemos ver la figura 13.

Figura 13: Posibles configuraciones en una red WAP



Como se ve en la figura 13, el cliente WAP se comunica con dos servidores. El WAP *Proxy*, traduce las peticiones WAP a peticiones WWW y viceversa (por ejemplo HTML a WML), así como codificar la información enviada al terminal en formato binario, que es el que entiende el cliente WAP. El servidor WTA, es un ejemplo de servidor que se comunica con el terminal directamente.

3.2.1 Los componentes de la arquitectura

Entre los componentes de la arquitectura de una red WAP, pueden definirse tres elementos principales:

3.2.1.1 Terminales

Para poder acceder y visualizar las diferentes páginas de forma remota se debe disponer de una terminal adecuada. Este tipo de terminales puede ser teléfonos móviles con soporte para WAP; PDAs o *handhelds* conectados a alguna tarjeta que permita enviar y recibir datos via radio.

Los terminales tienen las siguientes características:

- Lleva incorporado un *microbrowser*, que es el que realiza la traducción de las páginas WML, y las presenta en pantalla. Interpreta los hipervínculos mediante los que el usuario quiere saltar a otras páginas. Se encarga de generar la secuencia de acciones que lleve consigo un posible código *WMLScript*. Carga las imágenes que contenga la página (formato *.wbmp). Es decir es el encargado de que podamos realizar la navegación a través del terminal móvil, comportándose de forma

similar a los navegadores (browsers) desarrollados para PCs como el *Netscape Communicator* o el *Internet Explorer*.

- El estándar permite un alto grado de libertad a la hora de implantar los micro-navegadores. Esta variabilidad implica diferencias en la presentación de las páginas, lo que supone que una página que tenga un diseño óptimo para un determinado terminal, no pueda observarse con la misma calidad en otro terminal con diferente microbrowser.

- No todos los terminales se ajustan al estándar, lo que hace difícil el desarrollo de aplicaciones válidas para cualquier equipo.

- En la actualidad coexisten distintas versiones de WAP, por lo que existirán terminales con versiones más antiguas del estándar su *microbrowser*, que no podrán visualizar las mejoras añadidas por las nuevas versiones del estándar. La solución parece obvia, basta con cambiar el software del terminal, aunque quienes realicen el desarrollo no deben olvidar de este inconveniente, que por otro lado también presenta la "Internet fija".

- La pantalla tiene limitaciones debido a su reducido tamaño, a su resolución y a la calidad. Ello limita el uso de imágenes en color o con gran resolución.

- El interfaz de usuario presenta ciertas características que deben considerarse a la hora de diseñar aplicaciones WAP. Bien sea la baja utilización del teclado, el número de botones, el número de líneas y

caracteres que puede visualizarse sin realizar un *scroll* por la página, etc.

- Presenta una amplia interoperabilidad con otros servicios y portadores.

3.2.1.2 *Gateway*/*Proxy*

- Es el punto de acceso a la Web desde la red móvil

- Adapta los protocolos WML (ligero) y HTML (Web), descodificando los datos provenientes del terminal y codificando aquellos que vienen del servidor o traduciendo los contenidos de HTML a WML si es necesario

- Puede pesentar interoperabilidad ineficiente con terminales y aplicaciones.

- No siempre adapta los contenidos del mismo modo. En principio, procura reconocer el terminal con el que trabaja pero puede suceder que no sea adecuada la adaptación que realiza sin ser este un problema que genere el terminal en sí. También puede ocurrir con ciertas aplicaciones, que vean sus prestaciones degradadas debido a incompatibilidades originadas en el *gateway*.

3.2.1.3 Servidor

Almacena los contenidos que se ofrece en la Web. No tiene porqué ser propiedad del operador que ofrece el servicio WAP. Los contenidos

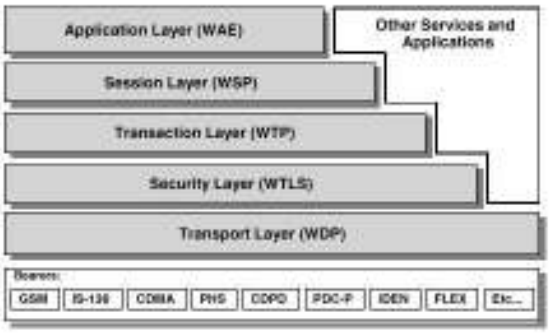
almacenados pueden estar en formato HTML o WML, en función del interés del proveedor de contenidos y del proveedor de servicios por que estén disponibles en WAP o únicamente en acceso Web. De no estar en este último formato se requerirá una traducción en el g*ateway*.

Un servidor establece su plan de negocio sobre la publicidad buscando generar el máximo tráfico posible a través de sus páginas. La publicidad puede ser vendida a terceros como "*banners*" o consistir en auto-promoción si quién gestiona el servidor es la propia empresa.

3.2.2 Capas del protocolo WAP

Los tres elementos identificados anteriormente son indispensables en la arquitectura de toda red WAP, sin embargo, esta se compone de un entorno escalable y extensible para el desarrollo de aplicaciones de telefonía móvil. Esto se realiza a través de un diseño del protocolo WAP mediante capas, en el que cada capa es accesible por la capa superior y por otros servicios y aplicaciones, en la figura 14 se puede ver la jerarquía de las capas del protocolo WAP.

Figura 14: Capas de la arquitectura del protocolo WAP



Esta arquitectura de capas permite a otros servicios utilizar las funciones de la pila del protocolo WAP, a través de interfaces definidas. Además puede haber otras aplicaciones exteriores, que pueden acceder directamente a las capas de sesión, transacciones seguras y transporte, ver la figura 14.

3.2.2.1 *WAE (Wireless Aplication Protocol)*

Aplicación de propósito general basada en la combinación de las tecnologías de la WWW y de la telefonía móvil. Su principal objetivo es establecer la interoperatibilidad entre los operadores y los proveedores de servicios para permitir la construcción de aplicaciones de una forma eficiente, independientemente del dispositivo en el que se ejecuten (móvil, PDA,...). WAE incluye un *micro browser* con las siguientes funciones:

- Un lenguaje específico optimizado para este tipo de terminales, WML.
- Un lenguaje script, WMLScript, que aporta inteleigencia al WML.
- Servicios de telefonía (WTA), e interfaces de programación (WTAI).
- Formatos de imágenes, información de calendario, agenda telefónica, etc.

3.2.2.2 WSP (*Wireless Session Protocol*)

Provee a la aplicación de un interfaz para el establecimiento de sesiones. Para ello se han creado dos tipos de protocolos diferentes: el primero crea los servicios orientados hacia la conexión mediante el servicio de transacciones (WTP), el otro permite acceder directamente sobre WDP, sin

la necesidad de establecer una conexión, mejorando el rendimiento de aplicaciones que no necesitan confirmación de envío de datos.

WSP actualmente dispone de servicios creados para aplicaciones sobre un *browser* (WSP/B). El diseño de WSP/B permite, por ejemplo, que un WAP *proxy* conecte a un cliente WSP/B, con un servidor HTTP.

3.2.2.3 WTP (*Wireless Transaction Protocol*)

Provee un simplificado protocolo diseñado para situaciones de poco ancho de banda, como lo es el caso de las comunicaciones inalámbricas.

Ofrece tres tipos de transacciones:

- Una para el envía de mensajes sin contestación por parte del servidor (*class* 0).
- Otra con notificación de recepción de mensaje (*class* 1),
- Una transacción que permite al receptor enviar un mensaje de resultado al emisor y a éste otro mensaje de confirmación al receptor.

Este protocolo también permite la concatenación de mensajes y la demora en el envío de notificaciones para minimizar el número de mensajes enviados.

3.2.2.4 WTLS (*Wireless Transport Layer Security*)

Protocolo de seguridad basado en el protocolo industrial estándar TLS (*Transport Layer Security*), formalmente conocido como SSL(*Secure Sockets*

Layer). Ha sido optimizado a partir de éste para ser usado en canales de comunicación de ancho de banda limitado. Características:

- Integridad de los datos: WTLS tiene herramientas que aseguran que los datos enviados por el terminal llegan al servidor sin sufrir ninguna variación.
- Privacidad: empleando WTLS los datos pueden ser encriptados.
- Certificación: permite establecer validaciones del terminal y del servidor de aplicaciones.
- Denegación: este protocolo puede rechazar datos que no han sido correctamente verificados.

WTLS también puede ser usado para la seguridad de comunicaciones entre terminales, como por ejemplo, para la autentificación de tarjetas de monedero electrónico.

Las funciones del WTLS pueden ser habilitadas o deshabilitadas por otras aplicaciones, atendiendo a las necesidades de seguridad y las características de la red (por ejemplo se podría deshabilitar en redes que ya lleven implementados protocolos de seguridad).

3.2.2.5 WDP (*Wireless Datagram Protocol*)

Como un servicio general de transmisión, WDP ofrece los diferentes servicios necesarios para que las capas de protocolos que tiene por encima

puedan realizar una comunicación transparente con alguno de los portadores disponibles.

Como los protocolos WDP proporcionan un interface a las capas de protocolos superiores, WAE, WSP y WTLS son capaces de funcionar de forma independiente de la red telefónica sobre la que trabajen, con tan solo adaptar WDP a las especificaciones de dicha red.

3.2.2.6 Bearers

Los protocolos WAP han sido diseñados para operar sobre una amplia variedad de servicios portadores o '*bearer*'. Estos '*bearers*' ofrecen diferentes niveles de calidad de servicio con respecto a tasas de errores, retrasos, etc.

El portador se encarga de transmitir los datos desde los dispositivos WAP hasta las operadoras de telefonía. WAP es independiente del portador de la información, de ello se encarga el WDP, que adapta el transporte de información a las diferentes formas posibles.

Algunos de estos portadores son el SMS (*Short Message Service*) empleado ampliamente para el envío de mensajes con cualquier teléfono móvil digital. Tiene la limitación de 160 caracteres por mensaje. EL CSD (*Circuit Switched Data*) que es el más extendido en los servicios basados en WAP y el GPRS(*General Packet Radio Service*), destinado en convertirse en el sistema escogido para todos los servicios, debido a sus relativa rapidez y a la eficacia de las conexiones, como predecesor del sistema UMTS.

3.2.2.7 Otros servicios

La arquitectura de capas permite utilizar las ventajas de su pila de protocolos a través de unas interfaces definidas a otros servicios y aplicaciones, que actualmente no estén definidas por WAP. Como por ejemplo se puede citar aplicaciones para correo electrónico, calendario, agendas, páginas amarillas, etc.

4. ACCESO SEGURO A INTERNET MOVIL

Todos conocemos el crecimiento que ha experimentado Internet y la telefonía móvil en estos últimos años, este crecimiento se ha debido en gran parte a la facilidad de uso y en el beneficio que obtienen los usuarios de los diferentes servicios, además de esto la reducción de precios que ha existido. Esto ha motivado que los usuarios demanden nuevos servicios a los operadores, en este capítulo se hará una reseña de las diferentes tecnologías de telefonía móvil (GSM, GPRS y UMTS) explicando la manera en la cual se realiza el proceso de autenticación y cifrado de la comunicación, en la última parte se expone el protocolo WAP que permitirá la navegación de Internet a través de móviles, haciendo énfasis en la seguridad que este posee introduciendo los diferentes formas de implementación de mecanismos de seguridad en la comunicación, realizando una comparación entre las diferentes alternativas existentes indicando sus características, ventajas y desventajas de cada una de ellas.

4.1 Seguridad en el estándar GSM

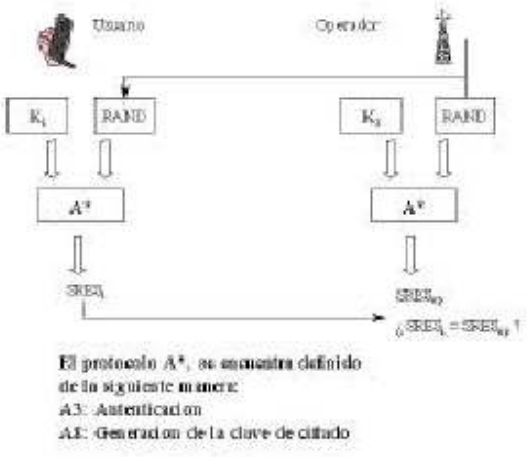
El estándar GSM es el sistema de telefonía móvil más usado alrededor del mundo (51 % del mercado compartido de todos los teléfonos celulares, tanto analógicos como digitales), con más de 215 millones de usuarios en América, Europa, Asia, África y Australia; dicho estándar hace uso de un conjunto de algoritmos criptográficos para proporcionar los mecanismos de seguridad del sistema (Autenticación y Confidencialidad), estos algoritmos son:

Tabla I: Algoritmos criptográficos utilizados en el estándar GSM

Código del Algoritmo	Descripción
A3	Algoritmo de autenticación
A5/1 – A5/2	Algoritmos de cifrado
A8	Algoritmo de generación de clave

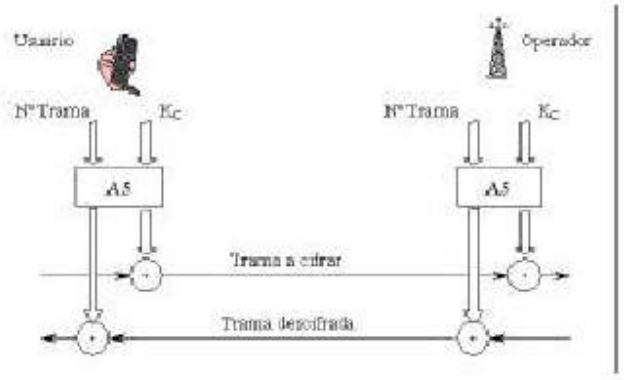
Dichos algoritmos fueron desarrollados de forma secreta; el procedimiento en el cual se describe el proceso de autenticación y generación de claves se muestra en la Figura 13, donde se observa que en ambos procesos el sistema hace uso de una clave secreta (Ki) que servirá como entrada al algoritmo correspondiente (A3 o A8). Dicha clave es conocida únicamente por el SIM (*Subscriber Information Module*) y el operador, estas claves son diferentes para el algoritmo autenticación y generación de claves empleando un número secreto aleatorio (RAND) generado por el operador.

Figura 13: Proceso de autenticación y generación de claves en el estándar GSM



La mayoría de los proveedores GSM utilizan un algoritmo denominado COMP128 tanto para A3 como para A8, este algoritmo es criptográficamente débil y no es difícil de romperlo y clonar teléfonos móviles. En Abril de 1998, un grupo de investigación de Berkeley publicó un análisis de COMP128. Este ataque puede ser llevado simultáneamente a cabo sobre tantos teléfonos en un rango de radio tan amplio como canales tenga la estación base con la que se lleva a cabo el delito. Demostrando en esta investigación que todas las implementaciones A8 que se habían examinado, incluyendo las pocas que no usaban COMP128, eran deliberadamente débiles.

Figura 14: Procedimiento de cifrado y descifrado en GSM



El procedimiento de cifrado esta basado en la suma OR exclusiva de los bits a transmitir, el algoritmo ocupado para la generación de las secuencias de cifrado y descifrado es secreto y se denomina A5, del cual existen dos versiones: A5/1 y A5/2. Este tiene dos entradas: el Nº de trama (22 bits) y la clave de cifrado KC (64 bits), con estas entradas el algoritmo desarrolla procedimientos matemáticos obteniendo a su salida dos secuencias binarias de 114 bits, una de las cuales se utiliza para cifrar y la otra para descifrar, el procedimiento anteriormente descrito se muestra en la Figura 14.

Entre los meses de Mayo y Agosto de 1999 se analizaron los algoritmos A5/1 y A5/2 encontrándose que los dos algoritmos encargados del cifrado en GSM son imperfectos, se encontró que el A5/2 es el más débil de los dos algoritmos de cifrado ya que este puede ser roto en tiempo real sin ningún problema; teniendo un factor de trabajo de aproximadamente 2^{16}.

4.2 Seguridad en los estándares GPRS y UMTS

GPRS (*General Packet Radio Services*) es un nuevo conjunto de servicios desarrollado por el ETSI (*European Telecommunication Standard Institute*) los cuales se añadirán a los actuales que posee GSM, actualmente se están introduciendo y básicamente añade conmutación de paquetes de datos a todos los niveles de la red GSM.

GPRS ofrece funciones de autenticación, control de accesos, confidencialidad de la identidad del usuario y confidencialidad de la información. Los algoritmos empleados en el proceso de autenticación son los mismos que los de GSM (A3 y A8), mientras que el algoritmo utilizado para el cifrado de los datos de usuario ha sido modificado debido a la naturaleza del tráfico de GPRS, dicho algoritmo denominado GPRS A5 fue definido por 5 personas en SAGE (*Security Advisor Group of Experts*) del ETSI y no se encuentra disponible de forma pública. A GPRS se le denomina como la generación 2.5, ya que es el paso intermedio a los nuevos sistemas de 3ª generación.

UMTS (*Universal Mobile Telephone System*) es el sistema de telefonía móvil de 3ª generación el cual se encontrará disponible pronto, UMTS será

capaz de alcanzar velocidades entre 384 Kbps. para entornos de redes de banda ancha y 2.0 Mbps. para entornos locales.

Respecto a los mecanismos de seguridad del sistema UMTS estos se encuentran en la fase de desarrollo, han sido propuestos diferentes mecanismos para proporcionar autenticación, confidencialidad y generación de claves.

4.3 Seguridad en el estándar WAP

La capa de transporte viene definida por el protocolo WDP (*Wireless Datagram Protocol*), este permite hacer uso de las mismas aplicaciones en diferentes tipos de portadoras (distintas frecuencias o distintos protocolos de acceso al medio) o señales de información. En la capa de seguridad se emplea el protocolo WTLS (*Wireless Transport Layer Security*) el cual es derivado del SSL 3.1, es basado en el sistema abierto TLS 1.0 proporcionando los elementos de seguridad de confidencialidad, integridad y autenticación; la verificación de la autenticación, no-repudio son dadas por una PKI (*Public Key Insfraestructure*).

La capa de transacción esta basada en WTP (*Wireless Transaction Protocol*) derivado del TCP, la función principal de esta capa es eliminar los datagramas no utilizados y preparar la información para la capa superior. WSP (*Wireless Session Protocol*), es el protocolo que se empleará en la capa de sesión y está preparado para agrupar varias operaciones WTP siendo encargado también del restablecimiento de las conexiones que excedan el tiempo de vida asignado al iniciar la conexión. La última capa, la de aplicación

define la interfaz de usuario en el teléfono, hace uso de: *Wireless Markup Language* (WML), *WMLScript* y *Wireless Telephony Application* (WTA).

Al igual que UMTS los mecanismos de seguridad de WAP se encuentran en una etapa de desarrollo aunque ya existen algunas herramientas que se apoyan en dicho estandar para ofrecer los elementos de confidencialidad, integridad, autenticidad y no-repudio. Así tenemos *W/Secure* y *Baltimore Telepathy* los cuales contienen una implementación de WTLS, existen diferentes forma de implementar dichos mecanismos de seguridad, entre los cuales tenemos:

- Autenticación mutua sobre la interfaz aire, la cual serviría para establecer
  parámetros importantes de seguridad.

- Cifrado interfaz aire, para emplear este tipo de cifrado es necesario hacer uso de diferentes claves de control junto con la información de señalización.

- Cifrado punto a punto: por medio de este tipo de cifrado la aplicación puede verificar las claves de administración sin ningún problema y de esta manera los datos que hace uso la aplicación nunca serán expuesta fuera de ella.

4.4 Mecanismos de seguridad

WAP ofrece una arquitectura flexible de seguridad, centrándose en proporcionar seguridad entre la conexión que posee un usuario y un servidor WAP, es decir, en general no ofrece mecanismos de seguridad extremo a extremo entre el usuario del terminal móvil y el servidor Web de Internet. Sin embargo, muchas aplicaciones requieren servicios de seguridad extremo a extremo (en particular es especialmente crítica la autenticación entre clientes y servidores Web). A continuación se evalúan diferentes opciones para ofrecer estos servicios extremo a extremo, estudiando el compromiso entre el nivel de seguridad requerido y la complejidad y coste de la solución adoptada. Se consideran las siguientes opciones:

- Confiar en el servidor WAP y utilizar el mecanismo de autenticación de la red móvil*:* En este caso se cede toda la autenticación del cliente a la propia red móvil, y el servidor WAP establece una conexión SSL con el servidor Web. Esta solución requiere confianza total en el servidor WAP, pero se puede implementar con facilidad y en la red móvil no es necesario utilizar el protocolo WTLS.

- Confiar en el servidor WAP y utilizar WTLS entre cliente y servidor WAP: Se aumenta la seguridad en la red móvil, pero nuevamente es necesaria una confianza total en el servidor. Requiere que los terminales móviles y el servidor WAP implementen WTLS.

- Utilizar una conexión WTLS con el servidor Web remoto: Esta solución no requiere confianza en el servidor WAP (las medidas de seguridad

se implementan extremo a extremo). A cambio, requiere que el servidor de Internet ofrezca un servidor WTLS.

- Proteger la comunicación a nivel de aplicación: Ciertas aplicaciones críticas requerirán servicios especiales de seguridad (como no repudio) que forzosamente se deben ofrecer a nivel de aplicación.

La Figura 15 muestra las opciones anteriores, analizadas en la Tabla II

Figura 15: Opciones para ofrecer servicios de seguridad extremo a extremo

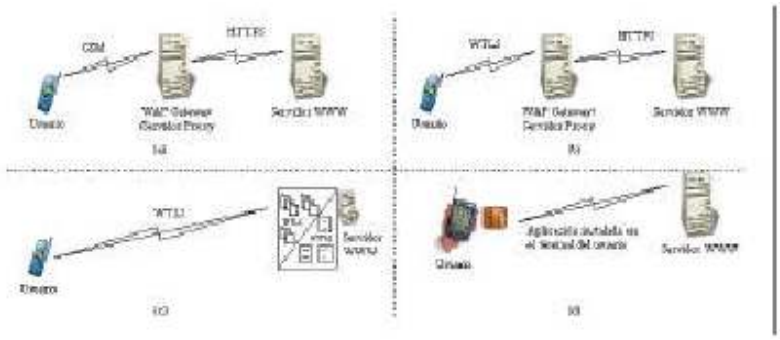


Tabla II: Características, ventajas y desventajas de las opciones de seguridad extremo a extremo

Opcion	Características	Ventajas	Desventajas
1	- Emplea características de autenticación de la red móvil. - Confianza en el servidor WAP y Proxy.	- No se debe emplear ningun otro método de autenticación para el transporte de la información.	- Requiere confianza en el servidor WAP y Proxy. -Considerar las debilidades encontradas en los algoritmos de autenticación y cifrado de GSM.

Continuación

2	- Emplea WTSL entre el móvil y el servidor WAP. - Confianza en el servidor WAP y Proxy. - Se necesita que los terminales móviles y el servidor implemente WTSL.	- Mayor seguridad en la comunicación ya que hace uso del protocolo WTSL. - No centralizado. - Hace uso de una PKI.	- Requiere confianza en el servidor WAP y Proxy. - Hace uso de ceritificados, lo cual incrementa los costos.
3	- Las medidas de seguridad se aplican a nivel de transporte empleando seguridad extremo a extremo. - El servidor de Internet debe ofrecer WTSL.	- Mayor seguridad en la comunicación ya que hace uso del protocolo WTSL. - No requiere confianza con el servidor WAP.	- Mayor complejidad ya que el servidor de Internet sirve como servidor WTSL. - Centralizado, toda la carga de trabajo se encuentra sobre el servidor.
4	-Seguridad a nivel de aplicación. - La aplicación es almacenada en el SIM	- La mejor solucion respecto a la seguridad. - Se tiene la facilidad de ofrecer servicios especiales.	- La complejidad aumenta, ya que ahora la aplicación junto con la información del móvil debe ser almacenada en el SIM. - Mayor complejidad en los terminales.

4.5 La capa de seguridad WTSL

La capa de seguridad en la arquitectura WAP es llamada la Capa de Seguridad de Transporte Inalámbrico (*Wireless Transport Layer Security* – WTLS). WTLS opera encima de la capa de transporte y es una capa modular que depende de los niveles de seguridad requeridos de una aplicación dada. Adicionalmente, WTLS provee una interfase para el manejo de conexiones seguras.

El principal objetivo de la capa WTLS es proveer privacidad, integridad de datos y comunicación entre dos aplicaciones que se comunican. Provee una funcionalidad similar a TLS (*Transport Layer Security*), e incorpora nuevas características tales como soporte de datagramas y negociación optimizada para portadores de red de bajo ancho de banda con largas latencias relativas.

4.5.1 Administración de la conexión WTLS

La administración de una conexión WTLS le permite al cliente conectarse con un servidor y acordar los protocolos a ser usados. El establecimiento de una conexión segura consiste de varios pasos, y tanto el cliente como el servidor pueden interrumpir la negociación en cualquier momento si los parámetros propuestos por el par no son aceptables. La negociación puede incluir parámetros de seguridad como algoritmos criptográficos, intercambio de llaves y autenticación.

La secuencia de primitivas para el establecimiento de una sesión segura, de negociación completa es mostrada en la figura 16.

La secuencia de primitivas para el establecimiento de una sesión segura de forma abreviada es mostrada en la figura 17.

Figura 16: Negociación completa

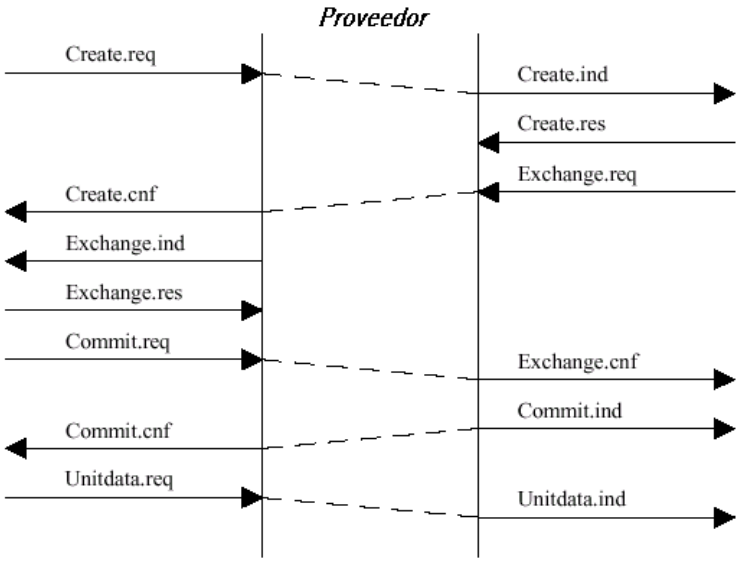
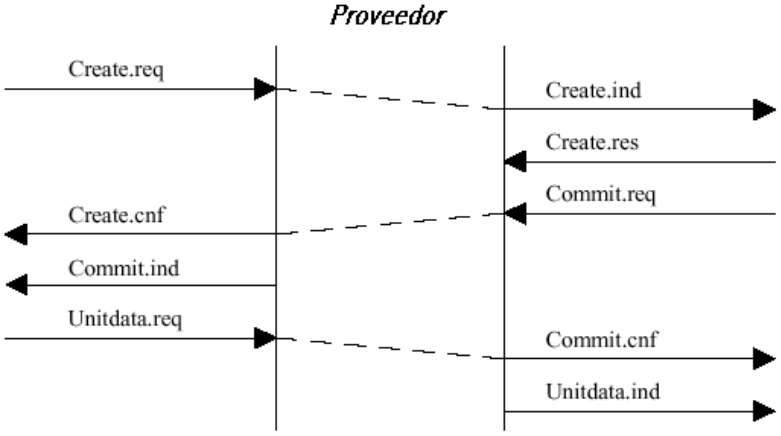


Figura 17: Secuencia de primitivas para el establecimiento de una sesión segura.



4.5.1.1 Primitivas del servicio

Tabla III: Primitivas de servicio de la administración de la conexión WTLS

Primitiva	Descripción
SEC-Create	Usada para iniciar el establecimiento de una conexión segura
SEC-Exchange	Usada en la creación de una conexión segura si el servidor desea ejecutar autenticación de llave publica o intercambio de llaves con el cliente
SEC-Commit	Esta primitiva es iniciada cuando la negociación es completada y cualquier par requiere conmutar a un estado de conexión de negociación nuevo.
SEC-Terminate	Usada para terminar la conexión
SEC-Exception	Usada para informar los otros niveles de alerta
SEC-Create-Request	Usada por el servidor para requerirle al cliente el inicio de un nuevo acuerdo de protocolos.

4.6 Protocolo de registro

El protocolo de registro toma mensajes a ser transmitidos, opcionalmente comprime los datos, aplica el algoritmo del código de autenticación de mensajes (*Message Authentication Code* – MAC) y transmite el resultado. Los datos recibidos son desencriptados, verificados y descomprimidos, entonces son entregados a los clientes de niveles superiores.

Son cuatro los protocolos de registro de cliente:

- El protocolo de cambio de cifrado.
- El protocolo de acuerdo de protocolos.
- El protocolo de alerta.
- El protocolo de aplicación de datos.

Si la implementación WTLS recibe un tipo de registro que no entiende, deberá ignorarlo.

4.6.1 Estado de conexión

El estado de una conexión WTLS es el entorno de operación del protocolo de registro de WTLS. Aquí se especifica el algoritmo de compresión, algoritmo de encriptación y el algoritmo MAC. Adicionalmente, los parámetros de estos algoritmos son conocidos: el secreto MAC y las llaves de encriptación de una conexión segura.

Siempre hay dos estados de conexión destacados: el estado actual y el estado pendiente. Todos los registros son procesados bajo el estado actual. Los parámetros de seguridad para el estado pendiente son configurados por el protocolo de acuerdo de protocolos (*handshake*) de WTLS, que es el que se encarga de convertir el estado pendiente a actual. El estado pendiente es entonces reinicializado como un estado vació.

Los parámetros de seguridad configurados en el estado de conexión se encuentran listados y descritos en la tabla IV.

Tabla IV: Parámetros de seguridad configurados en el estado de conexión

Parámetro	Descripción
Fin de conexión	Determina si esta entidad es considerada un cliente o un servidor en una sesión segura
Algoritmo de cifrado en masa	Algoritmo a ser usado para encriptar en masa. Esto incluye el tamaño de la llave, que cantidad de la llave es secreta, si es un cifrado de bloque o de flujo y el tamaño del bloque de cifrado (si es apropiado).
Algoritmo MAC	Usado para la autenticación de mensajes. Incluye el tamaño de la llave usada en el calculo MAC y el tamaño del *hash* retornado por el algoritmo.
Algoritmo de compresión	Encargado de comprimir los datos antes de la encripcion.
Secreto maestro	20 bytes ocultos compartidos por dos pares en una conexión segura
Cliente aleatorio	Valor de 16 bytes proveídos por el cliente
Servidor aleatorio	Valor de 16 bytes proveídos por el servidor
Modo de numeración de secuencia	Esquema usado en la comunicación de números de secuencia en una conexión segura. Puede ser implícita, explicita., o no usar ninguna.
Renovación de llave	Define cuan a menudo deben ser actualizados algunos parámetros del estado de conexión

Una vez estos parámetros de seguridad son configurados y las llaves generadas, los estado de conexión actuales pueden ser actualizados por cada registro procesado. Los elementos que incluye cada estado de conexión se encuentran listados y descritos en la tabla V.

Tabla V: Elementos incluidos en un estado de conexión

Elemento	Descripción
Estado de compresión	El estado actual del algoritmo de compresión
Secreto MAC del cliente de escritura.	El secreto usado para el calculo y verificación de los registros enviados por el cliente
Llave de cifrado del cliente de escritura	La llave usada para el cifrado y descifrado de los registros enviados por el cliente
Numero de secuencia del cliente de escritura.	Numero de secuencia usado por los registros enviados por el cliente. Los números de secuencia no pueden exceder 2^{16}-1. Cuando un nuevo estado de conexión es establecido, el numero de secuencia del primer registro es cero
Secreto MAC del servidor de escritura	El secreto usado para el calculo y verificación de los registros enviados por el servidor
Llave de encripcion del servidor de escritura.	La llave usado para el encripcion y desencripcion de registros enviados por el servidor. La llave debe ser actualizada de acuerdo a los parámetros de refresco de llaves.
Numero de secuencia del servidor de escritura.	Numero de secuencia usado por los registros enviados por el servidor. Cuando un nuevo estado de conexión es establecido, el numero de secuencia del primer registro es cero

4.6.2 La capa de registro

La capa de registro de WTLS recibe datos sin interpretar de las capas superiores en bloques no vacíos de tamaño máximo de 2^{16}-1.

4.6.2.1 Fragmentación

La capa de registro no fragmenta la información en bloques, ya que asume que la capa de transporte se encarga de la fragmentación y reensamble necesarios.

4.6.2.2 Compresión y descompresión de registros

Todos los registros son comprimidos usando el algoritmo de compresión definido por el estado de conexión actual. El algoritmo de compresión traduce el texto plano (*WTLSPlaintext*) en texto comprimido (*WTLSCompressed structure*). La encriptación y las funciones MAC traducen un texto comprimido en un texto cifrado. El proceso de descifrado lógicamente invierte el proceso.

4.6.2.3 Numeración de secuencia explicita

Cuando una numeración de secuencia explicita es usada, la verificación y desencripción de registros requiere de medidas especiales, como que debe ser usada con protocolos de transporte de datagramas significando que los registros pueden ser perdidos, duplicados o recibidos fuera de orden.

El receptor debe descartar los registros duplicados, lo cual puede ser realizado manteniendo un historial de los recibidos. Esto puede ser implementado usando una ventana corrediza. Por ejemplo una ventana de tamaño de 32 puede ser usada, la cual nos permite recibir mensajes con número de secuencia en el rango n-32 + 1 ... n donde n es el numero de secuencia actual, donde el numero de secuencia mas largo de los últimos registros recibidos no debe ser mayor a 32.

Los registros con números de secuencia menores o iguales a n-32 deben ser descartados. Un registro se considera recibido si no es descartado o ignorado. Un registro con un número de secuencia más grande a n-32 debe ser examinado. Si es recibido y el número de secuencia en el registro es más grande que n, n debe ser reemplazado con número de secuencia y la ventana es adelantada. Esto produce que la ventana sea incrementada casualmente, ya que siempre inicia de cero.

Cuando un *Handshake* inicia con intercambios de mensajes en texto plano, los números de secuencia inician de cero y son incrementados en uno por cada mensaje *Handshake*. Cuando un *Handshake* ocurre sobre una conexión segura, el numero de secuencia actual para la conexión segura es usado por los mensajes *Handshake* y son incrementados en uno por cada uno de ellos. Son configurados a cero por el mensaje del protocolo *ChangeCipherSpec*, que será explicado mas adelante. En las retransmisiones, los números de secuencia permanecen iguales como en el mensaje original. Cuando el numero de secuencia excede 2^{16} –1, la conexión segura debe ser cerrada.

4.7 Acuerdo de protocolos (*Handshake*)

El acuerdo de protocolos de WTLS esta compuesto por tres subprotocolos los cuales son usados para permitir a los pares acordar sobre los parámetros de seguridad para la capa de registro, la autenticación de ellos mismo, los parámetros de seguridad de una negociación inminente y las condiciones de reportes de error de cada par. El protocolo *Handshake* es responsable de la negociación de una sesión segura, la cual consiste de los siguientes componentes listados en la tabla VI.

Tabla VI: Componentes del protocolo de negociación

Componente	Descripción
Identificador de sesión	Es una secuencia arbitraria de bytes escogida por el servidor para identificar una sesión segura activa o reanudada.
Versión del protocolo	El numero de la versión del protocolo WTLS.
Certificado de par	Este elemento puede ser nulo.
Método de compresión	El algoritmo usado para la compresión de los datos anteriores a la encriptación
Cipher Spec	Especifica el algoritmo de cifrado de los datos en masa (tal como RC5, DES, etc.) y el algoritmo MAC (como SHA-1). También define atributos criptográficos como *mac_size*.
Secreto maestro	20 bytes secretos compartidos entre el cliente y el servidor
Modo de numero de secuencia	Esquema de numeración de secuencia la cual es usada en una conexión segura.

Continuación

Renovación de llave	Define cuan a menudo deben ser calculados valores de estado tales como la llave de encriptación y MAC.
Resumible	Bandera indicando si una sesión segura puede se usada al iniciar una nueva conexión segura.

Estos items son entonces usados al crear los parámetros de seguridad que son usados por la capa de registro cuando a los datos se les aplica protección. Muchas conexiones seguras pueden ser utilizadas usando la misma conexión segura a través de la característica de reanudación del protocolo *Handshake* de WTLS.

4.7.1 Protocolo *Change Cipher Spec*

El protocolo *Change Cipher Spec* se aplica en la transición de una señal en estrategias de cifrado. El protocolo consiste de un sencillo mensaje, el *ChangeCipherSpec* el cual es encriptado y comprimido bajo el actual (no el pendiente) estado de conexión. El mensaje consiste de un sencillo byte de valor igual a 1.

El *ChangeCipherSpec* es enviado por cliente o el servidor para notificar a la otra parte que registros subsecuentes serán protegidos bajo el recién negociado *CipherSpec* y las llaves. En la practica, el envió de este mensaje significa que el remitente a cambiado del estado corriente de escritura al estado pendiente. Cuando el *ChangeCipherSpec* es recibido, el receptor debe cambiar del estado corriente de lectura al estado pendiente. El *ChangeCipherSpec* es enviado durante el acuerdo de protocolos después de que los parámetros de seguridad han sido acordados, pero antes que el

mensaje de finalización sea enviado. Las implementaciones deben comprobar que esto sea cierto con el fin de que los mensajes subsecuentes sean protegidos bajo el recién negociado *CipherSpec*.

4.7.2 El protocolo de alerta

Uno de los tipos de contenidos soportados por la capa de registro de WTLS es el tipo alerta, los cuales determinan la inmediata terminación de una conexión segura si tienen un nivel fatal de alerta. En este caso, otras conexiones usando una conexión segura pueden continuar, pero la sesión identificada debe ser invalidada, previniendo el fallo de una sesión segura. Sin embargo, la invalidación de una sesión no es requerida en respuesta a una alerta fatal en ciertos escenarios. Una suma de chequeo es usada en las alertas, la cual es calculada del último registro recibido de la otra parte. El receptor de la alerta debe verificar que la suma de chequeo coincida con el primer mensaje por el enviado.

4.7.2.1 Alertas de cierre

El cliente y el servidor deben saber que una sesión segura esta terminada. Cualquiera de las partes puede iniciar el intercambio de mensajes de cierre. Los mensajes de cierre se encuentran listados y descritos en la tabla VII.

Tabla VII: Mensajes de cierre

Mensaje de cierre	Descripción
Connection_close_notify	Este mensaje notifica el deposito al cual el remitente no enviara mas mensajes usando este estado de conexión
Session_close_notify	Notifica que no se deben enviar mas mensajes usando este estado de conexión o sesión segura.

Cualquier parte puede iniciar un cierre enviando un *Connection_close_notify* o un *Session_close_notify*. Cualquier dato recibido después de una alerta de cierre es ignorado. Se requiere que la otra parte a su vez responda su propio *Connection_close_notify o Session_close_notify* para cerrar la conexión segura inmediatamente, descartando cualquier escrito pendiente. En el caso de *Session_close_notify*, el receptor debe también invalidar el identificador de sesión. No es requerido que el que inicio el cierre espere la alerta de respuesta antes de cerrar la lectura en una conexión segura. El nivel de alerta debe ser configurado como critico para *Connection_close_notify* y fatal para *Session_close_notify*.

4.7.2.2 Alertas de error

El manejo de los errores en el protocolo de negociación de WTLS es muy simple. Cuando un error es detectado, la parte que realiza la detección envía un mensaje a la otra parte. Antes de transmitir o recibir un mensaje de alerta fatal, las partes cierran inmediatamente las conexiones seguras. Los servidores y clientes, por así decirlo, olvidan cualquier identificador de sesión,

llaves y secretos asociados con la fallida conexión segura. Antes de transmitir o recibir un mensaje de alerta critico, ambas partes cierran inmediatamente las conexiones seguras pero pueden preservar los identificadores de sesión y usarlos para el establecimiento de nuevas conexiones seguras. Las alertas de errores se encuentran listadas y descritas en la tabla VIII.

Tabla VIII: Alertas de error

Alerta	Descripción
No_connection	Un mensaje fue recibido sin existir una conexión segura con el remitente. Este mensaje es fatal o critico y es enviado en texto claro.
Unexpected_message	Un mensaje inapropiado fue recibido. La alerta debe ser fatal o critica.
Bad_record_mac	Retornada si un registro es recibido con un MAC incorrecto. Este es un mensaje de advertencia y es enviado en texto claro.
Descryption_failed	El texto cifrado de WTLS (*WTLSCiphertext*) es descifrado de una manera inválida. Este mensaje es una advertencia y es enviado en texto claro.
Record_overflow	El registro *WTLSCiphertext* fue recibido con más bytes del largo permitido. Este mensaje es una advertencia y es enviado en texto claro.
Decompresion_failure	La función de descompresión recibe una entrada inapropiada, como datos que se expanden excesivamente. Este mensaje es una advertencia y es enviado en texto claro.

Continuación

Handshake_failure	El remitente fue incapaz de negociar con un conjunto aceptable de parámetros de seguridad dados por las opciones disponibles. Este es un error fatal.
Bad_certificate	El certificado fue corrupto, conteniendo firmas que no fueron verificadas correctamente.
Unsuppórted_certificate	El certificado fue de un tipo no soportado.
Certificate_revoked	El certificado fue revocado por el firmante.
Certificate_expired	El certificado ha expirado o no es valido actualmente.
Certificate_unknown	Algún certificado interpretado como inaceptable
Illegal_parameter	Campo en la negociación fuera de rango o inconsistente con otros campos. Esto es siempre fatal.
Unknown_ca	Cadena de certificado valida o cadena parcial que fue recibida, pero el certificado no fue aceptado porque el certificado CA no pudo ser localizado o no coincide con un CA confiable. Este mensaje siempre es fatal.
Access_denied	Un certificado valido fue recibido, pero cuando el control de acceso fue aplicado, el remitente decidió no proceder con la negociación. Este mensaje es siempre fatal.
Decode_error	Un mensaje no pudo ser decodificado porque algún campo estuvo fuera del rango específico o el largo del mensaje fue incorrecto. Este mensaje es crítico

	o fatal.

Continuación

Decrypt_error	Una operación criptográfica de acuerdo de protocolos fue fallida. Este mensaje debe ser enviado como fatal.
Unknown_key_id	Ninguno de los identificadores de llave (key_id's) del cliente es conocido o reconocido por el servidor. Esta es una alerta fatal.
Disabled_key_id	Todos los identificadores de llaves (key_id's) del cliente listados en el ClientHello.client_key son deshabilitados administrativamente. Esta es una alerta crítica.
Key_exchange_disabled	Para evitar que se intercambien llaves anónimas indeseables, esta función es administrativamente deshabilitada.
Session_not_ready	La sesión segura no esta lista para reanudar nuevas seguras conexiones debido a rezones administrativas tales como que el servidor se encuentra en mantenimiento. Esta es una alerta critica
Unknown_parameter _index	El cliente ha sugerido un intercambio de llaves que no pudo ser soportado por el servidor, pero el servidor no conoce le índice de parámetros suministrado.
Duplicate_finished _received	En un *handshake* optimizado o abreviado, el cliente ha enviado un segundo (reciente) mensaje finalizado. Este es un mensaje de advertencia.

Continuación

Export_restriction	Una negociación que no acata las restricciones de exportación fue detectada. Este es un mensaje siempre fatal.
Protocol_version	La versión del protocolo del cliente o el servidor que ha intentado negociar es reconocida, pero no soportada por el cliente o servidor. Este es un mensaje siempre fatal.
Insufficient_security	Retornado en lugar del *handshake_failure* cuando una negociación ha fallado específicamente porque el servidor requiere cifrado mas seguro que el soportado por el cliente. Este es un mensaje siempre fatal.
Internal_error	Un error interno no relacionado con el par hace imposible continuar, por ejemplo una falla en la localización de memoria. Este mensaje es fatal o critico.
User_canceled	El *Handshake* esta siendo cancelada por alguna razón no relacionada con fallas en el protocolo. Este es un mensaje de advertencia.
No_renegotiation	Enviado por el cliente en respuesta a un *hello request* o por el servidor es respuesta a *client hello* después de iniciada la negociación.

Para todos los errores en los cuales el nivel de alerta no es especificado explícitamente, la parte emisora puede determinar si es fatal,

critico o se trata de una advertencia. Si una alerta con un nivel critico o de advertencia es recibido, la parte receptora puede decidir si lo trata como un error fatal o no. Sin embargo, todos los mensajes que son transmitidos con un nivel fatal deben ser tratados inevitablemente como tales.

Las implementaciones pueden conservar un conteo de las alertas recibidas con niveles críticos o de advertencia, y tratarlos como fatales cuando cierto limite configurable es excedido.

Una alerta fatal solo termina la sesión a ser creada y mantiene intacta la sesión existente si el *Handshake* es conducida sobre una sesión segura. Sin embargo, pueden haber casos en que cerrar una sesión existente sea deseable. Una *Session_close_notify* debe ser enviada por el par si una de las partes decide terminar la sesión existente inmediatamente después de que una alerta fatal es recibida o enviada durante la negociación que intenta crear una nueva sesión. Bajo cualquier otra circunstancia, una alerta fatal es tratada normalmente como fue descrito antes.

Una sesión valida existente no debe ser invalidada si una alerta fatal en texto claro es recibida, aunque si debe serlo si la alerta no esta así. Una alerta de texto claro es aquella que no es ni encriptada ni autenticada.

4.7.3 Visión general del protocolo *Handshake*

Los parámetros criptográficos de una sesión segura son producidos por el protocolo *Handshake* de WTLS, el cual opera en la parte superior de la capa de registro de WTLS. Cuando un cliente WTLS y un servidor inician una comunicación, acuerdan la versión del protocolo, los algoritmos criptográficos seleccionados, autenticación y técnicas de cifrado de llave pública al generar secretos compartidos.

El protocolo *Handshake* de WTLS involucra los siguientes pasos:

- Intercambio de mensajes *hello* al acordar algoritmos e intercambio de valores aleatorios.

- Intercambio de parámetros criptográficos necesarios para permitir al cliente y al servidor acordar un secreto pre-maestro .
- Intercambio de certificados e información criptográfica para permitir al cliente y al servidor autenticarse ellos mismos.

- Generar un secreto maestro a partir del secreto pre-maestro e intercambiar valores aleatorios.

- Proveer parámetros de seguridad a la capa de registro.

- Permitir al cliente y al servidor verificar que sus pares han calculado los mismos parámetros de seguridad y que la negociación no ha sido interceptada.

Esto puede ser resumido de la siguiente forma: El cliente envía un mensaje *hello* de cliente, al cual el servidor le debe responder con un mensaje *hello* de servidor, de otra manera ocurrirá un error fatal y la conexión segura fallara. Los mensajes *hello* de cliente y de servidor establecen mejoras en las capacidades de seguridad entre ellos estableciendo los siguientes atributos: versión del protocolo, intercambio de llaves, cifrado, método de compresión, renovación de llaves y modo de numero de secuencia. Adicionalmente dos

valores aleatorios con generados e intercambiados: *ClientHello.random* y el *ServerHello.random*.

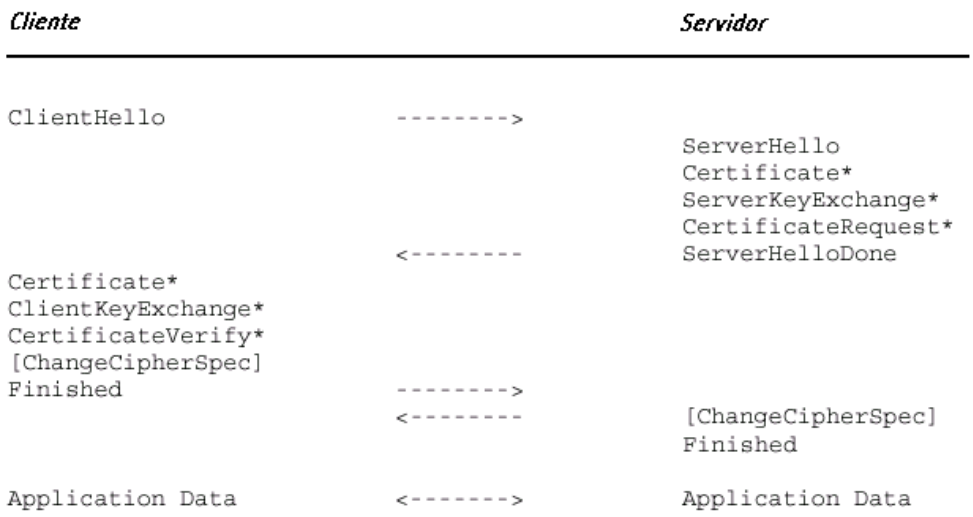
Después de enviar los mensajes *hello*, el servidor enviara un certificado, si este es autenticado. Adicionalmente, el mensaje de intercambio de la llave del servidor puede ser enviado si es requerido. El servidor puede requerir un certificado del cliente (ó conseguir uno de algún servicio de distribución de certificados). Ahora el servidor enviara el mensaje *hello* hecho, indicando que la fase de mensajes *hello* del *Handshake* esta completa.

El servidor entonces esperara la respuesta del cliente. Si el servidor ha enviado un mensaje de requerimiento de certificado, el cliente debe enviarlo. El mensaje de intercambio de llaves del cliente es ahora enviado si el certificado del cliente no contiene suficientes datos para intercambio de llaves, o si no fue enviado del todo. El contenido del mensaje dependerá del algoritmo de llave publica seleccionado entre el cliente *hello* y el servidor *hello*. Si el cliente es autenticado usando un certificado con capacidad de firmado (por ejemplo RSA), un mensaje de verificación de firma digital es enviado para verificar explícitamente el certificado.

En este punto, el mensaje *ChangeCipherSpec* es enviado por el cliente, y este copia el *CipherSpec* pendiente en el actual de escritura. El cliente envía entonces inmediatamente el mensaje de finalización bajo los nuevos algoritmos, llaves y secretos. Desde ahora el *CipherSpec* es configurado a 1 en los mensajes. Cuando el servidor recibe el mensaje *ChangeCipherSpec* también copia el *ChipherSpec* pendiente en el *ChipherSpec* actual de lectura. En respuesta, el servidor enviara su propio mensaje *ChangeCipherSpec* , configura su *ChipherSpec* actual de escrtura en un *ChipherSpec* pendiente, y envía su propio mensaje de finalización bajo el nuevo *CipherSpec*. En este

punto, el *HandShake* esta completo y el cliente y el servidor pueden comenzar a intercambiar los datos de la capa de aplicación. La figura 18 resume lo anterior (el * significa situaciones opcionales o que dependen de mensajes que no siempre son enviados).

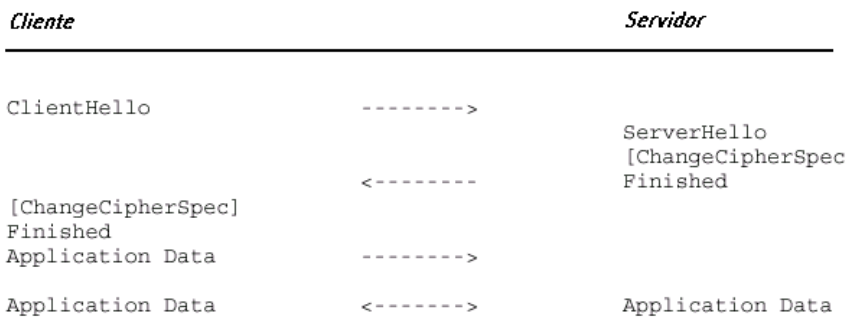
Figura 18: Flujo de mensajes de una negociación completa



Cuando el cliente y el servidor deciden resumir una sesión previa segura en lugar de negociar nuevos parámetros de seguridad, el flujo de mensajes es el siguiente: El cliente envía un *ClientHello* usando un identificador de sesión, de la sesión segura a ser resumida. El servidor entonces comprueba la sesión segura almacenada pensando en una correspondencia. Si esta es encontrada, y es servidor esta dispuesto a reestablecer una conexión segura bajo una sesión segura especifica, enviara un *ServerHello* con el mismo valor de identificador de sesión. En este punto el servidor debe enviar el mensaje *ChangeCipherSpec* y proseguir con el mensaje de finalización, al cual el cliente debe responder con su propio

ChangeCipherSpec y su propio mensaje de finalización. Una vez que el reestablecimiento esta completo, el cliente y el servidor pueden iniciar el intercambio de datos de la capa de aplicación. Adicionalmente, una nueva rata de refresco de llave puede ser negociada durante el *Handshake* abreviado para una conexión segura. Si el identificador no es encontrado, el servidor genera un nuevo identificador de sesion y el cliente TLS y el servidor ejecutan un *Handshake* completo. La figura 19 ilustra lo anterior.

Figura 19: Flujo de mensajes de una negociación abreviada



Un *Handshake* de secreto compartido significa que la nueva sesión segura esta basada en un secreto compartido ya implantado en ambas extremidades. En este caso el secreto compartido es usado como el secreto pre-maestro y el intercambio de la llave *Shared_Secret* es requerido por el cliente en el *ClientHello*. El flujo de mensajes es similar al del *Handshake* abreviado en la figura 19, excepto que el *ClientHello.session_*id esta vació. Tal y como otras sesiones seguras creadas en otro tipo de *Handshakes*, una sesion segura creada por un *Handshake* de secreto compartido puede ser reanudada.

Otra variación es que el servidor después de recibir el *ClientHello*, puede recuperar el certificado del cliente usando un certificado de un servicio de distribución de certificados o uno de sus propias fuentes. El proceso es ilustrado en la figura 20.

Figura 20: Flujo de mensajes de una negociación optimizada completa

5. EL FUTURO DE LOS PROTOCOLOS Y LAS PLATAFORMAS DE SERVICIOS EN LÍNEA

Este capítulo presenta el contexto en el que nace WAP, sus previsiones de desarrollo y crecimiento y el papel de cada uno de los agentes implicados en este estándar. Entender cuáles son sus ventajas competitivas y aquello hacia lo que evolucionará, ya sea en el desarrollo de sus mejoras más inmediatas, ya sea en la integración de otras soluciones existentes, resulta fundamental para situarse en el panorama actual.

5.1 Factores de evolución del estándar WAP

5.1.1 La convergencia

Como hemos anunciado, asistimos en estos días a dos eventos de distinta naturaleza. Por un lado, el "boom" de Internet, que cuenta ya con millones de usuarios en el mundo y permite disponer de una cantidad ingente de información como jamás antes. El otro gran "boom" es la convergencia Fijo-Móvil, que pone de manifiesto el auge de la venta de móviles en el mercado.

Resulta meritoria la implantación de Internet si se tiene en cuenta que el acceso requiere la instalación de un ordenador personal, cuyo coste no está subvencionado y es relativamente elevado. Sin embargo, conoce ciertas limitaciones como es la ausencia de movilidad. En efecto, el usuario está anclado a una línea fija de teléfono. A la necesidad de movilidad responde la telefonía móvil. Hasta hoy prestaba servicios de voz, de mensajería y algunos conocidos como servicios de valor añadido. Sin embargo era vital para los

usuarios el disponer de servicios de características similares a aquellos de los que disponían en sus hogares.

En cuanto al ancho de banda del que puede disponer el usuario, sigue siendo mayor el ofertado por el cable. Cabe destacar la llegada hasta los hogares de una oferta de servicios de cable que garantizan grandes velocidades de transmisión, dando entrada a todo tipo de servicios multimedia en tiempo real.

Sin embargo, si bien las expectativas de disponibilidad de ancho de banda en el móvil no hablan de rebasar estas cifras, sí anuncian por lo menos una convergencia de magnitudes, en la generaciones venideras (GPRS, UMTS).

5.1.2 Perspectivas

La segunda generación de móviles, GSM, ya permite el envío de información a una velocidad de 9.6 Kbps. que resulta ser insuficiente para aplicaciones multimedia en la Red. Está previsto el relevo tecnológico con la tercera generación de móviles conocida como UMTS.

Se prevé que permitirá el acceso a la información a una velocidad cercana a los 2 Mbps. y su implantación se realizará a finales de 2002. Requiere de una tecnología totalmente distinta en su esencia a la de GSM por lo que se han establecido pautas de transición. En principio, los usuarios pueden transferir datos teóricamente hasta los 144 Kbps gracias a GPRS desde finales del año 2000, y a 384 Kbps.l con EDGE desde el año 2001. GPRS (*General Packet Radio Services*) es una reutilización tecnológica de GSM. Cuenta con las mismas infraestructuras, con la mejora en el acceso al canal radio. Es por ello un paso intermedio idóneo, antes de UMTS.

A pesar de que estas tecnologías no alcanzan la velocidad de transmisión de UMTS, sí introducen el concepto de conmutación de paquetes del que no disponía GSM.

5.1.3 Agentes

5.1.3.1 Los operadores

WAP permite a los operadores con una baja inversión, aumentar sus beneficios, además de incrementar su número de abonados mediante la provisión de servicios valor añadido. De igual modo, resulta sencillo añadir de forma rápida y sencilla nuevas aplicaciones sin la necesidad de grandes infraestructuras adicionales o modificaciones en el terminal. Se incentivará así la competencia entre operadores a través de la diferenciación y personalización de servicios. Ello asentará la ventaja competitiva de los operadores, asegurando la fidelidad de los usuarios.

5.1.3.2 Los fabricantes

Se prevén cambios significativos en los fabricantes de terminales con la entrada de WAP. Así, veremos cambiar el aspecto externo de los aparatos, en su tamaño, su pantalla, sus botones y la autonomía y peso de sus baterías. Su capacidad de procesado también cambiará indefectiblemente. Las PDA´s (agendas personales), jugarán un papel fundamental en este escenario. En el Foro Wap se encuentra el 75% de los fabricantes de equipos que se compromete al desarrollo de terminales con nuevas y mejores capacidades. Un buen ejemplo de ello es la incorporación del *Microbrowser*.

Se trata de un programa que traducirá el código binario generado en WML almacenado en la Red a un contenido interactivo para el usuario.

5.1.3.3 Los contenidos

Obviamente, si la carrera hacia la provisión de servicios en WAP es importante para todos los operadores de telefonía, no lo es menos para los proveedores de contenidos. Les resulta de gran atractivo estar accesibles al usuario en todo momento en cualquier lugar. De ahí que tengan conciencia de la necesidad de desarrollar sus contenidos en formato WML y de acogerse a los menús existentes donde se agrupan todos los líderes del sector, como por ejemplo en *e-moción*. En este último caso aceptan respetar unas normas de presentación y de calidad que impone el operador. Resulta fundamental para ellos el posicionarse en un mercado al que terminarán accediendo la gran mayoría de los ciudadanos, sobre todo a la vista de los resultados de crecimiento. Con poco trabajo pueden extender su modelo de negocio de forma que incluya a todos los usuarios de móviles.

5.1.3.4 Los usuarios

Los usuarios finales contarán con un acceso sencillo y seguro a contenidos de Internet así como a servicios tales como mensajería, banca, noticias, ocio y otros muchos a través de su terminal. También tendrán acceso mediante WAP a las *Intranet* y a las bases de datos corporativas.

Además, debido al gran número de fabricantes, proveedores de contenidos y operadores garantizarán el carácter plural y el precio competitivo del servicio.

5.1.4 Las aplicaciones

El usuario tendrá a su alcance un amplio abanico de servicios, pudiendo estar reunidos en un menú. Abarcará desde el ocio hasta la información bursátil, pasando por deportes, comercio electrónico, aplicaciones Interactivas. Esta diversidad de servicios se obtiene gracias a las especificaciones del Foro WAP. Se buscó la creación de un estándar lo más abierto y flexible posible. En ello, es una réplica del modelo TCP/IP de Internet adaptada a las restricciones que impone un acceso radio. Por ejemplo, WML/WMLS (solución de seguridad de WML) recuerdan la simplicidad de HTML en su programación. La información está disponible con el método *PUSH/PULL* con la posibilidad por parte de los usuarios de interactuar por voz o datos.

Se espera que el navegador no se imponga como único software como uso de WAP, debido a las restricciones de tiempos y procesado.

Para promover el éxito de WAP en el entorno móvil, las aplicaciones y servicios en tiempo real requieren pequeñas cantidades de información esencial,. Valores bursátiles, noticias, el tiempo, viajes, resultados deportivos son algunas de las áreas en las que WAP responde a las necesidades de los usuarios. La filosofía básica de WAP consiste en tomar los servicios

habituales de Internet y adaptarlos al entorno móvil añadiendo valor al servicio básico existente.

Todos los operadores competirán por dar los más sofisticados e innovadores servicios de valor añadido. A su vez, es cada día mayor la seguridad y la fiabilidad en los protocolos WAP por lo que conocerá un gran auge el comercio electrónico con este soporte. Los usuarios llevarán a cabo sus transacciones, sus compras y sus movimientos financieros mediante el teléfono móvil. El entorno será entonces muy lucrativo, atrayendo a más inversores y proveedores de servicios. Se prevé pues un crecimiento sostenido en los años venideros.

5.2 Introducción a las nuevas tecnologías de servicios y aplicaciones

El tremendo interés y el gran desarrollo que ha promovido el WAP en el mercado de la transmisión inalámbrica de datos está obligando a los operadores, los creadores de infraestructuras y los fabricantes de terminales a colaborar de manera conjunta. La concordia que se vive en el WAP Forum ha permitido dejar atrás la diversidad de protocolos y plataformas para desarrollar un entorno común para el desarrollo de servicios avanzados de telefonía y acceso a Internet para el mercado inalámbrico.

Aunque el ancho de banda limita mucho las posibilidades de esta tecnología, todos los sectores de la industria -desde los desarrolladores de contenidos hasta los operadores- están trabajando más que nunca para explotar la amplia gama de oportunidades que les ofrece el WAP. Una de las tecnologías que más promete en combinación con el WAP es la *General packet radio Services* (GPRS) que gracias a un ancho de banda de 115 Kbps.

haría realidad la transmisión de contenidos multimedia. La GPRS es sólo un pequeño avance de lo que nos depara la tercera generación de telefonía móvil (3G), actualmente en proceso de estandarización. Si hacemos caso a sus impulsores, la capacidad para transmitir datos de esta nueva tecnología es tal que en combinación con el WAP sería posible mantener una videoconferencia con total normalidad desde un simple teléfono celular.

5.2.1 Los competidores de WAP

La competencia en protocolos WAP puede venir de numerosos frentes:

- *SIM Toolkit:* Es un método de desarrollo de aplicaciones que gira alrededor de la Tarjeta Inteligente (*Smartcard*). Permite la instalación de aplicaciones sobre GSM y SMS así como USSD. Las aplicaciones se distribuyen en Tarjetas Inteligentes. Su uso está bastante extendido. Cuatro de los mayores vendedores de Tarjetas Inteligentes son miembros del Foro WAP y existen en la actualidad grupos de trabajo cuyo objetivo es el desarrollo de interfaces entre las dos tecnologías (WAP, SIM-T). El papel que desempeñaran las Tarjetas Inteligentes en el futuro parece pasar por la seguridad y la personalización del usuario/operador en WAP.

- Windows CE: Es un sistema operativo multitarea con procesado en tiempo real, propiedad de Microsoft diseñado para terminales con restricciones de espacio y capacidad. Tiene funcionalidad multimedia y navegación por Internet con un servidor Web incorporado así como un *browser*.

- *Javacard-Sun Microsystems JavaCard* así como *JavaPhone*™ API: Que incorpora una máquina virtual Java™ (KVM). Se fabrican ya terminales

celulares móviles que podrán descargar funciones y características extra de Internet evitando así que los usuarios tengan que cambiar de aparatos con la frecuencia que se hacía hasta ahora cuando querían mejorar las prestaciones de los suyos y evitando una conexión permanente con el servidor para manejo de aplicaciones. Comprende ya un software personalizable según los recursos que se deseen emplear. Es un diseño seguro con un gran número de aplicaciones, copias de seguridad, que conserva el micro-browser, con conexión a PC. Rompe con el concepto de Cliente-Servidor.

- MexE: Es un estándar de entorno del terminal móvil de ejecución de aplicaciones por parte del operador o de los proveedores de servicios. Soporta una Máquina Virtual de Java™ en el móvil del usuario. La idea es la misma que en el *Javaphone*™: descargar la aplicación una sola vez, ejecutarla después cuando se desee, donde se desee. Se prevé que WAP termine asociándose a MExE.

5.2.2 La respuesta de WAP

WAP ofrece frente a estos otros sistemas:

- Un estándar abierto, no propietario.
- Independiente de la tecnología de Red sobre la que se usa.
- Un mecanismo de transporte optimizado para operadores de comunicaciones móviles.
- Ejecución de aplicaciones desde el servidor en contraposición a la filosofía de *softwares* instalados.
- Paradigma esencialmente idéntico al de Internet con su HTML.

El cuidadoso diseño de las especificaciones WAP, prestando atención a adaptarse a las prestaciones de los productos actuales, a las economías de escala en un mercado con una fuerte competitividad, garantiza una provisión de servicios a precios asequibles

Hoy por hoy, las áreas de interés cubren la seguridad extremo a extremo, interfaces de tarjeta inteligente, protocolos de transporte orientado a conexión, almacenamiento persistente, interfaces y tarificación, y tecnología PUSH/PULL.

Por otra parte, otro punto que suscita gran interés en el Foro WAP es la evolución hacia el soporte de servicios móviles multimedia. Tanto la versión 1.1 como la 1.2 de WAP son protocolos abiertos que permiten el transporte de diversos tipos de contenidos multimedia.

Sin embargo, ciertas aplicaciones basadas en flujos de información, exigen mejores prestaciones del sistema. A esta necesidad pretende responder GPRS, EDGE y más tarde UMTS. Se puede predecir un incesante aumento del ancho de banda requerido por las aplicaciones solicitadas por parte de los usuarios. Por ello se deberá optimizar el uso tanto de los terminales como de los recursos de la Red. Si se consigue que WAP tenga un gran éxito en los mercados de masa en la generación actual con sus mejoras (GPRS), quizá las redes de tercera generación únicamente necesiten mayor capacidad partiendo de las mismas aplicaciones.

Cabe señalar que WAP es en sí un protocolo que abre la puerta al acceso a Internet desde terminales móviles, por lo que, está en constante

evolución. En los próximos años sufrirá modificaciones debido a las capacidades que aportarán las futuras tecnologías de red (GPRS, UMTS). Esto dará lugar al nacimiento de nuevas versiones del estándar, de modo que pueda responder a las necesidades del mercado.

Lo que se conservará sin duda alguna de la filosofía sobre la que se ha diseñado WAP es que sea transparente a la tecnología que lo soporta, es decir, que los medios de transmisión que se utilicen sólo impliquen mejoras a la prestación del servicio, y no restricciones excluyentes entre ellas.

5.3 GRPS, caminando hacia la tercera generación

GPRS son las siglas de *General Packet Radio Service*, o Servicio General de Radio por Paquetes. Este nuevo sistema, que permite estar siempre conectado, se integra en la estructura de la red GSM mejorándola y aumentando la velocidad de transmisión de 64 a 115 Kbps., entre 5 y 11 veces superior a la de WAP. GPRS elimina el coste por conexión, facturándose por información descargados. En una primera fase, la velocidad sólo alcanzará los 50 Kbps, mientras que la capacidad del terminal será de 20 Kbph.

Esta nueva tecnología permite desdoblar la transmisión de voz y datos en diferentes canales que transmiten de forma paralela, permitiendo mantener conversaciones sin cortar la transmisión de datos. Cuando se trata de datos se estable un comunicación permanente mientras el terminal está conectado, lo que permite la transmisión continúa de la información a mayor velocidad. La información viaja por paquetes en lugar de circuitos conmutados como sucede en GSM, donde la voz se envía por un canal siempre abierto. En GPRS se puede elegir entre varios canales, de forma similar a como se

realiza en Internet. El aumento de la velocidad se produce porque los datos se comprimen y se envían a intervalos regulares, llamado conmutación por paquetes, lo que aprovecha mejor la banda de frecuencia.

La mayor ventaja de GPRS no es la tecnología en si misma sino los servicios que facilita. Los terminales de este nuevo sistema permiten personalizar funciones, desarrollar juegos interactivos, e incorporan aplicaciones para el intercambio de mensajes y correos electrónicos, a los cuales se podrá acceder directamente sin la necesidad de conectarse a Internet. Las pantallas, que serán de un tamaño mayor, serán táctiles, de alta resolución, con *zoom* e íconos que se activen de manera intuitiva pulsando sobre ellos con un puntero. Incorporan además una ranura para introducir la tarjeta de crédito con *chip* que facilitará las transacciones electrónicas más seguras. Con la tecnología GPRS se da un paso hacia la localización geográfica, en función de donde se encuentre el usuario, la operadora le puede ofrecer mayor información de la zona.

Los terminales serán de cinco tipos a corto plazo, en función del uso que le vaya a dar el usuario. Móviles similares a los actuales, con visor y resolución cada vez mayor, permitirán el uso de información escrita o gráfica de forma resumida. Terminales tipo agenda electrónica, con funciones mixtas de voz y datos, y pantallas de mayor tamaño y capacidad gráfica. Terminales tipo ordenador personal de mano (PDA) con pantalla plana de mayor formato y gran capacidad gráfica. Ordenadores portátiles que utilicen para la conexión inalámbrica un teléfono móvil GPRS. Y por último, dispositivos diversos con comunicación móvil y funciones especiales como sistemas de navegación para automóviles y tarjetas de comunicación inalámbrica en máquinas autoservicio.

5.4 UMTS, la cima del Internet móvil

Universal Mobile Telecommunications Systems es la tercera generación de móviles en toda su esencia. Su velocidad se sitúa entre 40 y 208 veces más que la de WAP.

Esta nueva tecnología permitirá todo tipo de comunicaciones, como videoconferencia y servicios multimedia, transmisión de imágenes de video en movimiento y sonido de alta fidelidad por redes móviles, correo electrónico, operaciones bancarias, publicidad personalizada, almacenamiento de información empresarial e incluso activación a distancia de ordenadores y electrodomésticos con tecnología *Bluetooth*. Los principales operadores están firmando acuerdos con los proveedores de contenidos para ofrecer una amplía gama de artículos y servicios a los consumidores

Las ventajas que aporta este nuevo sistema son velocidad en la transmisión y seguridad. El precio pagado en algunos países por las licencias de este sistema apunta que se trata de un mercado con un gran potencial, un gran negocio donde las operadoras de telecomunicaciones verán compensadas las inversiones realizadas.

Las previsiones del sector apuntan que en el año 2003 habrá 450 millones de usuarios de móviles, de los cuales 100 millones serán usuarios europeos de UMTS. Los nuevos aparatos de telefonía móvil, disponibles de manera generalizada en 2002, dispondrán de una pantalla más grande, en color e interactiva, y mucho de ellos incorporarán tecnología *Bluetooth* para que puedan conectarse sin hilos a otros aparatos, como por ejemplo controlar los electrodomésticos desde el teléfono. El teléfono móvil funciona comunicándose con la estación base más cercana, lo cual implica que la compañía de telefonía móvil sabe aproximadamente dónde se encuentra el

teléfono. Al saber donde se encuentra el usuario, la operadora podrá recomendarle tiendas o servicios cercanos que se adapten a sus necesidades.

5.5 i-Mode

5.5.1 ¿Qué es i-Mode?

Desde hace ya un par de años, Japón viene experimentando la revolución de la Internet móvil. Tanto es así, que cerca de 28 millones de japoneses acceden a la red en movilidad, gracias al i-*mode* del monopolio NTT DoCoMo, la operadora de comunicaciones móviles más grande de Japón. En febrero de 1999, la compañía lanzó al mercado el servicio "i-*mode*" (en el que curiosamente, la *i* no tiene ningún significado concreto, ya que en japonés, puede entenderse de muchas maneras diferentes), que permitía establecer una conexión continua con Internet a través de un teléfono móvil.

El i-*mode* es un sistema de transmisión que permite acceder a todo tipo de servicios basados en Internet, como banca *on-line*, búsqueda de información turística o de ocio sobre una ciudad, descarga de logos y melodías, etc. desde un teléfono móvil compatible.

El i-*mode* utiliza la transmisión de datos por paquetes (a 9600bps), lo que permite tarificar por la cantidad de datos transmitidos y recibidos en vez de por el tiempo de conexión.

5.5.2 ¿Cómo funciona?

Mientras el sistema de voz se conmuta por circuitos (es decir, es necesario marcar para establecer la conexión), el i-*mode* utiliza tecnología de conmutación por paquetes, lo que significa que, en principio, está "siempre conectado", a condición de que el usuario tenga cobertura. Al seleccionar un elemento i-*mode* en el menú del terminal, los datos suelen descargarse inmediatamente. Se evita la necesidad de marcar para establecer la conexión. Sin embargo, sí que hay que esperar hasta que los datos llegan a la terminal.

Por ponerlo de algún modo, este retraso es similar al que se experimenta en un ordenador personal cuando se hace clic en un enlace de Internet, o tras introducir una dirección en el navegador. Además, pueden producirse otros retrasos si el volumen de la información a descargar es muy elevado, o si se produce una sobrecarga en la red. El centro de transmisiones i-mode, controlado por la compañía japonesa DoCoMo, centraliza y gestiona todas las transmisiones realizadas a y desde terminales de este tipo.

Así, para contactar con un banco compatible con esta tecnología, la información circula por una red de transmisión de paquetes hasta el centro i-*mode* de DoCoMo, desde donde se establece una conexión por línea alquilada con el banco.

El mismo procedimiento se utiliza para acceder a proveedores de servicio de información i-*mode* u otros sitios web compatibles con esta tecnología, con la salvedad de que la comunicación entre el centro i-*mode* y estos sitios se hace directamente por Internet, como se puede apreciar en la figura 21.

Figura 21: Ilustración del mecanismo de comunicación entre el centro i-*mode* y un sitio *web* compatible.

5.5.3 ¿Qué diferencia hay entre WAP y i-*Mode*?

No es fácil comparar i-*mode* y WAP. En Japón, son dos servicios que compiten entre sí, algo que se espera acabe siendo una pauta generalizada a nivel global. Tanto el i-*mode* como el WAP son sistemas complejos y en realidad, lo único que se puede hacer es comparar sus funcionalidades, modelos de negocio, precios, marketing, etc.

Existen varias diferencias fundamentales en el modo de implementación, comercialización y en los precios de los servicios basados en i-mode o en WAP. Por ejemplo, el i-*mode* utiliza el cHTML, que al ser una derivación del HTML, resulta relativamente más sencillo de aprender para los desarrolladores de *webs* que el lenguaje "wml" del WAP.

Otra diferencia consiste en que, en la actualidad, el i-*mode* se implementa en Japón mediante un sistema de conmutación por paquetes, que

en principio está "siempre encendido", mientras que los sistemas WAP europeos utilizan la conmutación por circuitos.

La tarificación es otro factor que diferencia a los dos sistemas, un usuario de i-*mode* paga por la cantidad de información que descarga más las cuotas aplicables de servicio, mientras que los servicios WAP se tarifican por tiempo de conexión. Cabe aclarar que la conmutación por paquetes o por circuitos no es más que una diferencia técnica del sistema de telecomunicaciones desde el que se presta cada uno de los servicios, y no tiene nada que ver en principio con los estándares i-*mode* y WAP de por sí.

En principio, las páginas web programadas en i-*mode* o en WAP pueden proporcionarse sobre ambos tipos de sistemas.

5.5.4 ¿La clave del éxito?

En gran medida, el éxito del i-*mode* viene porque la operadora NTT-DoCoMo ha facilitado a los desarrolladores la creación de sitios web para esta tecnología, a continuación se presentan algunos puntos clave que pueden aclarar un poco el éxito del i-*mode* en Japón:

- Precio de venta al público relativamente reducido de los terminales con i-*mode*.

- 60 millones de abonados a la telefonía móvil.

- Penetración de ordenadores personales domésticos relativamente baja, precios de acceso a nodo local muy elevados.
- La tarificación por paquetes transmitidos y recibidos permite ofrecer el i-*mode* a precios relativamente reducidos.

- Sistema eficaz de micro facturación dentro de la factura telefónica, lo que facilita que los abonados paguen por sitios de valor añadido, y ayuda a que los *webmasters* vendan información a los usuarios.

- Marketing eficiente.

- El correo electrónico vuelve a ser la aplicación principal, como al principio de la expansión de Internet.

- Utiliza cHTML, lo que facilita la labor de creación de contenidos no sólo de los desarrolladores, sino también de los usuarios normales.

- Crecimiento exponencial de contenidos.

- Enlaces a sitios web asociados, lo que aumenta las posibilidades de vender contenidos y servicios.

5.5.5 Proveedores de servicios

Muchas son las compañías que han abierto filiales de i-*mode* ante el enorme éxito de esta tecnología en Japón. Desde bancos como el *Fukoka Bank*, que ofrece transferencias y estados de cuentas, agencias de valores como *Daiwa Securities* (Compra y venta de acciones, información bursátil actualizada, etc.), compañías de información sobre tarjetas de crédito como

DC *Card, Sumitomo* VISA *Card*, Compañías de seguros, como *Nippon Life Insurance*, Líneas aéreas, como *Japan Airlines*, agencias de viajes, agencias de reserva de entradas para espectáculos, de noticias e información, aplicaciones de bases de datos (información telefónica, diccionarios, guías de restaurantes), ocio, etc.

En definitiva, todo lo que existe en Internet existe o acabará existiendo en i-mode, según sea la demanda de sus usuarios. La sencillez de crear una página con esta tecnología facilita enormemente la entrada de distintos sectores.

5.5.6 Seguridad en i-*Mode*

Dado que el i-*mode* es básicamente comercio móvil (M-*commerce*), la seguridad es un factor esencial para su éxito.

La seguridad del i-*mode* se ha estructurado en varios elementos distintos para reforzarla: la seguridad del enlace de radio entre el terminal i-*mode* y la estación base móvil, que utiliza protocolos propietarios y una codificación controlada por DoCoMo; la seguridad de la conexión pública transparente a Internet entre los sitios i-*mode* y el terminal en la capa cHTML; la seguridad de las redes privadas; la seguridad de los enlaces de las redes privadas entre el centro i-*mode* y los proveedores de servicios especiales, como los bancos; y por último, la seguridad basada en contraseñas.

5.5.7 Hacia la tercera generación

En la actualidad, la velocidad de descarga de los datos i-*mode* es de 9,6 kbit/s, es decir, unas 6 veces más lenta que la de una conexión fija de RDSI. Sin embargo, en realidad los caudales de datos son mucho más lentos, especialmente en las áreas más pobladas, o cuando la red está "congestionada". Con la instalación de nuevo hardware y software de red para la tercera generación móvil (3G) se alcanzarán velocidades que podrán ser hasta 200 veces más rápidas.

DoCoMo está probando desde la primavera de 2001 un servicio de red de 3G denominado FOMA. Unos 4000 usuarios se han prestado para estas pruebas, aún pagando por determinados servicios. Las previsiones apuntan a que Japón será el primer país del mundo en introducir la 3G, dados los retrasos a los que se han tenido que enfrentar las compañías europeas que planeaban tenerlo funcionando para el presente año.

Se espera que los servicios de esta red se desplieguen a nivel general lo mas rápidamente posible, empezando en la zona de Tokio y ampliándose progresivamente hasta cubrir la mayoría de las áreas más pobladas de Japón.

La red proporciona unas tasas máximas de transmisión de 384 kbps en sentido descendente y 64 Kbps. en sentido ascendente, aunque de momento no se hayan podido alcanzar en el servicio en pruebas. La tasa media de descarga de servicio se acerca más bien a los 200 Kbps. en condiciones óptimas, y puede reducirse en gran medida por sobrecargas de la red o bajas coberturas.

Durante las pruebas, se han proporcionado tres tipos de terminales a los usuarios: terminales de voz e i-*mode*; terminales de voz, i-*mode* y vídeo, que incluyen una vídeo cámara y permiten realizar videoconferencias móviles; y, por último, una tarjeta PC para establecer transmisiones de datos a través de una PC sobre la red FOMA.

5.6 Bluetooth

Bluetooth es una tecnología de radio que se puede incorporar a teléfonos, ordenadores y otros dispositivos para conectarlos entre sí sin necesidad de cables, permitiendo transferir voz, datos e incluso imagen entre dispositivos que tengan incorporando esta tecnología.

Ericsson ha sido el impulsor de esta tecnología diseñada para transmitir datos a una velocidad por encima de 1 megabit con un alcance de hasta diez metros, extendiéndose a 100 metros con un amplificador opcional. Por ejemplo, una agenda electrónica portátil o un teléfono móvil puede ser sincronizada automáticamente con un ordenador antes de entrar en la oficina. El primer producto en comercializarse es un auricular de manos libres que se comunica inalámbricamente con un teléfono móvil.

Ericsson, junto con Nokia, Motorola, 3Com, IBM, Lucent, Microsoft, Toshiba e Intel formaron en 1998 el Grupo de Interés Especial Bluetooth para desarrollar este nuevo sistema. Un año más tarde, 650 compañías firmaron el Acuerdo de Suscriptores de Bluetooth, y actualmente son 1900 compañías de tecnología las que respaldan este sistema.

Ericsson estima que más de 200 millones de aparatos electrónicos incorporarán esta tecnología en el año 2002. Mientras que un año más tarde, habrá 600 millones de productos en todo el mundo, el triple que el año anterior

5.7 WAP 2.0

Ericsson, Nokia y Motorola anunciaron su apoyo a la nueva version del estandar WAP, el WAP 2.0 desarrollado por el WAP ForumTH. Las empresas también expresaron su intención de desarrollar productos, contenidos y servicios basados en este nuevo estándar.

La nueva generación de la especificación WAP en conjunto con unidades avanzadas, y otros equipos móviles aseguran un ambiente de desarrollo mejor para los servicios avanzados móviles. Basado en estándares de Internet bien establecidos incluyendo TCP y HTTP, así como los componentes necesarios específicamente adaptados para ambientes inalámbricos, WAP 2.0 suministrará una poderosa pero simple herramienta para desarrollar fácilmente una multitud de servicios nuevos.

WAP 2.0 ha adoptado como la base de su lenguaje el XHTML Basic. XHTML, desarrollado por el consorcio de la Red Mundial (W3C), es el lenguaje que será empleado para crear todos los contenidos, intencionado para el Internet fijo o el mundo de teléfonos móviles. Al cerrar la brecha entre el contenido fijo y móvil, XHTML acelera rápidamente el rango bajo el cual los servicios pueden ser creados y mejorados para los usuarios.

Otros estándares de Internet que han sido adoptados en WAP 2.0 incluyen *Cascading Style Sheets* (CSS), *Transport Layer Security* (TLS), HTTP y TCP. Al especificar el mejor manejo de los estándares en un ambiente inalámbrico, se alcanza una mejor experiencia del usuario. Por ello serán más diversos tanto el contenido como los servicios multimedia que estarán disponibles en redes 2.5G/3G, y estarán basados en estos estándares y por ello se integrarán con la tecnología WAP.

La incursión de WAP 2.0 incluye el primer lanzamiento de Servicios de Mensajes Multimedia (MMS), un servicio desarrollado conjuntamente con 3GPP, que permite a los usuarios enviar mensajes multimedia, combinando sonidos con imágenes y texto, de manera conjunta en un proceso parecido a los mensajes de texto SMS.

Adicionalmente, WAP 2.0 envuelve el WAP Push, que puede ser empleado para servicios en línea, en donde es importante para los usuarios el recibir la información en el momento de interés, en vez de ser forzado para que activamente se busque información.

Debido al factor de que WAP 2.0 es un estándar abierto e interoperable, será un componente muy valioso en ofertas futuras de servicios móviles. Las empresas creen que la Asociación GSM se beneficiará al incluir WAP 2.0 como una de sus piedras angulares al definir versiones futuras de sus iniciativas de Servicios-M.

"Ericsson esta orgulloso de haber contribuido de principio a fin con el nuevo WAP 2.0", dijo Lars Boman, Director de *Ericsson Mobile Internet Applications*. "La introducción de estándares de Internet, tales como XHTML, asegurarán la compatibilidad con contenido disponible en Internet. Conjuntamente con nuevas funcionalidades, tales como mensajes de

multimedia, abre nuevas posibilidades para operadores y desarrolladores de contenido. Nuestro enfoque es siempre crear productos amigables al usuario que faciliten estos nuevos servicios".

"El trabajo del Forum WAP nos ha traído un buen ajuste de estándares sólidos, de los cuales podemos empezar a desarrollar la siguiente generación de servicios móviles. Tecnologías como XHTML y las "*Cascading Style Sheets*" serán claves en entregar satisfacción a los usuarios, lo que es el ingrediente principal para el éxito en el mercado", dijo Aage Snorgaard, Vicepresidente de *Nokia Mobile Phones*. "En conjunto con los demás líderes de la industria, estamos comprometidos a cooperar y crear estándares abiertos con el objetivo de abrir el mercado para aplicaciones móviles y sus respectivos servicios".

5.8 Principios básicos del Comercio Móvil *(M-Commerce)*

M-*Commerce* puede definirse como el segmento de mercado que incluye a empresas preveedoras de tecnología para servicios móviles con comunicación inalámbrica dirigidos al usuario final.

Toda actividad comercial asociada a servicios móviles inalámbricos para el monitoreo de vehículos, equipos, personas, bienes y procesos pertenece al mundo de M-Commerce.

Dicho lo anterior, M-*Commerce* es el comercio electrónico efectuado desde un dispositivo móvil en una red móvil, y dado que nos encontramos en un momento en el que todo el mundo dispone de un teléfono móvil, incluidos grupos que en principio no fueron considerados como potenciales usuarios de este tipo de dispositivos, una pregunta importante es: ¿la utilidad de un

teléfono móvil o de un PDA -*Personal Digital Assistant*- se reduce a hablar a través del teléfono o a ser una mera agenda?

Aunque estamos en un momento que si ya de por sí se presenta poco alentador para el e-*commerce*, menos aún se presenta para el m-*commerce* hasta que no aparezca una tecnología lo suficientemente potente, segura y sobre todo accesible para el público en general.

El M-*Commerce* debe ser potente porque lo que busca cualquier usuario es valor añadido, es decir, que nos ayude a hacer gestiones tanto sencillas -como consultar el tráfico cada mañana según la ruta que solemos seguir-, como de mayor complejidad -hacer la compra del supermercado con el móvil desde el trabajo porque vemos que nos quedan unas cuantas horas de trabajo, no tenemos Internet y necesitamos urgentemente hacer esas compras de última hora-. Es época de demanda de servicios. Cada vez tenemos menos tiempo y queremos optimizarlo al máximo. ¿Por qué no aprovecharnos de las nuevas tecnologías?

Un ejemplo práctico podría ser el caso de un directivo al que le comunican por móvil que debe de viajar urgentemente a Amsterdam. A través del mismo móvil accede a la página de un operador de viajes o compañía aérea y adquiere los billetes en el mismo momento. Todas las facilidades que nos puedan proporcionar a precios asequibles son bienvenidas por todos. Ahí radica el éxito del comercio móvil (*m-commerce*), es decir, en entender las verdaderas necesidades de los clientes y dónde perciben éstos el valor añadido.

En este ejemplo he hecho expresa referencia a una operación de comercio móvil transaccional, pero no debemos olvidarnos del comercio móvil a nivel de comunicación. Esto es, las posibilidades de patrocinar o incluir publicidad en aquellas páginas a las que los usuarios puedan acceder a través del móvil supone una nueva herramienta de marketing, ya que permite personalizar los mensajes según tipo de público objetivo, promocionar las ofertas de la tienda tradicional, realizar campañas de comunicación con menores costos y cuya eficiencia es medible con datos estadísticos, etc.

Asimismo, el comercio móvil debe ser seguro. Por ejemplo, ¿que pasaría si algún *hacker* interfiriese una comunicación y nos robara información que es confidencial? Si no fuese segura la comunicación, si la tranquilidad que se tenga al utilizar una tecnología es nula, el éxito de dicha tecnología será muy poco.

O el caso más típico, si realizo un pago por medio del móvil, ¿podría alguien captar esos datos personales y confidenciales y robarnos el dinero que hay en nuestras cuentas bancarias? Una persona que conozca de la materia seguramente tendrá miedo a dar tus datos, pero si por el contrario, se conoce acerca del nivel de desarrollo de la seguridad en Internet y se sabe distinguir cuándo se encuentras en un sitio web seguro o no, no debería tenerse problema alguno en dar dichos datos. Claro que existe un peligro real, pero éste no es mayor que el que se corre cuando se proporcionan los datos de la tarjeta de crédito por teléfono.

Por otro lado, debe ser accesible para el público en general. Ahora bien, esto requeriría de una tecnología que se implantase con cierta estabilidad en el

mercado para que, a través de una mayor difusión, se logren economías de escala que redundaran en unos menores precios para los usuarios finales.

5.9 Las posibles mejoras del estándar WAP

WAP por su carácter de estándar abierto e intencionadamente dinámico, ofrece un sinfín de posibilidades de desarrollo. De hecho esta es la idea que intentaron preservar quienes realizaron el diseño del protocolo: dar libertad a los fabricantes de equipos, a los desarrolladores de aplicaciones y servicios así como a los operadores. Pretende pues el estándar WAP saber integrar los cambios en su entorno de modo que sus prestaciones mejoren incesantemente.

Se manejan varias líneas de trabajo en dichas mejoras:

- Integración de SIM Toolkit, tarjeta inteligente y WAP.
- Integración de MexE y WAP.
- Compresión.
- Aplicaciones sobre la capa de seguridad.
- Descarga de aplicaciones.
- APIs de habla.
- APIs para cada capa.
- *Streams* multimedia para portadores con mayor ancho de banda.

- Descarga de Librerías WMLScript.
- Servicios basados en localización.
- Transporte de datos orientado a conexión.
- Integración adicional con la Red Telefónica.
- Una arquitectura de seguridad más amplia, con tarjetas inteligentes, mejoras en la seguridad extremo a extremo, jerarquías de autoridades certificadoras.

6. PRINCIPIOS PARA EL DESARROLLO DE APLICACIONES EJECUTABLES EN AMBIENTE WAP

6.1 Generación dinámica de contenidos WAP

Tras la presentación de las características más relevantes de la tecnología WAP y justificando la importancia que en este ámbito cobra la personalización de contenidos ofrecidos, en este capítulo se analizarán los distintos modelos y tecnologías de procesamiento en servidor existentes (CGI, xSAPI, Servlets Java, etc) reflejando el "estado del arte" actual en lo referente a la generación dinámica de contenidos WAP. Además, se hace hincapié en otro importante aspecto de la personalización: la negociación del formato de presentación de los contenidos para su correcta visualización en terminales móviles concretos, presentando para ello una solución avanzada que se basa en la utilización de perfiles CC/CP y la transformación de documentos XML mediante XSL.

6.1.1 Agentes de usuario WAE

El nivel de aplicación de la arquitectura de protocolos de WAP, WAE, 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:

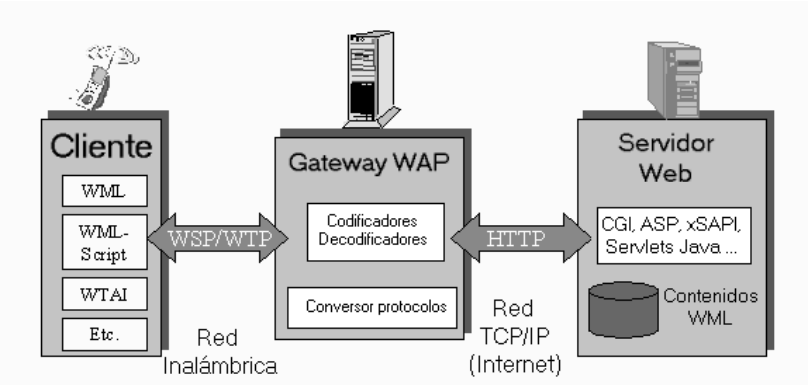
- Agentes de usuario, tales como micro navegadores, agendas telefónicas, editores de mensajes, etc.

- 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 usuario.

En la figura 23 se puede apreciar el modelo de comunicación entre un dispositivo ó terminal móvil con el servidor Web para acceder a un sitio, así como las tecnologías involucradas en cada componente del proceso de comunicación.

Figura 23: Modelo de comunicación WAE



6.1.2 Contenidos

En cuanto a los contenidos que se intercambian en transacciones WAP, el lenguaje WML define una estructura básica llamada carta (*card*), que representa una interacción con el usuario, y que sintacticamente contiene una primera parte no visible, con contenidos ejecutables, a la que le sigue una segunda parte de elementos visibles.

En este modelo, se define un mazo (*deck*) como un conjunto de cartas. Precisamente por ser ese conjunto una representación de las sucesivas interacciones con el usuario, un mazo sería el equivalente a un programa o documento WAP, y por ello tiene asociado un único URL (*Uniform Resource Locator*).

En la práctica, y a efectos de desarrollo, se puede comparar un mazo (*deck*, o documento WML) con un documento HTML que contiene numerosas secciones identificables sin ambigüedad (de hecho, en WML se usa para ello la sintaxis #etiqueta, lo que recuerda enormemente a las anclas nominales de HTML).

A continuación se muestra un ejemplo de la codificación WML de un mazo con dos cartas:

```
<wml>
	<card id="carta1" title="Ejemplo1">
		<do type="accept" label="Ir a carta 2">
			<go href="#carta2">
		</do>
	</card>
	<card id="carta 2" title="Ejemplo2">
	Esta es la segunda carta del mazo:
		<p>
		</p>
	</card>
</wml>
```

6.2 Modelo general de operación de WAP/WAE

El modelo típico de operación WAP se basa en el paradigma cliente/servidor multinivel, y tiene como característica distintiva el hecho de que las transacciones suelen utilizar como elemento intermediario un servidor/*gateway* WAP.

En este escenario (ver figura 23), cuando el usuario suministra una URL en el cliente (terminal móvil), un agente de usuario WAE usa el protocolo WSP para enviar solicitudes de contenidos y servicios a través de una red inalámbrica, en formato comprimido (y opcionalmente seguro), al *gateway*.

El *gateway* convierte dichas solicitudes a formato HTTP (o HTTPS si se utilizan transacciones seguras) 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.

Dentro del modelo descrito, las funciones del *gateway* están bastante definidas:

- 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.

Esta, aún siendo la configuración más habitual, no es la única. En efecto, existen servidores de contenidos y servicios WTA 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*).

6.3 Técnicas de procesamiento en servidor WWW para generación de contenidos WAP

Como se puede apreciar, el modelo general de operación cliente/servidor de WAP es muy similar al usado en el dominio WWW, y, de hecho, aquél ha sido específicamente diseñado para que sea posible aprovechar la infraestructura tecnológica existente en la Web para el aporte de contenidos.

Debe observarse además que en tanto se accede en última instancia a un servidor Web estándar, los contenidos WAP devueltos pueden ser estáticos (tienen existencia previa en un sistema de archivos local o remoto accesible por el servidor) o generados dinámicamente usando tecnologías de servidor suficientemente probadas en la Web, sin que el cliente WAP pueda establecer distinción alguna sobre el mecanismo de producción de contenidos usado realmente (tal como sucedía en los clientes Web).

En general, nunca se debería confiar en la efectividad de procesos de filtrado y conversión automática de contenidos Web a formato WAP (posibilidad anteriormente mencionada), debido a que estos filtros poco pueden hacer para adaptar al 100% contenidos que no tienen en cuenta –en origen- las características físicas especiales de los terminales móviles que realmente se van a usar como clientes. Significa esto que es siempre preferible aportar contenidos directamente en formato WAP (WML, WMLScript..etc).

Por otra parte, y como uno de los puntos más importantes que contribuirán al éxito de WAP, es preciso entender que se trata de una tecnología en la que el usuario final es un usuario registrado y conocido por el operador de red celular.

Esto es esencial en operaciones de comercio electrónico, máxime si se tiene en cuenta que al propio mecanismo de seguridad de las redes GSM se le añade el ofrecido por los servicios de seguridad de WTLS.

Quiere todo esto significar que en el caso de WAP, cobra más importancia que nunca la necesidad de personalizar los contenidos entregados al cliente, y, por tanto, al usuario final. Por ello, el modelo de

desarrollo a utilizar, sin duda alguna, será la entrega de contenidos WAP generados dinámicamente en el servidor Web.

Así pues, es necesario conocer cuáles son las técnicas de procesamiento en servidor disponibles en el dominio Web para su utilización en este nuevo ámbito, y conocer cómo se adaptan al mismo. Además, dado que nos encontramos en un entorno donde las redes inalámbricas introducen tiempos de latencia elevados, el rendimiento del mecanismo de generación dinámica es crítico para no añadir más penalizaciones en el tiempo de descarga.

Existen cinco mecanismos que permiten a los Servidores Web ofrecer contenidos dinámicos y procesar consultas de datos casi en tiempo real:

6.3.1 Server Side Includes (SSI)

Se trata de un mecanismo que permite integrar comandos especiales en documentos (originalmente HTML), de modo que los contenidos de la página resultante sean diferentes cada vez que un cliente acceda a la misma.

En este caso, el Servidor HTTP debe analizar los documentos antes de entregarlos, para poder así interpretar los posibles comandos SSI e incrustar dinámicamente bloques de datos en el flujo de datos correspondiente al documento finalmente enviado al cliente.

SSI es una técnica de primera generación, por lo que adolece de bastantes inconvenientes, entre los que destacan el bajo rendimiento y el carácter propietario del conjunto de comandos disponibles. Esta técnica se utilizaba en el dominio WWW en aquellos casos en los que se deseaba

implementar un mínimo comportamiento dinámico en servidores HTTP para sistemas Unix.

Tomando como base este mismo modelo de operación, han surgido técnicas de segunda generación, como es el caso de ASP (*Active Server Pages*, de Microsoft) y PHP (*Personal Home Pages)*, que ofrecen posibilidades muy superiores en lo que a generación dinámica de contenidos se refiere, y que, en su conjunto, se podrían considerar mecanismos SSI Mejorados.

ASP es hoy día la principal tecnología ofrecida por Microsoft para la generación dinámica de páginas Web, y es directamente reutilizable en el contexto WAP. Los servidores Web *Internet Information Server* y *Personal Web Server* ofrecen soporte nativo de páginas ASP. Básicamente, una Página ASP no es más que un archivo de texto con extensión .ASP que contiene una combinación de códigos de un lenguaje de descripción de contenidos (como HTML o WML) y sentencias en un lenguaje de script para servidores (En este caso concreto, se pueden usar *JavaScript* y *VBScript*).

Así las cosas, salvo en lo referente a la mayor flexibilidad que proporciona un lenguaje de *scripts*, este modelo en sí no parece diferir demasiado del modelo básico SSI. Sin embargo, la verdadera importancia de esta tecnología radica en el hecho de que ASP permite la integración de componentes COM (*Component Object Model*, Modelo de Objetos para Componentes) en el servidor. De esta manera, un *script* ASP avanzado puede limitarse a crear un flujo de control, desde el que se realiza la invocación de Componentes Activos de Servidor que implementan la lógica la negocio y la conectividad con diversas fuentes de datos (locales o remotas). El propio servidor IIS 4.0 proporciona varios componentes útiles en este sentido.

A continuación se listan algunas notas de interés para la generación de contenidos WML con ejemplos de implementación usando ASP:

- En primer lugar, se debe identificar en la cabecera HTTP de respuesta el tipo MIME del objeto devuelto:

```
<!--#include file="conn.asp">
<%
Response.ContentType= "text/vnd.wap.wml" %>
```

- En segundo lugar, la aplicación debe identificar el tipo de documento XML devuelto (recuerde que WML se basa en XML):

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

- Típicamente, el código ASP puede utilizar el modelo ADO (*ActiveX Data Objects*) para el acceso a elementos de una base de datos, de modo que se pueda ofrecer una carta WML que contenga, por ejemplo, un elemento *SELECT* con una lista de opciones:

```
<card id="carta1" title="Seleccione producto">
<%
        sqlQuery = "SELECT [Cod_Prod], [nombre] FROM Productos"
        set rsProductos = conn.execute(sqlQuery)
%>
<select name='prod'>
<%
        do while not rsProductos.eof
                response.write("<option value='" &
                        rsProductos("Cod_Prod") &"'>
                        "& rsProductos("nombre")
                rsProductos.MoveNext
loop %>
</select>
```

- Se debe evitar el uso del objeto *Session*, aunque sea necesario repetir consultas al SGBD, pues su funcionamiento depende del uso de *cookies*, que, pese a formar parte de la especificación, no siempre están soportadas en terminales móviles WAP de primera generación.

- En ocasiones, aún a riesgo de comprometer el principio de independencia del dispositivo contemplado por WML, es preciso realizar ajustes finos en la presentación para terminales específicos:

```
<% if InStr(Request.ServerVariables ("HTTP_USER_AGENT"),"MotorolaTimePort") then ...%>
```

6.3.2 La interface CGI (*Common Gateway Interface*)

La interfaz CGI, profusamente usada en el dominio WWW, posibilita la ejecución de aplicaciones externas en el servidor, cuya misión consiste en procesar la información suministrada por el usuario y generar al vuelo un documento WML (u objeto Wap, en general) que se devuelve al *gateway* WAP como respuesta, para su reenvío al cliente.

Estas aplicaciones pueden encargarse de forma autónoma de la gestión de los datos (procesamiento directo) , o bien actuar como pasarela *middleware* hacia otras aplicaciones especializadas (procesamiento indirecto).

El principal problema de las aplicaciones CGI es su pobre eficiencia en situaciones de carga elevada en el servidor. En efecto, cada vez que un cliente Web (el *gateway* WAP) hace una referencia a un programa CGI, se crea un proceso totalmente nuevo en el servidor para su ejecución : el sistema operativo debe inicializar un espacio de direcciones, leer el ejecutable de disco, cargar librerías dinámicas, iniciar la ejecución del programa,

inicializar sus variables, y, finalmente, leer y procesar los datos enviados por el cliente Web.

Si, además, el programa CGI está codificado en un lenguaje interpretado (por ejemplo, Perl), la situación se agrava aún más, ya que a todo lo anterior se ha de añadir una importante penalización de tiempo debida a la carga del intérprete y al proceso de análisis léxico y sintáctico del archivo fuente.

Para la codificación de aplicaciones Wap basadas en CGI, existen librerías de código Perl que simplifican enormemente el desarrollo al ofrecer funciones específicas para generar contenidos Wap. Por ejemplo en la librería wmlib.pl, la función *&PrintHeader* genera la siguiente salida WML:

```
Context-type: application/x-wap.wml\n\n
<?xml version="1.0"?>
<!DOCTYPE WML PUBLIC "-//WAPFORUM//DTD WML 1.0//EN"
"http://www.wapforum.org/DTD/wml.xml">
```

Como se puede observar, en el ejemplo, lo primero que se hace es lo que toda aplicación de servidor tiene por obligación: identificar el tipo MIME devuelto. En realidad este tipo se entrega al *gateway*, por lo que cabe usar tipos MIME propietarios para activar funciones específicas del *gateway*. Por ejemplo, en el caso del servidor UP.Link, se utiliza el tipo multipart/x-up-fax para activar el envío de los contenidos a un fax designado por el usuario/abonado.

6.3.3 Interfaces de programas de aplicación (APIs propietarias)

En este caso se trata de desarrollar aplicaciones de funciones similares a las de los programas CGI, pero accediendo directamente al código nativo

de los servidores. De este modo se incrementa el rendimiento de las aplicaciones y se reduce considerablemente la carga de los servidores.

Las APIs de referencia son las desarrolladas por Microsoft (ISAPI) y por *Netscape* (NSAPI), y son conceptualmente similares. Las aplicaciones xSAPI se implementan como librerías dinámicas (DLLs), con lo que se evita la creación de un proceso específico en el servidor para atender cada una de las peticiones de los clientes Web.

El uso de APIs en el servidor reduce notablemente la sobrecarga asociada a la ejecución de CGIs, pero no sin un coste : el código de una aplicación WAP implementada con xSAPI está directamente enlazado con el del servidor, por lo que cualquier *bug* podría, potencialmente, afectar al comportamiento del propio servidor. Por otra parte, no hay que olvidar que el código creado es propietario para cada servidor, por lo que se pierde la portabilidad de las aplicaciones WAP.

6.3.4 Interfaz CGI Asíncrona (*FastCGI*)

Se trata de una técnica reciente, y aún en evolución en el dominio WWW, que trata de combinar las ventajas de la interfaz CGI (simplicidad, aislamiento de procesos, solución estándar independiente del lenguaje y de la plataforma) con las derivadas de la utilización de APIs en el servidor (eficiencia y extensiones otras operaciones en el servidor).

La interfaz FastCGI es conceptualmente similar a la intefaz CGI, con dos importantes diferencias :

- Los procesos FastCGI son persistentes : no se destruyen tras finalizar el procesamiento de una petición, sino que entran en estado de espera de nuevas peticiones.
- El protocolo FastCGI multiplexa la información de entorno, la entrada estándar, la salida estándar y la salida de errores usando una única conexión bidireccional. En el caso de ejecución local se usa un pipe full-duplex, y en el caso de ejecución remota, el servidor utiliza una conexión TCP.

6.3.5 Servlets

Los *servlets* son programas codificados en Java que se ejecutan, como alternativa directa a CGI, en servidores HTTP especialmente diseñados para ello. Se trata, pues, de componentes de servidor, independientes de la plataforma, codificados en Java, y que permiten modificar dinámicamente las funciones del servidor. Así, los *servlets* ofrecen un entorno de desarrollo general para la creación de servicios basados en el paradigma solicitud/respuesta.

La utilización de *servlets* constituye una de las últimas tendencias en la programación de aplicaciones Web para servidores, y tienen una aceptación que crece día a día, y en consecuencia, son candidatos idóneos para el desarrollo de soluciones WAP en servidores HTTP.

Existe una única máquina virtual Java ejecutándose en el servidor HTTP, y los *servlets* sólo se cargan una vez, bajo demanda. Un *servlet* no se vuelve a cargar mientras no sufra ninguna modificación, y aún siendo éste el caso, no es necesario reiniciar el servidor.

Además, dado el carácter estándar del propio lenguaje Java, los *servlets* son directamente portables entre plataformas (independencia del Sistema Operativo y del tipo de servidor, en tanto en cuanto ofrezca soporte de *Servlets*)

A nivel de rendimiento, una de las principales características de los *servlets* es el hecho de que no precisan de la creación de un nuevo proceso para cada petición. En la mayoría de los sistemas, los *servlets* se ejecutarán en paralelo, dentro del mismo proceso del servidor. En estos casos, los *servlets* tienen una gran ventaja de rendimiento frente a las aplicaciones CGI e incluso aplicaciones FastCGI. Además, puesto que los *servlets* residen en memoria, posibilitan la compartición de información estática o persistente entre invocaciones.

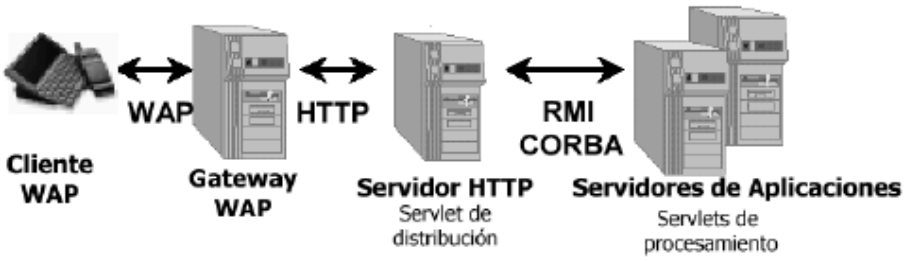
Ya en el apartado de características avanzadas, un *servlet* puede reenviar solicitudes a otros servidores, lo que permite equilibrar la carga (*load balancing*) o utilizar servidores específicos para la generación de determinados contenidos (*service request dispatching*).

A continuación se describe un ejemplo de este tipo, en el que se usa la tecnología de *servlets* para la generación de contenidos WAP usando varios servidores:

En este ejemplo (ver Figura 24), la lógica de la aplicación se reparte entre dos tipos de servidores: un servidor Web estándar y servidores de aplicaciones. El usuario establece, a través del *gateway* WAP, una conexión con el servidor HTTP. Aquí se activa un *servlet* que se encarga de localizar, mediante un mecanismo de distribución basado en RMI (Invocación de Métodos Remotos) o CORBA un servidor de aplicaciones para ejecutar el servicio solicitado (que se modela como una clase Java). Una vez localizada

dicha clase que representa al servicio, se establece la conexión entre el servidor WEB y el servidor de aplicaciones.

Figura 24: Ejemplo del uso de *servlets* para distribuir la lógica de la aplicación entre varios servidores



6.4 Eficiencia de la entrega de respuestas

Sea cual fuere el mecanismo elegido para la generación dinámica de contenidos WAP, es preciso tener en cuenta que existe un método para lograr una mayor eficiencia en la entrega de dichos contenidos al cliente final: se trata de los resúmenes (*digests*)

Los resúmenes, lejos de ser un extracto de los contenidos generados como parece sugerir el nombre, son, en este contexto, un formato MIME multiparte (*multipart/mixed*) que permite el envío de una o más entidades (mazos y otros contenidos WAP) en un único mensaje al *gateway* WAP.

Si se sabe de antemano que el usuario solicitará múltiples entidades, se pueden enviar todas en una única respuesta, con lo que el servicio percibido será mucho más ágil, ya que cada ciclo solicitud/respuesta HTTP

tiene una sobrecarga de tiempo mínima independiente de la cantidad de datos a transmitir.

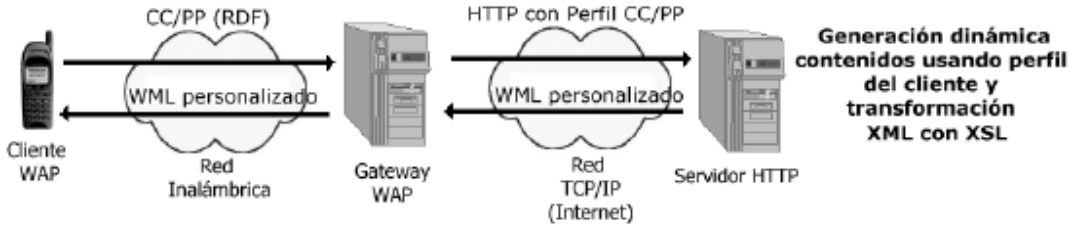
En cualquier caso, teniendo en cuenta que los servicios de creación de contenidos normalmente generan respuestas de longitud variable, se debe limitar el tamaño de los resúmenes a 1200 Bytes.

6.5 Personalización del formato de los contenidos

Resulta claro que las distintas técnicas de procesamiento en servidor ya vistas posibilitan la personalización de contenidos que se envían al cliente final; Sin embargo, aún queda por solucionar un aspecto importante de la personalización: el formato de presentación de los contenidos.

Cierto es que serán las propias técnicas de procesamiento las que se encarguen también de asignar el formato apropiado a los contenidos generados, aunque para conseguirlo, será necesario utilizar algún mecanismo que permita obtener las características físicas y la configuración actual del cliente y así poder establecer cuál es el formato más apropiado.

Figura 25: Generación dinámica de contenidos usando perfil del cliente y transformación XML con XSL



A continuación se expone uno de los mecanismos más avanzados de negociación de formato que se han propuesto para su aplicación en el contexto de WAP (Ver figura 25).

Por un lado, se contempla la adquisición de características del terminal, y, por otro, la aplicación del formato correspondiente. La primera cuestión queda resuelta con el envío de un documento Perfil de Cliente (en formato CC/PP) desde el propio terminal al *gateway* WAP. CC/PP (*Composite Capabilities/Preferences Profile*) es un formato de metainformación creado por el W3C. Se trata de un entorno estructurado en RDF, fácilmente extensible, que permite describir las características de un dispositivo y/o las preferencias del usuario.

El modo de solución de la aplicación del formato adecuado será la siguiente: En el dominio Web existe un mecanismo que separa la presentación de los contenidos: las hojas de estilo CSS. Así, si se tienen distintos tipos de clientes Web, bastará con aplicar la hoja de estilo apropiada para cada uno de ellos a los contenidos, que permanecen inalterados.

Esta solución no es válida para terminales WAP, puesto que en la especificación actual no se admite el uso de hojas de estilo. Esto significa que sólo se tiene la opción de transformar los datos en el propio proceso de generación, en lugar de usar hojas de estilo para asignarles un nuevo formato.

WML se basa en XML, por otra parte, puesto que la DTD de XML describe exactamente las semántica y el contenido de los códigos (usando una estructura arborescente), resulta sencillo transformar un archivo XML de un formato a otro. Únicamente será necesario usar un sistema de transformación, como XSL (en el que se define un árbol de reglas de

transformación), con lo que el problema estaría resuelto de una forma simple y conceptualmente elegante.

6.6 Desarrollar aplicaciones en ambiente WAP

Debido al número de elementos que interviene en WAP, el desarrollo de aplicaciones puede estar basado en diferentes lenguajes y sistemas. Los teléfonos

móviles basados en WAP únicamente entenderán WML y WMLScript. Pero al igual que ocurre con el desarrollo de aplicaciones para internet, podemos generar archivos WML dinámicamente mediante ASP o CGI, permitiendo de esta manera crear el WML como resultado de una operación realizada en el servidor.

La creación de estos servicios no reside únicamente en le servidor, la mayoría de los WAP *Gateways* permiten la incorporación de *servlets*, CGI, etc., lo que permitirá aplicaciones como por ejemplo, juegos *on-line* multijugador.

Para desarrollar aplicaciones basadas en WML y WMLScript existen entornos de trabajo que simulan el dispositivo WAP; de esta forma podremos crear y ejecutar programas sin necesidad de un teléfono móvil WAP, conexión, *gateway*, etc. Actualmente existen varios entornos de desarrollo (SDKs): Nokia ha creado el “Nokia Developer’s Kit”, Phone.com el “Up.Simulator”, etc. La mayoría de estos entornos pueden conseguirse de forma gratuita en las páginas de cada empresa. Con el SDK podremos crear

programas y ejecutarlos en el simulador, el cual se comporta como un teléfono móvil real. Permite acceder a aplicaciones WAP en internet con sólo incluir la dirección HTTP en la cual residen y evitar la conectar nuestro dispositivo WAP a través del *Gateway*, lo cual es complicado en algunas ocasiones, ya que la estructura necesaria de las compañías de telefonía móvil está, en muchos caso, en fase de prueba.

Existen otros productos de ayuda al desarrollo de aplicaciones WAP como son los simuladores de latencia, que nos permiten comprobar en nuestro entorno el comportamiento real en cuanto a la velocidad de transmisión de nuestras aplicaciones. Por otra parte, los convertidores de imágenes al formato WAP pasan de un formato GIF, BMP, etc. a WBMP, que es formato estándar de WAP.

6.6.1 WML

6.6.1.1 ¿Qué es WML?

Wireless Markup Language (WML) es un lenguaje de marcas (o etiquetas) enfocado a su uso en telefonía móvil WAP. Este lenguaje está basado en *Extensible Markup Language* (XML) siendo este último, a su vez, es una variación de SGML (*Standardised Generalised Markup Language*). XML nació del esfuerzo por crear documentos con formato que sean independientes del sistema en que se utilicen o visualicen.

Las etiquetas marcan las propiedades del fragmento de código o texto al que delimitan. Normalmente las etiquetas se utilizan por parejas: una

etiqueta de apertura, por ejemplo <card>, y otra de cierre </card>, en otros casos será una etiqueta unitaria, como es el caso de
.

Hay que tener en cuenta que XML, y por tanto WML, distinguen entre mayúsculas y minúsculas en sus etiquetas, nombres de variables, etc. También, aunque separemos las palabras o coloquemos el texto con espacios, tabuladores y/o retornos de carro, a la hora de la interpretación no tienen valor (solo se tendrán en cuenta los espacios pertenecientes a un texto, una tabla o similares), podría perfectamente ir todo el código en una misma línea.

6.6.1.2 Estructura básica WML

En las WEB, una página es cargada en el visor (o navegador) por lo que podemos considerar una página como la unidad mínima de información. Varias páginas pueden visualizarse usando marcos, lo cual hace una diferencia importante entre la página y el contenido. En los dispositivos WAP no se carga una página sino una baraja *(dec*k) de cartas *(car*d). Estas cartas contienen tanto la información como los elementos de navegación. Además un terminal WAP sólo puede mostrar una carta a la vez, aunque puede tener almacenadas varias (un grupo de cartas asociadas es lo que se conoce como baraja la cual, para la tecnología WAP constituye el "paquete" mínimo de información o contenido).

6.6.2 WMLScript

6.6.2.1 ¿Qué es WMLScript?

Es el lenguaje de *script* utilizado en las aplicaciones WAP, sus principales características son:

- WMLSript es un lenguaje de *script* en el lado del cliente, muy similar a *JavaScript*, basado en *ECMAScript* y es mucho más potente que WML.
- Es un lenguaje "débilmente orientado a tipos", por lo que será importante monitorizar el uso de las variables.
- WMLScript contiene todas las construcciones propias de cualquier lenguaje de programación como selecciones y repeticiones.
- WMLScript posee un conjunto de bibliotecas para realizar las tareas comunes.
- WMLScript presenta muy pocas posibilidades para el tratamiento de errores, aunque se controlen en gran medida comprobando las variables.

6.6.3 WML vs. WMLScript

Desde la perspectiva de la programación, WML y WMLScript son diferentes herramientas para realizar tareas distintas. WML es un lenguaje declarativo basado en marcas diseñado para expresar las características de una interfaz de usuario. WMLScript es un lenguaje imperativo basado en *scripts* diseñado para incluir construcciones propias de la programación lógica y funcional.

WML tiene un solo hilo de ejecución secuencial. No existe una pila de ejecución donde parámetros, punteros de instrucción y variables locales sean apiladas y desapiladas. Hay una pila histórica, pero contiene localizaciones - cartas y URLs. WMLScript tiene funciones y puedes llamar a una función desde otra. Esta ejecución ser realiza usando una pila, que contiene parámetros, el puntero de la instrucción actual y variables locales. Cuando llamas a una función son metidos en la pila ("PUSH") y cuando la función termina, son sacados ("POP") y desaparecen.

Todas las variables de WML son globales. En contraste, todas las variables de WMLScript son locales. WML no tiene declaración de procedimientos, mientras que WMLScript tiene un conjunto de instrucciones para la declaración de procedimientos tradicionales : if.... else...., loop...

WML tiene limitado la capacidad de manejo de datos. WMLScript, aunque es un lenguaje muy poco “tipado”, tiene varios tipos internos de datos, librerías para hacer conversiones entre distintos tipos de datos, *strings* y otras sofisticadas herramientas de manipulación de datos.

7. APLICACIÓN UTILIZANDO TECNOLOGÍAS ASP Y WAP

Este capítulo se basará en mostrar paso a paso como se desarrolla aplicación sencilla en ambiente WAP. El producto final será la simulación del funcionamiento de un Banco Virtual, el cual provee de las opciones básicas para el manejo de las cuentas y chequeras del usuario. La aplicación ha sido desarrollada utilizando la herramienta *Nokia Internet Toolkit 3.1*

7.1 Configuración del servidor

7.1.1 Tipos MIMES

Para que el servidor de HTTP sepa que debe servir las páginas wml como páginas WAP (y no como si fueran archivos de texto), debe indicársele previamente, para esto se utilizan los tipos MIME, con estos tipos indicamos al servidor como se debe comunicar con el cliente cuando se le solicita una página wml. La configuración de tipos MIME, es la siguiente:

Tabla X: Configuración de Tipos MIME

Contenido	Tipo MIME	Extensión
WML Source	text/vnd.wap.wml	wml
Compiled WML	Application/vnd.wap.wmlc	wmlc
WMLScript source	text/vnd.wap.wmlscript	wmls
Compiled WMLScript	Application/vnd.wap.wmlscriptc	wmlsc
Wireless Bitmap	image/vnd.wap.wbmp	wbmp

7.2 Definición de la base de datos de la aplicación

La aplicación ha sido desarrollada utilizando la base de datos *SQL Server* 2000, instalada sobre un servidor de *Windows* 2000, por lo que el acceso el servidor desde las páginas de la aplicación se ha hecho vía ADO (*ActiveX Data Object*), la cual es una librería que puede utilizarse cuando se trabaja en una plataforma *Windows* para acceder a la base de datos.

7.2.1 Definición de las tablas

7.2.1.1 Tabla "Cliente"

Tabla XI: Definición de la tabla "Cliente"

Campo	Descripción	Tipo de Dato
Cliente	Código de identificación del cliente	String (10)
Nombre	Nombre del Cliente	String (50)
Password	Palabra clave para ingreso al sistema	String (10)

En esta tabla deben definirse los datos generales de cada uno de los clientes que tenga el banco.

Ejemplo:

Cliente : ralvarado
Nombre : Rene Estuardo Alvarado
Password : ******** (no debe mostrarse al usuario)

7.2.1.2 Tabla "Cuenta"

Tabla XII: Definición de la tabla "Cuenta"

Campo	Descripción	Tipo de Dato
Cuenta	No. De la Cuenta	String (10)
Saldo	Saldo de la Cuenta	Decimal
Disponible	Cantidad de Dinero Disponible en la Cuenta	Decimal
Reserva	Cantidad de Dinero en Reserva de la Cuenta	Decimal
Status	Status de la cuenta	String
FechaApertura	Fecha de apertura de la cuenta	Entero

En esta tabla deben definirse las cuentas que se vayan creando para los clientes del banco, tomando en cuenta que Saldo = Disponible + Reserva, porque la reserva será algún valor por cheques de compensación, el cual se liberará totalmente hasta haber obtenido la respuesta del banco afiliado al cual corresponda el cheque, para saber que este cuenta con fondos suficientes para ser acreditado a la cuenta que se está afectando.

Ejemplo:

Cuenta : 1
Saldo : Q.250.00
Disponible : Q.150.00
Reserva : Q.100.00
Status : VIG {Cuenta Vigencia}
FechaApertura: 17/01/20003

7.2.1.3 Tabla "Status_Cuenta"

Tabla XIII: Definición de la tabla "Status_Cuenta"

Campo	Descripción	Tipo de Dato
Status	Status de la cuenta	String(10)
Descripción	Descripción del estatus	String(50)

En esta tabla van a definirse los distintos estados que pueda tener una cuenta en un momento dado, pueden ser vigente o bloqueada.

Ejemplo:

Cuenta : 0000000001
Status : BLO {Cuenta bloqueada}

7.2.1.4 Tabla "Cliente_Cuenta"

Tabla XIV: Definición de la tabla "Cliente_Cuenta"

Campo	Descripción	Tipo de Dato
Cliente	Código de cliente al que pertenece la cuenta	String (10)
Cuenta	No. De Cuenta	String (10)

En esta tabla van a definirse los clientes que son dueños de determinada cuenta, ya que pueden manejarse cuentas mancomunadas.

Ejemplo:

Cliente : ralvarado
Cuenta : 0000000001

7.2.1.5 Tabla "Chequera"

Tabla XV: Definición de la tabla "Chequera"

Campo	Descripción	Tipo de Dato
Chequera	Código de cliente al que pertenece la cuenta	String(10)
Cuenta	Número de Cuenta	String(10)
Status	Status de la chequera	String(10)
Fecha	Fecha de entrega de la chequera	Datetime

En esta tabla van a definirse las chequeras que se vayan asociando a una cuenta determinada.

Ejemplo:

Chequera : 0000000001
Cuenta : 0000000001
Status : VIG {Chequera Vigente}

7.2.1.6 Tabla "Status_chequera"

Tabla XVI: Definición de la tabla "Status_chequera"

Campo	Descripción	Tipo de Dato
Status	Status de la chequera	String(10)
Descripción	Descripción del estatus	String(50)

En esta tabla van a definirse los distintos Status que pueda tener una chequera en un momento dado, tales como Vigente ó Bloqueada.

Ejemplo:

Chequera : 1

Status : BLO {Chequera Bloqueada}

7.2.1.7 Tabla "Cheque"

Tabla XVII: Definición de la tabla "Cheque"

Campo	Descripción	Tipo de Dato
Cheque	No. De Cheque	String(10)
Cuenta	No. De Cuenta al que pertenece la chequera	String(10)
Chequera	No. De Chequera a la que pertenece el cheque	String(10)
Status	Status del cheque	String(10)
FechaPago	Fecha en la que el cheque fue pagado	Datetime

En esta tabla van a definirse los cheques correspondientes a una chequera de una cuenta determinada, así como el estado de cada uno de los cheques.

Ejemplo:

Cheque : 0000000001
Cuenta : 0000000001
Chequera : 0000000001
Status : COB {Cheque Cobrado}

7.2.1.8 Tabla "Status_Cheque"

Tabla XVIII: Definición de la tabla "Status_Cheque"

Campo	Descripción	Tipo de Dato
Status	Estado del cheque	String(10)
Descripción	Descripción del estado	String(50)

En esta tabla van a definirse los distintos estados que pueda tener un cheque en un momento dado, el cual puede ser : vigente, anulado, cobrado ó extraviado.

Ejemplo:

Status : COB
Descripción : Cheque cobrado.

7.2.2 *Script* de la base de datos

A continuación se lista el código del script para la creación de las tablas, llaves primarias y foráneas de la base de datos de la aplicación, este código debe ser ejecutado en SQL Server 2000 para la correcta creación de las estructuras y sus relaciones.

```
CREATE TABLE [dbo].[Status_Cheque] (
	[Status] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Cheque] (
	[Cheque] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Cuenta] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Chequera] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Status] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[FechaPago] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Chequera] (
	[Chequera] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Cuenta] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Status] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[Fecha] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Cliente] (
	[Cliente] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[Password] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

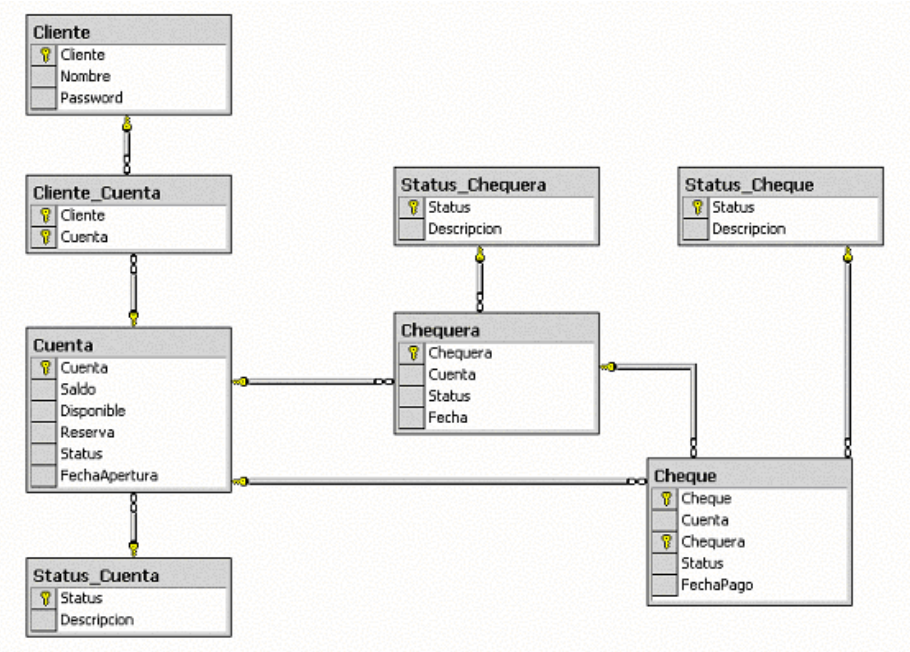
```
CREATE TABLE [dbo].[Cliente_Cuenta] (
	[Cliente] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Cuenta] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Cuenta] (
	[Cuenta] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Saldo] [decimal](9, 2) NULL ,
	[Disponible] [decimal](9, 2) NULL ,
	[Reserva] [decimal](9, 2) NULL ,
	[Status] [varchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[FechaApertura] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Status_Chequera] (
	[Status] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Status_Cuenta] (
	[Status] [varchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Descripcion] [varchar] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Status_Cheque] WITH NOCHECK ADD
	CONSTRAINT [PK_Status_Cheque] PRIMARY KEY  CLUSTERED
	(
		[Status]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Cheque] WITH NOCHECK ADD
	CONSTRAINT [PK_Cheque] PRIMARY KEY  CLUSTERED
	(
		[Cheque],
		[Chequera]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Chequera] WITH NOCHECK ADD
	CONSTRAINT [PK_Chequera] PRIMARY KEY  CLUSTERED
	(
		[Chequera]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Cliente] WITH NOCHECK ADD
	CONSTRAINT [PK_Cliente] PRIMARY KEY  CLUSTERED
	(
		[Cliente]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Cliente_Cuenta] WITH NOCHECK ADD
	CONSTRAINT [PK_Cliente_Cuenta] PRIMARY KEY  CLUSTERED
	(
		[Cliente],
```

```
		[Cuenta]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Cuenta] WITH NOCHECK ADD
	CONSTRAINT [PK_Cuenta] PRIMARY KEY  CLUSTERED
	(
		[Cuenta]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Status_Chequera] WITH NOCHECK ADD
	CONSTRAINT [PK_Status_Chequera] PRIMARY KEY  CLUSTERED
	(
		[Status]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Status_Cuenta] WITH NOCHECK ADD
	CONSTRAINT [PK_Status_Cuenta] PRIMARY KEY  CLUSTERED
	(
		[Status]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[Cheque] ADD
	CONSTRAINT [FK_Cheque_Chequera1] FOREIGN KEY
	(
		[Chequera]
	) REFERENCES [dbo].[Chequera] (
		[Chequera]
	),
	CONSTRAINT [FK_Cheque_Cuenta] FOREIGN KEY
	(
		[Cuenta]
	) REFERENCES [dbo].[Cuenta] (
		[Cuenta]
	),
	CONSTRAINT [FK_Cheque_Status_Cheque] FOREIGN KEY
	(
		[Status]
	) REFERENCES [dbo].[Status_Cheque] (
		[Status]
	)
GO

ALTER TABLE [dbo].[Chequera] ADD
	CONSTRAINT [FK_Chequera_Cuenta] FOREIGN KEY
	(
		[Cuenta]
	) REFERENCES [dbo].[Cuenta] (
		[Cuenta]
	),
	CONSTRAINT [FK_Chequera_Status_Chequera] FOREIGN KEY
	(
		[Status]
	) REFERENCES [dbo].[Status_Chequera] (
		[Status]
	)
GO
```

```
ALTER TABLE [dbo].[Cliente_Cuenta] ADD
	CONSTRAINT [FK_Cliente_Cuenta_Cliente] FOREIGN KEY
	(
		[Cliente]
	) REFERENCES [dbo].[Cliente] (
		[Cliente]
	),
	CONSTRAINT [FK_Cliente_Cuenta_Cuenta] FOREIGN KEY
	(
		[Cuenta]
	) REFERENCES [dbo].[Cuenta] (
		[Cuenta]
	)
GO

ALTER TABLE [dbo].[Cuenta] ADD
	CONSTRAINT [FK_Cuenta_Status_Cuenta] FOREIGN KEY
	(
		[Status]
	) REFERENCES [dbo].[Status_Cuenta] (
		[Status]
	)
```

7.2.3 Modelo Entidad – Relación

Figura 26: Modelo entidad-relación de la aplicación "Banco Virtual"



Descripción de la aplicación

7.3.1 Ingreso al sistema

Esta es la página de entrada al sistema, solicita al usuario el código del usuario (*ID Cliente*) y el *password*, permitiendo así la autenticación y reconocimiento del usuario por parte del sistema, el archivo físico de esta página es *pagInicio.asp*, el código fuente de esta página se muestra en la figura 28.

Figura 27: Página de Inicio

Figura 28: Código Fuente de *pagInicio.asp*

```
<%
        Response.ContentType = "text/vnd.wap.wml"
        session("charTelefono")=""
%>

<?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="login" title="Banco Virtual" newcontext="true">
        <p>
                <big>
                        <b>
                                Ingreso al Sistema
```

```
                        </b>

                </big>
        <br/><br/>
        ID Cliente :<input type="text" name="txtcliente" maxlength="10" />
        Password:<input type="password" name="txtpassword" maxlength="10" />

                <do type="accept" label="Aceptar">
                        <go method="post" href="pagVerificaLogin.asp">
                                <postfield name="txtcliente" value="$(txtcliente)"/>
                                <postfield name="txtpassword" value="$(txtpassword)"/>
                        </go>
                </do>
        </p>
</card>
</wml>
```

7.3.2 Verificación de autenticación

Esta página es llamada por la página *pagInicio.*asp, y contiene le código para ejecutar la validación del usuario en el sistema, recibe como parámetros el código del usuario y el *password* ingresados en la página anterior, el código fuente de esta página puede verse en la figura 30.

Figura 29: Usuario validado por el sistema

Figura 30: Código Fuente de *pagVerificarLogin.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="verificar" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                <%
                        dim cnn
                        dim rst
                        session("charCliente")=request.form("txtcliente")
                        Call ConectarBDD()
        strSQL = "SELECT nombre from cliente where cliente='" & session("charCliente") &_
         "' and password='" & request.form("txtpassword") & "'"
        rst.Open strSQL, cnn
        if  not( rst.eof=True ) then
        %>
                <big>
                <u><b>Bienvenido:</b></u><br/>
                </big>
                <table columns="1">
        <%
        //se muestran los datos del usuario
        do while not rst.EOF
                response.write ("<tr>")
                        response.write("<td>")
                        response.write rst("nombre")
                                session("charNombre")=rst("nombre")
                        response.write("</td>")
                response.write ("</tr>")
                rst.movenext
                loop
        %>
        </table>

        <!--muestra el menu de opciones-->
        <do type="accept" label="Ver Menú">
                <go href="pagMenu.asp"/>
        </do>
        <%
        //si no se logueo muestra un mensaje de error
        else
                response.write "<i>Cliente y Password no coinciden</i>"
        end if
        rst.close
        cnn.close
        %>
        <do type="accept" label="Salir">
                <go method="post" href="pagInicio.asp"></go>
                </do>
        </p>    </card></wml>
```

La instrucción <!--#include file="pagConectar.asp" --> que tiene incluida la página en el encabezado indica el llamado al código de la página *pagConectar.asp* en tiempo de compilación, esta página tiene la creación de los objetos de conexión a la base de datos y su código fuente se muestra en la figura 31.

Figura 31: Código Fuente de *pagConectar.asp*

```
<%
Sub ConectarBDD
   Dim strCnn

   ' Comprobar que la sesión está activa
   if session("charCliente")= "" then
            response.clear
            response.redirect "pagInicio.asp"
            response.end
   end if

   ' Abrir la conexión
   charConexion = "Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Initial Catalog=AppWAP;Data
Source=ROUTER"
   Set cnn = Server.CreateObject("ADODB.Connection")
   cnn.Open charconexion

   'Inicialización del objeto Recordset
   Set rst = Server.CreateObject("ADODB.Recordset")
End sub
%>
```

7.3.3 Menú de Opciones

Esta página muestra el menú con las opciones principales del sistema, el archivo físico de esta página es *pagMenu.asp*, el código fuente de esta página se muestra en la figura 33.

Figura 32: Vista del menú de opciones

Figura 33: Código Fuente de *pagMenu.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="consultasaldos" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                        <do type="accept" label="Consulta de Saldos">
                                <go href="pagConsultarSaldo.asp"/>
                        </do>
                        <do type="accept" label="Transferencias">
                                <go href="pagTransferencias.asp"/>
                        </do>
                        <do type="accept" label="Manejo de Chequeras">
                                <go href="pagConsultaCuenta.asp"/>
                        </do>
                        <do type="accept" label="Consulta de Cuentas">
                                <go href="pagMuestraCuenta.asp"/>
                        </do>
                        <do type="accept" label="Bloqueo de Cuenta">
                                <go href="pagCuentasVigentes.asp"/>
                        </do>
                        <do type="accept" label="Desbloqueo de Cuenta">
                                <go href="pagCuentasBloqueadas.asp"/>
                        </do>
                        <br/>
```

```
                        <i>Escoja la opción que desea realizar.</i>

                        <do type="accept" label="Salir">
                                <go method="post" href="pagInicio.asp"></go>
                        </do>
                        </p>
                </card>
</wml>
```

7.3.4 Consulta de saldos

Esta página muestra el listado de números de cuentas vigentes que tenga el usuario, es decir, las cuentas que estén habilitadas (no bloqueadas) para efectuar transacciones, el código fuente puede verse en la figura 35.

Figura 34: Vista de la página de consulta de saldos

Figura 35: Código fuente de *pagConsultarSaldo.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="consultasaldos" title="Banco Virtual">
	<p align="center">
	</p>
	<p>
			<%
				dim cnn
				dim rst
				//Conecta a la base de datos
				Call ConectarBDD()
				strSQL = "select cuenta from cliente_cuenta where cliente ='" &
session("charCliente") & "'" & _ " and status = 'VIG'"
				rst.Open strSQL, cnn
				if (not rst.eof=True) then
					%>
					<big>
					<u><b>Seleccione una cuenta:</b></u><br/>
					</big>
					<select title="cuenta" name="cuenta">
					<%
					do while not rst.eof
						Response.write "<option value='" & rst(0) & "'>" &
rst(0) & "</option>"
						rst.movenext
					loop
					%>
					</select>
					<do type="accept" label="Consultar Saldo">
						<go method="post"
href="pagConsultarSaldoCuenta.asp">
							<postfield name="cuenta"
value="$(cuenta)"/>
						</go>
					</do>
					<br/>
				<%
					//No hay cuentas
					else
				%>
					<b>No se encontro ninguna cuenta.</b><br/>
				<%
					end if
					rst.close
					cnn.close
			%>
			<br/><i>Presione regresar para volver a las opciones</i>
			<do type="prev" label="Regresar"><prev/></do>
			</p>
		</card>
</wml>
```

Al momento de consultar el saldo de la cuenta seleccionada, mostrará otra pantalla en la aplicación con el resultado de la búsqueda, cuyo código se muestra en la figura 37.

Figura 36: Resultado de la consulta de saldos

Figura 37: Código Fuente d*pagConsultarSaldoCuenta,.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="consultasaldoscuenta" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                        <%
                                dim cnn
                                dim rst
```

```
                                    'Conecta a la base de datos
                                    Call ConectarBDD()
                                    strSQL = "select disponible,reserva,saldo " & _
                                              "from cuenta " & _
                                                        "where cuenta = '" & Request.Form("cuenta")+"'"
                                    rst.Open strSQL, cnn

                                    if (not rst.eof) then
                                              %>
                                              <big>
                                              <u><b>Reserva:</b></u>
                                              </big>
                                              <% Response.write(rst(0)) %>
                                              <br/>

                                              <big>
                                              <u><b>Disponible:</b></u>
                                              </big>
                                              <% Response.write(rst(1)) %>
                                              <br/>

                                              <big>
                                              <u><b>Saldo:</b></u>
                                              </big>
                                              <% response.write(rst(2)) %>
                                              <br/>

                                    <% end if
                                      rst.close
                                      cnn.close
                                    %>
                          <do type="accept" label="Ver Menú">
                                                        <go href="pagMenu.asp"/>
                                              </do>
                          </p>
                </card>
</wml>
```

7.3.5 Transferencias

Esta página se utiliza para hacer transferencias de efectivo entre las cuentas no bloqueadas del usuario, pudiendo seleccionar las cuentas a afectar y se solicita que se ingrese la cantidad a transferir, el archivo físico de esta página es *pagTransferencias.asp* y su código fuente se muestra en la figura 39.

Figura 38: Vista de la página de transferencias

Figura 39: Código Fuente de *pagTransferencias.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="transferencias" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                <%
                        dim cnn
                        dim rst
                        'Conecta a la base de datos
                        Call ConectarBDD()
                        strSQL = "select a.cuenta from cliente_cuenta a " & _
                        "inner join cuenta b on a.cuenta=b.cuenta where a.cliente ='" & session("charCliente")
& "' " & _
                        " and status = 'VIG'"
                        rst.Open strSQL, cnn
                        if (not rst.eof=True) then
                                %>
                                <u><b>Cuenta a debitar:</b></u><br/>
                                <select title="cuenta1" name="cuenta1">
                                <%
                                do while not rst.eof
                                        Response.write "<option value='" & rst(0) & "'> " & rst(0) & "
</option>"
```

```
                                        rst.movenext
                                loop
                                %>
                                </select>

                                <br/>
                                <u><b>Cuenta a acreditar:</b></u><br/>
                                <select title="cuenta2" name="cuenta2">
                                <%
                                rst.movefirst
                                do while not rst.eof
                                        Response.write "<option value='" & rst(0) & "'> " & rst(0) & "
</option>"
                                        rst.movenext
                                loop
                                %>
                                </select>
                                Cantidad:<input type="text" name="txtcantidad" maxlength="10" />
                                <br/>
                        <do type="accept" label="Transferir">
                                        <go method="post" href="pagResTransferencia.asp">
                                                <postfield name="txtcantidad"
value="$(txtcantidad)"/>
                                                <postfield name="cuenta1" value="$(cuenta1)"/>
                                                <postfield name="cuenta2" value="$(cuent2)"/>
                                        </go>
                                </do>
                                <%
                                //No hay cuentas
                                else
                        %>
                                        <b>No se encontro ninguna cuenta.</b><br/>
                        <%
                                end if
                                rst.close
                                cnn.close
                %>
                <br/><i>Presione regresar para volver al menu</i>
                <do type="prev" label="Regresar"><prev/></do>
                </p>
        </card>
</wml>
```

Esta página llama a otra página que ejecuta el proceso de la transferencia y muestra el resultado, esta página es *pagResTransferencia.asp* y su código fuente de muestra en la figura 41.

Figura 40: Vista de la página de resultados de transferencias

Figura 41: Código Fuente de *pagResTransferencia.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="restransferencia" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                <%
                        dim cnn
                        dim rst
                        'Conecta a la base de datos
                        Call ConectarBDD()
                        strSQL = "exec pr_transferencias '" & Request.Form("cuenta1")+"'," & _
                                        "'" + Request.Form("cuenta2")+"'," & _
                                        Request.Form("txtcantidad")
                        rst.Open strSQL, cnn
                        if (not rst.eof) then
                                %>
                                <big>
                                <u><b>Resultado:</b></u>
                                <br/>
                                </big>
                                <% Response.write(rst(0)) %>
                                <br/>

                        <% end if
                                %>
                <do type="accept" label="Ver Menú">
                                <go href="pagMenu.asp"/>
                        </do>
```

```
                </p>
        </card>
</wml>
```

7.3.6 Manejo de Chequeras

Esta opción se utiliza para ver, consultar y cambiar de estado los cheques del usuario, primero se muestra el listado de cuentas habilitadas y se pide que se seleccione una de ellas, luego podrán visualizarse sus chequeras, el archivo físico de esta página es *pagConsultarCuenta.asp* y su código fuente puede verse en la figura 43.

Figura 42: Vista de la página de selección de cuenta

Figura 43: Código Fuente de *pagConsultarCuenta,asp*

```
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="consulta_cuenta" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
```

```
                        <%
                                dim cnn
                                dim rst
                                //Conecta a la base de datos
                                Call ConectarBDD()
                                strSQL = "select cuenta from cliente_cuenta where cliente ='" &
session("charCliente") & "'"
                                rst.Open strSQL, cnn
                                if (not rst.eof=True) then
                                        %>
                                        <big>
                                        <u><b>Seleccione una cuenta:</b></u><br/>
                                        </big>
                                        <select title="cuenta" name="cuenta">
                                        <%
                                        do while not rst.eof
                                                Response.write "<option value='" & rst(0) & "'>" &
rst(0) & "</option>"
                                                rst.movenext
                                        loop
                                        %>
                                        </select>
                                        <do type="accept" label="Mostrar Chequeras">
                                                <go method="post" href="pagMuestraChequera.asp">
                                                        <postfield name="cuenta"
value="$(cuenta)"/>
                                                </go>
                                        </do>
                                        <do type="accept" label="Seleccionar una Chequera">
                                                <go method="post" href="pagConsultaChequera.asp">
                                                        <postfield name="cuenta"
value="$(cuenta)"/>
                                                </go>
                                        </do>
                                        <br/>
                                <%
                                        //No hay cuentas
                                        else
                                %>
                                                <b>No se encontro ninguna cuenta.</b><br/>
                                <%
                                        end if
                                        rst.close
                                        cnn.close
                        %>
                        <br/><i>Presione regresar para volver a las opciones</i>
                        <do type="prev" label="Regresar"><prev/></do></p>
                </card> </wml>
```

7.3.6.1 Mostrar chequeras de una cuenta

La primera de las opciones del menú que se muestra luego de seleccionar una cuenta en el manejo de chequeras, es el listado de las chequeras de la cuenta seleccionada, se muestra el número de chequera, el estado actual de la misma y la fecha de solicitud de la chequera, el archivo físico

del listado es *pagMuestraChequera.asp* y su código fuente puede verse en la figura 46.

Figura 44: Vista del menú del manejo de chequeras

Figura 45: Vista de la página de "Mostrar Chequeras"

Figura 46: Código fuente de *pagMostrarChequera.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
```

Continuación

```
<!--#include file="pagConectar.asp" -->
<?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="muestra_chequeras" title="Banco Virtual">
	<p align="center"> </p>
	<p>
		<%
		dim cnn
		dim rst
		//Conecta a la base de datos
		Call ConectarBDD()
		strSQL = "select chequera,status,convert(varchar(10),fecha,103) fecha from chequera where
cuenta ='" & Request.Form("cuenta") & "'"
		rst.Open strSQL, cnn
		if (not rst.eof=True) then
			%>
			<i><b>Cuenta: <%Response.Write Request.Form("cuenta")%></b></i>
			<br/>
			<table columns="3">
				<tr>
					<td>Chequera</td>
					<td>Status</td>
					<td>Fecha</td>
				</tr>
			<%
			do while not rst.eof
				response.write ("<tr>")
				response.write("<td>")
					response.write rst("chequera")
				response.write("</td>")
				response.write("<td>")
					response.write rst("status")
				response.write("</td>")
				response.write("<td>")
					response.write rst("fecha")
				response.write("</td>")
				response.write("</tr>")
				rst.movenext
			loop
			%>
			</table>
			<br/>
		<%
			//No hay chequeras
			else
		%>
				<b>No se encontro ninguna chequera.</b><br/>
		<%
			end if
			rst.close
```

```
                        cnn.close
%>
<br/><i>Presione regresar para volver a las opciones</i>
<do type="accept" label="Ver Menú">
        <go href="pagMenu.asp"/>
</do>
<do type="prev" label="Regresar"><prev/></do></p>
</card></wml>
```

7.3.6.2 Consulta de chequeras

La segunda de las opciones del menú que se muestra luego de seleccionar una cuenta en el manejo de chequeras, permite seleccionar una chequera determinada, con la finalidad de mostrar sus cheques ó cambiar el estado actual de un cheque, el archivo físico de la página de consulta de chequeras es *pagConsultaChequera.asp*, y su código fuente puede verse en la figura 49.

Figura 47: Vista del menú del manejo de chequeras

Figura 48: Selección de chequera para una cuenta

Figura 49: Código Fuente de *pagConsultarChequera.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
```

Continuación

```
<?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="muestra_chequeras" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
```

```
                        <%
                                dim cnn
                                dim rst
                                //Conecta a la base de datos
                                Call ConectarBDD()
                                strSQL = "select chequera from chequera where cuenta ='" &
Request.Form("cuenta") & "'"
                                rst.Open strSQL, cnn
                                if (not rst.eof=True) then
                                        %>
                                        <i><b>Cuenta: <%Response.Write
Request.Form("cuenta")%></b></i>
                                        <br/>
                                        <i>Chequeras:</i>
                                        <br/>
                                        <select title="chequera" name="chequera">
                                        <%
                                        do while not rst.eof
                                                Response.write "<option value='" & rst(0) & "'>" &
rst(0) & "</option>"
                                                rst.movenext
                                        loop
                                        %>
                                        </select>
                                        <do type="accept" label="Mostrar Cheques">
                                                <go method="post" href="pagMuestraCheque.asp">
                                                        <postfield name="chequera"
value="$(chequera)"/>
                                                </go>
                                        </do>

                                        <do type="accept" label="Cambio de Estado de Cheques">
                                                <go method="post" href="pagConsultaCheque.asp">
                                                        <postfield name="chequera"
value="$(chequera)"/>
                                                </go>
                                        </do>
                                        <br/>
                                <%
                                        else
                                %>
                                        <b>No se encontro ninguna chequera.</b><br/>
                                <%
                                        end if
                                        rst.close
                                        cnn.close
                        %>
                        <br/><i>Presione regresar para volver a las opciones</i>
                        <do type="prev" label="Regresar"><prev/></do>
                        </p>
                </card></wml>
```

7.3.6.3 Mostrar cheques

Al momento de seleccionar una chequera, se muestra un menú que permite mostrar el listado de cheques para la chequera

seleccionada, mostrando el número de cheque, su estado actual y la fecha de cobro (si ya fue cobrado). El archivo físico que muestra el listado de cheques para una chequera determinada es *pagMuestraCheque.asp*, y su código fuente se muestra en la figura 52.

Figura 50: Vista del menú del manejo de chequeras

Figura 51: Vista de la página de listado de cheques

Figura 52: Código Fuente de *pagMuestraCheque.asp*

```
%Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->
```

Continuación

```
<?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="muestra_cheques" title="Banco Virtual">
	<p align="center">
	</p><p>
			<%
				dim cnn
				dim rst
				'Conecta a la base de datos
				Call ConectarBDD()
				strSQL = "select
a.cheque,b.descripcion,isnull(convert(varchar(10),a.fechapago,103),'') fechapago " &_
					"from cheque a inner join status_cheque b on a.status=b.status "
& _ 					"where a.chequera = '" & Request.Form("chequera") & "'"
				rst.Open strSQL, cnn
				if (not rst.eof=True) then%>
					<i><b>Chequera: <%Response.Write
Request.Form("chequera")%></b></i>
					<br/>
					<table columns="3">
						<tr>
							<td>Cheque</td>
							<td>Status</td>
							<td>Pagado</td>
						</tr>
					<%
					do while not rst.eof
						response.write ("<tr>")
						response.write("<td>")
							response.write rst("cheque")
						response.write("</td>")
						response.write("<td>")
							response.write rst("descripcion")
						response.write("</td>")
						response.write("<td>")
							response.write rst("fechapago")
						response.write("</td>")
						response.write("</tr>")
						rst.movenext
					loop
					%>
					</table>
					<br/>
				<%
					else
				%>
						<b>No se encontro ningun cheque.</b><br/>
				<%
					end if
					rst.close
					cnn.close
			%>
			<br/><i>Presione regresar para volver a las opciones</i>
			<do type="accept" label="Ver Menú">
				<go href="pagMenu.asp"/>
			</do>
			<do type="prev" label="Regresar"><prev/></do>
			</p>
```

```
</card></wml>
```

7.3.6.4 Cambiar estado de cheques

Al momento de seleccionar una chequera, se permite seleccionar un cheque determinado y cambiar su estado, siempre y cuando el cheque aún no haya sido cobrado, si es así, el sistema mostrará un mensaje indicando que el cheque ha sido cobrado y no puede cambiar su estado.

Sin embargo, cuando un cheque aún no ha sido cobrado, se permite declararlo como “robado” ó como “extraviado”, para que este no pueda ser cobrado por otra persona. El archivo físico de la página de la consulta de cheques es *pagConsultaCheque.asp*, y su código fuente se muestra en la figura 54.

Figura 53: Vista del menú del manejo de chequeras

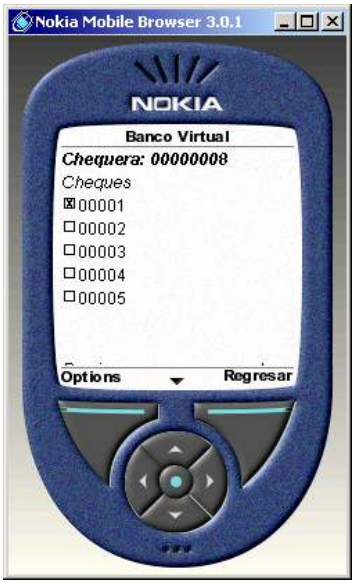


Figura 54: Vista de la página para seleccionar un cheque

Figura 55: Código Fuente de *pagConsultaCheque.asp*

```
<%Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="cambio_estaado_cheques" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                        <%
                                dim cnn
                                dim rst
                                'Conecta a la base de datos
                                Call ConectarBDD()
                                session("tmp_chequera")=Request.Form("chequera")
                                strSQL = "select cheque from cheque where chequera = '" &
Request.Form("chequera") & "'"
                                rst.Open strSQL, cnn
                                if (not rst.eof=True) then
                                        %>
```

```
                                        <i><b>Chequera: <%Response.Write
Request.Form("chequera")%></b></i>
                                        <br/>
                                        <i>Cheques</i>
                                        <br/>
                                        <select title="cheque" name="cheque">
                                        <%
                                        do while not rst.eof
                                                Response.write "<option value='" & rst(0) & "'>" &
rst(0) & "</option>"
                                                rst.movenext
                                        loop
                                        %>
                                        </select>
                                        <do type="accept" label="Declarar Cheque Robado">
                                                <go method="post"
href="pagCambioEstadoCheque.asp">
                                                        <postfield name="estadocheque"
value="R"/>
                                                        <postfield name="cheque"
value="$(cheque)"/>
                                                </go>
                                        </do>
                                        <do type="accept" label="Declarar Cheque Extraviado">
                                                <go method="post"
href="pagCambioEstadoCheque.asp">
                                                        <postfield name="estadocheque" value="E"/>
                                                        <postfield name="cheque"
value="$(cheque)"/>
                                                </go>
                                        </do>
                                        <do type="accept" label="Desbloquear Cheque">
                                                <go method="post"
href="pagCambioEstadoCheque.asp">
```

Continuación

```
        <postfield name="cheque" value="$(cheque)"/>
                                                </go>
                                        </do>
                                        <br/>
                                <%
                                        //No hay cheques
                                        else
                                %>
                                                <b>No se encontro ningun cheque.</b><br/>
                                <%
                                        end if
                                        rst.close
                                        cnn.close
                        %>
                        <br/><i>Presione regresar para volver a las opciones</i>
                        <do type="accept" label="Ver Menú">
                                <go href="pagMenu.asp"/>
                        </do>
                        <do type="prev" label="Regresar"><prev/></do>
                        </p>
                </card>
```

```
</wml>
```

Una vez haya sido seleccionado un cheque, se mostrará un menú con el listado de “estados” a los que el cheque puede ser cambiado, deberá seleccionarse una opción del menú y luego se mostrará el resultado de la operación, el archivo físico que ejecuta el proceso de cambio de estado al cheque es pagCambioEstadoCheque.asp y su código fuente se muestra en la figura 58.

Figura 56: Vista del menú de cambio de estado de cheques

Figura 57: Resultado de la operación de cambio de estado

Figura 58: Código Fuente de *pagCambioEstadoCheque.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="res_cambio_estado_cheque" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                        <%
                                dim cnn
                                dim rst
                                'Conecta a la base de datos
                                Call ConectarBDD()

                                strSQL = "exec pr_cambio_estado_cheques '" &
Request.Form("cheque")+"'," & _
                                        "'" + session("tmp_chequera") +"'," & _
                                                "'" + Request.Form("estadocheque") +"'"
```

Continuación

```
                                    rst.Open strSQL, cnn
                                    if (not rst.eof) then
                                              %>
                                              <big>
                                              <u><b>Resultado:</b></u>
                                              <br/>
                                              </big>
                                              <% Response.write(rst(0)) %>
                                              <br/>

                                    <% end if
                                       rst.close
                                       cnn.close
                                    %>
                          <do type="accept" label="Ver Menú">
                                                        <go href="pagMenu.asp"/>
                                              </do>
                          </p>
                </card>
</wml>
```

7.3.7 Consulta de cuentas

Esta opción muestra el listado de cuentas del usuario, mostrando el número de cuenta, el estado actual de la misma y la fecha de apertura de la cuenta, el archivo físico de esta página es *pagMuestraCuenta.asp*, el código fuente de esta página se muestra en la figura 60.

Figura 59: Listado de cuentas

Figura 60: Código Fuente de *pagMostrarCuenta.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="muestra_cuentas" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                <%
                        dim cnn
                        dim rst
                        //Conecta a la base de datos
                        Call ConectarBDD()
```

```
                    strSQL = "select a.cuenta,c.descripcion,convert(varchar(10),b.fechaapertura,103)
fechaapertura " & _
```

Continuación

```
                    "from cliente_cuenta a inner join cuenta b on a.cuenta=b.cuenta " & _
                    "inner join status_cuenta c on b.status=c.status " & _
                    "where a.cliente='"+session("charCliente")+"'"
                    rst.Open strSQL, cnn
                    if (not rst.eof=True) then
                              %>
                              <i><b>Listado de Cuentas</b></i>
                              <br/>
                              <table columns="3">
                                        <tr>
                                                  <td>Cuenta</td><td>Status</td><td>Fecha</td>
                                        </tr>
                              <%
                              do while not rst.eof
                                        response.write ("<tr>")
                                        response.write("<td>")
                                                  response.write rst("cuenta")
                                        response.write("</td>")
                                        response.write("<td>")
                                                  response.write rst("descripcion")
                                        response.write("</td>")
                                        response.write("<td>")
                                                  response.write rst("fechaapertura")
                                        response.write("</td>")
                                        response.write("</tr>")
                                        rst.movenext
                              loop
                              %>
                              </table>
                              <br/>
                    <%
                              //No hay cuentas
                              else
                    %>
                              <b>No se encontro ninguna cuenta.</b><br/>
                    <%
                              end if
                              rst.close
                              cnn.close
          %>
          <br/><i>Presione regresar para volver a las opciones</i>
          <do type="prev" label="Regresar"><prev/></do>
          </p>
          </card>
</wml>
```

7.3.8 Bloquear cuentas

Esta opción permite poner en estado “bloqueado” una ó mas cuentas del usuario, con la finalidad de no permitir que se puedan hacer transacciones con estas cuentas, para esto se pide que se escoja una cuenta del listado de cuentas que aparecen y luego se escoge la opción de “bloquear cuenta seleccionada”, para bloquear la cuenta. El archivo físico que permite seleccionar la cuenta es *pagCuentasVigentes.asp* y el archivo físico que ejecuta el proceso de bloqueo de la cuenta y muestra el resultado de la operación es *pagCambioEstadoCuenta.asp* y sus códigos fuentes pueden verse en las figuras 63 y 64 respectivamente.

Figura 61: Vista de la página para seleccionar una cuenta para ser bloqueada

Figura 62: Vista de la página con el resultado del proceso de bloqueo de la cuenta

Figura 63: Código Fuente de *PagCuentasVigentes.asp*

```
<% Response.buffer = true %>
```

Continuación

```
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="cuenta_vigente" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                        <%
```

```
                                dim cnn
                                dim rst
                                //Conecta a la base de datos
                                Call ConectarBDD()
                                strSQL = "select a.cuenta from cliente_cuenta a inner join cuenta b on
a.cuenta=b.cuenta " & _
                                "where b.status = 'VIG' and cliente ='" & session("charCliente") & "'"
                                rst.Open strSQL, cnn
                                if (not rst.eof=True) then
                                        %>
                                        <i>Cuentas Vigentes:</i>
                                        <br/>
                                        <select title="cuenta" name="cuenta">
                                        <%
                                        do while not rst.eof
                                                Response.write "<option value='" & rst(0) & "'>" &
rst(0) & "</option>"
                                                rst.movenext
                                        loop
                                        %>
                                        </select>

                                        <do type="accept" label="Bloquear Cuenta Seleccionada">
                                                <go method="post"
href="pagCambioEstadoCuenta.asp">
                                                        <postfield name="cuenta"
value="$(cuenta)"/>
                                                        <postfield name="estado" value="BLO"/>
                                                </go>
                                        </do>

                                        <br/>
                                <%
                                        //No hay cuentas
                                        else
                                %>
                                                <b>No se encontro ninguna cuenta vigente.</b><br/>
                                <%
                                        end if
                                        rst.close
                                        cnn.close
                        %>
                        <br/><i>Presione regresar para volver a las opciones</i>
                        <do type="prev" label="Regresar"><prev/></do>
                        </p>
                </card>
</wml>
```

Figura 64: Código Fuente de *PagCambioEstadoCuenta.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="res_cambio_estado_cuenta" title="Banco Virtual">
```

```
        <p align="center">
        </p>
        <p>
                        <%
                                dim cnn
                                dim rst
                                'Conecta a la base de datos
                                Call ConectarBDD()
                                strSQL = "exec pr_cambio_estado_cuenta '" & Request.Form("cuenta") &
"','" &  Request.Form("estado") & "'"
                                rst.Open strSQL, cnn
                                if (not rst.eof) then
                                        %>
                                        <big>
                                        <u><b>Resultado:</b></u>
                                        <br/>
                                        </big>
                                        <% Response.write(rst(0)) %>
                                        <br/>

                                <% end if
                                   rst.close
                                   cnn.close
                                %>
                        <do type="accept" label="Ver Menú">
                                                <go href="pagMenu.asp"/>
                                        </do>
                        </p>
                </card>
</wml>
```

7.3.9 Desbloquear cuentas

Esta opción permite poner en estado "vigente" una ó mas cuentas del usuario, con la finalidad de permitir que se puedan hacer transacciones con estas cuentas, para esto se pide que se escoja una cuenta del listado de cuentas que aparecen y luego se escoge la opción de "desbloquear cuenta seleccionada", para desbloquear la cuenta. El archivo físico que permite seleccionar la cuenta es *pagCuentasVigentes.asp* (ver código fuente en la figura 63) y el archivo físico que ejecuta el proceso de desbloqueo de la cuenta y muestra el resultado de la operación es *pagCambioEstadoCuenta.asp* y su código fuente se puede verse en las figura 67.

Figura 65: Vista de la página para seleccionar una cuenta para desbloquearla

Figura 66: Vista de la página con el resultado del proceso de desbloqueo de la cuenta

Figura 67: Código Fuente de *PagCuentasBloqueadas.asp*

```
<% Response.buffer = true %>
<%Response.ContentType = "text/vnd.wap.wml" %>
<!--#include file="pagConectar.asp" -->

<?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="cuenta_vigente" title="Banco Virtual">
        <p align="center">
        </p>
        <p>
                        <%
                                dim cnn
                                dim rst
                                //Conecta a la base de datos
                                Call ConectarBDD()
                                strSQL = "select a.cuenta from cliente_cuenta a inner join cuenta b
on a.cuenta=b.cuenta " & _
                                "where b.status = 'BLO' and cliente ='" & session("charCliente") &
"'"
                                rst.Open strSQL, cnn
                                if (not rst.eof=True) then
                                        %>
                                        <i>Cuentas Bloqueadas:</i>
                                        <br/>
                                        <select title="cuenta" name="cuenta">
                                        <%
                                        do while not rst.eof
                                                Response.write "<option value='" & rst(0) & "'>"
& rst(0) & "</option>"
                                                rst.movenext
                                        loop
                                        %>
                                        </select>

                                        <do type="accept" label="Habilitar Cuenta
Seleccionada">
                                                <go method="post"
href="pagCambioEstadoCuenta.asp">
                                                        <postfield name="cuenta"
value="$(cuenta)"/>
                                                        <postfield name="estado"
value="VIG"/>
                                                </go>
                                        </do>

                                        <br/>
                                <%
                                        //No hay cuentas
                                        else
                                %>
```

Continuación

```
                                        <b>No se encontro ninguna cuenta
bloqueada.</b><br/>
                                <%
                                        end if
                                        rst.close
                                        cnn.close
                        %>
                        <br/><i>Presione regresar para volver a las opciones</i>
                        <do type="prev" label="Regresar"><prev/></do>
                        </p>
                </card>
</wml>
```

CONCLUSIONES

1. El papel de WAP para el desarrollo de Internet móvil puede ser el mismo que jugaron en su día los navegadores para el desarrollo de Internet. En un futuro próximo se prevé que los usuarios de terminales móviles WAP puedan acceder a servicios similares a los que actualmente accede desde su residencia u oficina desde cualquier punto con cobertura móvil, aunque el protocolo WAP está principalmente destinado al comercio electrónico y a la consulta de información de carácter específico.

2. Se ha presentado la evolución de las redes de comunicaciones inalámbricas, explicando los factores que han influido dicha evolución, que responden en su mayoría a la oferta de nuevos servicios de telecomunicaciones, mas que por necesidad tecnológica. También se ha mostrado un resúmen de los nichos tecnológicos que pretenden cubrir las redes de 3ra y 4ta generación, mostrándose como las redes de 4ta generación vendrán a complementar mas que a substituir a las de 3ra generación, así como un análisis breve sobre las tecnologías TDMA y CDMA, lo que abre nuevas áreas de investigación y desarrollo en comunicaciones inalámbricas.

3. Es evidente la importancia del estudio de tecnologías relacionadas con la integración de Internet con la telefonía móvil, dado que estadísticamente existen 150 millones de usuarios de Internet mientras que existen 300 millones de teléfonos celulares.

Lo cual indica una gran oportunidad par hacer negocios enfocados en la política de encontrar una terminal en cualquier lugar y en cada persona, lo cual presenta nuevos desafíos.

Sin duda, WAP es una aproximación a lo que a integración se refiere, sin embargo no hay que dejar de lado que esta alternativa se encuentra operativa desde hace 2 años aproximadamente y aun no muestra una gran aceptación. Las empresas prestadoras de servicios atribuyen esto a la escasez de terminales que hagan mas amigable la navegación. Sin embargo no hay que dejar de lado que el desarrollo de equipos 3G esta totalmente en auge.

4. Se han detallado los diferentes mecanismos de seguridad que podrían emplearse para obtener seguridad extremo a extremo en los sistemas móviles, de los cuales el mas fácil de implementar, lo cual podría hacerlo mas viable, es la primera de las opciones, ya que hace uso de los protocolos de autenticación y cifrado actuales de GSM, haciendo que los costos sean menores ya que se ocupa la infraestructura utilizada por GSM, la segunda opción requiere confianza en el servidor WAP, además hace uso de certificados lo cual incrementa los costos, mientras que la tercera opción no requiere confianza en el servidor WAP pero la complejidad de administración del servidor es mayor ya que es un sistema centralizado.

   La cuarta y última opción es la mejor solución a nivel de seguridad ya que está basada en una aplicación que se encontrará almacenada en el teléfono móvil teniendo la oportunidad de implementar nuevos servicios; el detalle de estas opciones puede verse en la tabla II.

5. La evolución de WAP se verá condicionada por el desarrollo de los sistemas móviles de tercera generación UMTS, siendo previsible la aparición de soporte multimedia (música, voz, imagen de alta calidad, vídeo) a medida que UMTS permita su transmisión de forma fiable y con velocidades de transmisión elevadas, por otro lado, la similitud en el proceso de desarrollo de aplicaciones respecto a Internet fijo, junto con la multioperatividad de WAP con distintos sistemas móviles (*GSM, DECT, PHS, UMTS, Bluetooth*) permitirá que estos servicios se extiendan a gran velocidad La evolución no sólo supondrá un beneficio directo para las aplicaciones WAP por el importante aumento de ancho de banda que trae consigo, sino que además existirá integración con otros servicios como es el caso de GPS, lo que permitirá una personalización de contenidos sin precedentes: basada en la ubicación actual del terminal móvil, y, por ende, del usuario.

6. Las limitaciones de representación gráfica, así como de introducción de información desde una terminal móvil hacen que la adaptación de los contenidos de Internet para ser accesibles desde una terminal móvil obliguen a elaborar *scripts* específicos de filtrado de contenidos, así como de accesibilidad simplificada para la introducción de información: menús dinámicos, menús asistidos, etc. Se trata de soluciones transitorias, pero de utilidad, hasta que las posibilidades de los teléfonos móviles aumenten para ser equiparables a las de un ordenador personal.

7. Al momento de desarrollar aplicaciones, uno de los aspectos más notables de WAP es la posibilidad de reutilizar tecnologías maduras y suficientemente probadas como las existentes en el dominio WWW para ofrecer dichos servicios y resolver el problema de la generación

dinámica de contenidos, logrando de este modo un grado de personalización no contemplado hasta el momento, Además, por ser WAP una tecnología independiente del portador físico de acceso a la red inalámbrica, queda garantizada su vigencia para un futuro próximo, en el que existirá una evolución de las redes GSM actuales a redes GPRS y, posteriormente, UMTS.

RECOMENDACIONES

1. WAP es ideal para servicios de información donde el acceso a información a tiempo es crítico.

2. Asegurar que las políticas de seguridad están bien definidas y reforzadas en el entorno e-*business*, desde la seguridad en las aplicaciones de redes hasta los servidores y ordenadores portátiles. Definir e implementar individualmente políticas de seguridad en cada uno de los niveles, como se hace ordinariamente hoy en día, aunque suponga incrementar los costes y los problemas de complejidad, algo necesario ya que los procesos de negocios electrónicos deben ser seguros en su totalidad y en cada una de los niveles que conforman el sistema.

3. La seguridad en Internet es posible, y es recomendable contar con empresas especializadas en seguridad, para el análisis de necesidades y el mantenimiento y control de los niveles de seguridad

4. Para un desarrollo adecuado de aplicaciones basadas en WAP, debe procurarse vigilar el "peso" de la información, es decir, intentar mantener los tamaños por debajo del límite máximo, con ello se asegura que los dispositivos de baja potencia accedan al cotenido y se compense la lentitud de la conexión inalámbrica.

5. El contenido debe ser personalizado y que todos los servicios se acomoden a las preferencias de los usuarios, así como deben ser servicios inmediatos y que puedan obtenerse de una manera fácil y rápida.

BIBLIOGRAFÍA

1 ASP & WAP. **Descripción de los elementos WML.** http://www.programacion.com/asp/tutoriales/aspywap/wml2.htm. 25/11/2002.

2 Cortes, Angel. **El nuevo protocolo WAP 2.0**. http://facom.udp.cl/CEM/TDC/fichas/wap2.wap2.htm. 17/10/2002

3 Herramientas de desarrollo para soluciones WAP. **Nokia Wap Toolkit**. http://www.nokia.es/corporate/wap/sdk.html. 05/12/2002.

4 Las claves del futuro de la tecnología. **GRPS, caminando hacia la tercera generación**. 14/10/2002 http://expansiondirecto.com /tecnologia/ Informes/telefonia/grps.html. 14/10/2002.

5 Lee Meng, Wei y otros. ***Beggining WAP, WML & WMLScript***. 1era. Ed. Wrox Press Ltd, 2000. 628 pp.

6 Marquez, Santiago. **Ericsson Gateway 1.0**. http://wmlclub.com/tutoriales. 10/09/2002.

7 Navarro, Fausto. **Construyendo aplicaciones con WMLScript**. http://developer.earthweb.com/journal/techworkshop/051200_wmlscript.html. 15/12/2002.

8 Navarro, Fausto. **¿Qué es i-Mode?** http://movitienda.com/noticias/6069.htm. 16/10/2003.

9 Nieto Pérez, Iván. **Introducción al WAP.** http://www.elcodigo.net /tutoriales/wap/wap1.html. 15/08/2002.

10 Nuere Salgado, Leire. **M-Commerce** http://lafactoriadeinternet.com /trastienda/guia_31.shtml. 23/11/2002.

11 Pescador, Darío. **Tiembla WAP, i-Mode funciona**. http:// baquia.com/com/20000901/art00002.html. 10/11/2002.

12 Ríos Aguilar, Sergio. **Generación dinámica de contenidos WAP para terminales móviles**. http://www.aui.es/biblio /libros /mi2000/Sergio%20Rios.htm. 15/11/2002.

13 Teléfonos de última tecnología. **Historia del WAP.** http://www. terra.cl/especial/celulares/wap.cfm. 10/08/2002.

14 Van Der Henst, Christian. **WAP, el protocolo para aplicaciones inalámbricas.** http://www.maestrosdelweb. com/editorial /articulo.asp?wap. 03/08/2002.

15 WMLClub – Documentos. ***Wireless Aplication Protocol Architecture Specification***. http://wmlclub.com/docs/especwap2.0. 15/09/2002.