

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS ESCUELA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES



TESIS DE GRADO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

“Análisis y estudio de transacciones seguras con el Protocolo WAP y tecnologías para aplicaciones Web móvil”

APLICATIVO:

“Implementación y Desarrollo de una aplicación Wap-Commerce orientada al uso de tecnología celular.”

Autores:

MÓNICA ALEXANDRA RAMÍREZ VELASTEGUI
MARCELO ISMAEL GUDIÑO SILVA

Director: Ing. Jorge Caraguay

Ibarra, Febrero 2008

CERTIFICADO

Por medio del presente certifico:

Que los estudiantes de la Facultad en Ciencias Aplicada de la Escuela de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte, Srta. Mónica Alexandra Ramírez Velastegui CC 1002571485 y Sr. Marcelo Ismael Gudiño Silva CC 0201664042, son los autores intelectuales y materiales de la Tesis de Grado con el tema “Análisis y estudio de transacciones seguras con el Protocolo WAP y tecnologías para aplicaciones Web móviles”, esta certificación la confiero por haber desempeñado las funciones de Director de Tesis durante todo el tiempo que se ocupó en la elaboración y desarrollo de la mencionada tesis.

Ibarra, a 10 de Febrero de 2008

Ing. Jorge Caraguay Procel.

DEDICATORIA

Este trabajo va dedicado:

A los lectores en general esperando que el presente trabajo contribuya en parte al desarrollo tecnológico de la Universidad Técnica del Norte.

Mónica y Marcelo

AGRADECIMIENTO

A Dios, a nuestros padres, por su esfuerzo y sacrificio al ayudarnos a culminar con éxito nuestros estudios.

A la Universidad Técnica del Norte y en especial a la Facultad de Ingeniería en Ciencias Aplicadas por las facilidades brindadas para realizar el presente trabajo

Al personal docente de la Escuela de Ingeniería en Sistemas Computacionales por su incansable labor en beneficio de la juventud estudiosa del norte del país.

Al ingeniero Jorge Caraguay, que como Docente nos impartió sólidos conocimientos para nuestra carrera, motivándonos siempre a la investigación y como Director de Tesis fue el mentor de todo el proceso.

A la EMAPAI por facilitarnos sus instalaciones para la implantación del aplicativo de nuestra tesis.

Mónica y Marcelo

PRÓLOGO

El presente trabajo tiene como objetivo dar a conocer la arquitectura y los fundamentos teóricos del protocolo WAP; desarrollando aplicativos móviles, seleccionando la mejor plataforma existente en nuestro medio; realizando el análisis y estudio de las diferentes tecnologías y de los diversos usos que se puede dar en las empresas de nuestro medio.

En el Primer Capítulo se da un vistazo general del Protocolo WAP, se explica de una forma global su definición, su estructura, su funcionamiento y los servicios que ofrece el protocolo WAP.

En el Segundo Capítulo se estudia la capa de seguridad en las redes inalámbricas, la forma de funcionamiento de esta capa en la pila OSI de las redes móviles.

En el Tercer Capítulo se revisa la forma de obtener aplicaciones o transacciones seguras, analizando métodos de autenticación, encriptación, criptografía, algoritmos de protección de datos (por ejemplo el algoritmo RSA, encriptación mediante clave privada, encriptación mediante clave pública, entre otros.)

En el Cuarto Capítulo se analiza la arquitectura y diseño de la información móvil, los beneficios que prestan estas aplicaciones y el diseño de la arquitectura óptima para la realización de las aplicaciones móviles.

En el Quinto Capítulo se profundiza en los tipos de soluciones móviles existentes On-Line, Stand Alone y Smart Client, además tratamos una introducción a la telefonía móvil de nuestro medio.

En el Sexto Capítulo se estudia las tecnologías actuales de las aplicaciones móviles nativas, profundizamos especialmente en las plataformas Java y .Net, tanto en sus entornos de desarrollo, limitantes y alcances que nos ofrecen estas tecnologías.

En el Séptimo Capítulo estudiamos los estándares de diseño web-móvil implantados por la WAP 1.x y la WAP 2.x, aquí vemos las ventajas, los beneficios, la sintaxis del lenguaje; para implantar en nuestras aplicaciones.

En el Octavo Capítulo se estudia las tecnologías actuales de las aplicaciones móviles On-Line, especialmente en las plataformas Java, .Net y PHP, tanto en sus entornos de desarrollo, limitantes y alcances que nos ofrecen estas tecnologías.

En el Noveno Capítulo realizamos un análisis comparativo de las tecnologías existentes, aplicando elementos de evaluación escogemos las mejores plataformas para el desarrollo de los diferentes tipos de aplicativos.

En el Décimo Capítulo profundizamos en el desarrollo, la implantación y los resultados obtenidos con los tres aplicativos realizados, hacemos un estudio detallado de las fases de ingeniería de software de cada una de las aplicaciones.

En el último capítulo exponemos la hipótesis y su verificación, para finalizar proponemos nuestras conclusiones personales y recomendaciones del estudio.

En los anexos; complementamos con estudios realizados sobre los RFCs, las encuestas debidamente tabuladas que fueron realizadas tanto a empresas como a personas en general que apoyan a la implantación de este tipo de tecnología.

Esperamos que el presente trabajo, análisis, estudio, desarrollo e implantación que lo ponemos a su consideración: sea de beneficio a técnicos e investigadores y contribuya al desarrollo tecnológico de la Universidad Técnica del Norte, de la provincia y del país.

Los Autores

INTRODUCCIÓN

En esta Sociedad globalizada debemos mantenernos involucrados y conectados en el medio haciendo uso de los beneficios que nos traen las diferentes tecnologías existentes.

El uso de los celulares en nuestro medio es muy generalizado, las técnicas de marketing incluyen las promociones a través de estos medios. Por ello estamos seguros que crear un espacio que facilite la gestión comercial a través de la tecnología celular es de gran importancia, ya que se podría diversificar las formas de hacer negocios.

La transmisión de datos a través de canales móviles mediante el protocolo WAP ha adquirido una relevancia importante en la industria de la comunicación. Los medios de comunicación están manteniendo un ojo vigilante en la evolución de datos inalámbricos, y han puesto a la transmisión de datos inalámbricos como prioridad de su orden del día.

Dado que la tecnología celular esta convirtiéndose en un elemento más de la vida diaria, creemos que estas tecnologías deben incorporarse en las estrategias de comercio electrónico de las empresas de nuestro medio; además el abaratamiento de costos en la telefonía móvil capta cada vez más usuarios, lo que permite que involucremos esta

tecnología en el desarrollo de las aplicaciones de información y comercio electrónico específicamente.

Nuestro país en particular, no debería estar al margen de esta evolución en las comunicaciones, y las empresas deben evaluar cada vez más las opciones de aumentar la productividad, eficiencia y eficacia de sus negocios o servicios que prestan mediante esta tecnología; esta investigación apunta a enfocar el desarrollo de aplicaciones Wap como solución a la tendencia tecnológica y comercial.

De esta manera planteamos en nuestra tesis aplicaciones Wap para celulares con estándares seguros, que brinden un fácil acceso a la información, tengan una rápida y amigable interfaz. De esta manera esperamos crear una poderosa herramienta a la que el usuario podría acceder confiadamente a realizar transacciones de comercio, de negocios y/o de servicios, con la ventaja que lo haría desde cualquier lugar y en cualquier momento, utilizando la tecnología celular existente en nuestro medio.

RESUMEN

A medida que los usuarios se suman a la red, aparecen nuevas formas de conectarse que no necesariamente requieren de la presencia de una computadora. Ahora con la tecnología la mayoría de modelos de teléfonos celulares pueden navegar a través de páginas especialmente diseñadas para funcionar con este sistema. De esta manera podemos hacer una cantidad de cosas por medio de esta, como: mejorar la gestión de comercio, de negocios, de servicios o administrativa de empresas, hacer compras, consultar: información pública y privada de clientes, carteleras de cines y teatros, enviar correos, etc... y todo esto desde un teléfono celular!

En base a los resultados obtenidos al aplicar técnicas de investigación científica, se determinó que se podían desarrollar aplicaciones On-Line o Stand alone dependiendo de las necesidades planteadas tanto por las empresas como por los clientes, mediante la utilización de la tecnología WAP en nuestro medio.

En el presente trabajo se inicia con un estudio del protocolo WAP y las capas en los cuales es soportado WAE, WSP, WTP, WTLS, WDP. Luego se profundiza en la seguridad en la transmisión de datos encargada del WTLS y sus métodos de encriptación y compresión.

Después de una objetiva visión sobre el protocolo y sus estándares especificados en la OMA (Open Mobile Alliance) se pasa a estudiar las mejores políticas de diseño para lograr aplicaciones amigables y eficientes con temas como:

- ω Diseño centrado en el usuario

- ω Arquitectura de la información.

- ω Se estudia también las mejores prácticas propuestas por la W3C.

Continuamos con un estudio detallado las mejores tecnologías actuales de desarrollo tanto web-móvil como móvil instalables.

En el desarrollo web móvil se estudiará las principales tecnologías ASP.NET, JAVA y PHP y en stand alone J2ME como el único proveedor de servicios en este tipo de aplicaciones.

Se sigue con una comparación entre ventajas y desventajas entre cada una de las tecnologías y el estudio de sus fortalezas.

Se continúa con una práctica guía de desarrollo e implementación de aplicaciones ejemplos y manuales técnicos que darán una clara idea de cómo llevar el proceso desde el análisis de requerimientos hasta su instalación e implantación.

El crecimiento de los servicios de telecomunicaciones inalámbricas hace que el uso de telefonía móvil en nuestro medio sea parte de la vida diaria; y consideremos a este como una potente herramienta que nos permitirá estar conectados permanentemente.

'Internet WAP' establece una **nueva ruta de navegación** que permite que terminales móviles accedan a contenidos y servicios hasta ahora restringidos al 'Internet fijo'¹.

La **integración entre Internet y telefonía móvil** es posible gracias a la tecnología WAP, que aun siendo un protocolo, en realidad nos estamos refiriendo a la capacidad de entregar contenidos y servicios directamente en la pantalla de un dispositivo móvil.

La tecnología sigue avanzando a pasos agigantados y hay que saber cada día más; investigar y aprender para no quedarnos atrás.

¹ (Internet Fijo: Internet o red mundial que se accesa desde una PC de escritorio)

ÍNDICE DE CONTENIDO

DEDICATORIA	III
AGRADECIMIENTO	IV
PRÓLOGO	V
INTRODUCCIÓN	VIII
RESUMEN	X
ÍNDICE DE CONTENIDO	XIII
ÍNDICE DE FIGURAS	XVII
ÍNDICE DE TABLAS	XIX
MARCO TEÓRICO	1
CAPÍTULO I	1
PROTOCOLO WAP	1
1. INTRODUCCIÓN AL PROTOCOLO WAP	1
1.1. DEFINICIÓN DE WAP	1
1.1.1. Servicios que nos ofrece el Protocolo WAP.....	2
1.1.2. Estructura y funcionamiento del Protocolo WAP.....	3
1.2. PROTOCOLO WAP	5
1.2.1. Modelo WAP.....	5
1.2.2. Primitivas de Servicio.....	6
1.2.3. Principios Básicos de la arquitectura WAP.....	8
1.2.3.1. Capa de Aplicaciones (WAE).....	9
1.2.3.2. Capa de Sesión.....	12
1.2.3.3. Capa de Transacciones.....	13
1.2.3.4. Capa de Seguridad de Transporte.....	16
1.2.3.5. Capa de Datagramas.....	17
CAPÍTULO II	19
CAPA WSTL	19
2. SEGURIDAD EN REDES MÓVILES	19
2.1. HISTORIA DEL WTSL	19
2.1.1. Capa de Seguridad (WTSL).....	21
2.1.1.1. Las cuatro operaciones de criptografía.....	22
2.1.1.2. Las primitivas de servicio.....	24
2.1.2. WTLS Servicio de Transporte.....	27
2.1.2.1. SEC-Unitdata.....	27
2.1.2.2. SEC-Create.....	28
2.1.2.3. SEC-Exchange.....	31
2.1.2.4. SEC-Commit.....	31
2.1.2.5. SEC-Terminate.....	32
2.1.2.6. SEC-Excepción.....	32

2.1.2.7.	SEC-Create-Request.....	33
2.1.3.	Capa de Registros.....	34
2.1.3.1.	Compresión y Decompresión de Registros.....	34
2.1.3.2.	Protección de Payload(Carga explosiva) de Registro.....	35
2.1.4.	Protocolo Handshake (Negociación) de WTLS.....	35
CAPÍTULO III.....		39
TRANSACCIONES SEGURAS - ENCRIPCIÓN.....		39
3.	<i>PROTECCIÓN DE LAS TRANSMISIONES MEDIANTE ENCRIPCIÓN</i>	39
3.1.	CONCEPTOS BÁSICOS DE ENCRIPCIÓN.....	39
3.2.	CONCEPTO DE CRIPTOGRAFÍA.....	41
3.2.1.	Sistemas criptográficos.....	42
3.2.1.1.	Algoritmo RSA.....	43
3.2.2.	Como utilizar eficazmente la encriptación mediante clave privada.....	45
CAPÍTULO IV.....		47
ARQUITECTURA Y DISEÑO DE LA INFORMACIÓN MÓVIL.....		47
4.	<i>APLICACIONES MÓVILES</i>	47
4.1.	BENEFICIOS DE LAS APLICACIONES MÓVILES.....	47
4.2.	DISEÑO CENTRADO EN EL USUARIO.....	49
4.3.	ARQUITECTURA DE LA INFORMACIÓN MÓVIL.....	50
CAPÍTULO V.....		55
APLICACIONES MÓVILES CON TEGNOLOGÍA CELULAR.....		55
5.	<i>INTRODUCCIÓN A LA TELEFONÍA MÓVIL</i>	55
5.1.	HISTORIA DEL CELULAR.....	55
5.1.1.	Funcionamiento.....	56
5.1.2.	Tipos de redes de telefonía móvil.....	57
5.2.	SOLUCIONES ON-LINE.....	59
5.3.	SOLUCIONES STAND-ALONE.....	64
5.3.1.	TIPO DE CÓDIGO EJECUTABLE.....	65
5.3.1.1.	Código nativo.....	66
5.3.1.2.	Código manejado.....	67
5.4.	SOLUCIONES SMART-CLIENT.....	68
CAPÍTULO VI.....		70
TECNOLOGÍAS ACTUALES DE APLICACIONES MÓVILES NATIVAS.....		70
6.	<i>APLICACIONES MÓVILES NATIVAS</i>	70
6.1.	TECNOLOGÍAS MÓVILES.....	70
6.1.1.	Plataforma Java.....	73
6.1.1.1.	Introducción a J2ME.....	74
6.1.2.	MIDlets.....	76
6.1.2.1.	Herramientas de Desarrollo.....	84
6.1.3.	Plataforma .Net.....	86
6.1.3.1.	.NET Compact Framework.....	87
CAPÍTULO VII.....		89
STÁNDARES DE DISEÑO WEB – MÓVIL.....		89
7.	<i>ESTANDARES BASADOS EN LA WAP 1.X Y EN LA WAP 2.0</i>	89
7.1.	WML.....	89
7.1.1.	Introducción.....	89
7.1.2.	Ventajas:.....	91
7.1.3.	Desventajas:.....	92
7.1.4.	Características del Lenguaje WML.....	92
7.1.5.	Estructura de un documento WML.....	93
7.2.	XHTML.....	94
7.2.	Introducción.....	94
7.2.1.	Reglas Sintaxis.....	96

7.3.	WCSS.....	100
7.3.1.	Introducción.....	100
7.4.	AJAX EN LAS APLICACIONES WEB MÓVILES.....	106
CAPÍTULO VIII.....		109
TECNOLOGÍAS ACTUALES DE APLICACIONES WEB-MÓVILES.....		109
8.	<i>APLICACIONES WEB MÓVILES.....</i>	<i>109</i>
8.1.	COMPONENTES Y ESTRUCTURA DE UNA APLICACIÓN BÁSICA J2EE.....	109
8.2.	PHP.....	112
8.2.1.	Componentes y estructura de una aplicación básica PHP.....	112
8.3.	PLATAFORMA .NET.....	114
8.3.1.	Componentes y estructura de una aplicación básica de .Net.....	114
8.3.1.1.	Aplicaciones ASP.Net.....	115
8.3.1.2.	Qué es un WebForm.....	116
8.3.1.3.	ASP.NET en el desarrollo WEB Móvil.....	118
8.3.2.	Mecanismo de Renderizado Dinámico.....	119
8.3.2.1.	Introducción a los Mobile Internet Controls.....	120
CAPÍTULO IX.....		126
ANÁLISIS COMPARATIVO DE SOLUCIONES TECNOLÓGICAS ACTUALES.....		126
9.	<i>ELEMENTOS PARA LA EVALUACIÓN.....</i>	<i>126</i>
9.1.	APLICACIONES MÓVILES NATIVAS.....	127
9.1.1.	Análisis comparativo usando elementos de juicio.....	127
9.2.	APLICACIONES WEB-MÓVILES.....	128
CAPÍTULO X.....		131
APLICATIVOS.....		131
10.	<i>DESARROLLO E IMPLEMENTACIÓN DEL APLICATIVO.....</i>	<i>131</i>
10.1.	PROTOTIPO WAP-COMMERCE APLICACIÓN Web-Móvil.....	131
10.1.1.	DESCRIPCIÓN DE LA PROPUESTA.....	131
10.1.2.	Análisis previos.....	133
10.1.2.1.	Metodología del desarrollo.....	133
10.1.2.2.	Flujo de la información.....	134
10.1.2.3.	Diseño de tablas.....	135
10.1.2.4.	Funcionamiento del aplicativo wap-commerce.....	136
10.1.3.	Desarrollo.....	137
10.1.4.	Implantación.....	140
10.1.5.	Manual de usuario.....	147
10.2.	APLICATIVO EMAPAIWAP.....	155
10.2.1.	DESCRIPCIÓN DE LA PROPUESTA.....	155
10.2.2.	Análisis previos.....	157
10.2.2.1.	Metodología del desarrollo.....	157
10.2.2.2.	Flujo de la información.....	157
10.2.2.3.	Diseño de tablas.....	159
	Modelo de entidad relación.....	159
10.2.3.	Desarrollo.....	160
10.2.4.	Implantación.....	162
10.2.5.	Seguridades.....	169
10.2.6.	Manual de usuario.....	169
10.3.	PROTOTIPO VPC Stand-Alone (VENTAS POR CELULAR).....	178
10.3.1.	DESCRIPCIÓN DE LA PROPUESTA.....	178
10.3.2.	Análisis previos.....	180
10.3.2.1.	Metodología del desarrollo.....	180
10.3.2.2.	Flujo de la información.....	180
10.3.2.3.	Diseño de tablas.....	181
10.3.2.4.	Funcionamiento del aplicativo VPC.....	182
10.3.3.	Desarrollo.....	183
10.3.4.	Implantación.....	184
10.3.5.	Manual de usuario.....	187

CAPÍTULO XI	194
CONCLUSIONES Y RECOMENDACIONES	194
11. <i>VERIFICACIÓN DE LA HIPÓTESIS</i>	194
11.1. HIPÓTESIS PLANTEADA	194
11.2. VERIFICACIÓN.....	195
11.3. CONCLUSIONES.....	196
11.4. RECOMENDACIONES.....	199
BIBLIOGRAFÍA.....	201
ANEXOS.....	207
GLOSARIO	208
RFC Y ESTÁNDARES USADOS COMO BASE TEÓRICA.....	220
ESTUDIO DE MERCADO	230
<i>RESULTADOS DE LAS ENCUESTAS ORIENTADAS A LOS CLIENTES</i>	230
<i>RESULTADOS DE LAS ENCUESTAS ORIENTADAS A LAS EMPRESAS</i>	234
MANUAL TÉCNICO -- APLICATIVO EMAPAIWAP --	239
MANUAL TÉCNICO - - APLICATIVO VPC - -	253
MANUAL TÉCNICO -- APLICATIVO WAP-COMMERCE - -.....	297

ÍNDICE DE FIGURAS

FIGURA 1. ARQUITECTURA BÁSICA WAP.....	4
FIGURA 2. ARQUITECTURA WAP.....	5
FIGURA 3. COMPONENTES DEL CLIENTE DE WAE.....	11
FIGURA 4. EJEMPLO DE INTERCAMBIO DE PRIMITIVAS ENTRE CAPA SESIÓN Y TRANSACCIÓN	14
FIGURA 5. ARQUITECTURA DEL PROTOCOLO INALÁMBRICO DE DATAGRAMAS	18
FIGURA 6. SECUENCIA EN TRANSMISIÓN CLIENTE SERVIDOR	26
FIGURA 7. CRIPTOGRAFÍA DE CLAVE PRIVADA	40
FIGURA 8. CRIPTOGRAFÍA DE CLAVE PÚBLICA.....	41
FIGURA 9. POLÍTICA DE “MIENTRAS MAS SIMPLE MEJOR”	51
FIGURA 10. ARQUITECTURA DE LA INFORMACIÓN MÓVIL	53
FIGURA 11. TECNOLOGÍAS MÓVILES USADAS EN LA ACTUALIDAD	70
FIGURA 12. ARQUITECTURA PLATAFORMA JAVA.....	73
FIGURA 13. MODELO DE PERFILES DE J2ME.....	76
FIGURA 14. ESTADO DE UN MIDLET.....	80
FIGURA 15. CON .NET PODEMOS ACCEDER A EQUIPOS MÓVILES EN EL MERCADO, YA SEA CON APLICACIONES ONLINE O APLICACIONES EJECUTABLES.	86

FIGURA 16. ARQUITECTURA DEL COMPACT FRAMEWORK DE MICROSOFT.....	88
FIGURA 17. CÓDIGO BÁSICO DE EN WML.....	93
FIGURA 18. NAVEGADOR O EMULADOR MÓVIL.....	94
FIGURA 19. CÓDIGO DE XHTML.....	105
FIGURA 20. CÓDIGO DE WCSS.....	105
FIGURA 21. NAVEGADOR O EMULADOR MÓVIL.....	106
FIGURA 22. PAGINA DE EJEMPLO USANDO XHTML	110
FIGURA 23. VISTA EN EL DISPOSITIVO MÓVIL.....	110
FIGURA 24. SCRIPT AL LADO DEL SERVIDOR.....	111
FIGURA 25. TECNOLOGÍAS MÓVILES USADAS EN LA ACTUALIDAD	113
FIGURA 26. ESQUEMA DEL FUNCIONAMIENTO DE LA APLICACIÓN WEB.....	114
FIGURA 27. FLUJO DE INFORMACIÓN EN EL SISTEMA MÓVIL EMAPAIWAP.....	157

ÍNDICE DE TABLAS

TABLA 1. TIPOS DE PRIMITIVAS DE SERVICIOS	7
TABLA 2. NOMENCLATURA DE PARÁMETROS DE SERVICIOS.....	8
TABLA 3. EJEMPLO DE LA ESTRUCTURA DE BLOQUE CIFRADO	24
TABLA 4. LAS PRIMITIVAS DE SERVICIO.....	25
TABLA 5. PARÁMETROS DE PRIMITIVA UNITDATA.	27
TABLA 6. PARÁMETROS DE PRIMITIVA CREATE.	29
TABLA 7. PARÁMETROS DE PRIMITIVA EXCHANGE.....	31
TABLA 8. PARÁMETROS DE PRIMITIVA TERMINATE.....	32
TABLA 9. PARÁMETROS DE PRIMITIVA EXCEPTION.....	32
TABLA 10. PARÁMETROS DE PRIMITIVA CREATE – REQUEST	33
TABLA 11. ÍTEMS DE LA NEGOCIACIÓN DE UNA SESIÓN SEGURA	35
TABLA 12. ATRIBUTOS Y DESCRIPCIÓN EN FAR Y JAD.....	84
TABLA 13. DIFERENCIA ENTRE WEB Y MOBILE CONTROLS.	121
TABLA 14. ATRIBUTO DE LOS CONTROLES MOBILE WEB CONTROLS	121
TABLA 15. CÓDIGO GENERADO EN UNA WEBMOBILEFORMS.....	122
TABLA 16. CÓDIGO GENERADO DE UN WEBMOBILEFORMS SERVER. ...	123
TABLA 17. CÓDIGO GENERADO DE UN PANELMOBILECONTROL SERVER	123
TABLA 18. CÓDIGO GENERADO DE UN TEXTVIEWCONTROL SERVER....	123
TABLA 19. CÓDIGO GENERADO DE UN LISTCONTROL SERVER	124
TABLA 20. CÓDIGO GENERADO DE UN SELECTIONLIST SERVER	124
TABLA 21. CÓDIGO GENERADO DE UN PHONECALL SERVER.....	124
TABLA 22. CÓDIGO GENERADO DE UN LINK SERVER	125

MARCO TEÓRICO

CAPÍTULO I

PROTOCOLO WAP.



1. INTRODUCCIÓN AL PROTOCOLO WAP

1.1. DEFINICIÓN DE WAP

WAP es el acrónimo de *Wireless Application Protocol*, que podríamos traducir como Protocolo de Aplicación Inalámbrico. La tecnología WAP es realmente un estándar impulsado por la industria del sector de las telecomunicaciones con el objetivo de proporcionar un sistema avanzado de servicios de Internet para dispositivos móviles.

“El Protocolo de Aplicaciones Inalámbricas (WAP) es una serie de especificaciones que define a un protocolo para: la comunicación entre aplicaciones de servidor y clientes, la forma en que se pueden usar funciones específicas en el dispositivo móvil en la posición actual del cliente, y un lenguaje reforzado de marcas para aplicaciones inalámbricas, el lenguaje de marcas inalámbrico (WML)”².

² WAP forum, www.wapforum.com

En la actualidad el estándar WAP 2.0³ utiliza el lenguaje XHTML. MP para las aplicaciones.

WAP es escalable, permitiendo así todo lo que se refiere a teléfonos móviles, Palmtops (Ordenadores de mano), portátiles, y cualquier otro acceso a redes con dispositivos sin conexión física. El protocolo incluye especificaciones para las capas de la pila OSI de sesión y de transporte, así como funcionalidades de las aplicaciones para 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. Como cualquier estándar, las ventajas son múltiples a la hora de desarrollar aplicaciones, fabricar terminales o estructurar la red.

1.1.1. Servicios que nos ofrece el Protocolo WAP

El protocolo WAP se puede implementar no sólo sobre un terminal telefónico digital celular actual, sino en los de 3ª generación (UMTS) o en los inalámbricos (DECT), para ofrecer servicios de datos que van más allá de los que hoy se pueden conseguir o con prestaciones mejoradas.

Los servicios más inmediatos que se prevé se ofrezcan con WAP, son:

- ω Acceso a la información general disponible en Internet.
- ω Acceso al correo electrónico.

³ OMA www.openmobilealliance.org/index.html

- ω Acceso a bases de datos en las Intranets (información corporativa, de administración y de gestión).
- ω Noticias breves (financiera, deportiva, meteorológica, horarios, etc.).
- ω Directorios (páginas amarillas, etc.).
- ω Banca y comercio electrónico. (Aplicaciones E-Commerce)
- ω Juegos diversos.
- ω Entre otras aplicaciones.

1.1.2. Estructura y funcionamiento del Protocolo WAP

El modelo de programación WAP es similar al modelo de programación de la Web. Esto brinda muchos beneficios para la comunidad de desarrolladores tales como, trabajar con un modelo de programación familiar, de comprobada arquitectura, y con la habilidad en herramientas comunes (tales como servidores Web, herramientas XML, etc). La optimización y extensiones de este modelo han sido hechas para corresponder a las características del entorno inalámbrico. Donde es posible, los estándares existentes han sido adoptados o han sido usados como punto de partida para la tecnología WAP.

El contenido y las aplicaciones WAP son especificados en un conocido conjunto de formatos de contenido basado en los de la Web. El contenido es transportado usando un conjunto de protocolos de comunicación estándar basados en los de la Web. La terminal inalámbrica del usuario es coordinada por un micro navegador análogo al navegador estándar de la Web.

La tecnología WAP se basa en 3 elementos ⁴:

- ω Cliente: el cliente será el micro-navegador de un Terminal móvil. Este será el equivalente al navegador de un PC en el Internet "fijo" que todos conocemos.

- ω Pasarela: la pasarela constituye la interfaz entre la red inalámbrica y la red física. Es lo que se conoce como el WAP Gateway.

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

En la Figura 1 mostramos la arquitectura básica WAP.

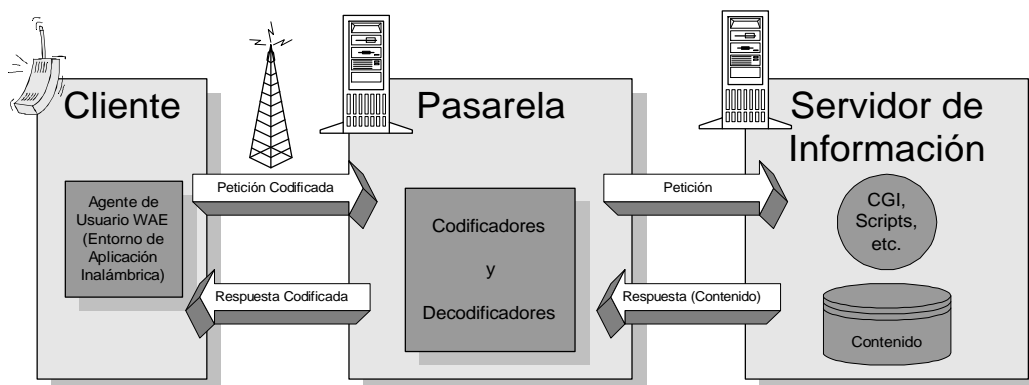


Figura 1. Arquitectura básica WAP⁵

⁴ El Protocolo Inalámbrico de Aplicaciones de LUIS ALBERTO VIELBA VILLEGAS

1.2. PROTOCOLO WAP

1.2.1. Modelo WAP.

Una vez analizada la estructura y funcionamiento del protocolo WAP, estudiaremos la arquitectura WAP. La arquitectura WAP está pensada para proporcionar un “entorno escalable y extensible para el desarrollo de aplicaciones para dispositivos de comunicación móvil”. Para ello, se define una estructura en capas, en la cual cada capa es accesible por la capa superior así como por otros servicios y aplicaciones a través de un conjunto de interfaces muy bien definidos y especificados⁶. En la Figura 2 se define la estructura de capas.

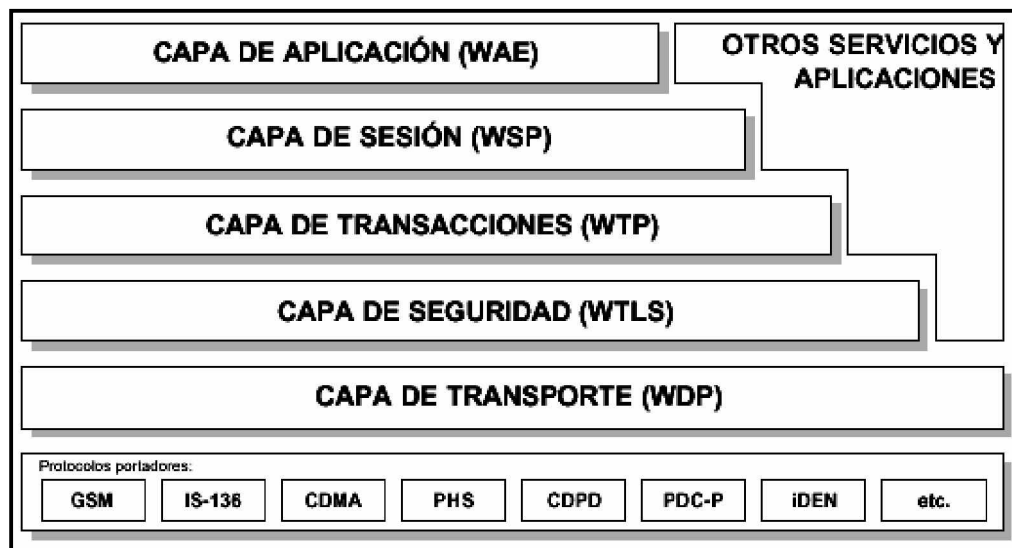


Figura 2. Arquitectura WAP⁷

⁵ <http://elcodigo.net/tutoriales/wap/wap1.html#punto2>

⁶ Arquitectura WAP Capas de arquitectura , Componentes en la red, GSM, GPRS de FRANKLIN GARCIA MARIN

⁷ CURSO DE DOCTORADO COMUNICACIONES MÓVILES DIGITALES “El Protocolo inalámbrico de aplicaciones WAP”

1.2.2. Primitivas de Servicio.

Primitivas de servicio representa el intercambio lógico de información entre la capa y las capas adyacentes a esta.

Se han definido un conjunto de primitivas de servicio para permitir la comunicación entre las capas integradas dentro del equipo cliente y las capas integradas en el equipo servidor.

A continuación en la Tabla1 analizaremos las primitivas en general.

Tipo	Abreviación	Descripción
<i>Request</i>	req	Se utiliza cuando una capa superior solicita un servicio de la capa inmediatamente inferior
<i>Indication</i>	Ind	Una capa que solicita un servicio utiliza este tipo de primitiva para notificar a la capa inmediatamente superior de las actividades relacionadas con su par, o con el proveedor del servicio

<i>Response</i>	Res	Este tipo de primitiva se utiliza para reconocer la recepción de la primitiva de tipo <i>Indication</i> de la capa inmediatamente inferior
<i>Confirm</i>	cnf	La capa que proporciona el servicio requerido utiliza este tipo de primitiva para notificar que la actividad ha sido completada satisfactoriamente.

Tabla 1 Tipos de Primitivas de Servicios⁸

Las primitivas de servicio usan tablas donde se definen qué parámetros son obligatorios y cuales no dependiendo de las diferentes tipos de primitivas:

A continuación en la Tabla2 describiremos la nomenclatura a usarse de aquí en adelante en el estudio de las primitivas.

M	Representa que el parámetro es obligatorio (Mandatorio)
C	Representa que el parámetro es condicional dependiendo del valor de otros parámetros.

⁸ CURSO DE DOCTORADO COMUNICACIONES MÓVILES DIGITALES “El Protocolo inalámbrico de aplicaciones WAP”

O	Representa que el parámetro es opcional dependiendo del usuario puede omitirse.
P	Representa que el parámetro es una opción del proveedor de servicio.
*	Representa que es determinado por la capa de protocolos mas baja.
()	Representa que el valor del parámetro es idéntico al valor del parámetro correspondiente del procedimiento del servicio primitivo

Tabla 2 Nomenclatura de Parámetros de Servicios⁹

1.2.3. Principios Básicos de la arquitectura WAP.

La pila de protocolos WAP comparte muchas características similares a Internet.

WAP divide el conjunto de funciones necesarias para asegurar el funcionamiento del sistema, en subconjuntos o capas funcionales basadas en criterios de homogeneidad de las funciones dentro de una misma capa. Estas capas son:

- ω Capa de Aplicación (WAE o Wireless Application Environment)
- ω Capa de Sesión (WSP o Wireless Session Protocol)
- ω Capa de Transacciones (WTP o Wireless Transaction Protocol)
- ω Capa de Seguridad en Transporte (WTLS o Wireless Transport Layer Security)

⁹ CURSO DE DOCTORADO COMUNICACIONES MÓVILES DIGITALES “El Protocolo inalámbrico de aplicaciones WAP”

ω Capa de Transporte (WDP o Wireless Datagram Protocol)

1.2.3.1. Capa de Aplicaciones (WAE)

La capa de aplicaciones WAE o *Entorno Inalámbrico de Aplicación (WAE)* es como su nombre lo dice un entorno de aplicación de propósito general basado en la combinación del WWW (*World Wide Web*) y tecnologías de Comunicaciones Móviles.

Este entorno incluye un *micro navegador*, que posee las siguientes funcionalidades:

- Para clientes que soportan WAP 1.x
 - ω Un lenguaje denominado WML similar al HTML, pero optimizado para su uso en terminales móviles, aunque el lenguaje XML tiene estándares más generales para todo tipo de aplicaciones.
 - ω Un lenguaje denominado *WMLScript*, similar al *JavaScript* (esto es, un lenguaje para su uso en forma de *Script*)
 - ω Un conjunto de formatos de contenido, que son un conjunto de formatos de datos bien definidos entre los que se encuentran imágenes, entradas en la agenda de teléfonos e información de calendario.

- Para clientes que soportan WAP 2.x
 - ω Un lenguaje denominado XHTML MP; es un subconjunto de XHTML, esta basado en XML y HTML.

 - ω Un conjunto de formatos de contenido; son un conjunto de formatos de datos bien definidos entre los que se encuentran imágenes, entradas en la agenda de teléfonos e información de calendario.

De esta forma, la arquitectura del Entorno Inalámbrico de Aplicaciones (en adelante WAE) está enfocado principalmente sobre los aspectos del cliente de la arquitectura WAP, esto es, de los puntos relacionados con los agentes de usuario que es todo aquel software o dispositivo que interpreta un contenido soportado por el dispositivo móvil (XHTML o WML). Esto es debido a que la parte que más interesa de la arquitectura es aquella que afecta principalmente a los terminales móviles, esto es, a aquellos puntos en los cuales van a estar ejecutándose los diversos agentes de usuario.

Basándonos en esta arquitectura, vamos a profundizar un poco más en los componentes de este Entorno Inalámbrico de Aplicación. Tal y como podemos observar en la Figura 3, se divide en dos partes, dos capas lógicas:

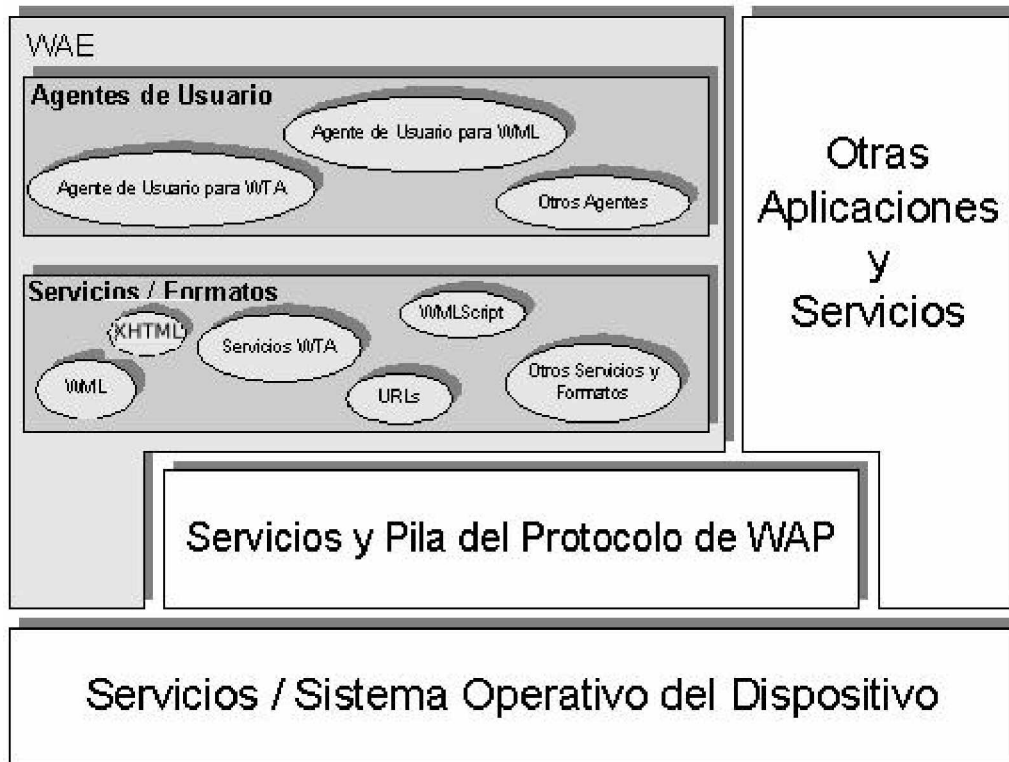


Figura 3. Componentes del Cliente de WAE

- ω Los Agentes de Usuario, que incluye aquellos elementos como navegadores, agendas telefónicas, editores de mensajes, etc.

- ω Los Servicios y Formatos, que incluyen todos aquellos elementos y formatos comunes, accesibles a los Agentes de Usuario, tales como XHTML, WML, WMLScript, formatos de imagen, etc.

Los dos Agentes de Usuario más importantes son el Agente de Usuario para WML y el Agente de Usuario para WTA.

1.2.3.2. Capa de Sesión

El *Protocolo Inalámbrico de Sesión* constituye la capa que se sitúa por debajo de la capa de Aplicación, proporcionando las siguientes funcionalidades:

- ω Establecer una conexión fiable entre el cliente y el servidor, y liberar esta conexión de una forma ordenada.

- ω Ponerse de acuerdo en un nivel común de funcionalidades del protocolo, a través de la negociación de las posibilidades.

- ω Intercambiar contenido entre el cliente y el servidor utilizando codificación compacta.

- ω Suspender y recuperar la sesión.

Este protocolo ha sido definido únicamente para el caso de la navegación. Esta implementación está realizada para el establecimiento de una conexión sobre la base de un protocolo compatible con HTTP1.1.

De esta forma, se han definido un conjunto de primitivas de servicio que permiten un intercambio lógico de información entre la capa de Sesión y capas adyacentes, para

permitir la comunicación entre la capa de sesión integrada dentro del equipo cliente y la capa de sesión integrada en el equipo servidor.

1.2.3.3. Capa de Transacciones

El *Protocolo Inalámbrico de Transacción* se establece para proporcionar los servicios necesarios que soporten aplicaciones de “navegación” (del tipo petición/respuesta), a este tipo de aplicaciones lo vamos a denominar como transacción. Este protocolo se sitúa por encima del *Protocolo Inalámbrico de Datagramas* y, de forma opcional, de la *Capa Inalámbrica de Seguridad de Transporte*, que serán estudiados posteriormente.

Las características de este protocolo las vamos a describir a continuación:

§ Proporciona tres clases de servicios de transacción:

- ω Clase 0: mensaje de solicitud no seguro, sin mensaje de resultado.
- ω Clase 1: mensaje de solicitud seguro, sin mensaje de resultado.
- ω Clase 2: mensaje de solicitud seguro, con, exactamente, un mensaje de resultado seguro.

§ La seguridad se consigue a través del uso de identificadores únicos de transacción, asentimientos, eliminación de duplicados y retransmisiones.

§ Seguridad opcional usuario a usuario.

§ De forma opcional, el último asentimiento de la transacción puede contener algún tipo de información adicional relacionada con la transacción, como medidas de prestaciones, etc.

§ Se proporcionan mecanismos para minimizar el número de transacciones que se reenvían como resultado de paquetes duplicados.

§ Se permiten las transacciones asíncronas.

Al igual que en el protocolo inalámbrico de sección, podemos decir que en este protocolo también existen primitivas de servicios que sustentan la comunicación entre dos capas de transacciones situadas en dos equipos distintos.

En la Figura 4 vamos a ver la concatenación de Primitivas de Servicio de Sesión y de Transacción para el caso de una petición-respuesta:

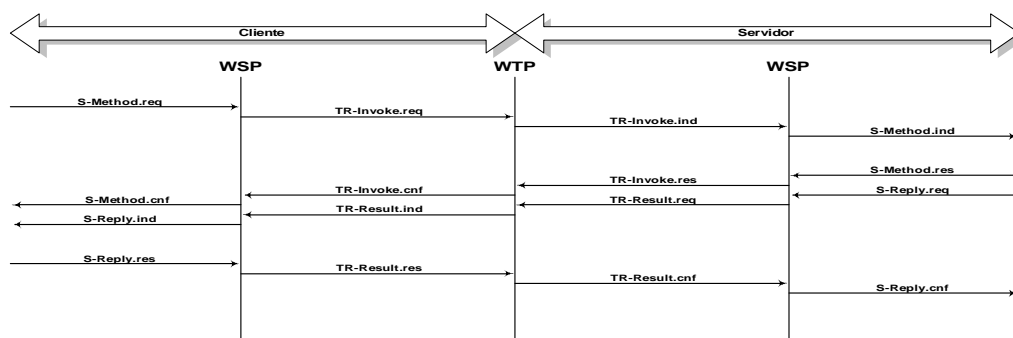


Figura 4. Ejemplo de intercambio de primitivas entre capa Sesión y Transacción ¹⁰

¹⁰ CURSO DE DOCTORADO COMUNICACIONES MÓVILES DIGITALES “El Protocolo inalámbrico de aplicaciones WAP”

A continuación las principales características de este protocolo:

- ω Transferencia de Mensajes. En este protocolo podemos distinguir dos tipos de mensajes: mensajes de datos y mensajes de control. Los mensajes de datos transportan únicamente datos de usuario, mientras que los mensajes de control se utilizan para los asentimientos, informes de error, etc. pero sin transportar datos de usuario.
- ω Retransmisión hasta el asentimiento, se utiliza para la transferencia fiable de datos desde un proveedor WTP a otro, en caso que haya pérdida de paquetes.
- ω Asentimiento de usuario, permite al usuario de este protocolo, confirmar cada mensaje recibido por el proveedor WTP.
- ω Información en el Último Asentimiento. Se permite, así pues, enviar información en el último, y únicamente en el último, asentimiento de una transacción. De esta forma, se puede enviar, por ejemplo, información del rendimiento proporcionado por el sistema durante la transacción realizada, etc.
- ω Concatenación y Separación. Podemos definir concatenación como el proceso de transmitir múltiples Unidades de Datos del Protocolo (PDU) de WTP en una Unidad de Datos del Servicio (SDU) de la red portadora,

mientras que separación es el proceso de separar múltiples PDUs de un único SDU (esto es, el proceso inverso al anterior).

1.2.3.4. Capa de Seguridad de Transporte

La *Capa Inalámbrica de Seguridad de Transporte (WTLS)*, constituye una capa modular, que depende del nivel de seguridad requerido por una determinada aplicación. Esta capa proporciona a las capas de nivel superior de WAP una interfaz de servicio de transporte seguro, que lo resguarde de una interfaz de transporte inferior.

El principal objetivo de esta capa es proporcionar privacidad, integridad de datos y autenticación entre dos aplicaciones que se comuniquen. Adicionalmente, la WTLS proporciona una interfaz para el manejo de conexiones seguras.

Al igual que los protocolos anteriores este posee primitivas de servicio que sustentan la comunicación entre dos capas situadas en dos equipos distintos.

Este protocolo tiene las siguientes características:

- ω Integridad de los datos. Este protocolo asegura que los datos intercambiados entre el terminal y un servidor de aplicaciones no han sido modificados y no es información corrupta.

- ω Privacidad de los datos. Este protocolo asegura que la información intercambiada entre el terminal y un servidor de aplicaciones no puede ser entendida por terceras partes que puedan interceptar el flujo de datos.

- ω Autenticación. Este protocolo contiene servicios para establecer la autenticidad del terminal y del servidor de aplicaciones.

- ω Denegación de servicio para datos que no han sido suficientemente verificados.

- ω Adicionalmente, el WTLS puede ser utilizado para la realización de comunicación segura entre terminales, por ejemplo para operaciones de comercio electrónico entre terminales móviles.

1.2.3.5. Capa de Datagramas

El *Protocolo Inalámbrico de Datagramas* (WDP) ofrece un servicio consistente al protocolo (Seguridad, Transacción y Sesión) de la capa superior de WAP, comunicándose de forma transparente sobre uno de los servicios portadores disponibles.

Este protocolo ofrece servicios a los protocolos superiores del estilo:

- ω Direccionamiento por número de puerto,
- ω Segmentación
- ω Re-ensamblado opcional
- ω Detección de errores opcional,

de modo que se permite a las aplicaciones de usuario funcionar de forma transparente sobre distintos servicios portadores disponibles. Para ello, se plantea una arquitectura de protocolo como el que se muestra en la Figura 5:

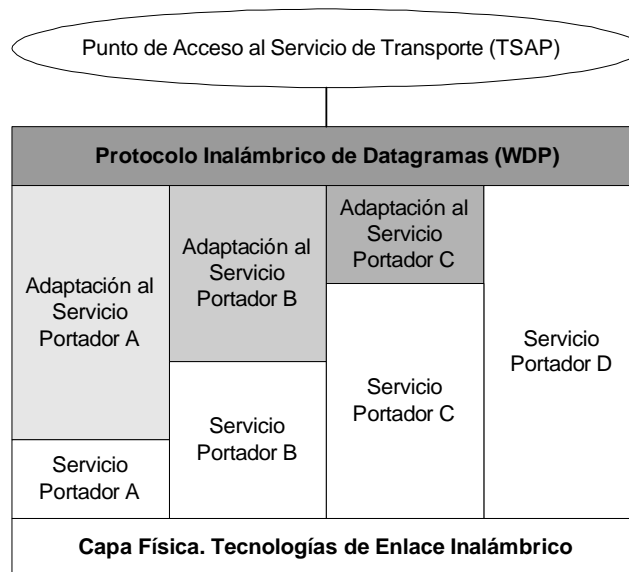


Figura 5. Arquitectura del Protocolo Inalámbrico de Datagramas¹¹

¹¹ CURSO DE DOCTORADO COMUNICACIONES MÓVILES DIGITALES “El Protocolo inalámbrico de aplicaciones WAP”

CAPÍTULO II

CAPA WSTL

2. SEGURIDAD EN REDES MÓVILES



2.1. HISTORIA DEL WSTL.

SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocolo que asegura la comunicación del tráfico web que usa HTTP, desde que un cliente envía un requerimiento, el servidor (web-server) responde con un documento (su pagina web). Es quizá el protocolo de seguridad más usado.

En la actualidad su uso más común es en el comercio electrónico. Donde una tienda electrónica vende un producto a cualquier usuario, el principal propósito de asegurar la conexión es que el usuario (cliente) envíe su información confidencial (Ej: número de tarjeta de crédito) de manera segura. Para tal caso solo es necesario que la tienda tenga un certificado digital, que generalmente compra a una autoridad certificadora reconocida, aunque esto no es necesario técnicamente.

SSL permite varios niveles de seguridad que pueden ser configurados desde el servidor los cuales son:

- ω Primero que ningún lado de la comunicación tenga certificado digital, en este caso ambos lados no podrán autenticarse.

- ω El segundo nivel y el mas usado, es que un lado (generalmente el servidor) tenga certificado digital, en este caso el cliente puede verificar la autenticidad del servidor pero no al contrario (B2C).

- ω Finalmente existe la posibilidad que ambos lados tengan certificado digital y los dos lados puedan autenticarse (B2B).

Los algoritmos usados en los la autenticación son RSA, o DSA. SSL provee servicio de autenticación de mensajes con un MAC (Message Authentication Code), integridad del tráfico con el mismo MAC, y confidencialidad de la información con un método de cifrado simétrico que puede ser RC4, TDES, AES, etc.

SSL inventado por Netscape en 1994 la versión 1 no fue liberada, sin embargo el mismo año liberaron la versión 2, para 1995 apareció la versión 3, desde 1997 pasó a manos de Internet Engineering Task Force (IETF) para conformar la versión 3.1 llamada TLS. ***En 1998 se acondiciona para inalámbricos llamado ahora WTLS.*** En la actualidad se cuenta con la versión libre OpenSSL que permite implementar SSL de manera gratuita.

Existe una versión comercial llamada SSLplus que soporta curvas elípticas¹².

¹² <http://computacion.cs.cinvestav.mx>

2.1.1. Capa de Seguridad (WTSL¹³)

La *Capa Inalámbrica de Seguridad de Transporte (WTLS)* es un protocolo basado en el estándar SSL, utilizado en el entorno Web para proporcionar seguridad en la realización de transferencias de datos.

Este protocolo ha sido especialmente diseñado para los protocolos de transporte de WAP y optimizado para ser utilizado en canales de comunicación de banda estrecha.

Para este protocolo se han definido las siguientes características:

- ω *Integridad de los datos.* Este protocolo asegura que los datos intercambiados entre el terminal y un servidor de aplicaciones no han sido modificados y no es información corrupta.

- ω *Privacidad de los datos.* Este protocolo asegura que la información intercambiada entre el terminal y un servidor de aplicaciones no puede ser entendida por terceras partes que puedan interceptar el flujo de datos.

- ω *Autenticación.* Este protocolo contiene servicios para establecer la autenticidad del terminal y del servidor de aplicaciones.

¹³

Wireless Transport Layer Security ó Capa Inalámbrica de Seguridad de Transporte

Adicionalmente, el WTLS puede ser utilizado para la realización de comunicación segura entre terminales, por ejemplo en el caso de operaciones de comercio electrónico entre terminales móviles.

La *Capa Inalámbrica de Seguridad de Transporte* (de aquí en adelante WTLS), constituye una capa modular, que depende del nivel de seguridad requerido por una determinada aplicación. Esta capa proporciona a las capas de nivel superior de WAP una interfaz de transporte seguro¹⁴.

El protocolo WTLS usa criptografía para el envío de datos, a continuación se expone las principales características y los atributos de encriptación.

2.1.1.1. Las cuatro operaciones de criptografía.

Digitally signed(firma digital), stream cipher encryption(criptación de codificación de flujo), block cipher encryption(criptación bloque cifrado), and public key encryption (criptación de clave pública) y son designadas como digitally-signed(firma digital), stream-ciphered(flujo cifrado), block-ciphered(bloque cifrado), and public-key-encrypted(criptación de clave pública), respectivamente.

El proceso de encriptación de datos es especificado dependiendo de la clave apropiada designada antes de la especificación del tipo de datos.

¹⁴ (www.infoteca.com.uy , UNIVERSIDAD DE VALLADOLID E.T.S.I.TELECOMUNICACIÓN. Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática)

La clave de criptografía esta dispuesta por el estado de sesión actual.

Digitally signed (Firma digital): En la firma digital one-way son usadas funciones hash como entrada para el algoritmo de la firma.

El elemento de la firma digital es codificado como un vector, donde el largo es especificado por el algoritmo y por la clave de la firma.

Stream-ciphered (Flujo Cifrado): En Stream-ciphered, el texto plano es exclusivo con una idéntica cantidad de salida generada para la secuencia de criptografía con una clave generada con un número pseudorandómico.

Block –ciphered (Bloque cifrado): En la encriptación block cipher, todos los blocks de texto plano se encripta como un block de ciphertext. Todos los datos encriptados por block cipher son hechos en modo CBC (Cipher Block Chaining), y todos los ítems block-ciphered serán exactamente múltiplos del largo de cipher block.

Public-key-encrypted(encriptación de clave pública): En la encriptación de clave pública, el algoritmo de clave pública es usado para encriptar los datos de tal manera que se lo pueda desencriptar solo obteniendo la clave raíz que género la clave pública.

El elemento encriptado con clave pública es codificado en un opaque vector <0..216 -1> donde el vector es especificado por la firma del algoritmo y una clave.

Como se muestra en la Tabla 3 un ejemplo de la estructura de bloque cifrado:

```
block-ciphered struct {  
  
    uint8 field1;  
  
    uint8 field2;  
  
    digitally-signed opaque hash[20];  
  
} UserType;
```

Tabla 3 Ejemplo de la estructura de bloque cifrado¹⁵

El contenido hash es usado como entrada para la firma del algoritmo, luego la estructura entera encriptada en un bloque Cifrado.

El largo de esta estructura, en bytes podría ser exactamente múltiplo del largo del Chipre-block.

2.1.1.2. Las primitivas de servicio

A continuación en la Tabla4 Las primitivas de servicio que sustentan la comunicación entre dos capas situadas en dos dispositivos distintos son:

¹⁵ (WAP-261-WTLS-20010406-a)

Primitiva	Descripción
<i>SEC-Unitdata</i>	Esta primitiva se utiliza para intercambiar datos de usuario entre los dos participantes. Sólo puede ser invocada cuando existe previamente una conexión segura entre las direcciones de transporte de los dos participantes.
<i>SEC-Create</i>	Esta primitiva se utiliza para iniciar el establecimiento de una conexión segura.
<i>SEC-Exchange</i>	Esta primitiva se utiliza en la creación de una conexión segura si el servidor desea utilizar autenticación de clave pública o intercambio de claves con el cliente.
<i>SEC-Commit</i>	Esta primitiva se inicia cuando el <i>handshake</i> ¹⁶ se completa y cualquiera de los equipos participantes solicita cambiar a un nuevo estado de conexión negociado.
<i>SEC-Terminate</i>	Esta primitiva se utiliza para finalizar la conexión.
<i>SEC-Exception</i>	Esta primitiva se utiliza para informar al otro extremo sobre las alertas de nivel de aviso.
<i>SEC-Create-Request</i>	Esta primitiva se utiliza por el servidor para solicitar al cliente que inicie un nuevo <i>handshake</i> .

Tabla 4 Las primitivas de servicio

La secuencia para establecer una sesión segura es mostrada en la Figura 6.

¹⁶ Término utilizado para denominar el intercambio de primitivas entre cliente y servidor con el objetivo de establecer una sesión segura. Posteriormente veremos este intercambio de primitivas.

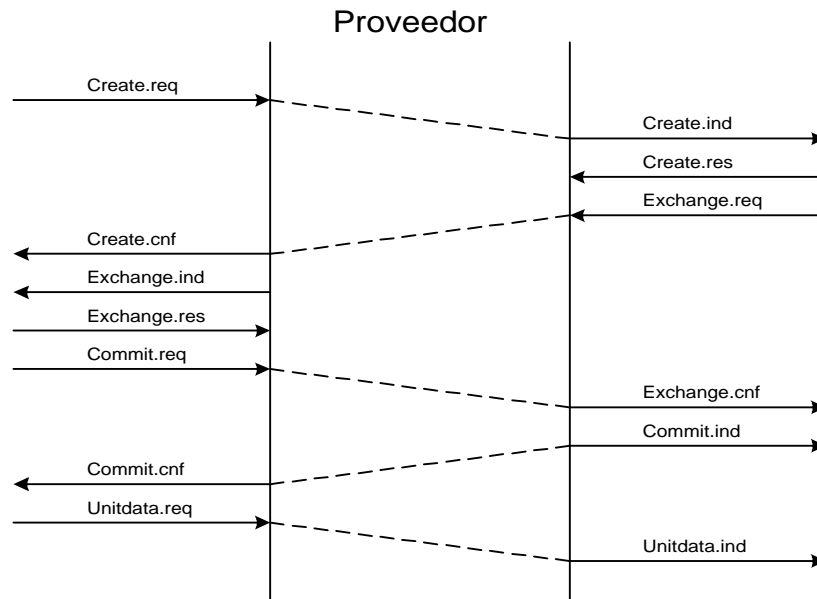


Figura 6. Secuencia en transmisión cliente servidor¹⁷

- ω El proceso inicia cuando un cliente intenta iniciar una comunicación y su aplicación envía una primitiva Create.req.
- ω El servidor recibe con un Create.ind que indica la recepción de una petición de iniciar una conexión segura y responde con un Create.res para confirmar el Create.ind y envía un Exchange.req que propone la creación de la conexión segura.
- ω El cliente primero confirma el Create.res con un Create.cnf y luego informa de la recepción Exchange.req con un Exchange.ind luego envía el Exchange.res para confirmar a su par y a continuación se inicia un Commit.req porque se a completado el **hanshake** y se puede pasara un nuevo estado.

¹⁷ (WAP-261-WTLS-20010406-a – pag 20)

- ω El servidor genera un Exchage.cnf y genera un Commit.ind

- ω El cliente recibe la indicación y envía el primer Unidata.req para empezar el intercambio de datos.

- ω El servidor genera un Unitdata.ind.

2.1.2. WTLS Servicio de Transporte

Descripción de las Primitivas y su funcionamiento.

2.1.2.1. SEC-Unitdata

Esta primitiva es usada para intercambiar datos entre los pares usuarios. SEC-Unitdata puede ser únicamente invocada cuando hay una conexión segura existente entre las direcciones de pares (pares son las partes involucradas en la conexión) existente. Ver Tabla5

Parameter	Primitive	SEC-Unitdata	
		<i>req</i>	<i>ind</i>
Source Address		M	M(=)
Source Port		M	M(=)
Destination Address		M	O(=)
Destination Port		M	O(=)
User Data		M	M(=)

Tabla 5 Parámetros de primitiva Unitdata¹⁸.

¹⁸ WAP-261-WTLS-20010406 – pag 21

- ω Source Address.- Identifica el origen de la petición.

- ω Source Port.- Identifica el puerto desde donde ha sido enviado este mensaje.

- ω Destination Address.- Identifica el par hacia donde se dirige esta unidad de dato.

- ω Destination Port.- Identifica el puerto a cual este mensaje es enviado.

- ω User Data.- Es el dato que se desea transmitir¹⁹.

La administración de la conexión del WTLS permite al cliente conectarse con el servidor y acordar el protocolo a usar.

La seguridad en la conexión consiste en varios pasos y el cliente o el servidor pueden interrumpir la negociación en cualquier momento (Ejemplo: Si algún parámetro enviado por el par es inaceptable). La negociación puede incluir parámetros de seguridad (Ejemplo: algoritmos criptográficos y extensas claves) claves de intercambio y autenticación. Ambos servicios de servidor o cliente pueden terminar la conexión en cualquier momento.

2.1.2.2. SEC-Create

¹⁹ (OMA Open Mobile Alliance technical_wap2_0-20021106 WAP-261-WTLS-20010406-a)

Esta primitiva es usada para iniciar una conexión segura y establecerla. Ver Tabla6.

Parameter	Primitive	SEC-Create			
		<i>req</i>	<i>ind</i>	<i>res</i>	<i>cnf</i>
Source Address	M	M(=)			
Source Port	M	M(=)	-	-	
Destination Address	M	O(=)	-	-	
Destination Port	M	O(=)			
Client Identities	O	C(=)	-	-	
Proposed Key Exchange Suites	M	M(=)	-	-	
Proposed Cipher Suites	M	M(=)	-	-	
Proposed Compression Methods	M	M(=)	-	-	
Sequence Number Mode	O	C(=)	M	M(=)	
Key Refresh	O	C(=)	M	M(=)	
Session Id	O	C(=)	M	M(=)	
Selected Key Exchange Suite	-	-	M	M(=)	
Selected Cipher Suite	-	-	M	M(=)	
Selected Compression Method	-	-	M	M(=)	
Server Certificate	-	-	O	C(=)	

Tabla 6 Parámetros de primitiva create²⁰.

- ω Source Address.- Identifica la dirección de origen.
- ω Source Port.- Identifica el puerto desde el cual se envía el mensaje.
- ω Destination Address.- Identifica la dirección de destino.
- ω Destination Port.- Identifica el puerto de destino.

²⁰ (WAP-261-WTLS-20010406 – pag 20)

- ω Client Identities.- Identifica el origen sin importar la manera de transporte. Este parámetro puede ser usado por el servidor para buscar el correspondiente certificado de cliente.

- ω Proposed Key Exchange Suites.- Incluye la clave de intercambio propuesta por el cliente.

- ω Proposed Cipher Suites.- Incluye la clave cifrada propuesta por el cliente.

- ω Proposed Compression Methods.- Incluye los métodos de compresión propuestos por el cliente.

- ω Sequence Number Mode.- Define como la secuencia numérica será usada en esta conexión segura.

- ω Key Refresh.- Define la frecuencia con que la encriptación y protección de claves son renovadas dentro de la conexión segura.

- ω Sesión Id.- Identifica la sesión segura. Esta es única por servidor.

- ω Selected Key Exchange.- Identifica las claves de intercambio escogidas por el servidor.

- ω Selected Cipher Suite.- Identifica las claves de cifrado escogidas por el servidor.

- ω Selected Compression Method.- Identifica los métodos de compresión escogidas por el servidor.
- ω Server Certificate.- Es la clave pública que certifica al servidor.

2.1.2.3. SEC-Exchange

Esta primitiva es usada en la creación de una conexión segura si el servidor desea ejecutar autenticación de clave pública o intercambio de claves con el cliente.

El cliente certifica si esta clave pública está certificada. Ver Tabla7

Parameter	Primitive	SEC-Exchange			
		<i>req</i>	<i>ind</i>	<i>res</i>	<i>cnf</i>
Client Certificate		-	-	M	M(=)

Tabla 7 Parámetros de primitiva Exchange²¹.

2.1.2.4. SEC-Commit

Esta primitiva es iniciada cuando el contrato es completado y ambos pares requieren escoger en una nueva negociación de estado de conexión.

²¹ (WAP-261-WTLS-20010406 – Pag 21)

2.1.2.5. SEC-Terminate

Esta primitiva es usada para terminas la conexión. Ver Tabla8

Parameter	Primitive	SEC-Terminate	
		<i>req</i>	<i>Ind</i>
Alert Description		M	M(=)
Alert Level		M	M(=)

Tabla 8 Parámetros de primitiva terminate²².

- ω Alert Descripción identifica la razón por la cual se ha terminado la sesión.
- ω Alert Level define si tanto la sesión (fatal) o solo conexión (crítica) es terminada.

2.1.2.6. SEC-Excepción

Esta primitiva es usada para informar acerca de advertencias del nivel de alertas. Ver Tabla9

Parameter	Primitive	SEC-Exception	
		<i>req</i>	<i>ind</i>
Alert Description		M	M(=)

Tabla 9 Parámetros de primitiva exception²³.

²² (WAP-261-WTLS-20010406 – Pag 21)

²³ (WAP-261-WTLS-20010406 – Pag 21)

- ω Alert Descripción identifica la causa de la alerta.

2.1.2.7. SEC-Create-Request

Esta primitiva es usada por el servidor que requiere que el cliente inicie un nuevo contrato. Ver Tabla10

Parameter	Primitive	SEC-Create-Request	
		<i>req</i>	<i>ind</i>
Source Address		O	C(=)
Source Port		O	C(=)
Destination Address		O	C(=)
Destination Port		O	C(=)

Tabla 10 Parámetros de primitiva create – request²⁴

- ω Source Address.- Identifica el origen.
- ω Source Port.- Identifica el puerto de origen.
- ω Destination Address Identifica a la dirección cliente a la cual se ha enviado.

Este parámetro es necesario cuando la primitiva es usada en estado de sesión nulo.

²⁴ (WAP-261-WTLS-20010406 – pag 21)

- ω Destination Port.- Identifica el puerto de destino²⁵.

2.1.3. Capa de Registros

Esta capa recibe datos no interpretados de las capas superiores los cuales están en forma de block de caracteres con un máximo de $2^{16}-1$.

2.1.3.1. Compresión y Decompresión de Registros.

Todos los registros son comprimidos usando el algoritmo de compresión definido en el actual estado de conexión.

Siempre hay un algoritmo de compresión activo; sin embargo, inicialmente es definido como null.

El algoritmo de compresión traslada de la estructura WTLSPlainText en la estructura WTLSCompressed. Esto significa que el fragmento WTLSPlainText es comprimido y luego copiado. Otros campos (tal como el fragmento de largo) son actualizados si se necesita.

²⁵ (OMA Open Mobile Alliance technical_wap2_0-20021106 WAP-261-WTLS-20010406-a)

2.1.3.2. Protección de Payload (Carga explosiva) de Registro.

La encriptación y las funciones MAC trasladan de la estructura `WTLSCompressed` en una cifrada `WTLSCiphertext`.

La descriptación invierte el proceso.

2.1.4. *Protocolo Handshake (Negociación) de WTLS*

El Protocolo de Negociación esta compuesto por 3 sub-protocolos los cuales son usados para permitir a los pares que acuerden sobre los parámetros de seguridad para la capa de registro, autenticación, instancia de los parámetros de seguridad, y reportar condiciones de error de cada uno.

Este Protocolo es responsable de la negociación de una sesión segura. Ver Tabla 11

Item	Description
Session Identifier	An arbitrary byte sequence chosen by the server to identify an active or resumable secure session.
Protocol Version	WTLS protocol version number.
Peer Certificate	Certificate of the peer. This element of the state may be null.
Compression Method	The algorithm used to compress data prior to encryption.
Cipher Spec	Specifies the bulk data encryption algorithm (such as null, RC5, DES, etc.) and a MAC algorithm (such as SHA-1). It also defines cryptographic attributes such as the <code>mac_size</code> .
Master Secret	20-byte secret shared between the client and server.
Sequence Number Mode	Which sequence numbering scheme (off, implicit, or explicit) is used in this secure connection.
Key Refresh	Defines how often some connection state values (encryption key, MAC secret, and IV) calculations are performed.
Is Resumable	A flag indicating whether the secure session can be used to initiate new secure connections.

Tabla 11 Ítems de la negociación de una sesión segura²⁶

²⁶ (WAP-261-WTLS-20010406)

Estos ítems son usados para crear los parámetros de seguridad para ser usada por la capa de registros cuando protege los datos de aplicación.

Muchas conexiones de seguridad pueden ser instanciadas usando la misma sesión segura a través de la característica de reanudación del Protocolo de negociación WTLS.

Los parámetros de la sesión segura son producidos por el protocolo WTSL Handshake (Contrato), el cual opera sobre WTLS Capa de Registro. Cuando un cliente y un servidor WTLS empiezan a comunicarse, acuerdan la versión del protocolo, seleccionan los algoritmos de criptografía, opcionalmente otra autenticación, y usan técnicas de encriptación de clave-pública para generar el modo de acceso secreto.

El protocolo de negociación involucra los siguientes pasos:

Intercambio de mensaje hello para acordar en los algoritmos,

Intercambio de valores randómicos.

Intercambio de parámetros de criptografía necesarios para permitir al cliente y servidor para acordar la pre llave secreta.

Intercambio de certificados e información de criptografía para permitir al cliente y servidor que se autentiquen.

Generación de llave-maestra secreta de la pre-llave y el intercambio de valores randómicos.

Proveer parámetros de seguridad a la capa de registro.

Permitir al cliente y al servidor verificar que su par fue calculado con los mismos parámetros de seguridad y que el contrato ocurrió sin saboteo de algún ataque.

Estos objetivos son archivados por el protocolo de contrato, el cual se resume de la siguiente manera:

El cliente envía un mensaje hello al cual el servidor debe responder con un mensaje server hello, o en caso contrario un error fatal va a ocurrir y la conexión segura fallara.

Los mensajes hello tanto del cliente como de l servidor son usados para establecer la seguridad entre el cliente y el servidor.

Los mensajes hello establecen los siguientes atributos:

- ω Protocol Version
- ω Key Exchange Suite
- ω Cipher Suite
- ω Cipher Suite

- ω Key Refresh
- ω and Sequence Number Mode
- ω Y adicionalmente dos valores randómicos son generados

A continuación del mensaje hello, el servidor enviara su certificado, esto es si está autenticado.

Adicionalmente, puede ser enviada una clave de servidor como mensaje, si esta es requerida.

El servidor puede solicitar un certificado del cliente, si es apropiado para la clave de intercambio seleccionado.

Ahora el servidor enviará un mensaje hello hecho, indicando que la FACE mensaje-hello del contrato esta completa correctamente.

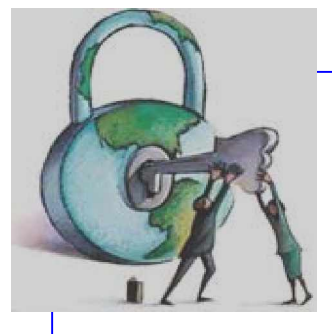
Después de esto el servidor esperará por una respuesta del cliente.

El cliente procede iniciar el envío de datos de aplicación²⁷.

²⁷ (OMA Open Mobile Alliance technical_wap2_0-20021106 WAP-261-WTLS-20010406-a)

CAPÍTULO III

TRANSACCIONES SEGURAS - ENCRIPCIÓN



3. PROTECCIÓN DE LAS TRANSMISIONES MEDIANTE ENCRIPCIÓN

3.1. CONCEPTOS BÁSICOS DE ENCRIPCIÓN.

La encriptación es una técnica que nos permite transformar un mensaje inicial en otro con el mismo significado pero con la diferencia que este está en una codificación diferente es decir que no será legible ni entendible fácilmente para cualquiera a menos que se lo descifre con un proceso reverso al que se usó para la encriptación.

Desde tiempos pasados la encriptación ha sido utilizada por el hombre para comunicarse evitando que información valiosa pueda llegar a manos de alguien que no debería tener acceso a esta. Entonces se inventaban nuevos símbolos los cuales eran conocidos solo por las partes que se comunicaban, con esto aunque el mensaje fuera robado no podía ser entendido.

En la actualidad con el avance de la tecnología las comunicaciones han evolucionado tanto hasta globalizarse en una forma exponencial, llegando a usar el medio inalámbrico para efectuar la transmisión de mensajes. La importancia de estos mensajes es igual o mayor que en tiempos pasados ya que hoy en día se transmite

mensajes en los cuales están códigos de cuentas bancarias, mensajes de secretos de estado, secretos de procesos empresariales, personales privados , etc.

Si se empleara métodos similares al pasado para tener diferentes símbolos o lenguajes entre cada par de individuos que deseen comunicarse seria casi imposible con el sin número de comunicaciones que se realizan a diario entre millones de personas.

La criptografía entra a escena con sus modelos criptográficos de clave pública y privada varios algoritmos complejos que logran simplificar la manera que nuestros mensajes sean secretos y solo puedan ser comprendidos por su destinatario, todo esto en un ambiente informático en el cual el uso de computadores nos facilita el trabajo pesado.

La principal diferencia entre la encriptación pública y privada es que en la privada cada par debe tener su única clave privada que les servirá tanto para encriptar como para desencriptar y si uno de los dos desea comunicarse con alguien mas debe tener una clave diferente así con cualquier otro llegando a crearse un gran problema al desear comunicarse seguramente con miles de personas se debería crear miles de claves privadas. Un ejemplo en la Figura 7

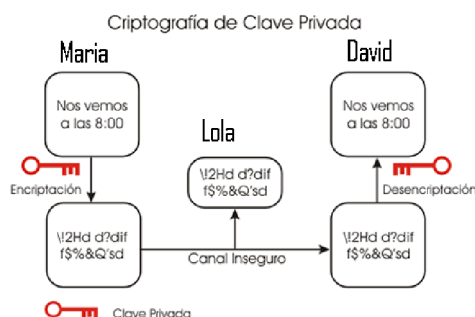


Figura 7. Criptografía de clave privada²⁸

²⁸ http://www.diphuelva.es/contenido_basico.asp?idContenido=777

En contrario con lo que sucede con la clave pública en la que un usuario tiene una única clave pública con la que cualquiera puede usar para enviarle mensajes a este. Ver Figura 8

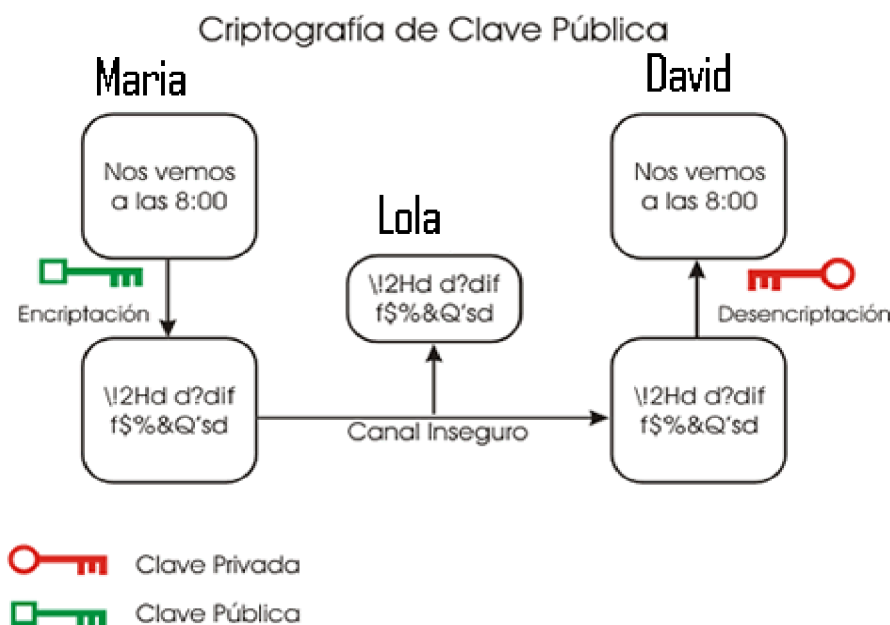


Figura 8. Criptografía de clave pública²⁹

3.2. CONCEPTO DE CRIPTOGRAFÍA

Criptografía es el proceso de diseñar sistemas basados en técnicas matemáticas para obtener una comunicación segura en canales que no lo son. La criptografía da lugar a diferentes tipos de sistemas criptográficos que permiten asegurar los principios de la seguridad informática: confidencialidad, integridad, disponibilidad, autenticación y no repudio de emisor y receptor. Así pues, los protocolos de seguridad permiten llevar al

²⁹ http://www.diphuelva.es/contenido_basico.asp?idContenido=777

cabo las metas de seguridad, pero la criptografía es la principal herramienta matemática de la que se sirven dichos protocolos para brindar los servicios necesarios que protegen tanto a los datos como a la comunicación.

3.2.1. Sistemas criptográficos

Los componentes de un sistema criptográfico o criptosistema son cinco:

1. El espacio de caracteres del texto original M .
2. El espacio de caracteres del texto cifrado C .
3. El espacio de la clave de cifrado K .
4. Una familia de transformaciones de cifrado $E_k; M \rightarrow C$.
5. Una familia de transformaciones de descifrado $D_k; C \rightarrow M$.

Cada transformación de cifrado E_k está definida por un algoritmo de cifrado E_k , que es común al tipo de transformación de la familia, y una clave K que la distingue del resto de transformaciones. Lo mismo sucede para las transformaciones de descifrado D_k . Para una K dada debe cumplirse que D_k sea inversa de E_k , y por tanto que $D_k(E_k(M))=M$ para cualquier texto original M .

Todo criptosistema ha de cumplir tres requisitos generales:

1. Las transformaciones de cifrado y descifrado han de ser eficientes para cualquier clave.
2. El sistema tiene que ser fácil de usar.
3. La seguridad del sistema sólo debe depender de lo bien guardada que esté la clave, y no de lo bien preservado que se encuentre el algoritmo, que, antes bien, debe ser público.

A continuación se dará una clara visión de lo que es el sistema criptográfico RSA y el funcionamiento del mismo. El motivo de estudio de este algoritmo es porque la capa anteriormente vista WTLS lo usa como sistema criptográfico interno

3.2.1.1. Algoritmo RSA.

El algoritmo de clave pública RSA fue creado en 1978 por Rivest, Shamir y Adlman matemáticos del MIT y es el sistema criptográfico asimétrico más conocido y usado.

Estos señores se basaron en el artículo de Diffie-Hellman sobre sistemas de llave pública, crearon su algoritmo y fundaron la empresa RSA Data Security Inc., que es actualmente una de las más prestigiosas en el entorno de la protección de datos.

El RSA es un algoritmo de encriptación que nos permite tener privacidad en cuanto a quien puede leer nuestros textos se basa en una simple teoría de funciones, que usa una clave pública y una privada este es el secreto del éxito de este algoritmo.

Se basa en algunas operaciones matemáticas con las cuales codifica nuestro texto

Primero genera un número primo muy grande en este caso (E) que viene a ser la clave pública luego con este número el algoritmo genera otro primo muy grande la clave privada (D).

Vamos los pasos lógicos en los que se guía el algoritmo:

1.- Al decodificar un mensaje codificado es obvio que se obtendrá el mensaje.

$$M=D(E(M))$$

2.- D Y E ya fueron explicadas la forma de obtenerlas.

3.- Si revelamos E de forma pública todos podrán codificar el mensaje pero solo el usuario que tenga la D o sea la clave privada podrá recuperar el mensaje.

Para encriptar con RSA podríamos proceder de la siguiente manera:

Al aplicar la clave pública sobre determinado texto o mensaje en cualquier tipo que se este transmitiendo (con las cabeceras MIME se puede transportar a través de la web varios y diferentes tipos de archivos como gráficos: jpg, bmp, gif. Archivos de MS Office o de Open Office como: doc, odt, ppt. Etc) al aplicar el algoritmo con su clave a cada carácter o byte del cual este conformado el archivo convirtiéndolo en un mensaje

encriptado. El proceso de desencriptación es aplicar el mismo algoritmo pero con la clave privada a cada byte y volver a obtener el byte original.

3.2.2. Como utilizar eficazmente la encriptación mediante clave privada.

El concepto de criptosistemas de clave pública fue introducido en 1976 por Diffie y Hellman.

Con este nuevo método cada usuario tiene un par de claves. Una la mantiene secreta (privada), que es la que se corresponde con D_k mientras que la otra la pública (la E_k).

Veamos a continuación cómo se puede transmitir un mensaje cifrado sin comprometer su integridad por un canal inseguro. Supongamos, por ejemplo, que Maria quiere enviar un mensaje a su amigo David y no desea que nadie más lo pueda leer. Maria posee una clave pública E_{maria} y una privada D_{maria} , y David tiene las correspondientes E_{david} y D_{david} .

Para enviar un mensaje M , Maria lo encripta con la clave pública de David, E_{david} , y se lo envía David por el canal inseguro. De esta manera sólo David lo podrá leer con su clave privada D_{david} , David ya que sólo él conoce esa clave. Se cumplirá la relación $D_{david}(E_{david}(M)) = M$, y el texto quedará inteligible.

No obstante este esquema no garantiza que el emisor del mensaje M sea Nuria, pues cualquiera podría encriptar el mensaje con la clave pública de David y enviárselo bajo un nombre falso. De hecho, Laura, que se lleva muy mal con Nuria y quiere destruir su amistad con David tiene intención de hacer pasar un falso mensaje M' como si fuera de Nuria. ¿Es posible evitar esta situación con nuestro sistema de clave pública? La respuesta es afirmativa, pero sólo con un tipo de criptosistema de clave pública se lo use de una manera eficaz. Veamos, pues, cuál sería el esquema que se debería seguir. Si prescindimos por un momento de la necesidad de privacidad, Maria podría mandar el mensaje M a David encriptado con su clave privada D_{maria} . Cuando éste le llegue a David, lo podrá desencriptar con la clave pública E_{maria} y comprobar por tanto su autenticidad. Esto es posible porque existe conmutatividad entre las claves D y E , es decir, se cumple la relación $D(E(M)) = E(D(M))$ (donde los subíndices se han omitido por motivos de claridad). Pero esto no garantiza la privacidad del mensaje. Sin embargo, para satisfacer esta nueva condición no hay más que encriptarlo posteriormente con la clave pública de David, enviando $E_{\text{david}}(D_{\text{maria}}(M))$. Sin embargo, habrá que tener la precaución que los espacios de cifrado y de texto original sean iguales entre sí. Por ejemplo, los dos espacios podrían ser el alfabeto castellano.

CAPÍTULO IV

ARQUITECTURA Y DISEÑO DE LA INFORMACIÓN MÓVIL

4. APLICACIONES MÓVILES



4.1. BENEFICIOS DE LAS APLICACIONES MÓVILES.

Las aplicaciones móviles presentan un gran reto en el mundo de las telecomunicaciones, y este está vinculado con la creación de servicios, con la finalidad de prestar mejores servicios que son de interés para los usuarios finales o para el cliente.

Para hacer un análisis de los beneficios que nos presentan las aplicaciones móviles nos centraremos en los tres tipos de miembros vinculados con la telefonía inalámbrica en el desarrollo de este tipo de aplicaciones.

- ω Usuario Final
- ω Operadores o Empresas
- ω Desarrolladores de las aplicaciones y contenido

Para los usuarios.- Los teléfonos celulares son las herramientas dominantes de las comunicaciones y al mismo tiempo, la Internet es una plataforma privilegiada para la información. Al adoptar un protocolo común (WAP) el usuario final es el que más se beneficiará ya que se le proporcionarán más servicios de valor agregado, los cuales

serán de fácil acceso y fáciles de utilizar directamente desde cualquier dispositivo inalámbrico.

Es decir que puede permitir el acceso a la información, cuando quiera y en cualquier lugar. Librando la información de las restricciones de un escritorio o Pc cercanos, las personas pueden recuperar, interactuar, o intercambiar la información.

Tomando en cuenta que este tipo de aplicaciones proporciona mayor conectividad. Ya que en la actualidad alrededor de 1/3 de la población del planeta tiene acceso al Internet a través de un dispositivo móvil, considerado que este número es dos veces mayor que el número de computadoras personales conectadas (PCs). Además se espera que para el 2010, la mitad de la población del planeta tendrá el acceso al Internet a través de un dispositivo móvil.

Operadores o Empresas.- Los operadores podrán diferenciarse del resto de sus competidores comerciales al lanzar servicios especiales, como por ejemplo, servicios E-commerce, servicios bancarios, compra-venta de acciones y servicios de directorio, entre otros, mediante las soluciones On-Line.

Adicionalmente, el protocolo permite personalizar diferentes aplicativos en teléfonos celulares, desarrollando las aplicaciones Stand-Alone, personalizadas a las necesidades del cliente y de la empresa. Esto incrementará los ingresos y hará adquirir nuevos clientes, mientras que al mismo tiempo reducirá los costos excesivos.

Con los servicios proporcionados por la conectividad inalámbrica se permite a los servicios aprovechar las capacidades del dispositivo móviles.

Desarrolladores de la aplicación.- Con este tipo de aplicaciones podemos acceder a Internet al mundo en línea, mediante la comunidad inalámbrica. Las aplicaciones pueden desarrollarse beneficiándose totalmente de la interfaz del usuario final, debido a que el navegador WAP en cada dispositivo inalámbrico será capaz de controlar cómo el contenido se mostrará y visualizará. Además, los desarrolladores no tienen por qué preocuparse ya que WAP es un estándar abierto con una ruta de migración hacia el futuro.

4.2. DISEÑO CENTRADO EN EL USUARIO

Al momento de desarrollar aplicaciones móviles un punto importante es tener la idea de mantener un diseño centrado en el usuario esta es una manera inteligente de acercarse al usuario y poder interactuar con él.

El desarrollador de las aplicaciones inalámbricas debe entender las necesidades del usuario y crear procesos del proyecto que cumpla con sus metas y expectativas.

Empezar con el contexto. Piense sobre cómo y donde las personas actuarán con la aplicación.

Pensar como los clientes actúan con un dispositivo móvil, proporciona el primer paso para la creación de una aplicación móvil eficaz.

Este es uno de los primeros pasos más importantes al momento de desarrollar una aplicación móvil, ya que la implementación de la aplicación deberá soportar dispositivos celulares de diferentes características y diferentes recursos y de acuerdo a este se procesa y diseña la interfaz, para brindar al usuario un incremento en la rapidez y la amigabilidad en su experiencia con la aplicación, con esto se pretende atacar los que han sido principales obstáculos para el uso de este tipo de sistemas la pobre presentación de contenidos dinámicos por los limitados espacios y recursos, dificultando el aprovechar las excelentes prestaciones de los celulares como son: La movilidad, largo alcance y uso masivo de los celulares para comercio electrónico.

Otro factor importante al momento de desarrollar soluciones móviles esta en conocer a las personas a quien va dirigida la aplicación.

Es importante que el desarrollador de la aplicación móvil entienda como los usuarios finales trabajan con los dispositivos para que este pueda anticiparse cómo ellos querrán navegar su sitio, y así poder proporcionarles una navegación rápida y eficaz.

4.3. ARQUITECTURA DE LA INFORMACIÓN MÓVIL

Uno de los principales consejos a tomarse en cuenta en las aplicaciones móviles es mantener la Política “Mientras mas simple mejor”.

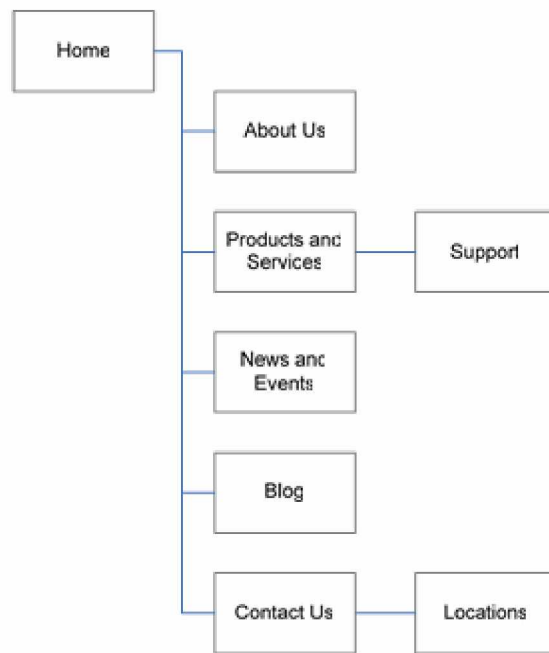


Figura 9. Política de “Mientras mas simple mejor”³⁰

La mejor arquitectura móvil es guardar la información lo más simple que sea posible, debido a la presentación de los dispositivos móviles.

Los factores a tomar en cuenta en esta técnica son los siguientes:

§ **Limitar las opciones:** Tomar el contenido relevante para un usuario móvil y el resto desecharlo. Esto facilita la navegación al usuario y evita el riesgo que este se pierda en el sitio.

³⁰ Mobile Web Developer’s Guide Part I: Creating Simple Mobile Sites for Common Handsets

§ **Crear la arquitectura del sitio simple, organizado y dividido por categorías.**

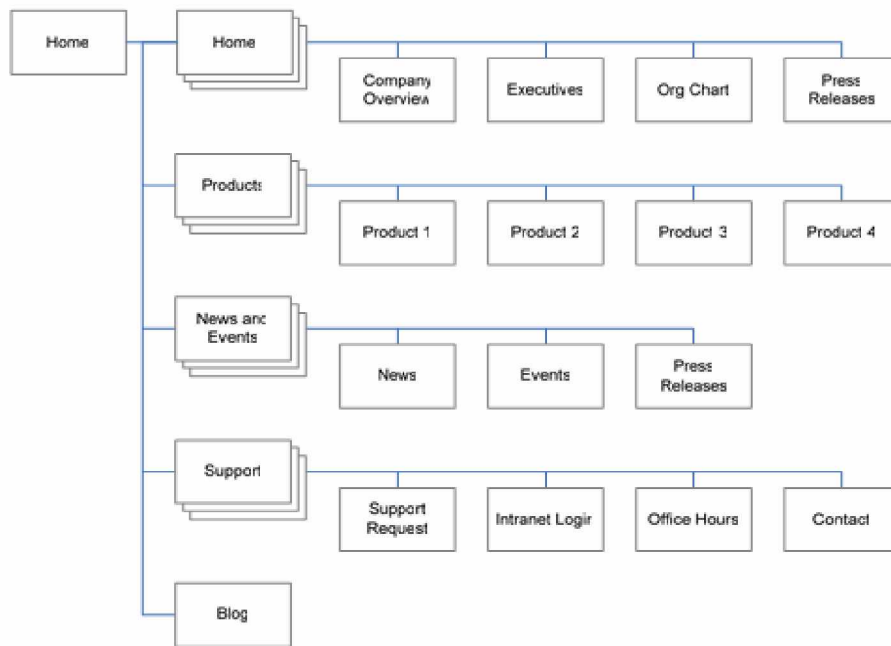
En lo que sea posible presentarle una interfaz amigable solo con lo necesario para realizar la aplicación para lo que fue creada.

§ **Limite las categorías tanto como sea posible:** Se desorientan los usuarios cada vez más cuando ellos entran con mayor profundidad al sitio. Con un sitio de Web Móvil, de acuerdo a estudios se dice que la tolerancia de un usuario a menudo es aproximada de cinco niveles, es por eso que mientras menos niveles es mejor.

§ **Intentar limitar los link** de las páginas para presentar más información debido a que el teclado de los dispositivos móviles es muy restringido y limitado.

§ **Priorizar los links por actividades o popularidad:** Ordenar los link de acuerdo a la frecuencia de acceso permite que la información que se desee será vista con mayor frecuencia por el usuario.

Aquí podemos ver una arquitectura simple de un sitio para móvil, dividido en categorías y tomando en cuenta sus limitantes.

Figura 10. Arquitectura de la información móvil³¹

El desarrollador una vez que determina sus requisitos y define su estructura de la categoría y etiquetas, podrá obtener un Sitio Web simple. Esto lo da un alto nivel la apreciación global de su espacio de información.

Otro punto importante en el desarrollo de las aplicaciones móviles esta el **Diseño de las aplicaciones móviles** ya que cuando se diseña para móviles es muy limitado comparar este tipo de aplicaciones con las diseñadas para Sitios web orientadas a escritorio.

El desarrollador puede crear experiencias móviles ricas pero siempre se debe tomar en cuenta las capacidades y limitantes del dispositivo.

³¹ Mobile Web Developer's Guide Part I: Creating Simple Mobile Sites for Common Handsets

En este tipo de aplicaciones se puede encontrar con ciertos delimitantes debido a los excesos de tiempos, compatibilidad del dispositivo reducida y soporte de estilos incoherente.

CAPÍTULO V

APLICACIONES MÓVILES CON TEGNOLOGÍA CELULAR

5. INTRODUCCIÓN A LA TELEFONÍA MÓVIL.

5.1. HISTORIA DEL CELULAR³²



La **telefonía móvil**, también llamada **telefonía celular**, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red.

El teléfono móvil o celular (designación usada en Latinoamérica) fue inventado en 1947 por la empresa norteamericana AT&T, pero no se hizo portátil de manera práctica hasta 1983 cuando Motorola culmina el proyecto DynaTAC 8000X, que es presentado oficialmente en 1984. El DynaTAC 8000X pesaba cerca de 1kg, tenía un tamaño de 33.02x4,445x8,89 centímetros (13 x 1,75 x 3,5pulgadas), y rendía una hora de comunicación y ocho horas en *stand-by*, con pantalla de LED.

El teléfono móvil consiste en un dispositivo de comunicación electrónico con las mismas capacidades básicas de un teléfono de línea telefónica convencional. Además de

³² <http://es.wikipedia.org>

ser portátil es inalámbrico al no requerir cables conductores para su conexión a la red telefónica.

La red de telefonía móvil consiste en un sistema telefónico que mediante la combinación de una red de estaciones transmisoras-receptoras de radio (repetidores ó también llamados estaciones base) y una serie de centrales telefónicas de conmutación, se posibilita la comunicación entre terminales telefónicos portátiles (teléfonos móviles) o entre terminales portátiles y teléfonos de la red fija tradicional.

5.1.1. Funcionamiento.

El empleo de la palabra *celular* referido a la telefonía móvil, deriva del hecho que las estaciones base enlazan vía radio los teléfonos móviles con los controladores de estaciones base, están dispuestas en forma de una malla, formando lo que en inglés se ha denominado como *cell* y en Latinoamérica se ha traducido erróneamente (aunque ya aceptado socialmente) como *células*, debido a la doble traducción de la palabra inglesa original (siendo *celdas* una traducción más correcta, haciendo alusión a la forma geométrica que forma la malla de telecomunicaciones). Así, cada repetidor está situado en los vértices de estas celdas y tiene asignado un grupo de frecuencias de transmisión y recepción propio. Como el número de frecuencias es limitado, con esta disposición es posible reutilizar las mismas frecuencias en otras celdas, siempre que no sean adyacentes, para evitar interferencia entre ellas.

No obstante, el término *móvil* es de uso generalizado en España, mientras que *celular* se usa ampliamente en Iberoamérica. De la misma forma que en Reino Unido se dice *mobile phone*, y en EEUU se conoce como *cell phone*.

Hay que dejar constancia que en castellano el término "móvil", aplicado a este tipo de teléfonos, es incorrecto. Según el DRAE, "móvil" es aquello "que puede moverse o se mueve por sí mismo", condición que no se da por ahora en ningún teléfono. "Móvil" no es sinónimo de "portátil", que es el término que corresponde.

5.1.2. Tipos de redes de telefonía móvil

- ω **Red de telefonía móvil analógica (FDMA/FDD)**. Como su propio nombre indica, en esta red la comunicación se realiza mediante señales analógicas tanto en el tramo radioeléctrico como en el terrestre. En su primera versión funcionó en la banda radioeléctrica de los 450 MHz, trabajando posteriormente en la banda de los 900 MHz.

- ω **Red de telefonía móvil digital (TDMA/FDD)**. En esta red la comunicación se realiza mediante señales digitales, lo que permite optimizar tanto el aprovechamiento de las bandas de radiofrecuencia como la calidad de transmisión. Su exponente más significativo en el ámbito público es el estándar

GSM, así como la tercera generación de telefonía móvil o 3G, UMTS. Funciona en las bandas de 850/900 y 1800/1900 MHz en el estándar GSM y en la banda de 2000 MHz en UMTS. En el 2004 llegó a los 1000 millones de usuarios. Hay otros varios estándares digitales, presentes en América y Asia, con distintas denominaciones, como por ejemplo el sistema PCS en Japón, el IS-95 en EEUU y algunos países de América Latina (el cual utiliza la primera versión comercial de la tecnología CDMA, utilizada también por las posteriores tecnologías 3G), o el D-AMPS en EEUU. Esta no es una lista extensiva. En el ámbito privado y de servicios de emergencias como policía, bomberos y servicios de ambulancias se utilizan los estándares Tetrapol y Terrestrial Trunked Radio (TETRA) en diferentes bandas de frecuencia.

La evolución del teléfono móvil ha permitido disminuir su tamaño y peso, desde ese primer teléfono móvil en 1983 que pesaba cerca de 1kg, a los actuales más compactos y con mayores prestaciones de servicio. El desarrollo de baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores, la incorporación de software más amigable, hacen del teléfono móvil un elemento muy apreciado en la vida moderna.

El avance de la tecnología ha hecho que estos aparatos incorporen funciones que no hace mucho parecían futuristas, como juegos, reproducción de música MP3 y otros formatos, correo electrónico, SMS, agenda electrónica PDA, fotografía digital y video digital, video llamada, navegación por Internet y hasta Televisión digital. Las compañías de telefonía móvil ya están pensando nuevas aplicaciones para este pequeño

aparato que nos acompaña a todas partes. Algunas de esas ideas son: medio de pago, localizador e identificador de personas. Siempre hay que tener en cuenta los grandes avances sufridos desde el primer teléfono móvil hasta el actual.

5.2. SOLUCIONES ON-LINE.

Las aplicaciones On-Line conocidas también como aplicaciones Web Móviles o también WAP pues usan este protocolo para comunicarse, son aquellas que se ejecutan en el dispositivo celular utilizando el mini-browser que disponen estos dispositivos.

Actualmente, existe una demanda cada vez mayor por parte de los usuarios en lo referente a una disponibilidad incondicional de la Web, pero la realidad en el mercado es otra ya que aunque la oferta de dispositivos móviles está creciendo de forma asombrosa en los últimos años, ofreciéndonos infinidad de dispositivos desde los que llevar a cabo operaciones que normalmente realizábamos desde el equipo de sobremesa, existen limitaciones a la hora de acceder a los servicios desde esos dispositivos móviles.

En la mayoría de la ocasiones, el resultado es una experiencia de usuario poco satisfactoria al encontrarnos con numerosos problemas para acceder a la Web desde los dispositivos móviles.

La Web móvil llega a lugares donde el cable no puede llegar, lugares que anteriormente eran impensables. Esto trae consigo nuevas oportunidades para realizar negocios, para trabajar, para gestionar nuestro tiempo de ocio, y para muchas otras cosas, lo que nos permite incrementar nuestra capacidad de movimiento.

La Web móvil se presenta como un auténtico reto tanto para usuarios como para desarrolladores ya que, por un lado, el usuario encuentra problemas al intentar acceder a los sitios Web desde los dispositivos móviles, y por otro, los proveedores de contenido encuentran dificultades para crear sitios Web que funcionen adecuadamente en todos los tipos de dispositivos y configuraciones.

Es importante tener en cuenta que existen grandes diferencias entre usuarios móviles y usuarios fijos, como son los diferentes tipos de contenido que manejan, las capacidades de los dispositivos que utilizan (pantallas pequeñas) y el contexto en el cual el usuario recibe el contenido.

El acceso a la información desde cualquier lugar, en cualquier momento e independientemente del dispositivo utilizado puede alcanzarse a través de aplicaciones que se adapten dinámicamente a las necesidades del usuario, a las capacidades del celular y a las condiciones del entorno.

Dotar a las aplicaciones de **movilidad** permitirá a los usuarios utilizar diferentes dispositivos para acceder a la misma información. Como usuarios, podremos elegir la

forma de interactuar con estas aplicaciones en función de nuestras necesidades y de las características del dispositivo utilizado. Para ello es necesaria una infraestructura global basada en estándares que permita la interoperabilidad³³.

Ahora nos centremos en una visión general de cómo es el desarrollo y funcionamiento de las aplicaciones On-Line.

Estas aplicaciones no consumen recursos del dispositivo para el procesamiento pues todo el proceso se realiza de lado del servidor y el browser se encarga de presentar los resultados que pueden estar en WML o en XHTML-MP desde la nueva versión de WAP 2.0.

Este tipo de aplicaciones tiene su gran ventaja en que nos permiten acceder a gran parte de la potencialidad de la web como es:

- ω Correo electrónico.
- ω Blogs Personales.
- ω Canales de noticieros.
- ω Acceso a servicios de comercio electrónico.
- ω Servicios de IRC(Internet Relay Chat) esto si el dispositivo tiene las características necesarias.

- ω Acceso a información pública o privada.

³³ [http:// www.w3c.es](http://www.w3c.es)

Y un gran sin número de aplicaciones que esta limitado solo por la imaginación y creatividad de los desarrolladores y empresas de negocios.

Listamos a continuación algunas de las ventajas de las aplicaciones On-Line.

La gran ventaja que disponen estas aplicaciones contra las web es la facilidad de acceso desde cualquier lugar y en cualquier momento en el que haya cobertura de la red del operador celular en el que se tenga el servicio.

Un clásico ejemplo es el de un ejecutivo que termina una jornada intensa de trabajo y debe embarcarse en un viaje en avión cuando esta a punto de llegar a su destino recuerda que debió haber hecho las reservaciones respectivas del hotel , rentado un auto, pasajes de regreso entonces intenta llamar por teléfono y las líneas se encuentran congestionadas!!!.

Ahora el mismo ejemplo pero con la variante que el ejecutivo hace uso de sistemas de los sistemas WAP mientras se encuentra en vuelo y navega hacia un sitio de reservaciones de hoteles se da tiempo escoger el hotel, la habitación hasta puede mirar fotos de ellas, de igual manera para rentar el auto y hasta tiene tiempo de elegir el horario de vuelo y comprar el pasaje de regreso.

Este ejemplo nos da una idea de lo útiles que pueden ser las aplicaciones On-Line o WAP.

Ahora veremos un poco las desventajas sobre todo en cuando a las dificultades que enfrentan los desarrolladores como son el muy reducido espacio en el cual deben diseñar las aplicaciones e intentar proveer interfaces amigables sin desperdiciar un centímetro.

Otra desventaja es que las redes inalámbricas no llegan a tener velocidades de banda ancha como las redes físicas o redes inalámbricas para PC. Aunque en la actualidad agosto del 2007 una tecnología esta entrando con fuerza y lo a hecho ya en países como México y Uruguay, estamos hablando de 3G **Tercera generación** esta tecnología permite navegar a más de 3 megas de velocidad. Este impresionante avance es de una utilidad magnífica para nuestras aplicaciones On-Line pero volviendo a la realidad la gran mayoría se podría asegurar que un 95% de usuarios no dispone aun conexión 3G, por lo que el desarrollo debe estar orientado a esa gran mayoría.

Pero para poder desarrollar aplicaciones On-Line para móviles sin encontrarnos con problemas es necesario una total adecuación de los contenidos a los dispositivos apoyándonos en los estándares de desarrollo como se verá en capítulos posteriores.

5.3. SOLUCIONES STAND-ALONE³⁴.

Las aplicaciones Stand-alone son aquéllas que se desarrollan para ser instaladas y ejecutadas sobre el equipo móvil en cuestión y que funcionan en forma desconectada de Internet o de un servidor central.

Para desarrollar una solución de este tipo, debemos generar un paquete ejecutable en el formato correcto para el tipo de sistema operativo sobre el cual será instalado, así como también la versión del mismo.

Cada sistema operativo es diferente y hasta el hardware sobre el que se ejecuta es distinto; es por ello que una aplicación desarrollada para Palm OS es completamente distinta de una para Pocket PC o para Symbian OS.

Las ventajas de este tipo de soluciones son:

- ω Ejecución veloz.

- ω Aprovechamiento de características de bajo nivel de cada equipo.

- ω Uso de todas las herramientas, controles y accesos que ofrece el dispositivo.

³⁴ Libro PDF Desarrollos Moviles con .NET

- ω Manejo de memoria.

- ω Soporte de sincronización con un equipo de escritorio.

- ω Se puede trabajar sin necesidad de estar conectado.

Las desventajas son:

- ω Se deben desarrollar diferentes versiones para cada sistema operativo.

- ω Se deben instalar manualmente en cada equipo

- ω No pueden soportar grandes cantidades de información para búsquedas o almacenamiento.

5.3.1. TIPO DE CÓDIGO EJECUTABLE

Cuando trabajamos con aplicaciones que se ejecutan directamente sobre el equipo, distinguimos dos tipos de código para generar: código nativo y código manejado.

5.3.1.1. Código nativo

Desarrollar una aplicación en código nativo implica que el archivo ejecutable que instalaremos en el equipo está expresado en código ensamblador entendible por el sistema operativo y por el procesador del equipo.

Esto no quiere decir que nosotros desarrollemos bajo ensamblador (assembler), el compilador que utilizemos para desarrollar realizará la traducción a dicho código.

Las ventajas de desarrollar bajo código nativo son:

- ω Mayor rapidez de ejecución.
- ω Código más compacto.
- ω Acceso al 100% de las capacidades del equipo.
- ω No se requiere la instalación de ningún agregado para la ejecución.
- ω Se posee acceso directo a memoria y de bajo nivel.

Las desventajas son:

- ω En cada sistema operativo o hardware, se necesita recompilar el proyecto y generar ejecutables distintos.

- ω Si trabajamos con dos equipos distintos, debemos tener en cuenta diferencias de hardware en algunas funciones.

- ω Se posee acceso directo a memoria (esto puede traer problemas).

5.3.1.2. Código manejado

El código manejado surgió como solución a los problemas que traía el código nativo.

Cuando trabajamos con este tipo de código, lo que se genera al compilar el proyecto no es código nativo entendible por el hardware y el sistema operativo del equipo, sino que es un código que es entendible por un aplicativo intermedio entre nuestro programa y el hardware, llamado Máquina Virtual.

Esta máquina virtual interpreta el código manejado y lo convierte en tiempo real (Just in Time) a código nativo subordinado al sistema y hardware en que se encuentra.

Este tipo de desarrollo es el más utilizado. Y la maquina virtual más usada por los distribuidores de dispositivos celulares es la KVM (Kilo Virtual Machine) por lo que

esta es una de las razones por lo que hemos escogido Java como lenguaje de desarrollo para nuestras aplicaciones.

5.4. SOLUCIONES SMART-CLIENT³⁵

Una aplicación Smart Client(cliente inteligente) junta lo mejor de dos mundos, Stand-alone y Online.

Este tipo consta de aplicaciones ejecutables que se distribuyen e instalan en los equipos, pero que también utilizan la conexión para comunicarse e interactuar con un servidor.

La “inteligencia” radica en que la aplicación debe ser capaz de seguir ejecutándose aun cuando el equipo pierda conexión con el servidor (generando buffer de Información, por ejemplo).

Algunas ventajas de utilizar Smart Clients:

- ω Junta lo mejor del mundo conectado y del desconectado.

- ω Permite consultar grandes capacidades de información y hacer uso de funciones de bajo nivel de los equipos.

- ω Permite seguir trabajando cuando el equipo se desconecta.

³⁵ Libro PDF Desarrollos Moviles con .NET

Algunas desventajas:

- ω Es más difícil a la hora de desarrollar las aplicaciones, al pensar de qué manera baja la aplicación online u offline sin que el usuario pueda percibir la diferencia.

- ω Se debe crear el cliente basándose en cada tipo y versión de sistema operativo.

- ω Se debe distribuir e instalar el cliente en todos los equipos.

- ω Necesita dispositivos de altas características en recursos.

CAPÍTULO VI

TECNOLOGÍAS ACTUALES DE APLICACIONES MÓVILES

NATIVAS

6. APLICACIONES MÓVILES NATIVAS

6.1. TECNOLOGÍAS MÓVILES



Uno de los primeros avances de la telefonía móvil en el sector de las comunicaciones se dio con la aparición de la tecnología WAP y por ende WAP exige la presencia de una puerta de enlace encargado de actuar como intermediario entre Internet y el terminal. Esta puerta de enlace o gateway es la que se encarga de convertir las peticiones WAP a peticiones web habituales y viceversa. Las páginas que se transfieren en una petición usando WAP no están escritas en HTML, si no que están escritas en WML, o XHTML.




	GSM (Global System for Mobile communications) <i>(Sistema Global para las comunicaciones Móviles)</i>
	GPRS (Global Packet Radio System) <i>(Servicio General de Paquetes de Radio.)</i>
	UMTS o 3GSM (Universal Mobile Telephone System) <i>(Sistema Universal para la Telefonía Móvil)</i>

Figura 11. Tecnologías móviles usadas en la actualidad

Los avances de la telefonía móvil nos llevaron a las tecnologías conocidas como generaciones 2 , 2.5 y en la actualidad 3, que hacen uso de las tecnologías GSM , GPRS y UMTS respectivamente. GSM es una conexión telefónica que soporta una circulación de datos, mientras que GPRS es estrictamente una red de datos que mantiene una conexión abierta en la que el usuario paga por la cantidad de información intercambiada y no por el tiempo que permanezca conectado.

En la actualidad, y tras varios años de incertidumbre que produjeron la quiebra de numerosas compañías de telecomunicaciones, se está implantando progresivamente móviles de 3ra generación. Esta nueva tecnología ha supuesto un salto cualitativo en la calidad de las comunicaciones.

UMTS representa la tercera generación (3G) de telecomunicaciones. Esta nueva tecnología incorporada a la telefonía móvil supone un importante avance para el mundo de las telecomunicaciones, ya que permite disponer de nuevos servicios multimedia caracterizados por una alta calidad de imagen y una elevada velocidad en la transmisión de datos. Los servicios asociados con la tercera generación proporcionan la posibilidad de transferir tanto voz como datos (una llamada telefónica) y solo datos (como la descarga de programas, intercambio de correo-e y mensajería instantánea).

Esta tecnología es un servicio de comunicaciones inalámbricas que permite estar conectado de forma permanente a Internet a través del teléfono móvil, el ordenador de bolsillo y el ordenador portátil. La tecnología 3G propone una mejor calidad y fiabilidad, una mayor velocidad de transmisión de datos y un ancho de banda superior (que incluye la posibilidad de ejecutar aplicaciones multimedia). Con velocidades de datos de hasta siete veces más rápida que una conexión telefónica estándar.

Actualmente ya existe una amplia oferta de dispositivos móviles en el mercado y los fabricantes distribuyen nuevos modelos constantemente. Como resultado de esta gran variedad de dispositivos, se debe afrontar el problema cada vez que se desarrolla una aplicación móvil haciendo uso de las nuevas herramientas de desarrollo existentes entre las más destacadas en la actualidad

ω Plataforma Java

ω Plataforma .Net

Las soluciones móviles están mostrando sus beneficios para la gestión de las empresas, comercio, en la mejora de la productividad, en la creación de nuevos servicios.

El desarrollo de Aplicaciones móviles permite estar a la vanguardia en comunicaciones y sistemas de información

6.1.1. Plataforma Java

Juntos, el lenguaje de programación Java, la máquina virtual y las APIs, forman la plataforma Java, la cual, en su versión 2, se puede encontrar en tres ediciones:



Figura 12. Arquitectura plataforma Java³⁶

³⁶ JAVA 2 MICRO EDITION. MANUAL DE USUARIO Y TUTORIAL. Agustín Froufe Quintas; Patricia Jorge Cárdenas (Editorial Ra-ma)

- ω Java 2 Standard Edition (J2SE). Diseñada para computadoras de escritorio, puede trabajar en sistemas operativos como: Windows, Linux, MacOS, Solaris, OS x.

- ω Java 2 Enterprise Edition (J2EE). Plataforma para aplicaciones multiusuario o empresariales. Se basa en J2SE y agrega apis para trabajo en el servidor.

- ω Java 2 Micro Edition (J2ME). Conjunto de tecnologías y especificaciones desarrolladas para dispositivos pequeños como los teléfonos celulares y PDA (palm, agendas electrónicas). Utiliza derivados de componentes J2SE, como son una máquina virtual más pequeña y un conjunto de apis menos potentes.

6.1.1.1. Introducción a J2ME

A diferencia de J2SE, la micro edición de Java no es sólo una pieza de software ni una simple especificación, J2ME es una plataforma, una colección de tecnologías y especificaciones diseñadas para diferentes partes del mercado de los dispositivos pequeños.

J2ME se divide en configuraciones, perfiles y paquetes opcionales. Las configuraciones son especificaciones que detallan una máquina virtual y un conjunto base de apis que pueden ser usadas en cierta clase de dispositivos. La maquina virtual puede ser completa, como la describe la especificación o algún derivado de ella.

Un perfil trabaja sobre una configuración pero agrega APIs específicas para hacer un entorno completo de construcción de aplicaciones. Usualmente incluyen apis para el ciclo de vida de las aplicaciones, interfaz de usuario y almacenamiento persistente.

Actualmente J2ME tiene disponibles dos configuraciones:

- ω **Connected Limited Device Configuration (CLDC)**. Para dispositivos pequeños con conexiones de red intermitentes, como teléfonos celulares y PDA.

- ω **Connected Device Configuration (CDC)**. Diseñada para aparatos más grandes (en términos de memoria y poder de procesamiento) con conexiones de red robustas.

- ω **MIDP** (Mobile Information Device Profile) es el perfil con mayor desarrollo en la plataforma Java, aunque se está investigando el PDA Profile. MIDP incluye apis de interfaz de usuario, de ciclo de vida de aplicaciones y algunas de almacenamiento persistente.

Los dispositivos implementan una "pila" de software completa, la cual consiste usualmente en una configuración, un perfil y APIs opcionales.

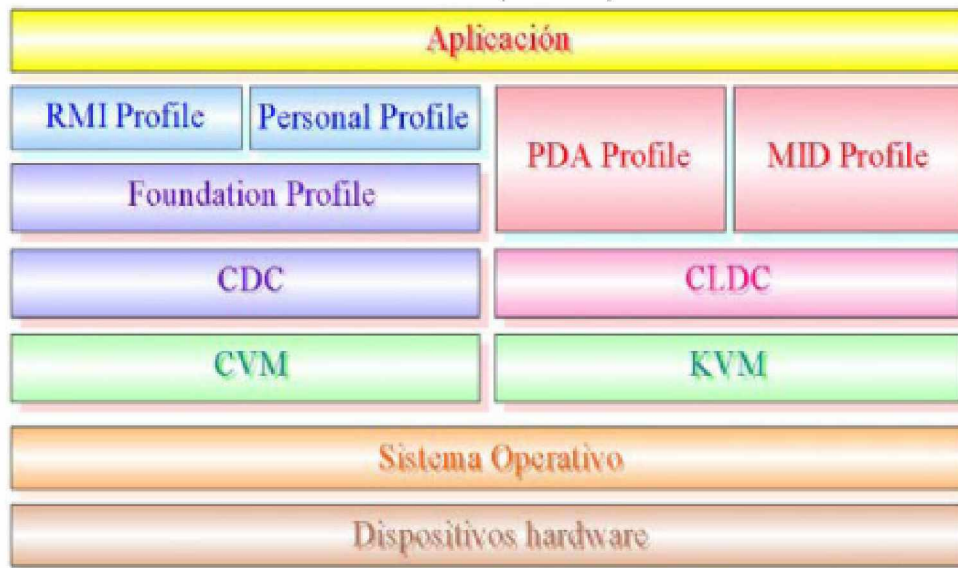


Figura 13. Modelo de perfiles de J2ME³⁷

6.1.2. MIDlets

Las aplicaciones J2ME desarrolladas bajo la especificación MIDP, se denominan MIDlets. Las clases de un MIDlet, son almacenadas en bytecodes java, dentro de un fichero .class. Estas clases, deben ser verificadas antes de su puesta en marcha para garantizar que no realizan ninguna operación no permitida. Este preverificación, se debe hacer debido a las limitaciones de la máquina virtual usada en estos dispositivos (KVM). Para mantener esta máquina virtual lo más sencilla y pequeña

³⁷ Arquitectura dirigida por modelos para J2ME de Luis Enrique Corredera de Colosa

posible, se elimina esta verificación, y se realiza antes de la entrada en producción. La preverificación se realiza después de la compilación, y el resultado es una nueva clase, lista para ser puesta en producción.

Las aplicaciones JAVA que se ejecutan en dispositivos que implementan MIDP se denominan MIDlets. Un MIDlet consta de al menos una clase JAVA, que debe derivar de la clase base abstracta **javax.microedition.midlet.MIDlet**. El tiempo de ejecución de un MIDlet viene controlado por una serie de métodos definidos en dicha clase y que, forzosamente, todas las clases derivadas deben implementar.

Un grupo de MIDlets relacionados pueden agruparse en un **MIDlet suite**. Todos los MIDlets de un **suite** se agrupan e instalan en un dispositivo como si fuesen un único elemento, de forma que sólo pueden desinstalarse y eliminarse en conjunto. Los MIDlets agrupados en un **suite** comparten tanto recursos estáticos como dinámicos:

- ω En tiempo de ejecución, si los dispositivos soportan ejecución concurrente, todos los MIDlets se ejecutarán en la misma máquina virtual. Por tanto, todos los MIDlets comparten las mismas instancias de todas las clases, así como los recursos cargados en la máquina virtual. Entre otras cosas, esto supone que se pueden compartir datos entre MIDlets y que las primitivas de sincronización no sólo deben usarse para proteger frente al acceso

concurrente dentro de cada MIDlet, sino también el producido por otros MIDlets asociados al mismo **suite**.

- ω El almacenamiento persistente es gestionado de forma global para todos los MIDlets del **suite**. Por tanto, los MIDlets pueden acceder tanto a sus propios datos persistentes como a los de otras clases del MIDlet.

Los MIDlets, son empaquetados en ficheros .jar. Se requiere alguna información extra, para la puesta en marcha de las aplicaciones. Esta información se almacena en el fichero de manifiesto, que va incluido en el fichero .jar y en un fichero descriptor, con extensión .jad. Un fichero .jar típico, por tanto, se compondrá de:

- ω Clases del MIDlet

- ω Clases de soporte

- ω Recursos (imágenes, sonidos...)

- ω Manifiesto (fichero “.mf”)

- ω Descriptor (fichero “.jad”)

Un fichero .jar puede contener varios MIDlets. Esta colección de MIDlets, se suele

llamar MIDLet Suite. Esta unión de varios MIDLets en una distribución, permite compartir recursos tales como imágenes o sonidos y, por tanto, optimizar los recursos del dispositivo.

Un MIDlet durante su ejecución pasa por 3 estados diferentes:

- ω **Activo:** el MIDlet está actualmente en ejecución.

- ω **Pausa:** El MIDlet no está actualmente en ejecución. En este estado el MIDlet no debe usar ningún recurso compartido. Para volver a pasar a ejecución tiene que cambiar su estado a Activo.

- ω **Destruído:** El MIDlet no está en ejecución ni puede transitar a otro estado. Además se liberan todos los recursos ocupados por el MIDlet.

En la Figura 14 se muestra los diferentes estados del midlets.

Figura 14. Estado de un Midlet³⁸

Como vemos en el diagrama, un MIDlet puede cambiar de estado mediante una llamada a los métodos `MIDlet.startApp()`, `MIDlet.pauseApp()` o `MIDlet.destroyApp()`.

El gestor de aplicaciones (AMS) cambia el estado de los MIDlets haciendo una llamada a cualquiera de los métodos anteriores. Un MIDlet también puede cambiar de estado por sí mismo.

³⁸ <http://leo.ugr.es/J2ME/MIDP/aplicaciones.htm>

Ahora vamos a ver por los estados que pasa un MIDlet durante una ejecución típica y cuáles son las acciones que realiza tanto el AMS como el MIDlet. En primer lugar, se realiza la llamada al constructor del MIDlet pasando éste al estado de Pausa durante un corto período de tiempo. El AMS por su parte crea una nueva instancia del MIDlet.

Cuándo el dispositivo está preparado para ejecutar el MIDlet, el AMS invoca al método MIDlet.startApp() para entrar en el estado de Activo. El MIDlet entonces, ocupa todos los recursos que necesita para su ejecución. Durante este estado, el MIDlet puede pasar al estado de Pausa por una acción del usuario, o bien, por el AMS que reduciría en todo lo posible el uso de los recursos del dispositivo por parte del MIDlet. Tanto en el estado Activo como en el de Pausa, el MIDlet puede pasar al estado Destruído realizando una llamada al método MIDlet.destroyApp(). Esto puede ocurrir porque el MIDlet haya finalizado su ejecución o porque una aplicación prioritaria necesite ser ejecutada en memoria en lugar del MIDlet. Una vez destruido el MIDlet, éste libera todos los recursos ocupados.

Los archivos de manifiesto y el .JAD son archivos de texto con la siguiente estructura:

```
nombre_atributo: valor_atributo
```

El nombre del atributo y su valor están separados por dos puntos, siendo el espacio adicional opcional. Todos los atributos que son relevantes para la instalación de MIDlets comienzan por el prefijo "MIDlet-". Una lista completa de atributos, junto con una breve descripción de sus valores asociados, aparece en la Tabla 12. Los valores en

las columnas JAD y JAR indican que el atributo es obligatorio (M), opcional (O) o ignorado (I).

Nombre de atributo	JAR	JAD	Valor y significado
MIDlet-Name	M	M	El nombre del suite integrado en el archivo JAR. El usuario podrá ver este nombre
MIDlet-Version	M	M	El número de versión del suite empaquetado en el archivo JAR. Los número de versión tienen la forma a.b.c, donde valores mayores indican versiones más recientes, teniendo los números de la izquierda mayor precedencia
MIDlet-Vendor	M	M	El nombre del fabricante del suite. Se trata de texto libre.
MIDlet-n	M	I	Atributo que describe cada uno de los MIDlets que integran el suite. El valor numérico n comienza en 1 y sirve para identificar cada uno de los MIDlets
MIDlet-Description	O	O	Una descripción del suite, para servir de información a los usuarios
MIDlet-Icon	O	O	Un icono que representa al suite durante la instalación o configuración. Se debe tratar de una imagen en formato png (Portable Network Graphics)
MIDlet-Info-URL	O	O	Dirección URL de un archivo que contiene información adicional sobre el suite. El contenido del archivo se

			mostrará al usuario, de forma que éste pueda decidir si desea o no instalar dicho suite.
MIDlet-Data-Size	O	O	Mínima cantidad de espacio de almacenamiento persistente para que el suite funcione de forma correcta. Se trata de espacio usado por el suite para almacenar datos de larga duración. Se especifica en bytes. Si no se proporciona este atributo, se asume que el suite no precisa almacenamiento persistente
MIDlet-Jar-URL	I	M	La dirección URL del archivo JAR que contiene el suite
MIDlet-Jar-Size	I	M	El tamaño del JAR que contiene el suite, en bytes
MIDlet-Install-Notify	I	O	Una dirección URL usada para indicar condiciones de error o de éxito en la instalación del suite. Se trata de un atributo no incluido en la especificación MIDP, pero soportado por J2ME.
MIDlet-Delete-Confirm	I	O	Mensaje a mostrar en caso que el suite vaya a ser borrado del dispositivo. Con este atributo ocurre lo mismo que se indicó respecto al anterior: se trata de un atributo no incluido en la especificación de MIDP
MIDlet-specific attributes	O	O	Los desarrollados de MIDlets pueden proporcionar atributos específicos

MicroEdition-Profile	M	I	La versión de especificación MIDP con la que el suite puede trabajar. Cuando aparecen varias versiones deben separarse mediante espacios. Las versiones especificadas en este atributo se comparan con el valor de la propiedad microedition.profiles para determinar la compatibilidad del suite con la versión de MIDP disponible
MicroEdition-Configuration	M	I	La configuración J2ME precisada por el suite. El valor de este atributo se compara con la propiedad microedition.configuration para determinar la compatibilidad

Tabla 12 Atributos y descripción en FAR y JAD.³⁹

6.1.2.1. Herramientas de Desarrollo

En el mercado existen varias herramientas válidas para construir programas bajo el standard J2ME que nos pueden ayudar a la hora de crear nuestros MIDlets. En el desarrollo de este proyecto estudiaremos un par de ellas que explicaremos a continuación:

1. La primera de ellas es un entorno de desarrollo de Java Netbeans, además de incluir las APIs propias de la configuración CLDC y el perfil MIDP (Mobile Edition).

³⁹ <http://leo.ugr.es/J2ME/TOOLS/index2.htm#herramientas>

En esta herramienta es posible realizar todas las fases del desarrollo de aplicaciones MIDP:

- ω Disponemos de un editor de texto totalmente integrado donde crear el código fuente.
- ω Una vez creado el código del MIDlet es posible su compilación ya que el entorno dispone de todas las librerías necesarias para ello.
- ω El proceso de preverificación se realiza automáticamente después de la compilación.
- ω El entorno también nos da la posibilidad de empaquetar el MIDlet por separado o dentro de una MIDlet suite.
- ω Por último, las fases de ejecución y depuración también la podemos realizar con esta herramienta ya que nos permitirá ejecutar el MIDlet sobre un emulador, ya que trae integrada la herramienta J2ME Wireless Toolkit . Como vemos, esta herramienta engloba todas las fases de desarrollo en un mismo entorno.

2. La segunda herramienta es el J2ME Wireless Toolkit 2.0 que es simplemente un emulador al que le proporcionamos las clases java ya creadas y podemos ver el MIDlet en ejecución.

Se trata de una plataforma ideada por Sun. Es un conjunto de herramientas que hace posible crear aplicaciones para teléfonos móviles y otros dispositivos inalámbricos. Aunque está basado en MIDP 2.0 (Mobile Information Device Profile), Wireless Toolkit también trae una amplia colección de paquetes opcionales que lo convierten en una herramienta de desarrollo bastante potente.

6.1.3. Plataforma .Net

En este capítulo haremos un repaso general de todas las tecnologías disponibles, y qué tipo de soluciones podemos desarrollar y sobre cuáles podremos trabajar con las herramientas que nos provee .NET.

En la Figura 15 podemos ver qué tipo de soluciones nos permiten desarrollar las herramientas de desarrollo de Microsoft y en qué tipo de equipos podemos ejecutarlo.

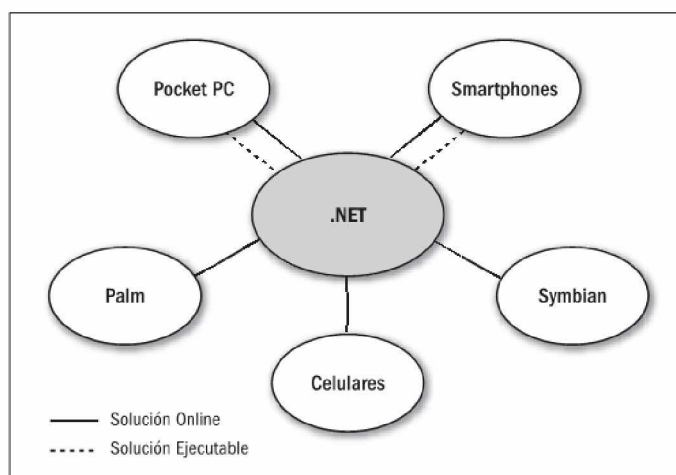


Figura 15. Con .NET podemos acceder a equipos móviles en el mercado, ya sea con aplicaciones online o aplicaciones ejecutables.⁴⁰

⁴⁰ DESARROLLOS MÓVILES CON .NET editorial mpediciones CAP 1

Con la salida de .NET y el principio de la programación en código manejado en cuanto a herramientas Microsoft, surgió la necesidad de recrear el mismo concepto para los equipos móviles. Así surgió .NET Compact Framework, un subset del .NET Framework conocido en aplicaciones de escritorio y aplicaciones web. Las aplicaciones bajo este paradigma se conocen como Smart Device Application.

Como podemos ver en el gráfico anterior este framework no soporta el desarrollo de aplicaciones instalables para celulares.

6.1.3.1. .NET Compact Framework

Desde la salida de Visual Studio 2003, surgió esta versión compacta y reducida del .NET Framework. Consta de un subset de clases (y cada clase, un subset de métodos y propiedades) del .NET Framework mayor, e incorpora algunos namespaces dedicados específicamente al trabajo sobre equipos móviles.

La última versión es la .NET Compact Framework 1.0 Service Pack 2. Esta versión compacta ocupa generalmente entre 2 y 3 MB en memoria del equipo, contra los casi 24 MB de la versión mayor.

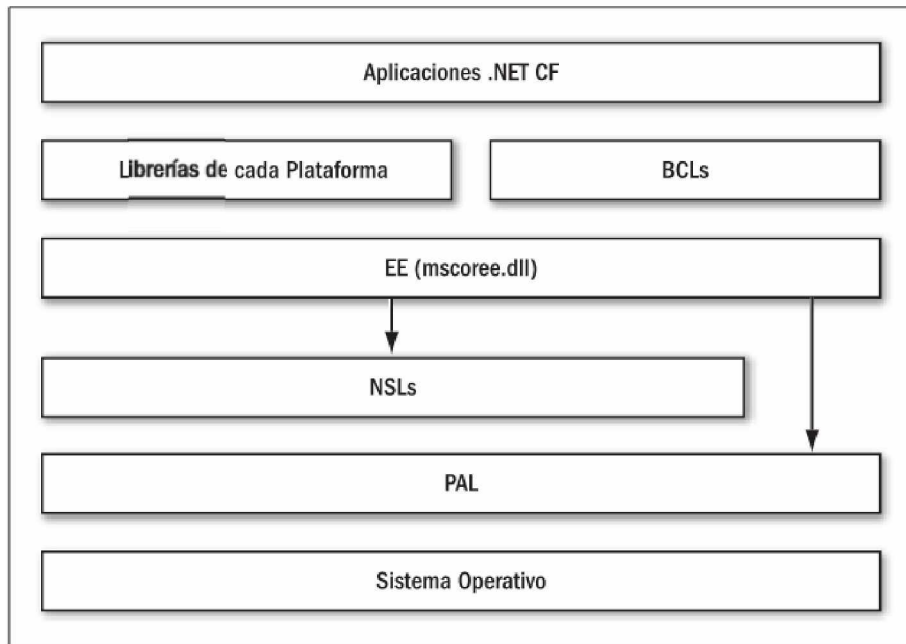


Figura 16. Arquitectura del Compact Framework de Microsoft⁴¹

En primer término hay que definir el entorno de desarrollo donde trabajar para realizar desarrollos móviles con .NET. Para el caso de desarrollos para SD (otra abreviatura muy utilizada, de Smart Devices o dispositivos inteligentes) la única opción será Visual Studio.NET(VS.NET).

⁴¹ DESARROLLOS MÓVILES CON .NET editorial mpediciones CAP 1

CAPÍTULO VII

STÁNDARES DE DISEÑO WEB – MÓVIL.

7. ESTANDARES BASADOS EN LA WAP 1.X Y EN LA WAP 2.0



7.1. WML

7.1.1. Introducción

WML(Wireless Markup Language-Lenguaje inalámbrico de marcas) es el primer lenguaje estándar para dispositivos inalámbricos. Es soportado por la mayoría de fabricantes de celulares y demás dispositivos que soportan tecnología WAP.

El rol de WML en las aplicaciones de Internet móvil es similar al HTML en las aplicaciones web. Los sitios WAP están escritos en WML como los WEB en HTML.

WML 1.x es muy similar a HTML. Ambos usan tags y están escritos en texto plano y siguen estándares de la WAP 1.x

Algunos funcionamientos de WML son específicos para dispositivos WAP. Por ejemplo WML provee a los desarrolladores una manera de programar eventos en los

softkeys de teléfonos móvil. Esta característica no esta soportada por HTML obviamente por que las PC no tienen teclas de celular.

WML soporta script de cliente. Este lenguaje es llamado WMLScript. Su sintaxis esta basada en JavaScript.

La WAP 1.x es una vieja versión del estándar WAP. La más reciente versión es WAP 2.0.

El lenguaje de marcas definido en WAP 2.0 es XHTML-MP(XHTML-mobile Profile). Este es un subconjunto de XHTML usado en la web. XHTML-MP soporta la versión mobile de estilos en cascada (WCSS). Este es un subconjunto de CSS2 usado en la web.

La mayoría de dispositivos nuevos publicados bajo la WAP 2.0, son compatibles con la WAP 1.x, los nuevos dispositivos bajo la WAP 2.0 pueden mostrar tanto XHTML-MP como WML⁴².

⁴² www.developershome.com/wap/wml/default.htm

Para desarrollar un sitio WAP y estamos considerando WML como lenguaje de marcas tenemos ventajas como desventajas, vamos a listar las que a nuestra consideración nos parece las principales:

7.1.2. Ventajas:

- ω Soporta WMLScript lenguaje script del lado del cliente, que nos permite validar formularios de entrada en el cliente mejorando la velocidad de respuesta en validaciones del tipo formularios de registro de datos.

- ω Control de eventos disparados por las teclas del celular.

- ω Definición de variables.

- ω Soporta timers para efectuar diversos controles.

7.1.3. Desventajas:

- ω La principal desventaja es que no es soportada por browsers WEB con lo que las páginas no serian estándares en un futuro donde todo en la informática tiende a estandarizarse.

- ω No esta dentro del estándar para desarrollar contra WAP 2.0.

7.1.4. Características del Lenguaje WML

El lenguaje esta basado en el XML pero con tags o marcas más restringidas, leído e interpretado por un micro navegador instalado en el dispositivo WAP.

La mayor diferencia entre WML y HTML es que el elemento básico en la navegación de HTML es una página mientras que en WML es una carta. Un archivo WML puede contener múltiples cartas y estas formar una baraja.

Cuando un usuario se dirige a un sitio WAP el browser carga el archivo WML o XHTML que contiene una carta o una baraja. Solo una carta será mostrada cada vez en el dispositivo móvil.

Si el usuario se dirige a otra carta de la misma baraja, el browser móvil no tiene que enviar una nueva petición hacia el servidor puesto que el archivo que contiene la baraja ya está guardado en el dispositivo inalámbrico.

WML y XHTML están diseñados de esta manera debido a la alta latencia que tiene las comunicaciones inalámbricas, descargarse una baraja puede disminuir los viajes de ida y vuelta hacia el servidor.

En una carta pueden colocarse texto, imágenes, colores, etiquetas, campos de entrada etc.

7.1.5. Estructura de un documento WML

En la Figura 17 presentamos el código básico de “hola mundo” en lenguaje WML.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
  <card id="card1" title="WML Tutorial">
    <p> hola mundo </p>
  </card>

  <card id="card2" title="WML Tutorial">
    <p>Bienvenido al mundo WML</p>
  </card>
</wml>
```

Figura 17. Código básico de en WML

El resultado lo presentamos en la 0 en diferentes modelos de emuladores.



Figura 18. Navegador o emulador móvil

Para propósitos de esta tesis no se profundizará en el estudio de WML ya que el aplicativo estará desarrollado según el estándar de la WAP 2.0.

7.2. XHTML

7.2. *Introducción*

XHTML MP (Lenguaje Extensible de Marcas Perfil Móvil). Es el lenguaje definido en el estándar WAP 2.0. Wap 2.0 es la especificación más reciente creada por la WAP Forum (ahora Open Mobile Alliance [OMA]). La especificación de WAP CSS (WAP Estilos en Cascada o WCSS) está también definida en la WAP 2.0 y trabaja junto con

XHTML MP. Con WCSS se puede cambiar la apariencia de presentación de las paginas XHTML MP.

XHTML MP es un subconjunto de XHTML, el cual es una versión estricta de HTML.

XHTML MP es XHTML Basic (también un subconjunto de XHTML) con algunos elementos adicionales y atributos de la versión completa de XHTML.

El objetivo de XHTML Mobile Profile es juntar las tecnologías de Internet mobile y de la World Wide Web. Antes de la llegada de XHTML Mobile Profile los desarrolladores WAP creaban sus sitios WAP utilizando WML y WMLScript, mientras los desarrolladores Web lo hacían usando HTML / XHTML y CSS.

Con el manifiesto de XHTML Mobile Profile, el mundo inalámbrico de marcas, y el mundo de red global convergen. XHTML Mobile Profile y WAP CSS dan a los desarrolladores de aplicaciones inalámbricas en Internet más y mejor control de la presentación. Sin embargo el gran avance es que ahora las dos tecnologías pueden ser utilizadas para crear ambos sitios versión inalámbrica y versión web de nuestros sitios de internet. Se puede usar cualquier web browser para ver las aplicaciones WAP 2.0 durante el proceso de desarrollo y prototipo.

7.2.1. Reglas Sintaxis.

Al ser XHTML MP un subconjunto de XHTML, las reglas de sintaxis XHTML MP siguen los mismos patrones:

1. Todos los tags deben estar debidamente cerrados.

Todas las tags en XHTML MP deben cerrar apropiadamente. Por ejemplo, el siguiente código no es correcto en XHTML MP dado que `</p>` no existe o está extraviado.

```
<p>XHTML MP tutorial 1
```

```
<p>XHTML MP tutorial 2
```

```
<p>XHTML MP tutorial 3
```

Esto es correcto en XHTML MP:

```
<p>XHTML MP tutorial 1</p>
```

```
<p>XHTML MP tutorial 2</p>
```

```
<p>XHTML MP tutorial 3</p>
```

Algunas tags en XHTML MP no vienen en pares ya que no hay contenido a ser encerrado. Un ejemplo es el tag `
` que es usado para dar un salto de línea. Para cerrar estos tags se debe colocar `"/` en el final del tag antes de `>`.

El siguiente código es incorrecto en XHTML MP.

Salto de línea

Este es correcto en XHTML MP:

Salto de línea

2. Tags y atributos deben estar en minúsculas.

En XHTML MP el código de marcas es case-sensitive. Todos los tags y atributos deben estar en minúsculas.

El siguiente código es incorrecto ya que los tags <P></P> y los atributos (ID) están en mayúsculas

```
<P ID="p1">XHTML MP tutorial 1</P>
```

```
<P ID="p2">XHTML MP tutorial 2</P>
```

```
<P ID="p3">XHTML MP tutorial 3</P>
```

El siguiente código es correcto XHTML MP.

`<p id="p1">XHTML MP tutorial 1</p>`

`<p id="p2">XHTML MP tutorial 2</p>`

`<p id="p3">XHTML MP tutorial 3</p>`

3. Todos los valores de atributos deben estar encerrados entre comillas.

Los valores de todos los atributos deben estar encerrados entre comillas en el estándar XHTML MP. Se puede usar comillas simples o comillas dobles.

El siguiente código es incorrecto en XHTML MP.

`<p id=p1>XHTML MP tutorial 1</p>`

`<p id=p2>XHTML MP tutorial 2</p>`

`<p id=p3>XHTML MP tutorial 3</p>`

Este es correcto en XHTML MP:

`<p id="p1">XHTML MP tutorial 1</p>`

`<p id="p2">XHTML MP tutorial 2</p>`

`<p id="p3">XHTML MP tutorial 3</p>`

También puede usar comillas simples

```
<p id='p1'>XHTML MP tutorial 1</p>
```

```
<p id='p2'>XHTML MP tutorial 2</p>
```

```
<p id='p3'>XHTML MP tutorial 3</p>
```

4. No se permite atributos minimizados.

En el caso de algunos atributos hay solo un valor posible. En HTML, se puede dejar este atributo sin colocar ningún valor en tales casos. Esto es llamado minimización de atributos.

Por ejemplo, en el siguiente código de marcas, se tiene definido un check box que es inicializado en el estado checked. Como el atributo checked solo puede ser inicializado en el estado “checked”, se puede omitir el valor que sería “checked” esto en HTML.

```
<input type="checkbox" checked />
```

Sin embargo, la minimización no está permitida en XHTML MP. El código anterior es incorrecto en el estándar XHTML MP. Lo correcto es inicializar todo atributo con su respectivo valor tal como esto:

```
<input type="checkbox" checked="checked" />
```

5. No se permiten tags cruzadas.

Los tags deben estar anidados apropiadamente en XHTML MP. Tags superpuestos no esta permitido. El siguiente código esta incorrecto en el estándar XHTML MP dado que los tags se sobreponen.

```
<p><b>XHTML MP tutorial 1</p></b>  
<i><p>XHTML MP tutorial 2</i></p>  
<p><b><i>XHTML MP tutorial 3</p></i></b>
```

El siguiente código es correcto los tags están anidados apropiadamente.

```
<p><b>XHTML MP tutorial 1</b></p>  
<p><i>XHTML MP tutorial 2</i></p>  
<p><b><i>XHTML MP tutorial 3</i></b></p>
```

7.3. WCSS

7.3.1. Introducción

WCSS (WAP Cascading Style Sheet o WAP CSS) es la versión mobile de CSS. Y es un subconjunto de CSS2 (Estilos en cascada de la World Wide Web) con algunas especificaciones WAP.

WAP CSS es el compañero de XHTML Mobile Profile. Ambos están definidos en el estándar de especificación WAP 2.0, el cual fue creado para formar parte de la WAP Forum (Actualmente la OMA).

La idea de CSS es simple, poner estilos y disposiciones de la información en el documento los cuales están definidos por las reglas de CSS.

Para cambiar la apariencia del documento en el browser solo se necesita modificar el WCSS apropiadamente.

Se puede controlar la apariencia del documento en los diferentes dispositivos móviles haciendo uso del WCSS y no se necesita modificar el contenido del archivo.

Una de las principales carencias de WML es no soportar WCSS por lo que la WAP 2.0 aconseja usar el binomio XHTML MP y WCSS.

Reglas de sintaxis en WCSS

La sintaxis de WCSS no es difícil de aprender. Las sentencias de WCSS consisten en un selector, una propiedad, y un valor de la propiedad.

El selector especifica el elemento afectado por el estilo. Típicamente, el selector es un elemento XHTML MP al cual se quiere aplicar el estilo.

La propiedad especifica el estilo que se quiera dar. El valor de la propiedad es el valor que se asigna a cierta propiedad. La sintaxis de una sentencia WAP CSS sería así:

```
selector {property: property_value}
```

Este es un ejemplo de una sentencia WCSS.

```
p {text-align: right}
```

La sentencia de estilo anterior le dice al navegador WAP que alinee todo texto encerrado dentro de tags <p> </p> horizontalmente a la derecha.

Múltiples propiedades pueden ser asignadas y deben separarse por punto y comas de la siguiente manera:

```
selector {property1: property_value1; property2: property_value2; ... propertyN:  
property_valueN}
```

A continuación otro ejemplo de sentencia WCSS. Este estilo le dice al navegador WAP que alinee el texto a la derecha y que cambie el color del texto a azul.

```
p {text-align: right; color: blue}
```

Esta permitido especificar mas de un selector. Para separar el selector se utiliza un a coma. La sintaxis es la siguiente:

```
selector1, selector2, ... selectorN {property1: property_value1}
```

En el siguiente ejemplo el estilo encierra el texto entre pares de tags `<p></p>` y `<h1></h1>` para alinear a la derecha y cambiar el color a azul.

```
p, h1 {text-align: right; color: blue}
```

Como aplicar estilos WCSS a un documento XHTML MP

Hay tres maneras de aplicar estilos WCSS a un documento XHTML MP

- ω Vinculando a un estilo en cascada externo.

- ω Colocando el estilo WCSS en la cabecera.

- ω Especificando el WCSS con un atributo de estilo.

Para separar el estilo WCSS y el documento XHTML MP, se debería emplear la primera forma. Con la segunda y tercera forma. Los estilos WCSS son puestos dentro del documento XHTML MP y no se puede disfrutar de los beneficios de separar la presentación y el contenido.

Link a Estilo de Códigos en Cascada externo

El elemento <link> es usado para referenciar a un WCSS externo. Este elemento puede ser colocado en la cabecera del documento, su sintaxis es la siguiente.

```
<link href="url_of_style_sheet" rel="stylesheet" type="text/css"/>
```

El atributo href. Especifica la URL al estilo WAP WCSS. No hay necesidad de cambiar los valores asignados al atributo rel y al type. El atributo rel especifica la relación entre el documento XHTML MP y el WCSS y el atributo type especifica el tipo MIME.

En las Figura 19 y Figura 20 mostramos un ejemplo que especifica como vincular un XHTML MP con estilos en cascada externos:

[\(linkToWCSSFileEg1.xhtml\)](#)

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile
1.0/EN" "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <link href="linkToWCSSFileEg1.css" rel="stylesheet"
    type="text/css"/>
    <title>WCSS Tutorial</title>
  </head>
  <body>
    <h1>Table of Contents</h1>
    <hr/>
    <p>
      WCSS Tutorial Chapter 1<br/>
      WCSS Tutorial Chapter 2<br/>
      WCSS Tutorial Chapter 3<br/>
    </p>
  </body>
</html>
```

Figura 19. Código de XHTML.

[\(linkToWCSSFileEg1.css\)](#)

```
hr {
  color: blue
}

p {
  font-style: italic
}
```

Figura 20. Código de WCSS.

En el estilo en cascada del ejemplo anterior, se especifica el color de la regla horizontal a azul y el estilo de fondo del texto en párrafo y tipo de letra Italia.

En la Figura 21 está como se vería el ejemplo en un navegador o emulador móvil.

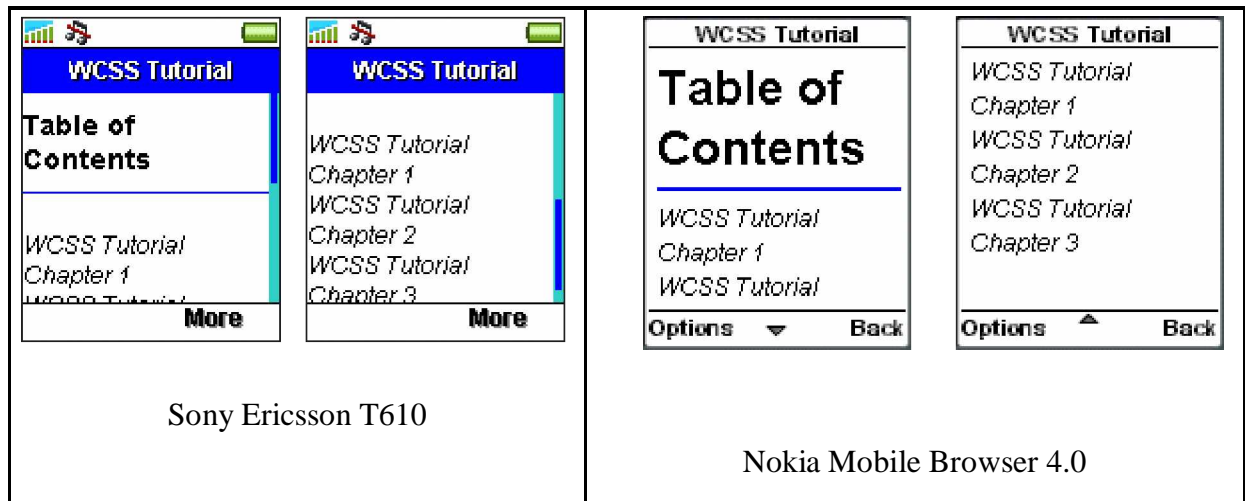


Figura 21. Navegador o emulador móvil

7.4. AJAX EN LAS APLICACIONES WEB MÓVILES

Al tiempo que las aplicaciones web son cada vez más populares y gozan de una creciente aceptación por los usuarios, AJAX será todavía mucho más importante y relevante para la Web Móvil, donde la conectividad y el teclado son elementos condicionantes de la experiencia de usuario.

Tan determinante, o más si cabe, que la propia capacidad de la red es sin duda el tiempo de respuesta. La propia naturaleza síncrona de las aplicaciones Web (petición-respuesta) no favorece precisamente la interactividad y resulta especialmente doloroso en redes móviles donde la capacidad y, sobretodo, la latencia hacen de este modo de

funcionamiento su principal defecto. De hecho, la pobre interactividad de las aplicaciones web en movilidad es uno de los principales factores limitadores para el desarrollo de Internet Móvil hoy en día.

AJAX permitirá mitigar este problema gracias a la comunicación asíncrona con el servidor. La capacidad o la latencia dejarán de ser un problema para las aplicaciones web en movilidad que utilicen AJAX.

AJAX también contribuirá a mejorar la usabilidad de las aplicaciones móviles, aliviando tareas que desde el teléfono resultan tediosas, como por ejemplo la entrada de texto. Con ayuda de AJAX el relleno y comprobación automáticos de campos de texto podrán ser una realidad y permitirán un nuevo concepto dentro de las aplicaciones móviles. Rellenar campos de un formulario desde el teléfono por ejemplo, ya no será algo implanteable.

AJAX es ya un importante paso adelante en el desarrollo de aplicaciones web y su importancia va en aumento. Precisamente, Internet Móvil será el escenario donde AJAX desplegará todo su potencial. El principal motivo del éxito de AJAX radica en que, en sí mismo no es una tecnología nueva sino la utilización conjunta de un grupo de tecnologías maduras y bien conocidas por los desarrolladores: XHTML/CSS, XML, Document Object Model (DOM), JavaScript y XMLHttpRequest.

Numerosos fabricantes de navegadores para móviles ya han anunciado que soportarán AJAX, si no lo están haciendo ya. Por otra parte, AJAX cuenta ya con un importante apoyo dentro de la comunidad de desarrolladores, lo que lo hace todavía más

interesante. Por último, AJAX presenta una ventaja adicional frente a otras plataformas como Java y es su capacidad para acceder a funcionalidad del propio teléfono que está simplemente vetada para aplicaciones Java, como por ejemplo añadir archivos al sistema de ficheros o interactuar con el salva pantallas del teléfono.

AJAX promete ser de gran apoyo para las aplicaciones móviles y seguramente será soportado por la gran mayoría de dispositivos, mientras tanto queda esperar y desarrollar las aplicaciones con las tecnologías actuales lo mas livianas que se pueda.

CAPÍTULO VIII

TECNOLOGÍAS ACTUALES DE APLICACIONES

WEB-MÓVILES.

8. APLICACIONES WEB MÓVILES



8.1. COMPONENTES Y ESTRUCTURA DE UNA APLICACIÓN BÁSICA J2EE.

La plataforma Java nos ofrece un potente framework de trabajo cuando se trata de aplicaciones Stand-Alone las cuales como se pudo observar en el capítulo anterior nos ofrecen diversas posibilidades al desarrollar nuestras aplicaciones como por ejemplo el que se pueden conectar con un servidor a través de Internet e interactuar de acuerdo a las respuestas de este.

En cuanto a las aplicaciones On-Line no hay nada específico aún con lo que podamos trabajar.

Una arquitectura que se puede utilizar si se quiere tener a Java del lado del servidor WAP sería usar JSP's o Servlets y para el lado del cliente WML y WMLScript o XHTML MP y WCSS .

En la Figura 22 un ejemplo demostrativo de cómo sería una aplicación usando JSPs y XHTML MP:

(file.xhtml)

```

<!-- <?xml version="1.0" encoding="UTF-8"?> -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Ingreso de Datos</title>
  </head>
  <body bgcolor="#C0E3F8">
    <form name="frm_Consulta" action="sConsulta.java">
      <h3>-- EMAPAI -- </h3>
      <h4><p>Patr&oacute;n a Buscar: </p> </h4>
      <p><input type="radio" checked="yes" name="lista_datos" value="1" ></input>CUENTA </p>
      <p><input type="radio" name="lista_datos" value="2" checked="yes"></input>CI/RUC </p>
      <h4><p>Inserte Dato:</p></h4>
      <p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="txt_datos" value="" ></input></p>
      <p><input type="submit" value="Consultar" name="btn_Consultar" ></input></p>
      <p><a href="index.xhtml">Regresar</a></p>
    </form>
  </body>
</html>

```

Figura 22. Pagina de ejemplo usando XHTML

En la Figura 23 muestra como se vería en el celular.



Figura 23. Vista en el dispositivo móvil

El código anterior es el que se presentaría en el mini Browser y esta codificado bajo el estándar XHTML MP. Como se puede apreciar hay componentes input tales como text, submit, reset los cuales están dentro de un formulario que será enviado a la Jsp indicada (sConsulta.java).

En la Figura 24 el código del script de lado del servidor.

(s_Conсульта.java)

```
public class s_Conсульта extends HttpServlet {
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Servlet s_Conсульта</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Servlet NewServlet at " + request.getContextPath () + "</h1>");
    out.println("</body>");
    out.println("</html>");
    out.close();
}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

/** Returns a short description of the servlet.
*/
public String getServletInfo() {
    return "Short description";
}
}
```

Figura 24. Script al lado del servidor

Esta página procesa la petición del cliente, en este caso para subir el archivo y presentar los respectivos mensajes de éxito o error.

8.2. PHP.

8.2.1. Componentes y estructura de una aplicación básica PHP

Primeramente cabe recalcar que PHP es un motor potente y versátil para crear sitios-web, y sitios WAP dinámicos.

Para muchos la combinación de PHP al lado del servidor WAP es la mejor opción para desarrollar sitios web móviles y para el lado del cliente WML y WMLScript o XHTML MP y WCSS.

Como podemos ver PHP es un lenguaje de programación que al igual que con la tecnología .Net o J2EE podemos desarrollar aplicaciones web móviles, solo que con ciertos limitantes al momento de desarrollar este tipo de aplicaciones, con respecto a las otras dos tecnologías como:

- ω No existe un buen soporte para aplicaciones empresariales.

- ω No existe entornos específicos para desarrollar en PHP.

- ω No es 100% Orientado a Objetos.

- ω Las aplicaciones PHP son interpretadas cada vez que se realiza una petición por lo que se ejecutan más lento que las precompiladas en Java ó .NET.

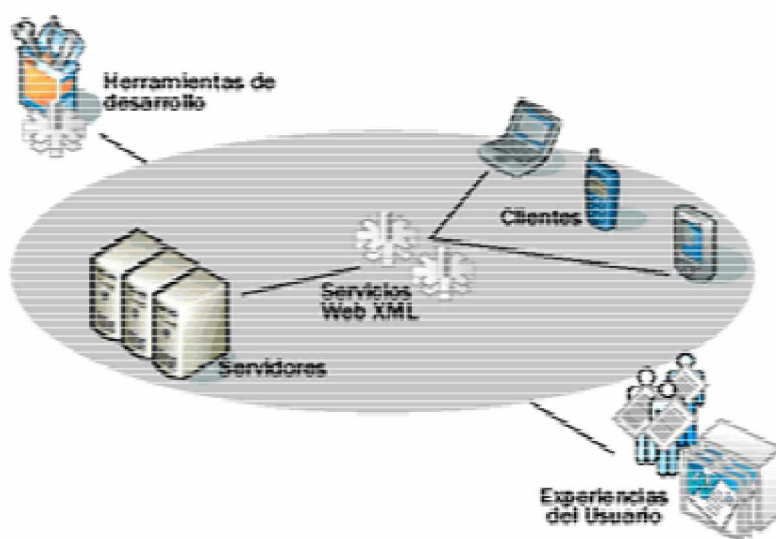


Figura 25. Tecnologías móviles usadas en la actualidad⁴³

Podemos ver que PHP tiene muchas otras cualidades, entre otras, permitir la autenticación de usuarios, el uso de XML, la creación dinámica de imágenes, creación dinámica de archivos y más. Es además fácilmente extensible, pequeño y sencillo. Una maravilla desde el punto de vista de algunos desarrolladores, al momento de desarrollar aplicaciones web.

⁴³ Proyecto de Teclado sobre una PDA para Personas con Parálisis Cerebral capítulo 6 Visual Studio .NET 2003 para dispositivos móviles

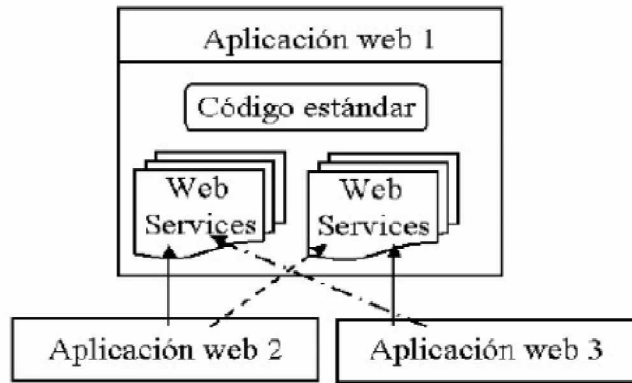


Figura 26. Esquema del funcionamiento de la aplicación Web ⁴⁴

8.3. PLATAFORMA .NET.

8.3.1. Componentes y estructura de una aplicación básica de .Net

Algunas de las ventajas son el uso de “codebehind” que nos facilita mucho en el trabajo con las diferentes capas como son entre la capa de presentación y la capa de lógica de negocios.

Tenemos dos tipos de archivos los .aspx que son en los cuales se coloca el diseño de las paginas controles asp y código html, aquí es donde colocamos todo el diseño y no nos preocupamos de nada de la lógica de negocios.

El otro tipo de archivo es .cs o .vb dependiendo del lenguaje que se desee utilizar que puede ser C# o Visual Basic respectivamente.

⁴⁴ Proyecto de Teclado sobre una PDA para Personas con Parálisis Cerebral capitulo 6 Visual Studio .NET 2003 para dispositivos móviles

Otras de las bondades de las aplicaciones ASP.NET es que los controles de formulario ASP nos facilitan de muchas de las tareas habituales como validación, desplegar listas o tablas enlazadas a bases de datos etc.

Como herramienta de desarrollo esta Visual Studio .NET con su potente entorno de trabajo y soportado por el Framework y el CLR.

8.3.1.1. Aplicaciones ASP.Net

Una aplicación ASP.NET se define como todos los archivos, páginas, manejadores, módulos y códigos ejecutables que se pueden invocar desde un directorio virtual y sus subdirectorios en un servidor de aplicaciones Web.

Web Forms es una función de ASP.NET que puede utilizar para crear la interfaz de usuario para sus aplicaciones Web. Las páginas de Web Forms le ofrecen un modelo de programación potente y preciso para construir interfaces habilitadas para la Web. En cambio, ASP.NET está construida sobre .NET Framework, de manera que todo el marco está disponible para cualquier aplicación ASP.NET. La autoría de sus aplicaciones puede estar en cualquier lenguaje compatible, incluyendo Microsoft Visual Basic, Visual C# y JScript .NET.

Una página de Web Forms consiste en dos partes: los elementos visuales (HTML, controles de servidor y textos estáticos) y la lógica de programación de la página. Visual

Studio almacena cada uno de estos componentes en un archivo por separado. Los elementos visuales se crean en un .aspx y el código en un archivo de clase por separado, llamado archivo de clase con código atrás (.aspx.vb o .aspx.cs). También es posible crear los elementos visuales y el código en el mismo archivo, a lo que en algunas ocasiones se denomina una página Web Forms con “archivo único”.

La lógica para las Web Forms consiste en el código que creo para interactuar con la forma. La lógica de programación reside en un archivo separado del archivo de la interfaz. Este archivo se refiere como el archivo con “código atrás” (code behind) y tiene una extensión ".aspx.vb" o ".aspx.cs". La lógica escrita en el archivo con código atrás puede estar escrita en Visual Basic o Visual C#. Los archivos de clase código atrás para todas las páginas Web Forms en un proyecto están compilados en el proyecto del archivo de biblioteca de vínculos dinámicos (.dll).

8.3.1.2. Qué es un WebForm

Una página Web Forms presenta información para el usuario en cualquier explorador o dispositivo de cliente e implementa la lógica de la aplicación utilizando un código del lado del servidor. El resultado de una página Web Forms puede contener casi cualquier lenguaje que sirve para http, incluyendo HTML, XML, WML y ECMAScript (JScript, JavaScript).

Las páginas Web Forms están basadas en la tecnología ASP.NET de Microsoft en cuyo código que se ejecuta en el servidor dinámicamente genera un resultado de la página

Web para el explorador o dispositivo de clientes. El marco de la página ASP.NET elimina los detalles de la implementación de la separación del cliente y del servidor inherentes en las aplicaciones basadas en el Web al elevar un modelo unificado para responder a los eventos del cliente en un código que se ejecuta en el servidor.

Cuando se compila la unidad de Web Forms, ASP.NET analiza la página y su código, genera una clase nueva dinámicamente, y luego compila esta clase nueva. La clase generada dinámicamente se deriva de la clase Page ASP.NET, pero se extiende mediante controles, su código y el texto HTML estático en el archivo .aspx.

Durante el tiempo de ejecución, la clase Page procesa solicitudes entrantes y responde al crear dinámicamente HTML y agilizarlo de vuelta al explorador. Si la página contiene controles Web (como es común), la clase Page derivada actúa como contenedor para los controles y las instancias de los controles se crean durante el tiempo de ejecución, y así mismo hace llegar el texto HTML al flujo.

Page contiene los siguientes objetos.

- ω Server: Obtiene el objeto Server, que es una instancia de la clase HttpServerUtility.

- ω Application: Obtiene el objeto de la Application para la solicitud Web actual.

- ω Session: Obtiene el objeto Session actual que proviene de ASP.NET.

- ω Request: Obtiene el objeto HttpRequest para la página solicitada. HttpRequest. permite que ASP.NET lea los valores de HTTP que envió un cliente durante una solicitud Web.

- ω Response: Obtiene el objeto HttpResponse asociado con Page. Este objeto le permite enviar datos de respuesta HTTP a un cliente, además contiene información acerca de la respuesta.

8.3.1.3. ASP.NET en el desarrollo WEB Móvil

Pero la ventaja que nos interesa más es que ASP.NET es capaz de crear páginas multidispositivo.

Cuando llevamos Internet al terreno de los dispositivos móviles nos damos cuenta de la heterogeneidad de éstos:

- ω Diferentes lenguajes (HTML, WML, CHTML...)

- ω Diferentes formas (tamaño de pantalla, teclado...)

- ω Diferentes capacidades (mostrar imágenes, colores...)

Con estas diferencias viene la pregunta ¿Cómo se puede mantener una aplicación de servidor para todos los diferentes tipos?

La opción de ASP.NET para hacer páginas multidispositivo son los Mobile Internet Controls, controles de formulario similares a los normales en cuanto a su manejo.

Pero son capaces de renderizarse de forma diferente dependiendo del dispositivo que los visualice.

El programador se olvida que la página es para varios dispositivos y trabaja en una versión única.

- ω Los controles se adaptan automáticamente.
- ω El nivel de adaptación puede ser muy alto
- ω No sólo se adaptan al tipo de dispositivo, puede llegar a adaptarse al modelo concreto.

8.3.2. Mecanismo de Renderizado Dinámico.

Después de diseñar las páginas, IIS (Internet Information Server) realiza un adaptación en tiempo real.

Se tienen unos ‘Adaptadores de Dispositivo’ que son capaces de traducir el control creado al lenguaje apropiado (XHTML, WML, CHTML, HTML. entre otros..).

Estos adaptadores se fijan en las capacidades del dispositivo para optimizar el código de los controles, se analiza el dispositivo que ha pedido la página luego se elige el adaptador adecuado y se traduce el control.

Por estas ventajas se ha elegido ASP.NET como tecnología de desarrollo para las aplicaciones WAP móvil.

Si bien es cierto los estándares de la WAP 2.0 nos aconsejan desarrollar con XHTML MP al momento de hacer una aplicación universal para el sin número de modelos de celulares.

8.3.2.1. Introducción a los Mobile Internet Controls

Los controles móviles se utilizan de forma similar a los controles normales HTML.

Se representan mediante etiquetas y representan controles de formulario, pero los nombres llevan un prefijo que especifica que son Mobile Web Controls.

En la Tabla 13 se muestra los tags que caracterizan a los web y mobile controls.

HTML:	<code><input name="TextNombre"/></code>
Mobile:	<code><mobile:TextBox id="TextNombre"/></code>

Tabla 13 Diferencia entre web y Mobile Controls.

Además, estos controles no se procesan en el cliente.

Por sus características peculiares, es el servidor quien tiene que procesarlos, así que para indicarlo, a todos los controles se les tiene que añadir un atributo extra: `runat="server"`.

```
<mobile:TextBox id="TextNombre"
runat="server"></mobile:textBox>
```

Tabla 14 Atributo de los Controles Mobile Web Controls

Ejemplo

La estructura de una página

Código de página generado en Visual Studio .net 2005

```
<HEAD></HEAD>
```

```
<body>
```

```
    <mobile:Form id="Form1" runat="server">  
        <mobile:Label id="labelHola"  
            runat="server">  
            ¡Hola Mundo!  
        </mobile:Label>  
    </mobile:Form>
```

```
</body>
```

Tabla 15 Código generado en una WebMobileForms.

Los controles se crean en un formulario con etiquetas, pero cada página .aspx tiene un código por detrás (CodeBehind) que define el comportamiento. Ese código reside en otro archivo y es orientado a objetos. Aunque también es posible incrustar código en la misma página aspx (entre los símbolos <% %>) No se recomienda.

La página es en ese archivo CodeBehind una clase.

Cada control y formulario dentro de la página son atributos de la clase.

Así cada control tiene propiedades, métodos.

Los eventos de los controles están reflejados como métodos de la clase (Page_Load).

- Form

Control Form Mobile

<pre><mobile:Form id="F1" runat="server"/></pre> <p>Para pasar a ser formulario activo</p> <pre> this.ActiveForm = "F1"</pre> <p>Paginar el formulario: Atributo Paginate = true</p>

Tabla 16 Código generado de un WebMobileForms server.

- Panel

Control Panel Mobile
<pre><mobile:Panel id="P1" runat="server"></pre> <p style="text-align: center;">[Controles]</p> <pre> </mobile:Panel></pre> <p>Cobra especial importancia si es necesario generar controles dinámicamente</p> <p style="text-align: center;">Atributo Controls</p>

Tabla 17 Código generado de un PanelMobileControl server

- TextView

Control TextView Mobile
<pre><mobile:TextView runat="server">Texto</mobile:TextView></pre> <p>Textos grandes, compatible con la paginación del formulario</p>

Tabla 18 Código generado de un TextViewControl server

- List

Control List Mobile
<pre><mobile:List id="List1" runat="server"> <Item text="Item1" value="1"/> </mobile:List></pre>

Tabla 19 Código generado de un ListControl server

- SelectionList

Control SelectionList Mobile
<pre><mobile:SelectionList id="sel1" runat="server"> <Item text="Item1" value="1"/> </mobile:SelectionList></pre>

Tabla 20 Código generado de un SelectionList server

- PhoneCall

Control PhoneCall Mobile
<pre><mobile:PhoneCall id="call1" runat="server" PhoneNumber="23458961"> Texto </mobile:PhoneCall></pre> <p>Atributos Text y PhoneNumber</p> <p>Al hacer click sobre ella realiza una llamada telefónica</p>

Tabla 21 Código generado de un PhoneCall server

- Link

Control Link Mobile
<pre><mobile:Link id="link1" runat="server" NavigateUrl="http://wap.deusto.es">Texto</mobile:Link></pre>
Atributos Text y NavigateUrl

Tabla 22 Código generado de un Link server

Y más controles como Label, Command, TextBox, Calendar.

CAPÍTULO IX

ANÁLISIS COMPARATIVO DE SOLUCIONES TECNOLÓGICAS ACTUALES

9. ELEMENTOS PARA LA EVALUACIÓN



Cuando pensamos en desarrollar una solución para dispositivos móviles, lo primero que hay que considerar es qué tipo de solución será conveniente para cada necesidad.

Esto dependerá si se requiere una aplicación que se pueda instalar en el celular o una aplicación con la que podamos navegar mediante el celular

En nuestra tesis vamos a desarrollar las dos tipos de soluciones existentes:

- ω Soluciones Stand-alone

- ω Soluciones Online.

Para las aplicaciones móviles vamos a escoger una herramienta de desarrollo que mas se ajuste a satisfacer las necesidades.

9.1. APLICACIONES MÓVILES NATIVAS

9.1.1. Análisis comparativo usando elementos de juicio

Como hemos visto en los capítulos anteriores las dos plataformas usadas en la actualidad son Java y .Net, para soluciones de aplicaciones móviles instalables con la diferencia que .Net solo puede ser usado para aplicaciones en dispositivos Smart Clients.

Luego del estudio optamos por escoger la **Plataforma Java** debido a que la mayoría de los dispositivos celulares existentes en nuestro mercado soportan esta tecnología al tener una KVM, de esta manera pretendemos ofrecer nuestro servicio a la mayor cantidad de usuarios, esta plataforma la usaremos para las aplicaciones stand-alone, debido a que la plataforma .Net no tiene soporte para aplicaciones instalables en los teléfonos móviles solo soporta aplicaciones móviles para Smart Clients que soporten una instalación de Framework CE(compact edition) lo cual un celular con características normales no tiene.

9.2. APLICACIONES WEB-MÓVILES

9.2.1. Análisis comparativo usando elementos de juicio

Como estudiamos en los capítulos anteriores en el mundo de desarrollo de aplicaciones web móviles existen tres tecnologías de los últimos tiempos que mas sobresalen ya que estas tres tecnologías ofrecen un modelo de desarrollo de componentes empresarial (J2EE, .Net, PHP), y se basan o se complementan con los diseños de estandarización soportados por la WAP 2.0.

A la par con estas tecnologías de desarrollo, podemos realizar aplicaciones web-móviles siguiendo los estándares ya analizados en el diseño de este tipo de aplicaciones usando WML y WMLScript soportados por WAP 1.x o XHTML soportados por la WAP 2.0

Luego del estudio se llega a una clara tendencia por .NET y Java junto con los estándares de diseño para aplicaciones web-móviles, quedaría a decisión del desarrollador escoger la opción a usar basándose en aspectos propios de los requerimientos de la aplicación y cuestiones del cliente final.

Como base de datos escogimos SQL 2000 por las licencias adquiridas por el cliente, cabe recalcar que los aplicativos pueden ser cambiados a cualquier version de base de datos.

Optamos por escoger la **Plataforma .Net** para el desarrollo del prototipo WAP-COMMERCE debido a que Microsoft ofrece una serie de tecnologías ideales para el desarrollo de aplicaciones Web Móviles como son:

- ω Controles prediseñados que facilitan el desarrollo e implementación.
- ω Controles adaptadores que permiten filtrar el dispositivo y escoger una interfaz para cada uno dependiendo de sus características.
- ω Un poderoso IDE Visual Studio.Net para depurar, diseñar, configurar y ejecutar.
- ω Manejo de código de negocios y de presentación por separado usando codebehind.

Para el desarrollo del aplicativo implantado en EMAPAI usaremos las tecnologías estándares de la WAP 2.0. Como son XHTML-MP para el lado del cliente y Servlets del lado del servidor.

Las razones para escoger estas tecnologías en la aplicación EMAPAI-WAP son:

- ω Al ser un sistema de uso masivo se requiere aminorar costos al usuario final por lo que XHTML-MP es una mejor opción por ser mas liviano que páginas .aspx

- ω XHTML-MP es soportado en todos los dispositivos que soporten WAP 2.0 publicado en el 2002 por lo cual en la gran mayoría de celulares lo soportan.

- ω Al usar controles nativos se logra ganar en rapidez de navegación.

- ω Estas tecnologías permiten usar herramientas totalmente libres y no pagar ninguna licencia.

CAPÍTULO X

APLICATIVOS



10.DESARROLLO E IMPLEMENTACIÓN DEL APLICATIVO.

10.1. PROTOTIPO WAP-COMMERCE APLICACIÓN WEB-MÓVIL

10.1.1.DESCRIPCIÓN DE LA PROPUESTA

El prototipo de comercio electrónico con tecnología WAP, Web móvil WAP_COMMERCE, posee una conexión a la base de datos parametrizable, además cuenta con una interfaz de desarrollo bastante amigable y sencillo de navegar.

Este prototipo fue realizado en tecnología .Net, con Visual Studio .Net 2005 siguiendo los estándares que trae incorporado este entorno de desarrollo, y usando la última tecnología existente en servicios de navegación WAP.

Accedemos a una base de datos SQL Server 2000.

El prototipo surge de la idea de optimizar tiempo y dinero, facilitando con el uso de celulares el comercio electrónico. Tecnología que en nuestro medio aun no esta explotada.

El sistema WAP-COMMERCE tiene como objetivo principal, crear una tienda virtual conformada por diferentes empresas agrupadas por categorías, dando las facilidades al cliente de comparar precios entre las diferentes empresas y un rápido acceso a listas de productos disponibles con la respectiva información. El cliente puede agregar productos a un carrito de compras luego de esto puede realizar una transacción de compra ingresando sus datos de tarjeta de crédito, otra de las categorías es Hoteles en el cual el cliente luego de seleccionar un hotel puede realizar la reservación de habitaciones filtrando la disponibilidad por fechas.

Este aplicativo fue diseñado para cumplir los siguientes requerimientos:

- ω Mostrará categorías disponibles.

- ω Mostrará empresas agrupadas por las categorías existentes.

- ω Mostrará información de productos de determinada empresa.

- ω En caso de hoteles mostrará un calendario para poder seleccionar la fecha de la cual se visualizará las habitaciones disponibles.

- ω Se validará que la fecha seleccionada sea valida. (Superior a la fecha actual).

- ω El sistema detectará las características del browser y de acuerdo a esto filtrará la renderización de cada control.

- ω Validará los datos de tarjeta de crédito del cliente contra una tabla que simulará a OPTAR.

- ω Realizará las actualizaciones correspondientes de acuerdo a la transacción realizada.

- ω Se le dará al usuario la posibilidad de buscar por el criterio nombre de producto.

- ω Se deberá considerar un límite del valor de compra que será establecido para cada empresa.

10.1.2.Análisis previos

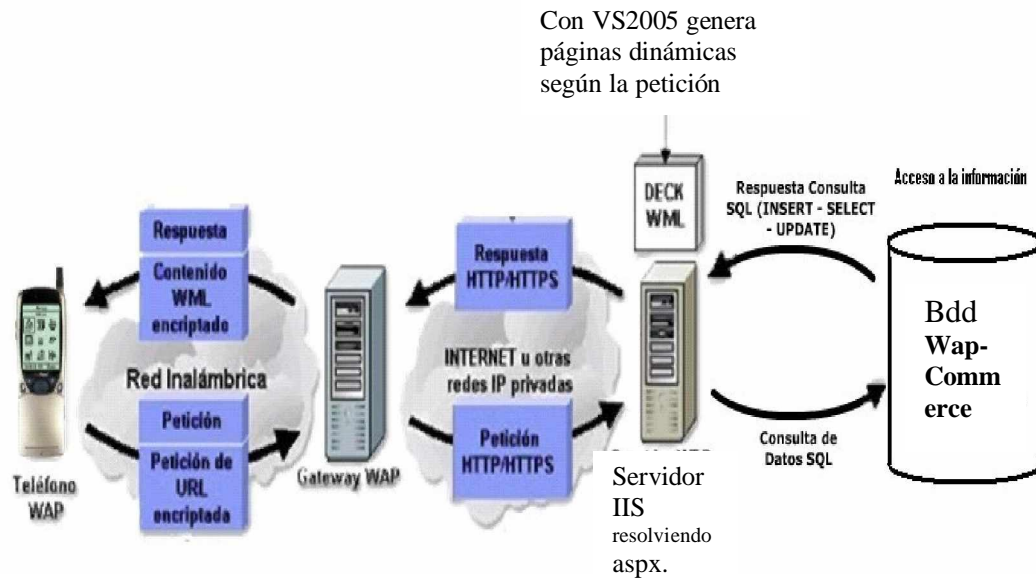
10.1.2.1. Metodología del desarrollo

Nuestro prototipo de comercio electrónico se inició concentrando esfuerzos en la captación de información teórica necesaria para la realización de esta aplicación.

Es un elemento central de la metodología que se genera a partir del intercambio de ideas entre quienes nos encontramos en el desarrollo de este prototipo. Este valor agregado se vuelca principalmente en los resultados finales, siendo al mismo tiempo

una alternativa eficiente e implantable en cualquier empresa que desee incursionar en este medio.

10.1.2.2. Flujo de la información



Flujo de información en el sistema móvil WapCommerce

En el gráfico observamos que: las transacciones WAP utilizan el mismo modelo básico de una operación WEB, siendo la principal diferencia que el teléfono y el gateway WAP sustituyen en conjunto al navegador Web.

Supongamos que el usuario presiona una tecla de un teléfono WAP requiriendo la siguiente

URL: `http://localhost:1654/WapCommerce/index.aspx`

- ω El usuario utiliza un teléfono WAP para solicitar una URL.

- ω El navegador WAP crea una petición que contiene la URL e información que identifique al suscriptor y las envía al gateway WAP.

- ω El gateway WAP interpreta la petición.

- ω El servidor Web IIS (Internet Information Server) interpreta a su vez la petición y determina que recuperar. Si la URL especifica un archivo estático, el servidor Web lo recupera.

- ω El servidor Web coloca un encabezado HTTP o HTTPS en el archivo estático y lo manda de nuevo al gateway WAP.

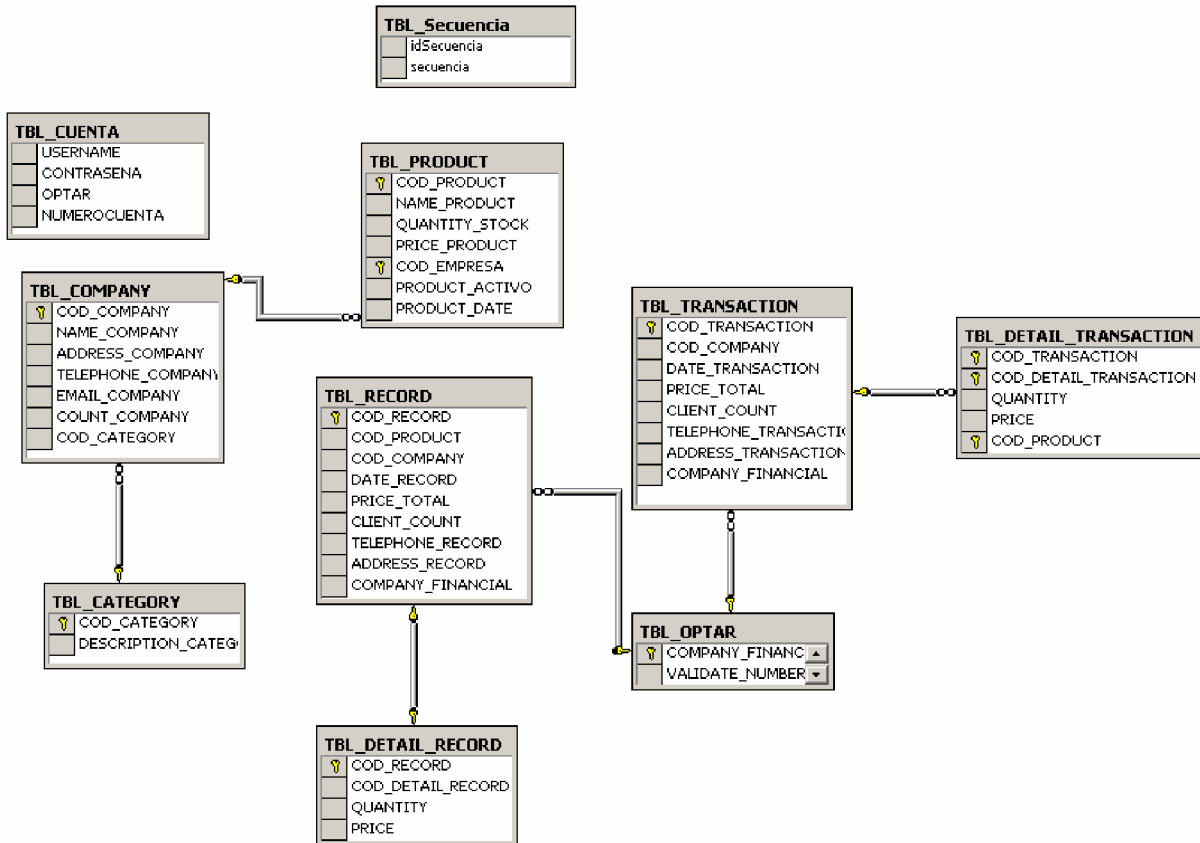
- ω El gateway WAP interpreta la respuesta, valida el aspx mobile, genera una respuesta (quitando el encabezado HTTP o HTTPS) y lo envía al teléfono WAP.

- ω El navegador WAP interpreta la respuesta y despliega el contenido al usuario.

10.1.2.3. Diseño de tablas

Modelo de entidad relación

Nuestro prototipo accede a una conexión de base de datos, a continuación el modelo entidad relación del diseño de la base de datos.



10.1.2.4. Funcionamiento interno del aplicativo Wap-Commerce.

Una transacción en el sistema wap-commerce inicia cuando el cliente llena el formulario de compra con los principales datos de su tarjeta de crédito como son número, pin de seguridad y número de CI del beneficiario.

Es ingresado en el formulario un número de CI para que se indique la persona beneficiaria que podrá ir a retirar la compra; previa verificación de su CI.

En una aplicación real de este tipo se deberá realizar un convenio con OPTAR (Operadora de Tarjetas de crédito) para realizar la validación de la cuenta y disponibilidad en las diferentes entidades de crédito.

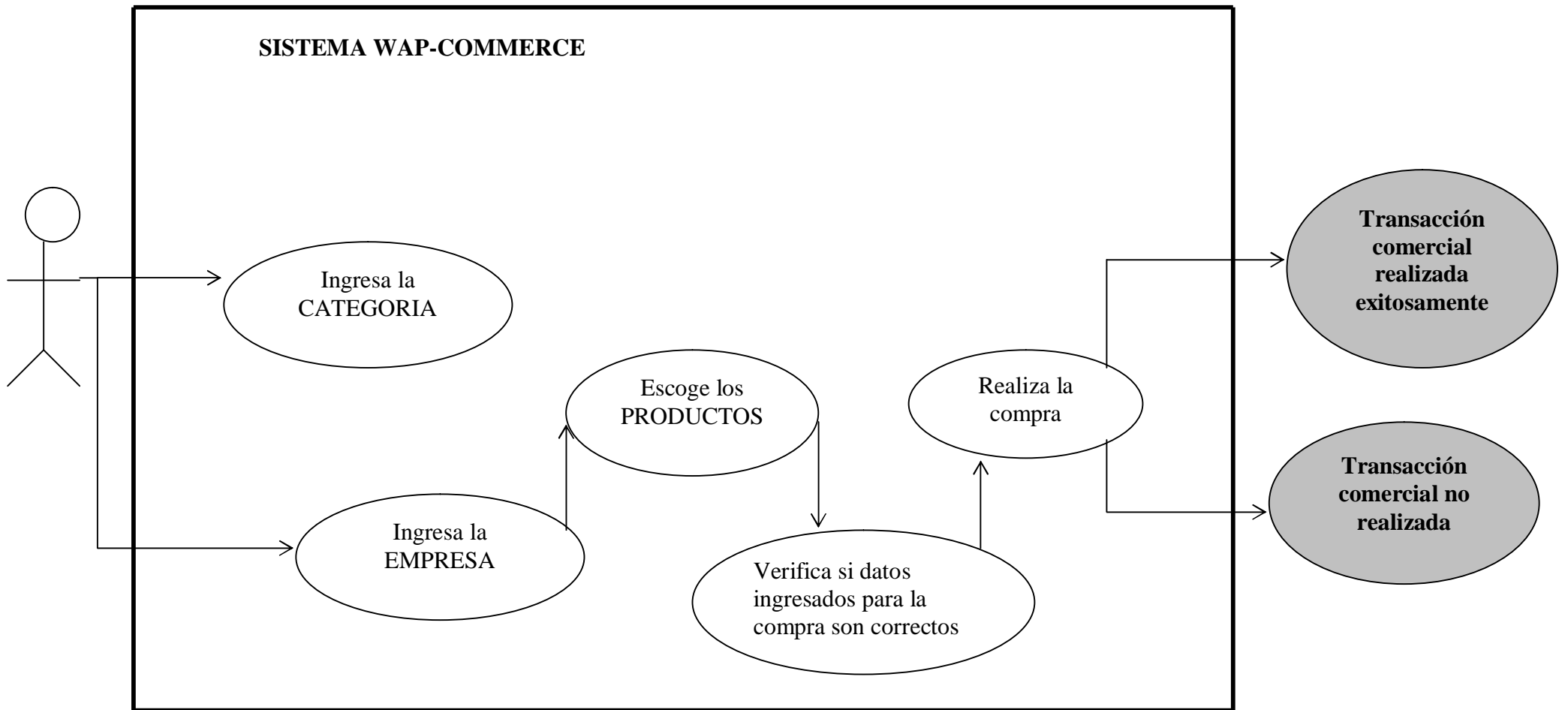
El sistema tiene el control hasta que el cliente llena el formulario de compra donde pasa el control a OPTAR y este devuelve un OK ó ERROR con las respectivas indicaciones de resultado en ese momento el control regresa al sistema que dará al usuario final las respectivas informaciones.

En cuestiones de seguridad se aconseja usar métodos de encriptación al pasar los datos de la tarjeta de crédito a OPTAR para evitar phishing o cualquier intento de ataque en este punto que sería el objetivo de ataque más crítico.

10.1.3.Desarrollo

Dentro del código desarrollado para la aplicación indicamos a continuación las páginas principales, para mostrar información consumida de la base de datos en los celulares, las páginas son con extensión aspx, la descripción con detalle lo encontramos en el manual técnico del aplicativo, **ANEXO VII**.

A continuación en la figura mostramos el diagrama de casos de uso del aplicativo prototipo de comercio electrónico



A continuación el diagrama de clases para ver como esta estructurado el código del sistema



10.1.4. Implantación

Para implantar este prototipo se toma en cuenta que la aplicación fue desarrollada en aspx Visual Studio .Net 2005 accediendo a las base de datos SQL 2000.

Para esto necesitamos instalar los siguientes requerimientos.

Ø Requerimientos:

ω Sistema Operativo:

Windows XP Profesional

ω Base de Datos:

SQL Server Standard 2000

ω Servidor WEB:

IIS (Internet Information Server) 5.1

ω Lenguaje de Programación:

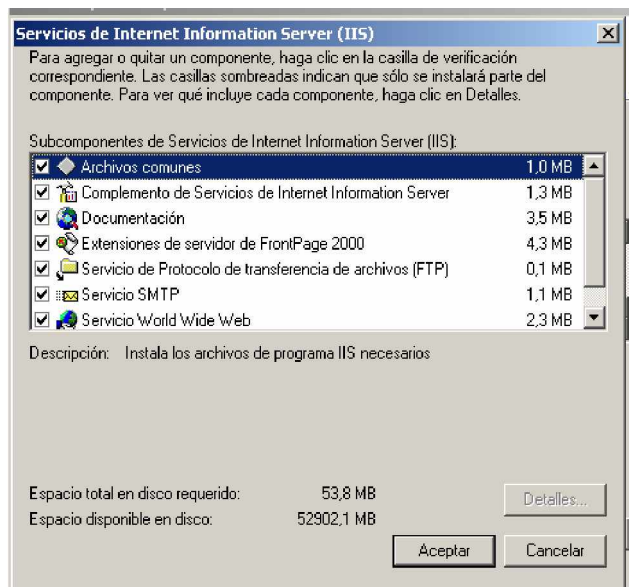
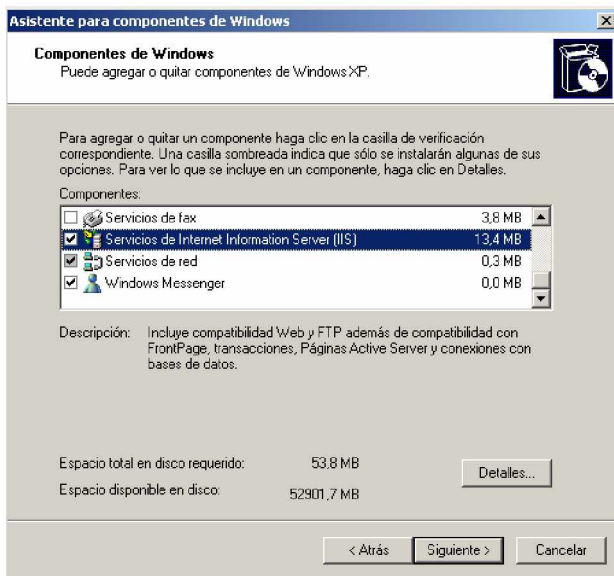
Visual Studio .Net 2005 (C#).

Ø Configuración de IIS 5.1 (Internet Information Server):

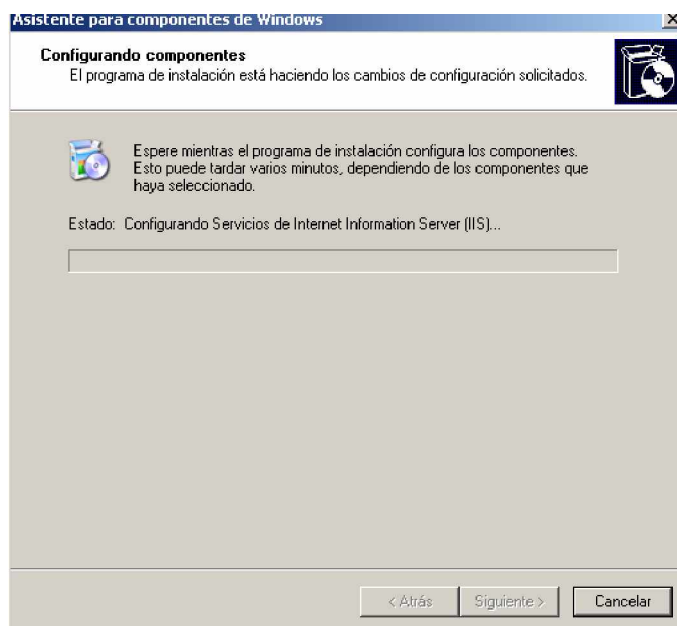
Para la configuración del servidor Web debemos seguir los siguientes pasos.

Instalación de IIS

Para instalar IIS instalamos componentes de Windows. Pulsamos SIGUIENTE sobre la pantalla escogiendo la opción Servicios de Internet Information Server, y escogiendo todos los detalles de IIS



A continuación se instalan todos los componentes de IIS



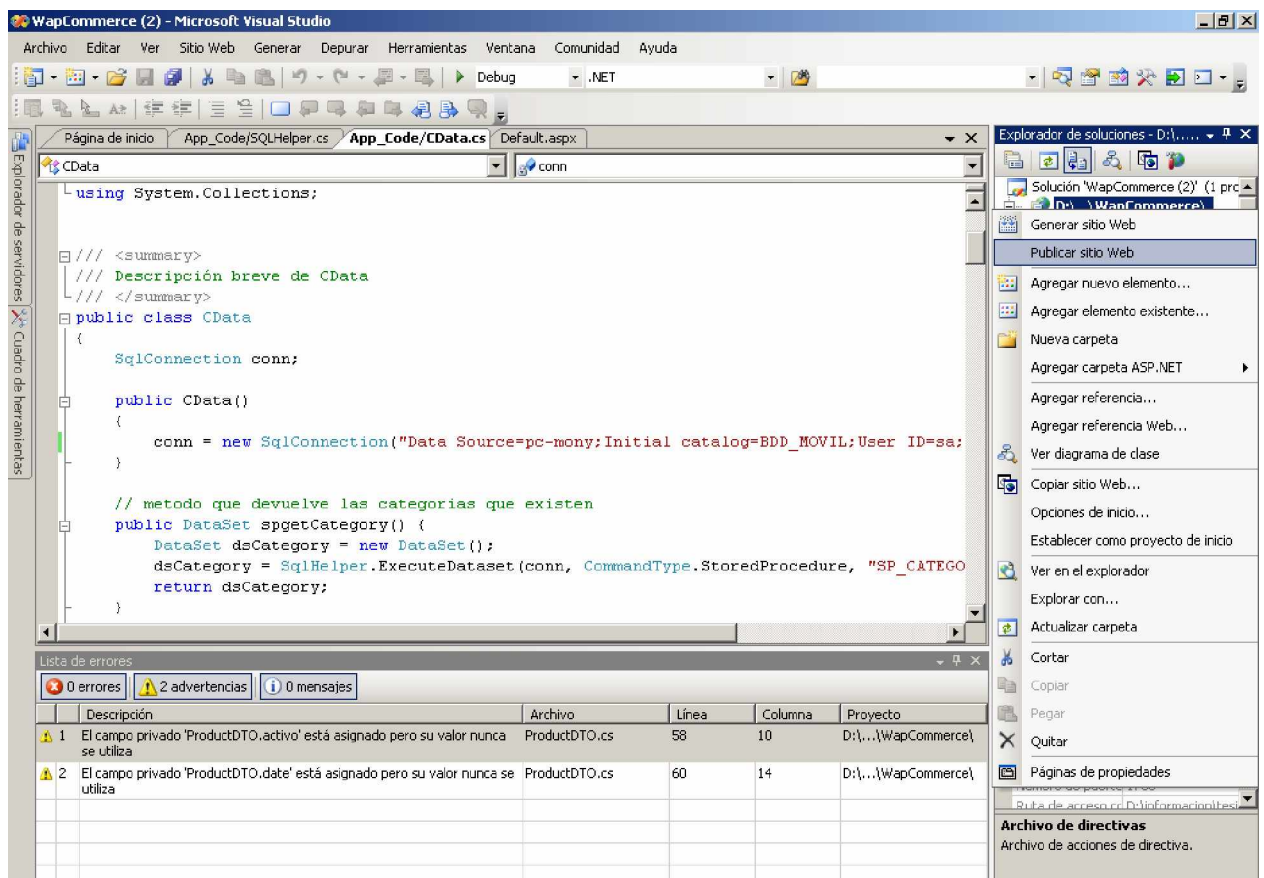
Y finaliza la instalación del servidor:



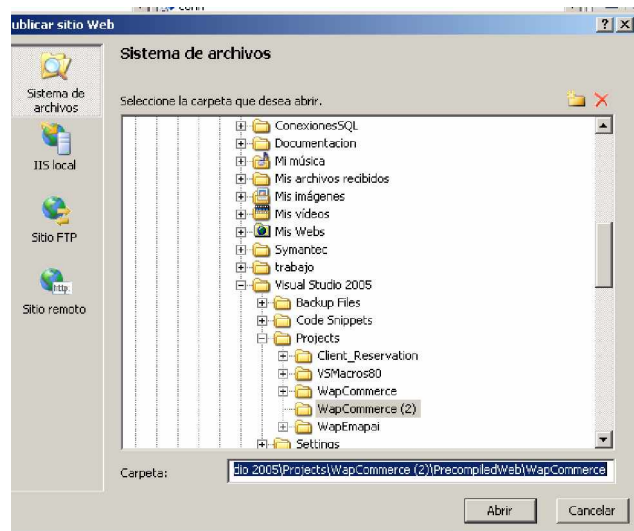
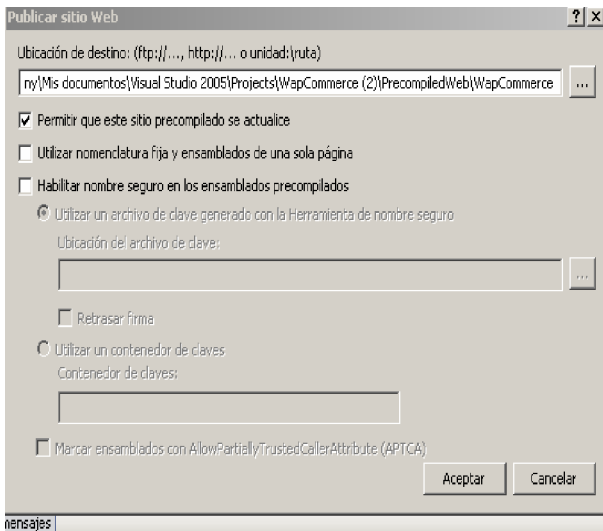
Ø **Configuración de la aplicación :**

Para que la aplicación WAP-Commerce pueda ejecutarse desde el servidor de aplicaciones IIS debemos seguir los siguientes pasos.

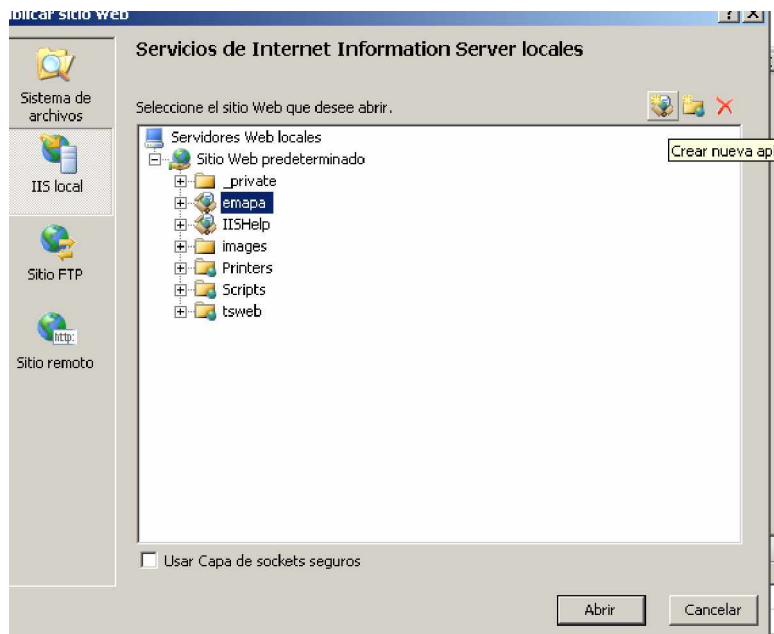
Desde el entorno de desarrollo (VS2005) sobre la aplicación Web Móvil pulsamos el botón derecho y escogemos Publicar Sitio Web



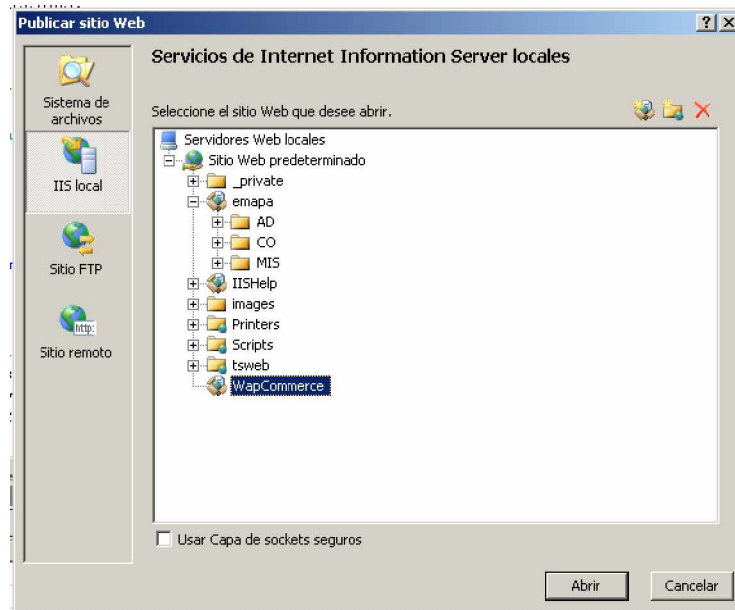
Escogemos el lugar a donde lo vamos a publicar



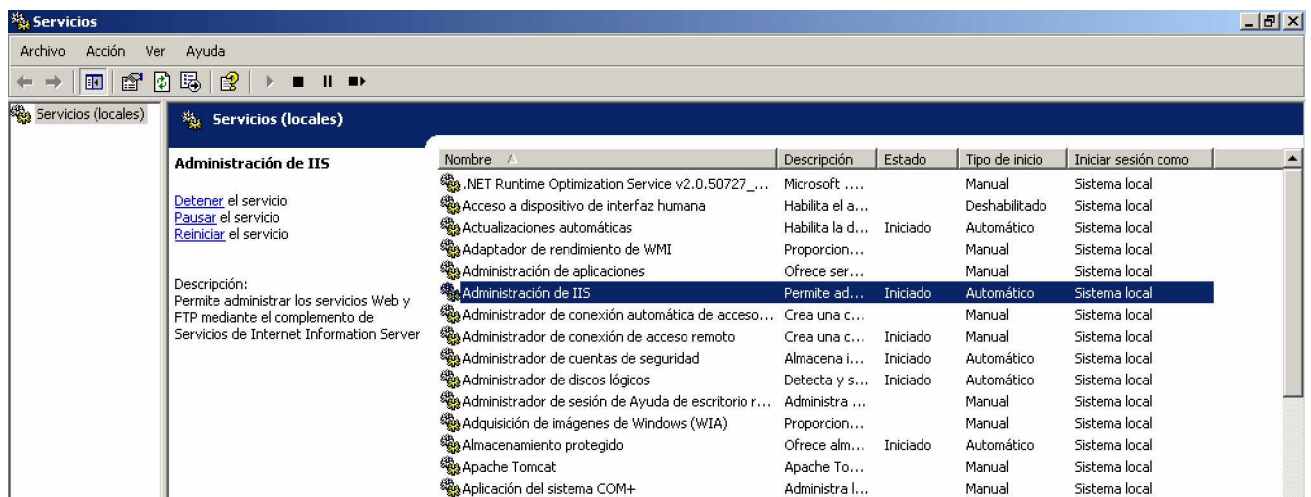
Escogemos la opción de IIS local y creamos una Nueva Aplicación Web en Sitio Web Predeterminado.



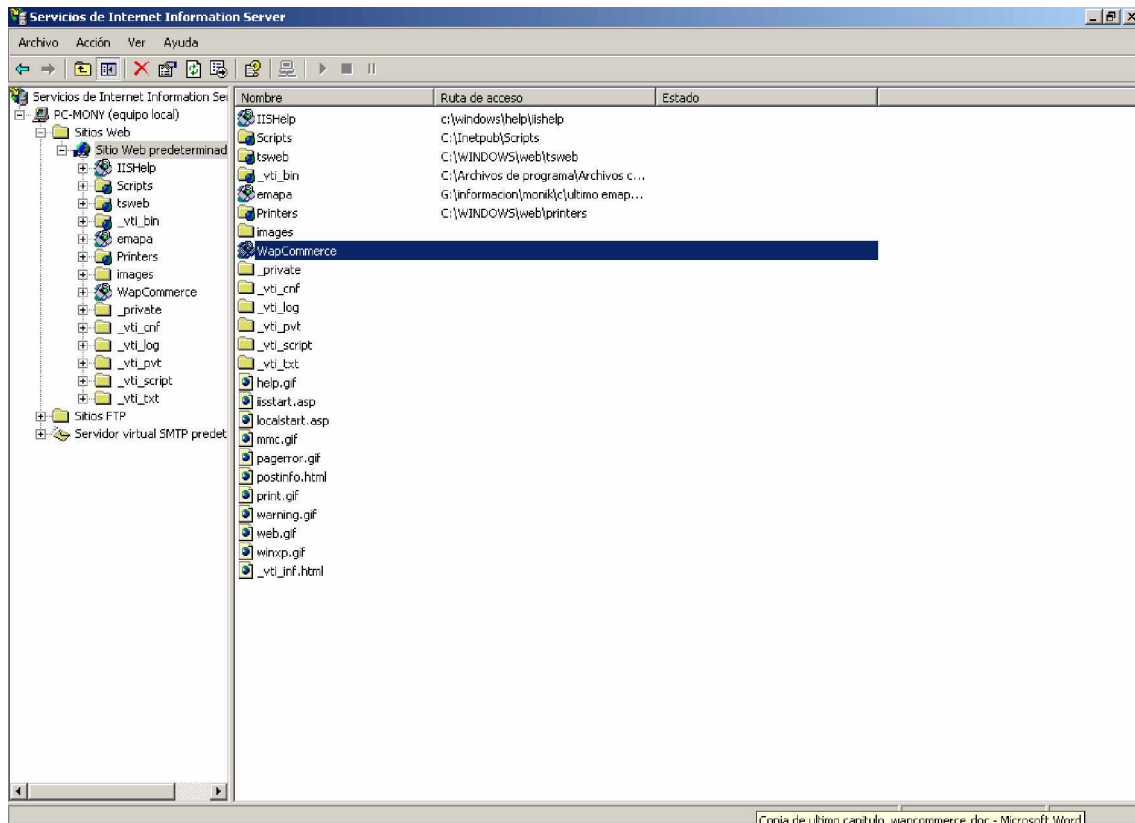
Para nuestro ejemplo lo llamaremos WapCommerce.



Y finalmente aceptamos así se crea nuestro sitio web, y configuramos nuestra aplicaciones, para que nuestra aplicación se ejecute debemos tomar en cuenta que el servicio de Administrador IIS este iniciado



Si ingresamos a herramientas administrativas – Servicios de Internet Information Server en sitio web predeterminado debe aparecer nuestra aplicación en este caso WapCommerce



Ø Configuración para el acceso a la base de datos :

En la Base de Datos no se requiere ninguna configuración adicional. Lo que se debe tomar en cuenta es que en el Servidor SQL Server debe estar iniciado.

Para el acceso a la base de datos creamos la siguiente cadena de conexión:

```
conn = new SqlConnection("Data Source=pc-mony;Initial catalog=BDD_MOVIL;User ID=sa;password=sa");
```

Para acceder a la base de datos, lo hacemos con el usuario sa y su contraseña respectiva, lo cual es transparente al usuario final, al mismo que podemos dar permisos que nosotros deseemos sobre esta base de datos.

Ø **Mayor seguridad**

Para tener una mayor seguridad en nuestro servidor de aplicación instalamos el antivirus para proteger de virus y troyanos.

Además de activar Firewall para proteger el ingreso de intrusos

Y algo importante en la aplicación para tener mayor seguridad en las transacciones comerciales simulamos mediante la tabla TBL_OPTAR, el proceso de autenticación que realiza con los datos de usuario, tarjeta de crédito y password para poder realizar con éxito la transacción.

10.1.5. Manual de usuario

El sistema móvil WAP_COMMERCE, puede ser ejecutado desde un terminal inalámbrico (teléfono móvil, Palm, Pda, etc), habilitado para usar el protocolo WAP, que debe contener un micro navegador o WAP Browser que sirve de interfaz de usuario.

La presentación del sistema lo hemos realizado en un emulador Openwave.

Para ejecutar la aplicación seguimos los siguientes pasos:

1. Conectamos la dirección en donde se encuentra nuestra página de inicio y **OK**.



En la pantalla principal escogemos la categoría que deseamos. Esto sirve para poder seleccionar las empresas de acuerdo a la rama o categoría escogida.

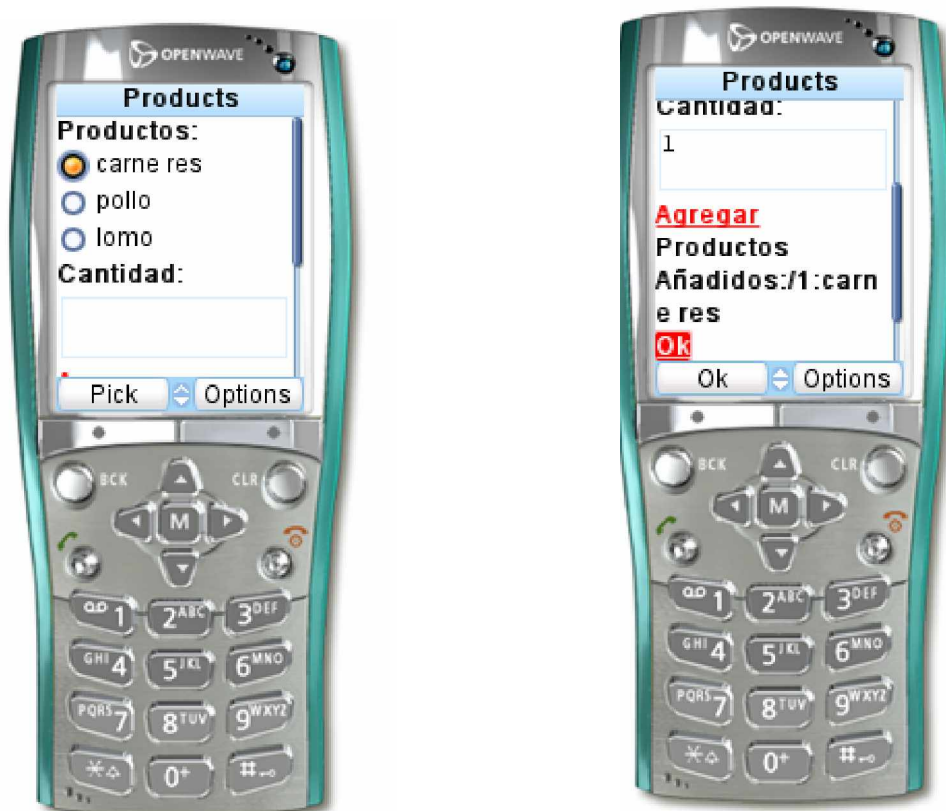
Al seleccionar esta categoría se despliegan sobre la siguiente pantalla todas las empresas que ofrezcan este tipo de servicio de la categoría escogida por ejemplo escogimos la categoría mercados.



Seleccionamos la empresa en la que deseemos realizar la compra y pulsamos ok.

En la siguiente pantalla se despliegan todos los productos que tenga dicha empresa seleccionada por ejemplo aquí los productos del SMX

Además que en esta pantalla existe un cuadro de texto que nos permite ingresar la cantidad que deseemos hacer la compra y los botones de agregar y ok, al fin de la pantalla visualizamos la lista de productos con sus cantidades respectivas de las compras que se esta realizando



Al pulsar sobre el botón Ok se despliega la pantalla donde nos muestra la lista entera de lo solicitado y además un formulario que permite ingresar la compañía o entidad financiera que tengamos, el número de la tarjeta de crédito.



Al aceptar se simula que es validado estos datos ingresados y se realiza la transacción satisfactoriamente en caso de no pasar la validación de los datos no se realiza la transacción comercial



Es el mismo proceso en caso de haber seleccionado las categorías Mercados, Ferreterías, y Farmacias existe una pequeña variación al seleccionar Hoteles.

Al igual luego de seleccionar la categoría en este caso hoteles se muestra una pantalla donde se despliegan los hoteles que ofrecen este servicio.



Al seleccionar el link Ok se muestra la pantalla donde mostramos un calendario para poder escogerla fecha en el que deseamos realizar la reservación de la habitación del hotel que es un control WebMobil propio de VS 2005. En este emulador se muestra un link



Este link muestra algunas opciones para poder escoger un día específico.



Luego de seleccionar el día sigue con el proceso visto anteriormente muestra las habitaciones disponibles para el día seleccionado



Luego se procede a realizar el proceso de transacción explicado anteriormente.

10.2. APLICATIVO EMAPAIWAP

10.2.1.DESCRIPCIÓN DE LA PROPUESTA

Esta aplicación Web móvil esta optimizada para contenido WAP, posee una conexión a la base de datos parametrizable, además cuenta con una interfaz de desarrollo bastante amigable y sencillo de navegar.

Esta aplicación fue realizada en tecnología XHTML, siguiendo los estándares de la WAP 2, usando la última tecnología existente en servicios de navegación WAP,

Accede a una base de datos SQL Server 2000.

El sistema EMAPAIWAP tiene como objetivo principal, mostrar datos informativos del cliente, de las facturas y las planillas emitidas en tiempo real. También dará la opción de que el cliente pueda insertar sugerencias, observaciones, quejas o recomendaciones identificándose por cuenta o CI.

Este aplicativo fue diseñado para cumplir los siguientes requerimientos:

- ω Permitirá consultar los principales datos de la factura por cuenta ó CI.

- ω Permitirá consultar los principales datos de la cuenta ingresando el número de cuenta.

- ω Permitirá visualizar las cuentas asignadas a una persona ingresando el número de cédula.

- ω Permitirá consultar el estado de la cuenta, ingresando el número de cuenta.

- ω Se podrá visualizar un histórico de consumos y valores pagados de determinada cuenta.

- ω Se podrá observar un gráfico estadístico generado dinámicamente del histórico de consumo de la cuenta.

- ω Se podrá ingresar sugerencias, observaciones, quejas o recomendaciones a la base de datos de gestión de la empresa, identificándose con su número de cuenta de usuario ó CI.

- ω Validará que el número de cuenta ingresada o la cédula de ciudadanía sean correctos.

El sistema se ha dado a conocer tanto a la Empresa como a los usuarios en general que utilizarán el mismo, por las ventajas en tiempo y dinero que facilita este aplicativo, se determina que el sistema les va a beneficiar en gran parte a la EMAPAI y a sus usuarios, por lo que se encuentra en producción y a entera satisfacción.

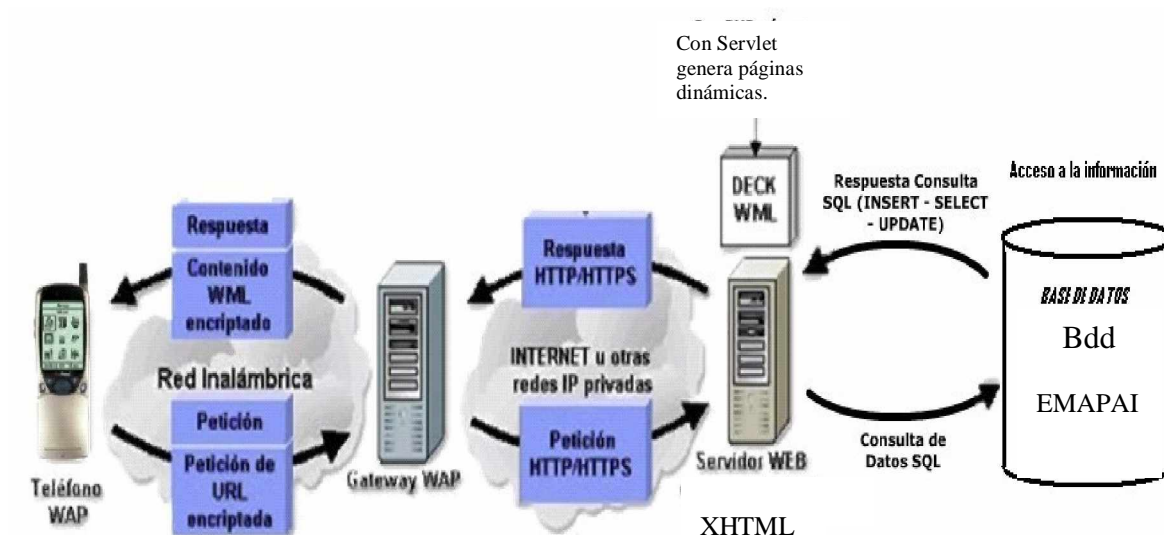
10.2.2. Análisis previos

10.2.2.1. Metodología del desarrollo

Nuestra aplicación se inició concentrando esfuerzos en la captación de información necesaria proporcionada por parte de la EMAPAI mediante la base de datos de gestión de la empresa.

Es un elemento central de la metodología que se genera a partir del intercambio de ideas, experiencias y sugerencias entre las partes involucradas en el desarrollo y administración de la empresa, la universidad y quienes nos encontramos en el desarrollo de este aplicativo. Este valor agregado se vuelca principalmente en los resultados finales, siendo al mismo tiempo una alternativa eficiente para le EMAPAI.

10.2.2.2. Flujo de la información



Flujo de información en el sistema móvil EmapaIWap

En el gráfico observamos que: las transacciones WAP utilizan el mismo modelo básico de una operación WEB, siendo la principal diferencia que el teléfono y el gateway WAP sustituyen en conjunto al navegador Web.

Supongamos que el usuario presiona una tecla de un teléfono WAP requiriendo la siguiente

URL: <http://emapai.ec.nu>

- ω El usuario utiliza un teléfono WAP para solicitar una URL.

- ω El navegador WAP crea una petición que contiene la URL e información que identifique al suscriptor y las envía al gateway WAP.

- ω El gateway WAP interpreta la petición.

- ω El servidor Web tomcat interpreta a su vez la petición y determina que recuperar. Si la URL especifica un archivo estático, el servidor Web lo recupera.

- ω El servidor Web coloca un encabezado HTTP o HTTPS en el archivo estático y lo manda de nuevo al gateway WAP.

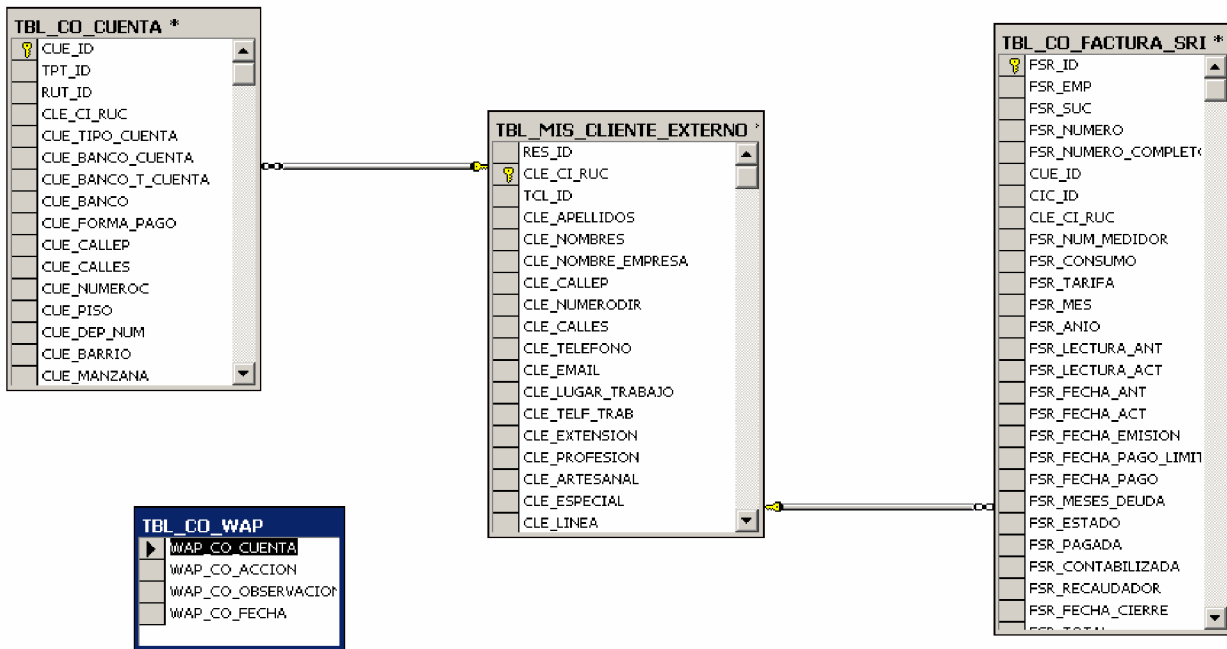
- ω El gateway WAP interpreta la respuesta, valida el XHTML, genera una respuesta (quitando el encabezado HTTP o HTTPS) y lo envía al teléfono WAP.

El navegador WAP interpreta la respuesta y despliega el contenido al usuario.

10.2.2.3. Diseño de tablas

Modelo de entidad relación

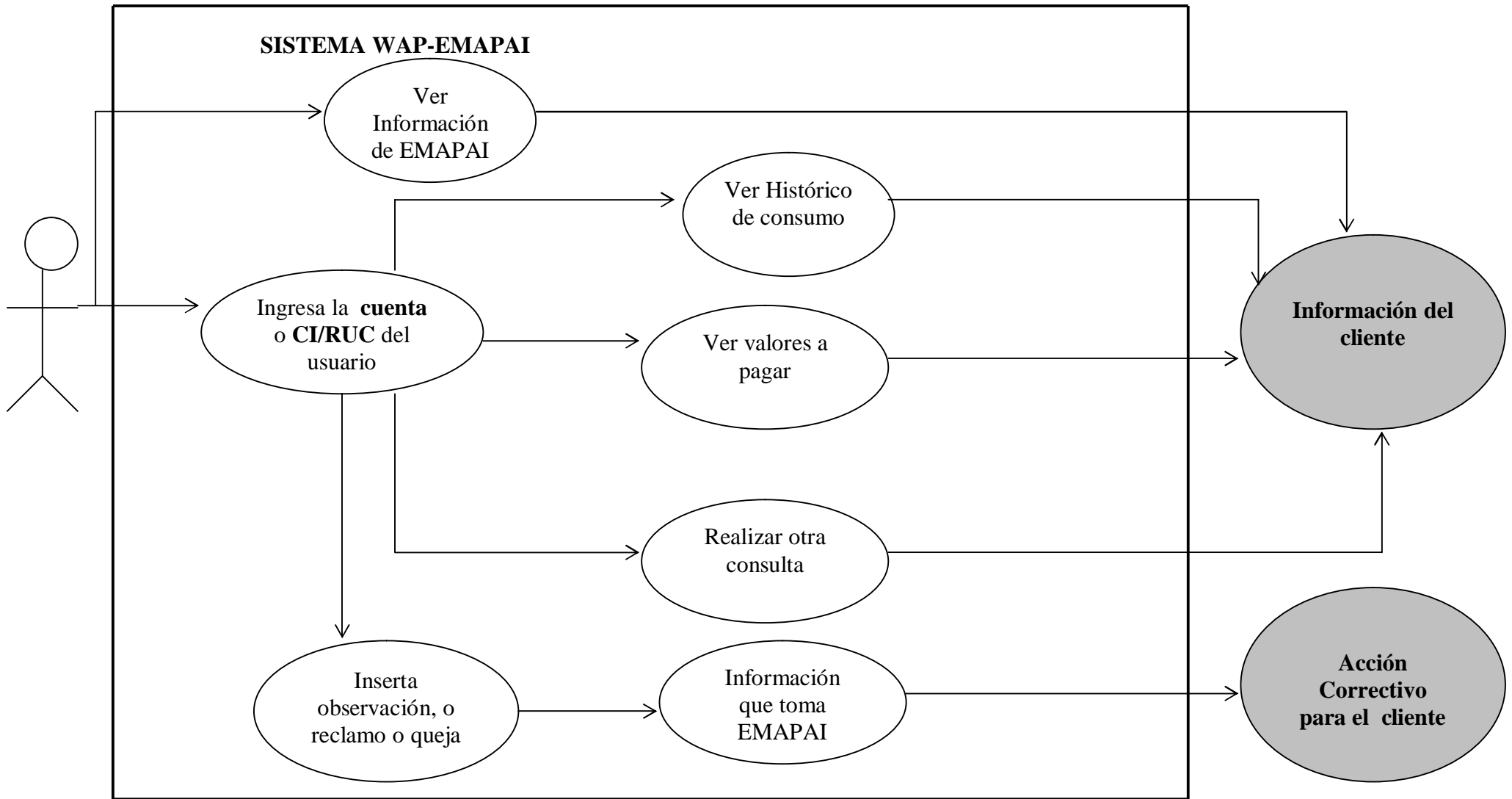
Para la implantación de nuestro aplicativo accedimos a ciertas tablas de la base de datos de gestión de la EMAPAI.



10.2.3.Desarrollo

Dentro del código desarrollado para la aplicación indicamos a continuación las páginas principales, para mostrar información en los celulares las páginas son con extensión XHTML y para la conexión y acceso a la base de datos se la realizó en servlets archivos con extensión java, la descripción con detalle lo encontramos en el manual técnico del aplicativo, **ANEXO V**.

A continuación en la siguiente figura mostramos el diagrama de casos de uso del sistema implantado en la EMAPAI



10.2.4.Implantación

Para implantar en la EMAPAI este aplicativo se tomo en cuenta que la aplicación fue desarrollada en XHTML accediendo mediante los servlets a las base de datos.

Para esto necesitamos instalar los siguientes requerimientos.

Ø Requerimientos:

ω Sistema Operativo:

Windows XP Profesional

ω Base de Datos:

SQL Server Standard 2000

ω Servidor WEB:

Tomcat 5.5

ω Lenguaje de Programación:

XHTML

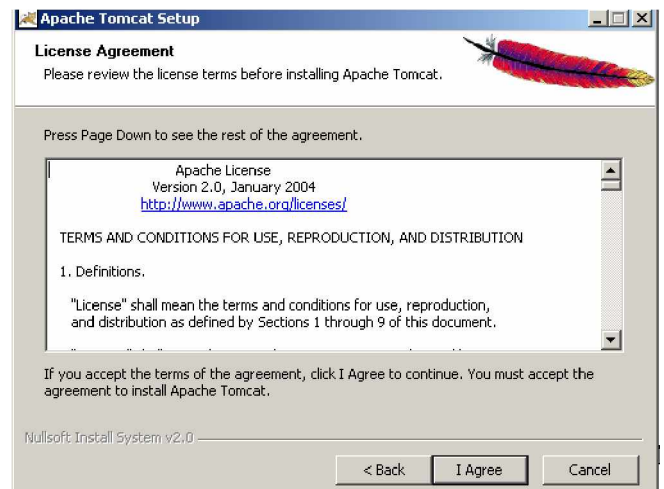
JAVA (Servlets)

Ø Configuración de Tomcat :

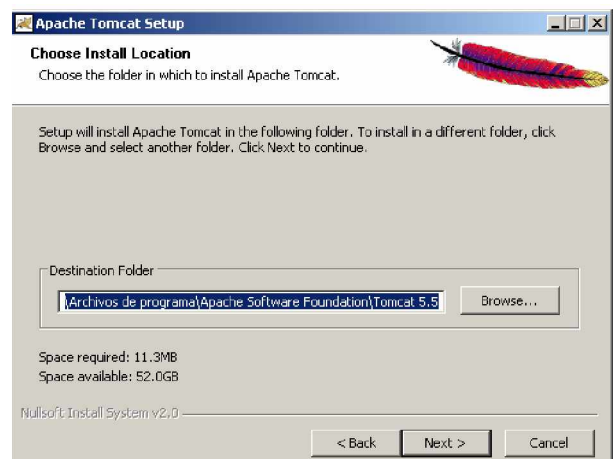
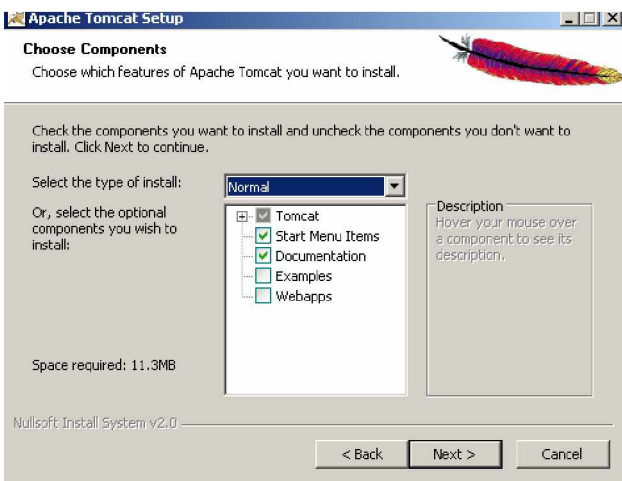
Para la configuración del servidor Web debemos seguir los siguientes pasos.

Ø **Instalación de Tomcat 5.5.15**

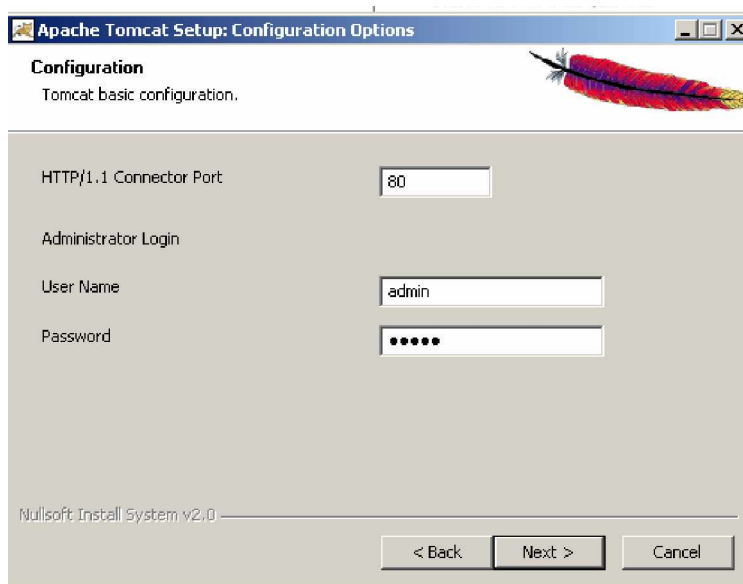
Pulsamos NEXT sobre la pantalla siguiente, y aceptamos las licencias:



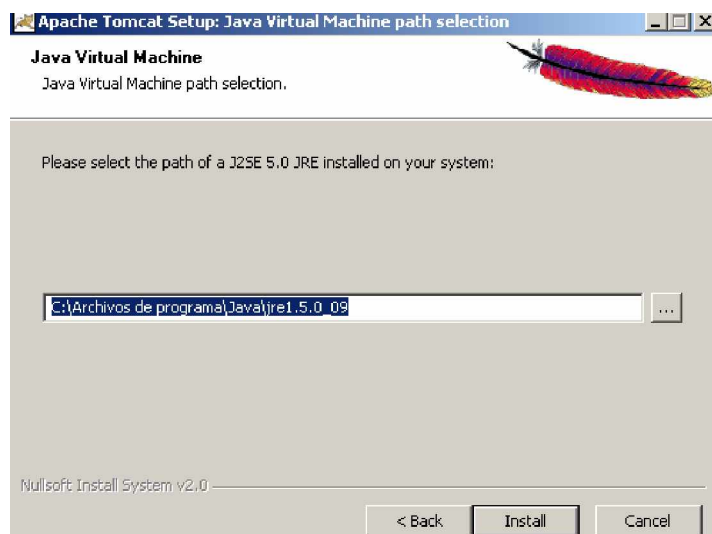
Seleccionamos la forma de instalación Normal y el directorio donde se instalará



Seleccionamos el puerto 80 y escribimos la contraseña del administrado de Tomcat



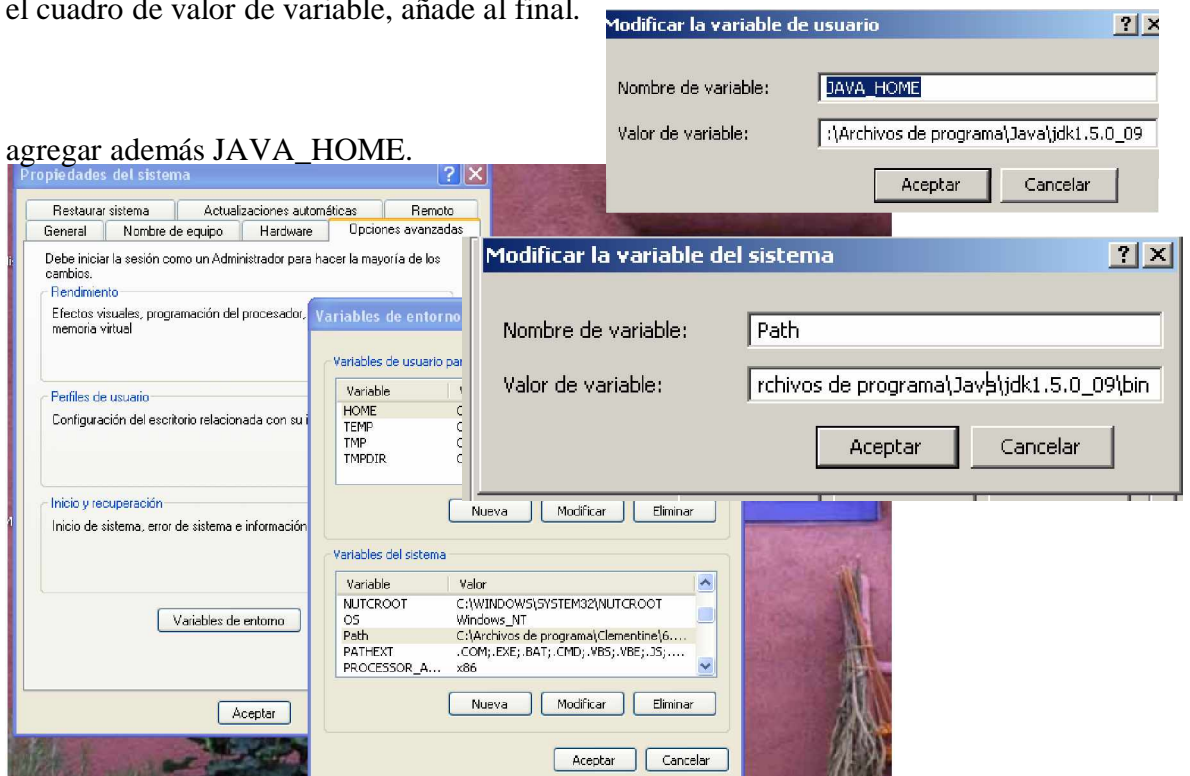
Seleccionamos la ubicación del J2SE y presionamos INSTALAR



Así concluimos con la instalación del Tomcat a continuación agregamos las variables de entorno para el correcto funcionamiento de la aplicación.

1. Pulsar con el botón derecho del ratón sobre "Mi PC" y selecciona "Propiedades".
2. Selecciona la pestaña "Opciones avanzadas" y pulsa en el botón inferior "Variables de entorno".
3. Tanto en el cuadro superior como el inferior, busca una variable que se llama "Path", haz click sobre ella y pulsa el botón "Modificar" (si aparece en los dos cuadros, haz lo siguiente dos veces).
4. En el cuadro de valor de variable, añade al final (cuidado, SIN BORRAR NADA).
5. Buscar otra vez en la lista de variables si existe "CLASSPATH". Si no existe, no pasa nada, sigue con el siguiente punto. Si existe, click sobre ella y pulsa el botón "Modificar" (si aparece en los dos cuadros, haz lo siguiente dos veces). En el cuadro de valor de variable, añade al final.

6. agregar además JAVA_HOME.



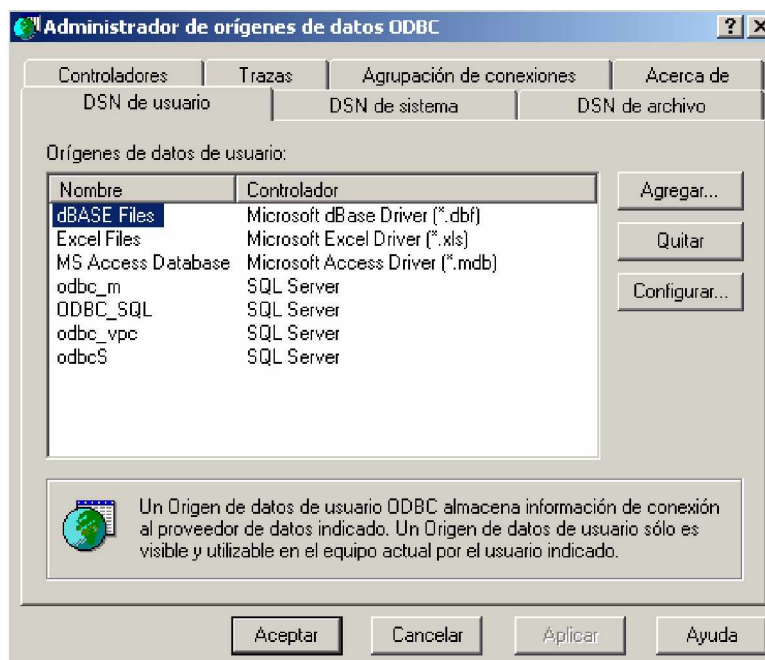
De esta manera se encuentra configurada las variables de entorno.

Ø **Configuración para el acceso a la base de datos :**

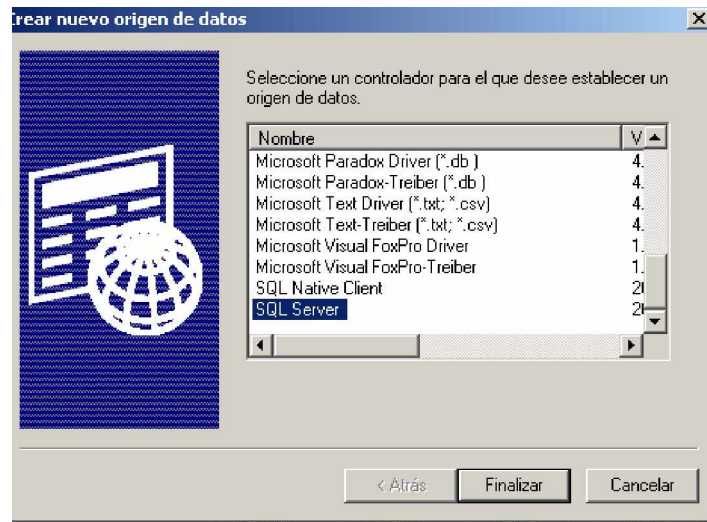
En la Base de Datos no se requiere ninguna configuración adicional. Lo que se debe tomar en cuenta es que en el Servidor SQL Server debe estar iniciado.

Para el acceso ala base de datos debemos crear un ODBC de la siguiente manera:

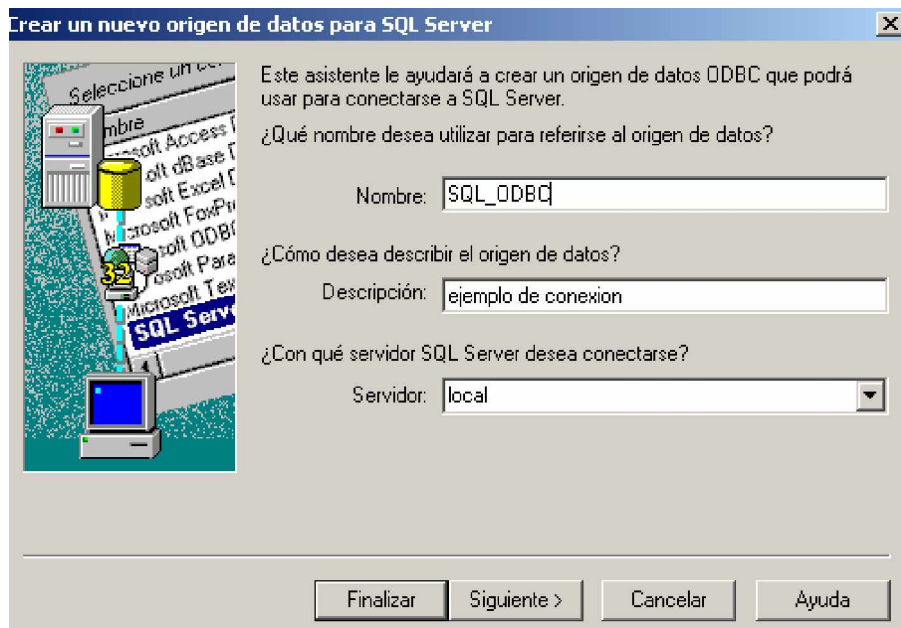
1. Creamos un Origen de datos ,pulsamos agregar:



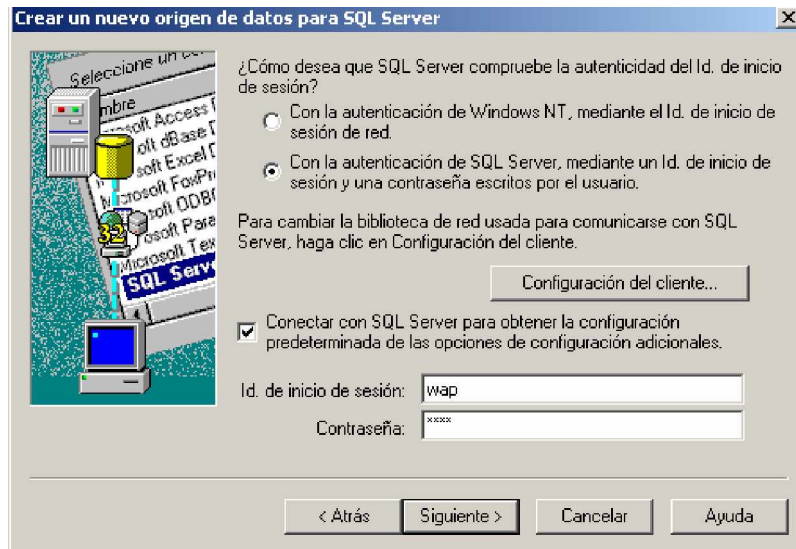
2. Seleccionamos el controlador para esta caso SQL Server y pulsamos Finalizar



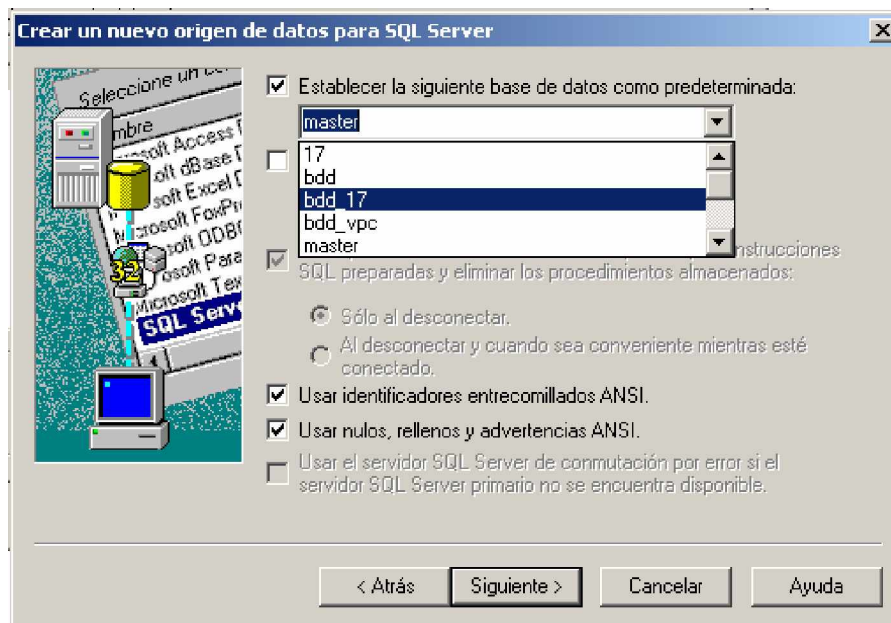
A continuación configuraremos el origen de datos: En Nombre escribiremos el nombre del origen de datos, en descripción pondremos una descripción sobre el odbc y en servidor seleccionaremos el servidor SQL al que nos vamos a conectar



A continuación escogemos la autenticación de inicio de sesión para ese caso autenticación SQL ,y escribimos el nombre del usuario y la contraseña del usuario para iniciar la sesión SQL



A continuación escogemos la base de datos a la que vamos a conectar.



Finalizamos y aceptamos así hemos creado el origen de datos.

10.2.5.Seguridades

Ø Acceso a la base de Datos

Para acceder a la base de datos, creamos un usuario y una contraseña, lo cual es transparente al usuario final, el mismo que tienen solo permisos de consultas e inserciones solo en la tabla TBL_CO_WAP.

Ø Mayor seguridad

Para tener una mayor seguridad en nuestro servidor de aplicación instalamos el antivirus para proteger de virus y troyanos.

Además de activar Firewall para proteger el ingreso de intrusos

10.2.6.Manual de usuario

El sistema móvil para la EMAPAI , puede ser ejecutado desde un terminal inalámbrico (teléfono móvil, Palm, Pda, etc), habilitado para usar el protocolo WAP, que debe contener un micro navegador o WAP Browser que sirve de interfaz de usuario.

La presentación del sistema lo hemos realizado en un emulador Openwave.

Para ejecutar la aplicación seguimos los siguientes pasos:

2. Conectamos la dirección en donde se encuentra nuestra página de inicio y **OK**.

Enter address:
<http://emapai.ec.nu>



1. Si escogemos la opción **INFORMACION. GENERAL** ingresamos a una página en donde tenemos la información general de la empresa.
2. Si escogemos la opción **CONSULTA VALORES:** ingresamos a una página en donde ponemos escoger un patrón de búsqueda para ver la información que nos proporciona EMAPAI de determinado patrón de búsqueda.

3. Si escogemos la opción **DATOS DE CUENTA:** ingresamos a una página en donde ingresamos la cuenta específica y observamos información general de esa cuenta.

4. Si escogemos la opción **OBSERVACIONES:** ingresamos a una página en donde ingresamos la cuenta ó la cedula y podemos insertar alguna duda, sugerencia, queja, reclamo o emergencia.

3. Seleccionamos el link Información General y pulsamos sobre **SELECT**



Aquí visualizamos información general sobre EMAPAI al pulsar sobre el link regresar, regresamos al menú principal de la aplicación.

4. Seleccionamos el link Consulta Valores y pulsamos sobre **SELECT**



En esta pantalla podemos escoger un patrón de búsqueda sea esta Número de Cuenta o número de Cédula para poder ver la información, ingresamos el datos escogido y pulsamos sobre el botón consultar. (En caso de haber escogido la Cuenta se desplegara la siguiente pantalla):



En esta pantalla mostramos datos de la cuenta tales como:

- ω **Cuenta:** Muestra el número de cuenta seleccionado.

- ω **CI/RUC:** Muestra la Cédula o RUC del dueño de la cuenta.

- ω **Nombre:** Muestra el nombre del cliente dueño de la cuenta.

- ω **Consumo:** Muestra en consumo en m³ del cliente en ese mes.

- ω **Total a Pagar:** Muestra la cantidad que tiene que pagar por consumo de agua Potable.

- ω **Pague Hasta:** Muestra la fecha límite de pago.

- ω **Factura Emitida:** Muestra la fecha de emisión de la factura.

- ω A demás tenemos un link de ver histórico al pulsar **SELECT** sobre esta opción representa una información de los últimos tres meses detallando mes, consumo y total pagado



En esta pantalla encontramos un link en el que podemos observar un grafico estadística del historial de los últimos tres meses.



(En caso de haber escogido CI/RUC se despliega la siguiente pantalla)



Esta pantalla visualiza las cuentas que tiene determinada cedula ingresando a cada una de las cuentas podremos ver la información de la cuenta.

Si regresamos al menú principal de la aplicación encontramos la opción DATOS DE CUENTA si pulsamos SELECT sobre esta opción se muestra una pantalla similar a esta:



Aquí podemos ingresar la cuenta y pulsamos sobre el botón Ver y se nos muestra la siguiente pantalla:



- ω **Dirección:** Muestra la dirección donde se encuentra ubicada esta cuenta.

- ω **Cliente:** Muestra el nombre del dueño de la cuenta

- ω **Medidor:** Muestra el número de medidor asignado a esa cuenta.

Si regresamos al menú principal de la aplicación encontramos la opción OBSERVACIONES si pulsamos SELECT sobre esta opción se muestra una pantalla similar a esta:



Al ingresar a esta pantalla esta nos permite insertar la observación, queja, duda, sugerencia o reclamo ingresando sea la cuenta o la cédula correcta del usuario.



10.3. PROTOTIPO VPC STAND-ALONE (VENTAS POR CELULAR)

10.3.1.DESCRIPCIÓN DE LA PROPUESTA

Esta aplicación móvil esta diseñada para ejecutarse en un celular que tenga en su sistema una KVM (Kilo Virtual Machine), posee una conexión remota a un servidor Web a través del protocolo http solo para determinadas opciones pues otras pueden funcionar desconectadas, además cuenta con una interfaz de desarrollo bastante amigable y sencilla de navegar.

Las peticiones enviadas son procesadas por servlets del lado del servidor web y estos a su vez se conectan con la base de datos procesando las consultas y actualizaciones necesarias.

Esta aplicación fue realizada en tecnología J2ME, siguiendo los estándares y patrones de diseño, convirtiéndose esta, en una aplicación estandarizada.

El SGBD usado es SQL Server 2000.

El sistema VPC tiene como objetivo principal, vender entradas de eventos deportivos de una manera virtual. El cliente podrá escoger el partido o evento para observar la descripción o escoger el tipo de entrada disponible para dicho evento.

Este aplicativo fue diseñado para cumplir los siguientes requerimientos:

- ω Mostrará los eventos deportivos disponibles.

- ω Mostrará los tipos de entradas disponibles del evento seleccionado (General, Tribuna, o Palco).

- ω Mostrará información descriptiva del evento.

- ω Validará los datos de tarjeta de crédito del cliente contra una tabla que simulará a OPTAR.

- ω Realizará las actualizaciones correspondientes de acuerdo a la transacción realizada.

- ω Se deberá considerar un límite del valor de compra que será establecido para cada empresa.

- ω Mostrará la ayuda de cada pantalla de la aplicación.

- ω Deberá mostrar los tipos de entrada mientras exista un número disponible de entradas en la base de datos.

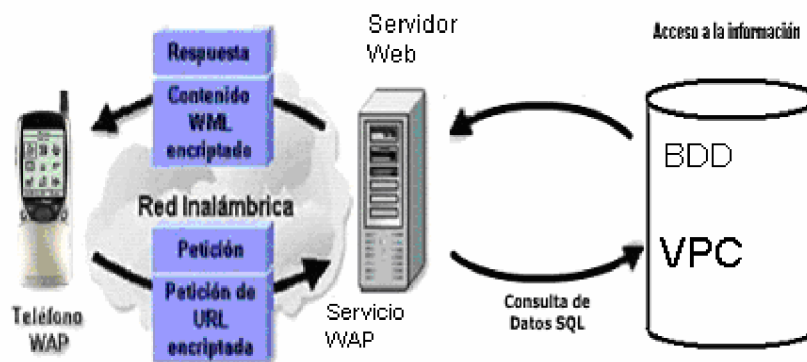
10.3.2. Análisis previos

10.3.2.1. Metodología del desarrollo

Esta aplicación tiene como objetivo mostrar las ventajas de las aplicaciones stand-alone en que pueden trabajar tanto conectadas como desconectadas y los tiempos de respuesta son mucho mejores pues el servidor web solo se encarga de enviar en forma de streams (flujos) los resultados y no de cómo se va a presentar la información.

El sistema es un prototipo modelo de un sistema para realizar ventas de entradas a eventos deportivos específicamente a Partidos de Fútbol. Para la realización del pago se ingresa un número de tarjeta de crédito y un pin para ser validados contra OPTAR (operadora de tarjetas de crédito).

10.3.2.2. Flujo de la información



Flujo de información en el sistema móvil VPC

Las transacciones en este modelo de aplicaciones se realizan de un modo idéntico al de una aplicación web.

- ω El cliente instalado en el celular hace una petición por el protocolo http.

- ω El Servidor web procesa esa petición

- ω Y aquí viene la diferencia en vez de enviar paginas web con encabezados html escribe encabezados de texto plano con algún criterio de discriminación del contenido de respuesta.

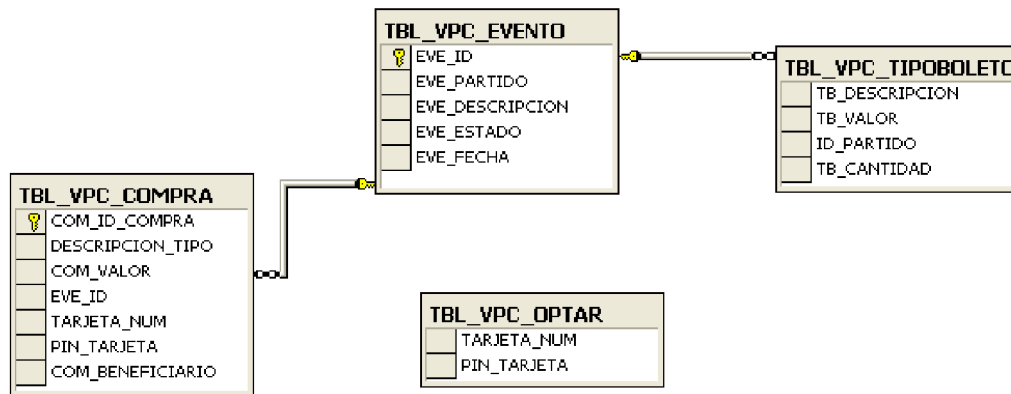
- ω El cliente recibe como un flujo de entrada la respuesta que el servidor escribe y dibuja la interfaz y presenta los datos según esa respuesta.

A diferencia con las aplicaciones WAP en la que es necesario un gateway como traductor de encabezados.

10.3.2.3. Diseño de tablas

Modelo de entidad relación

Para la implantación de nuestro aplicativo accedimos a las siguientes tablas de la base de datos bdd_vpc.



10.3.2.4. Funcionamiento del aplicativo VPC.

Una transacción en el sistema VPC inicia cuando el cliente llena el formulario de compra con los principales datos de su tarjeta de crédito como son número, pin de seguridad y número de CI del beneficiario.

Es ingresado en el formulario un número de CI para que se indique la persona beneficiaria que podrá ir a retirar la compra; previa verificación de su CI.

En una aplicación real de este tipo se deberá realizar un convenio con OPTAR (Operadora de Tarjetas de crédito) para realizar la validación de la cuenta y disponibilidad en las diferentes entidades de crédito.

El sistema tiene el control hasta que el cliente llena el formulario de compra donde pasa el control a OPTAR y este devuelve un OK ó ERROR con las respectivas indicaciones de resultado en ese momento el control regresa al sistema que dará al usuario final las respectivas informaciones.

En cuestiones de seguridad se aconseja usar métodos de encriptación al pasar los datos de la tarjeta de crédito a OPTAR para evitar phishing y/o cualquier intento de ataque en este punto que seria el objetivo de ataque más crítico.

10.3.3.Desarrollo

Dentro del código desarrollado para la aplicación indicamos a continuación las páginas principales, para mostrar información en los celulares usamos midlets archivos con extensión java y que extienden de MIDlet e implementan CommandListener y para la conexión y acceso a la base de datos se la realizo en servlets archivos con extensión java, la descripción con detalle lo encontramos en el manual técnico del aplicativo,
ANEXO VI.

10.3.4.Implantación

Para implantar este aplicativo se tomo en cuenta que la aplicación fue desarrollada una parte en u servidor web accediendo mediante los servlets a las base de datos.

Para esto necesitamos instalar los siguientes requerimientos.

∅ Requerimientos:

ω Sistema Operativo:

Windows XP Profesional o Windows NT 2003

ω Base de Datos:

SQL Server Standard 2000

ω Servidor WEB:

Tomcat 5.5

ω Lenguaje de Programación:

JAVA (Servlets)

∅ Configuración de Tomcat :

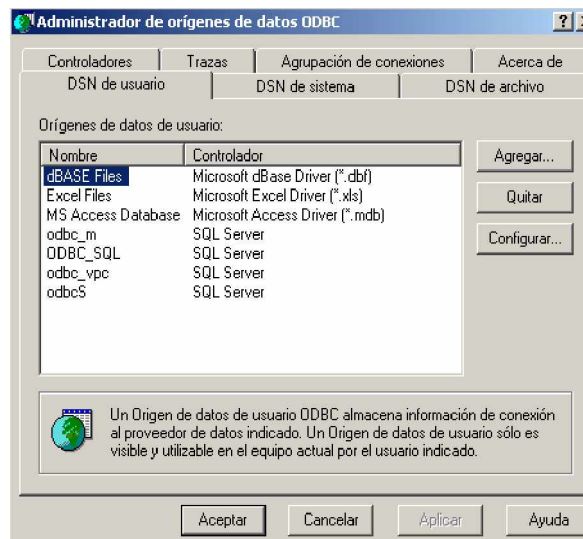
La configuración e instalación del servidor Tomcat se vio en punto 10.1 de este capítulo.

Ø **Configuración para el acceso a la base de datos :**

En la Base de Datos no se requiere ninguna configuración adicional. Lo que se debe tomar en cuenta es que en el Servidor SQL Server debe estar iniciado.

Para el acceso a la base de datos debemos crear un ODBC de la siguiente manera:

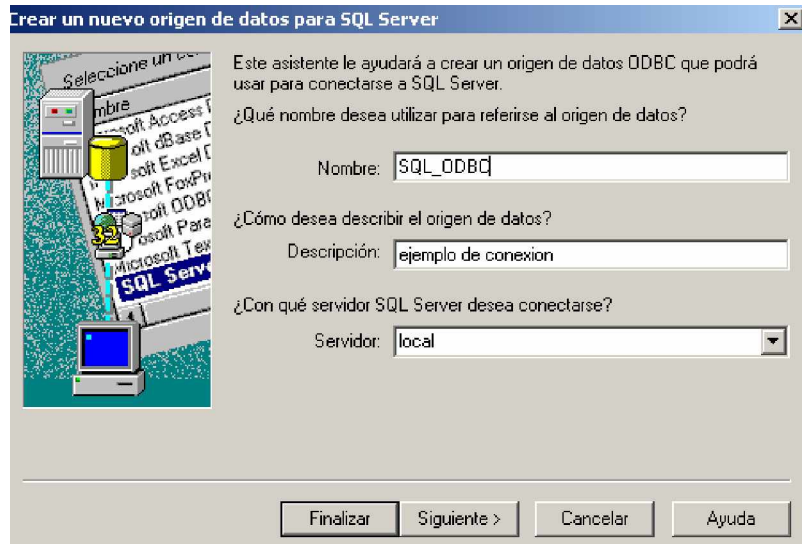
1. Creamos un Origen de datos ,pulsamos agregar:



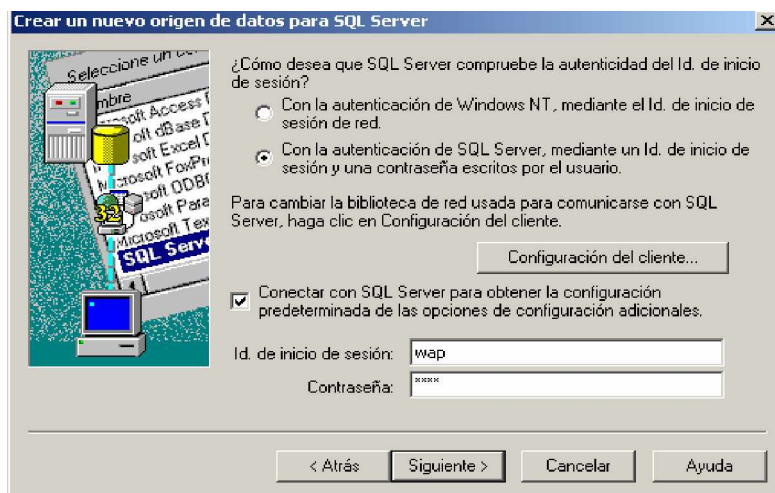
2. Seleccionamos el controlador para esta caso SQL Server y pulsamos Finalizar



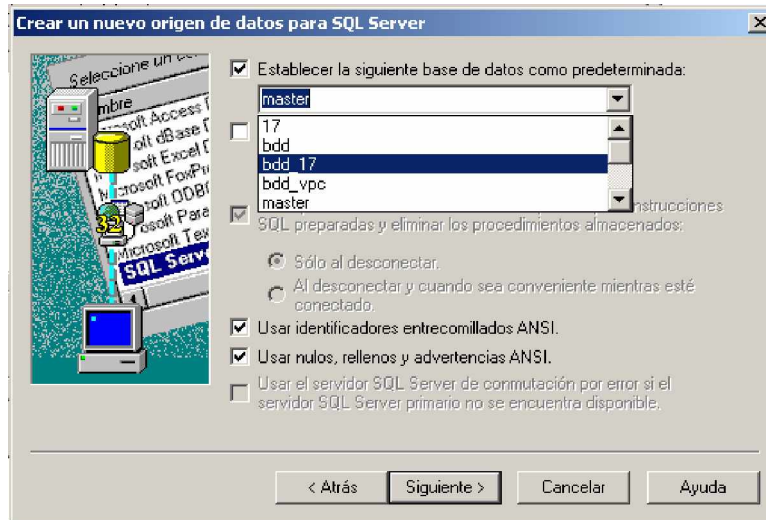
A continuación configuraremos el origen de datos: En Nombre escribiremos el nombre del origen de datos, en descripción pondremos una descripción sobre el odbc y en servidor seleccionaremos el servidor SQL al que nos vamos a conectar



A continuación escogemos la autenticación de inicio de sesión para ese caso autenticación SQL y escribimos el nombre del usuario y la contraseña del usuario para iniciar la sesión SQL



A continuación escogemos la base de datos a la que vamos a conectar.



Finalizamos y aceptamos así hemos creado el origen de datos.

10.3.5.Manual de usuario

El sistema móvil VPC , puede ser ejecutado desde un terminal inalámbrico (teléfono móvil, Palm, Pda, etc), que tenga instalada la maquina virtual que viene por defecto y un 90% de marcas de celulares.

La presentación del sistema lo hemos realizado en un emulador Sun Wirless Toolkit.

Para ejecutar la aplicación seguimos los siguientes pasos:

- ω Iniciamos la aplicación dando ok sobre el archivo VPC.jar.



1. Si escogemos la opción *inicio* ingresamos a una página en donde tenemos listados los partidos a realizarse en la fecha siguiente.
2. Si escogemos la opción *ayuda* encontramos una lista de temas tipos de posibles problemas que se le puede presentar al usuario.
3. Si escogemos la opción *AcercaDe* encontramos información sobre los autores y versión del producto.

- ω Seleccionamos la opción inicio.

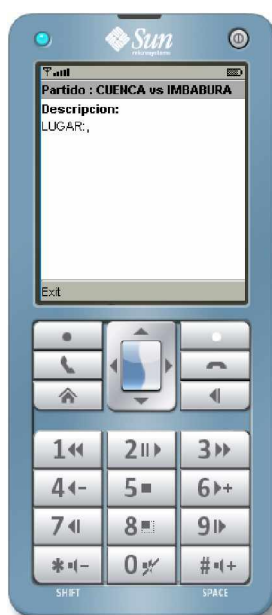


Aquí visualizamos la lista de encuentros disponibles.

Seleccionamos un partido y escogemos menú.

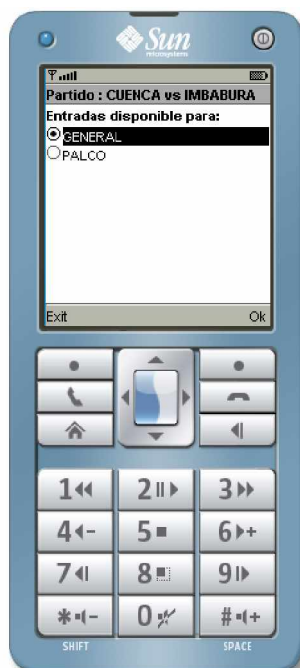


En esta pantalla podemos escoger la opción del menú que deseemos si escogemos Ver este mostrara una pantalla que escribe una breve descripción del partido seleccionado.



En caso de haber escogido OK en la pantalla donde se listan los partidos de fútbol de la temporada.

Esta pantalla permite visualizar los tipos de entradas disponibles para el partido seleccionado.



Los posibles tipos de entrada son:

- ω General
- ω Palco
- ω Tribuna

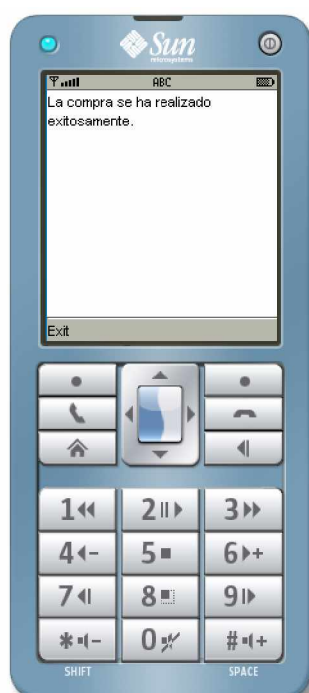
Si se han agotado las entradas en uno de estos tipos de entrada no se mostrara.

Al escoger un tipo pasamos a un formulario para llenar información sobre la compra como numero de tarjeta de crédito, pin de seguridad y CI del beneficiario que va a retirar las entradas.



En esta pantalla al terminar de llenar los datos correspondientes pulsamos en ok iniciando transacción de registro de la compra. Los datos de la tarjeta de crédito serán validados en un registro de usuarios de OPTAR.

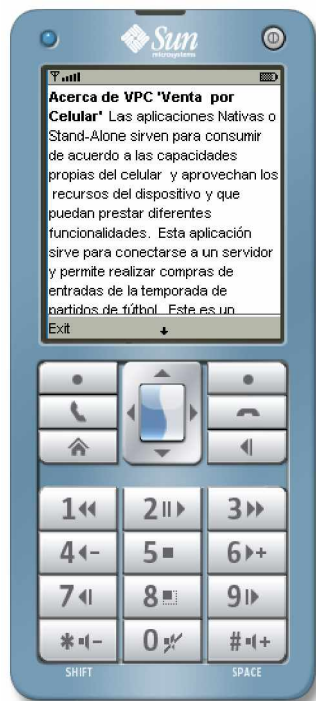
Si la compra se realiza correctamente se muestra la pantalla de éxito.



2. Seleccionamos la opción Ayuda.



3. Seleccionamos la opción AcercaDe donde obtenemos la siguiente pantalla.



Si en cualquier pantalla pulsamos exit regresaremos al menú de inicio.

CAPÍTULO XI

CONCLUSIONES Y RECOMENDACIONES

11. VERIFICACIÓN DE LA HIPÓTESIS



11.1. HIPÓTESIS PLANTEADA

Variables Independientes

Si las aplicaciones Wap para celulares:

- ω Implementaran estándares seguros,
- ω Brindaran un fácil acceso a la información, y
- ω Tendrían una rápida y amigable interfaz;

Variables Dependientes

Se lograra crear una poderosa herramienta (Aplicación) a la que el usuario podría acceder con confianza a realizar transacciones de comercio, de negocios y/o de servicios, con la ventaja que lo haría desde cualquier lugar y en cualquier momento, utilizando la tecnología celular existente en nuestro medio.

11.2. VERIFICACIÓN

La hipótesis queda comprobada ya que con el estudio de la capa WTLS se determinó que los niveles de seguridad al usar métodos de encriptación y compresión en la transmisión de datos se obtiene aplicaciones con estándares seguros. Además con la asignación de roles con funciones específicas limitamos cualquier acceso indebido a la base de datos.

Como se ha determinado en las encuestas del 100% de la muestra el 98% tiene celular que es la principal herramienta para acceder a las aplicaciones WAP con lo que se asegura brindar un fácil acceso a la información en cualquier lugar y en cualquier momento.

Además con el uso de los estándares de diseño y la aplicación de las mejores prácticas descritas por la W3C para el desarrollo e implementación de aplicaciones web-móviles se ha logrado tener una interfaz amigable para el usuario.

Comprobando que la implantación de sistemas para dispositivos celulares es de gran beneficio tanto para la empresa como para los clientes. La innovación en el uso de la tecnología móvil no ha sido tomada en cuenta como una alternativa rápida, eficiente y eficaz de entregar información desde cualquier punto geográfico, y demostramos que está es una muy buena alternativa, por lo estudiado en los capítulos anteriores.

Finalmente la favorable acogida que tuvo la implantación del sistema EMAPAIWAP en la Empresa Municipal de Agua Potable y Alcantarillado de Ibarra tanto por la empresa como por los clientes respaldan los puntos anteriores.

En base a este razonamiento lógico, se logró demostrar la hipótesis.

11.3. CONCLUSIONES

Estudio Teórico

- WAP proporciona una solución inteligente y específica al problema del acceso a Internet desde un terminal móvil.

- Con la utilización de la tecnología WAP, se proporciona la posibilidad de establecer una conexión segura entre un dispositivo celular y un servidor web.

- Wap cuenta con un entorno de desarrollo inalámbrico (WAE) que permite crear servicios y aplicaciones avanzadas.

- Wap cuenta con un protocolo de transacciones seguras (WTLS) que provee seguridad en la transmisión de datos.

- El estándar Wap 2.0 usa XHTML-MP como principal recomendación para el diseño de aplicaciones por su similitud con HTML llevando a una unificación de tecnologías.
- En el estudio de aplicaciones móviles encontramos aplicaciones on-line que representan las web-móviles y las aplicaciones stand alone que representan las instalables en el cliente.
- La tendencia para desarrollo de aplicaciones web-móviles es el uso de XHTML MP al lado del cliente y servlets al lado del servidor, o aplicaciones desarrolladas con .Net
- La tendencia para desarrollo de aplicaciones stand alone es el uso de los midlets de J2ME.

Aplicativo

- El sistema móvil WAP-COMMERCE es una aplicación de tipo Web-Móvil, principalmente está destinado a simular la realización de un carrito de compras de diferentes categorías (Mercados, Ferreterías, Farmacias y Hoteles), simplemente con un terminal móvil.

- Nuestro aplicativo Wap-Commerce utiliza adaptadores de contenido para según las características del dispositivo crear la interfaz dinámicamente.
- Los aplicativos desarrollados en esta tesis: WAP-COMMERCE, VPC y EMAPAIWAP tienen una interfaz amigable para el usuario.
- Nuestra aplicación implantada en la EMAPAI, permite obtener información en tiempo real, es decir, online. Al igual que nuestros prototipos Wap-Commerce y VPC simulan obtención de información en tiempo real.
- El sistema móvil de la EMAPAI es una aplicación de tipo Web-Móvil, principalmente está destinado a la consulta de información de carácter específico
 - § Valores a pagar por Cuenta o por su Cédula
 - § Fecha de emisión de la factura
 - § Fecha limite de pago
 - § Meses de deuda
 - § Estado de la cuenta
 - § Historial de consumo de los 3 últimos meses
 - § Gráfico estadístico
 - § Ingreso de observaciones, daños, dudas o emergencias., simplemente con un terminal móvil.
- El sistema móvil VPC (Ventas por Celular) es una paliación de tipo Stand-Alone, principalmente está destinado a simular la realización de

una compra de Entradas de Fútbol de la temporada, de carácter específico

- § Visualiza los partidos de cada temporada
- § Visualiza los tipos de entradas disponibles para cada partido
- § Realiza la compra ingresando los datos correctos, simplemente con un terminal móvil.

11.4. RECOMENDACIONES

- En la actualidad, la tecnología móvil está convirtiéndose en un elemento más en la vida diaria, podemos recomendar que dichas tecnologías deben incorporarse en las estrategias de la gestión administrativa, comercial, y/o servicios.
- Se puede aprovechar todas las ventajas del entorno inalámbrico, y diseñar e implementar aplicaciones: de negocios, comerciales, didácticas, etc. que aprovechen las posibilidades de este nuevo medio.
- Se debería aprovechar la tendencia de baja en los costos de transmisión, para incorporarse a los beneficios que brinda la tecnología inalámbrica junto con la tecnología móvil.

- Se debería seguir el ejemplo de EMAPAI en abrir las puertas a nuevas tecnologías que pueden brindar servicios útiles a la población, en las empresas de nuestro medio.

- Debería difundirse publicidad sobre el uso de las aplicaciones WAP.

BIBLIOGRAFÍA

Libros:

- ω Klander Larsv **A prueba de Hacker** Editorial ANAYAMULTIMEDIA, año 1998, España.

- ω Jayson Falkner, Ben Galbraith, Romin Irani, Casey Kochmer, Meeraj Moidoo, Sathya Narayana, Krishnaraj Perrumal y Jonh Timney **Fundamentos Desarrollo Web con JSP** de Editorial ANAYAMULTIMEDIA, año 2002, Madrid-España.

- ω Editorial ANAYAMULTIMEDIA **Desarrollo Web con ASP.Net**, año 2003, Madrid-España.

- ω Payne **Aprendiendo ASP.Net en 21 días** de Payne Edición 2005.

- ω Sergio Calvez Rojas **J2ME** , Universidad de Málaga, año 2003.

- ω Mohammad Akif, Steven Bordead, Andrei Cioroianu, James Hart, Eric Jung, Dave Writz **Java y XML** de, Editorial ANAYAMULTIMEDIA, año 2002, Madrid-España.

- ω Mohammad Akif, Steven Bordead,
Andrei Cioroianu, James Hart,
Eric Jung, Dave Writz

Fundamentos XML de, Editorial
ANAYAMULTIMEDIA, año 2002, Madrid-
España.

- ω Alexandra Jácome

Tesis para la obtención del Título de Ingeniería
en Sistemas Computacionales de Alexandra
Jácome con el tema; Sistema Móvil para la
gestión administrativa para el Instituto
Superior Alberto Enríquez G.

Libros virtuales:

- ω Editorial mpediciones.

Desarrollos móviles con .Net

- ω Mobile Top Level Domain

Mobile Web Developer's Guide
Published in 2007

- ω Dr. Angel Bravo M

**El Protocolo inalámbrico de aplicaciones
(WAP)**

ω Creativa Commons

Tutorial para aplicaciones móviles J2ME con NetBeans y Mobility Pack,

ω Agustín Froufe Quintas; Patricio Jorge Cárdenas

J2ME. Java 2 Micro Edition, Manual de usuario y Tutorial, (Editorial Rama)

ω Luis Enrique Corredera de Colsa

Arquitectura dirigida por modelos para J2ME para la obtención de Doctorando en Ingeniería del software Universidad Pontificia de Salamanca en Madrid

ω OMA

WAP Wireless Telephony Application
Version 08-Sep-2001 Wireless Application
Protocol WAP-266-WTA-20010908-a

ω Universidad de Granada del
Departamento de Ciencias de la
Computación

Instalación y primeros pasos en Java2
Micro Edition

ω Omar Salvador

Desarrollo de aplicaciones para dispositivos móviles utilizando J2ME de del Instituto Tecnológico de San Juan del Rio, Querétaro. –Universidad de Guadalajara

ω Francisco Martínez

Desarrollo de aplicaciones JAVA para móviles: J2ME y Herramientas de Desarrollo elaborado en junio 2006, del Departamento de Lenguajes y Sistemas Informativos Universidad Sevilla

Nuevas plataformas para terminales móviles Las Telecomunicaciones y la Movilidad en la Sociedad de la Información.

Internet:

- ω <http://avivancos.com/desarrollo-web-movil/>

Tutorial de aplicaciones
WebMóviles

- ω <http://www.ajax-en-el-movil.html>

Guía de Ajax

- ω <http://www.w3.org/2005/01/BPWCCharter/>

Mejores prácticas desarrollo móvil

- ω <http://www.w3c.es/Divulgacion/Guiasbreves/webmovil>

Guías de desarrollo

- ω www.anieto2k.com/2006/04/11/guia-breve-de-web-movil/

Manual de WebMóvil

- ω <http://leo.ugr.es/J2ME/REDES/index.html>

Manual de J2ME

ω <http://coepa.info/glosario>

Glosario Informáticos

ω <http://www.marcelopedra.com.ar/glosario.htm>

Glosario Informáticos

ω <http://www.developershome.com/>

Guia completa de desarrollo WAP

ω <http://www.openmobilealliance.org/index.html>

Referencia a los estándares WAP

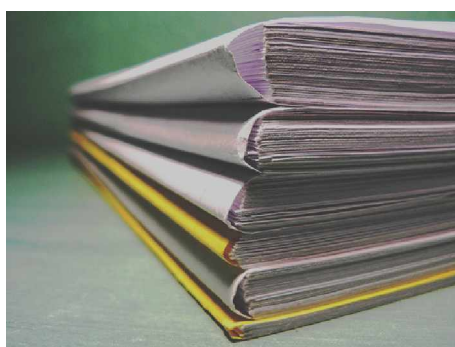
ω <http://www.w3.org/TR/2006/PR-mobile-bp-20061102/>

Mejores prácticas de desarrollo

ω www.wapforum.com/tech/affiliates/wap/wapindex.html

Referencia a protocolo WAP

ANEXOS



ANEXO I

GLOSARIO

ADSL: Asymmetric Digital Subscriber Line: Línea de Suscripción Asimétrica Digital. Tecnología que mejora el ancho de banda de los hilos del cableado telefónico convencional que transporta hasta 16 Mbps (megabits por segundo) gracias a una serie de métodos de compresión.

Ajax: Asynchronous JavaScript And XML. Es una técnica de desarrollo web para crear aplicaciones web interactivas. Así como DHTML, Ajax no es una tecnología en si, sino un termino que se refiere al uso de un grupo de tecnologías juntas, como son por ejemplo: XHTML (o HTML), Javascript, XMLHttpRequest entre otras. La idea es que las páginas sean más dinámicas cargando información del servidor de forma “escondida”, logrando que la página web no tenga que ser recargada cada vez que un usuario cambia información de un formulario, por ejemplo. De esta forma se logra que la página web incremente su interactividad, velocidad y usabilidad.

Apache: Servidor web de distribución libre. Fue desarrollado en 1995 y ha llegado a ser el más usado de Internet.

API: Acrónimo de Application Program Interface [Interfaz de programa de aplicación].

ASP: Acrónimo en inglés de Active Server Pages. Páginas de Servidor Activo. Son un tipo de HTML que además de contener los códigos y etiquetas tradicionales, cuenta con programas (o scripts) que se ejecutan en un servidor Microsoft Internet Information Server antes que se desplieguen en la pantalla del usuario. Por lo general este tipo de programas realizan consultas a bases de datos, siendo los resultados de éstas los que el usuario final obtiene. La extensión de estos archivos es ".asp."

ASP.NET: Es una tecnología Web del lado del servidor, desarrollada por Microsoft. ASP.NET toma el enfoque de la programación orientada a objetos para la ejecución de páginas Web. Cada elemento en una página en ASP.NET se trata como un objeto y corre sobre el servidor. Una página ASP.NET se compila en un lenguaje intermedio con un compilador CLR (Common Language Runtime) de .NET. Luego, un compilador JIT convierte el código intermedio a código de máquina, y eventualmente ese código se corre sobre el procesador. Dado que el código corre directamente sobre el procesador, las páginas se cargan mucho más rápido que las clásicas en ASP, en las que el VBScript o JScript embebido tiene que ser continuamente interpretado y cacheado. El ASP.NET se usa para crear páginas Web y servicios Web, y es parte integral de la nueva visión .NET de Microsoft.

Bit: Abreviatura de BInary digiT (dígito binario). El bit es la unidad más pequeña de almacenamiento en un sistema binario dentro de una computadora. Equivalente a un "sí" (0) o un "no" (1) binarios. La unión de 8 bits da lugar a un byte.

Bps (bits por segundo): Unidades de medida de la capacidad de transmisión de las redes de telecomunicaciones. Un bit sería cada paquete de información de unidades binarias.

Byte: Agrupación de 8 bits consecutivos que el ordenador maneja simultáneamente. Puesto que cada 'bit' puede representar dos estados (0 y 1), un byte puede representar 256 (2 elevado a 8) caracteres diferentes.

Criptografía: Ciencia dedicada al estudio de técnicas capaces de conferir seguridad a los datos. La encriptación es fundamental a la hora de enviar datos a través de las redes de telecomunicaciones con el fin de conservar su privacidad.

CSD: Circuit Switched Data (Datos de circuitos conmutados) = datos GSM/llamada de datos normal. En las redes convencionales GSM, las velocidades de transmisión de datos podían alcanzar un máximo de 9,6 kbps. Esta velocidad era adecuada en el pasado, cuando los teléfonos móviles se utilizaban principalmente para llamadas de voz. Sin embargo la estructura de la red GSM permite utilizar las frecuencias de radio, que se dividen en intervalos de tiempo, de modo más eficaz.

D-AMPS (Digital Advanced Mobile Phone Service): Servicio de teléfono celular, que usa la tecnología TDMA.

DCD : Acrónimo de Data Carrier Detect [Detección de portadora de datos] La señal de portadora es una señal básica que permite comprobar si la comunicación puede establecerse y posteriormente si los módems están conectados entre sí.

DECT:(Digital Enhanced Cordless Telecommunications) Tecnología digital inalámbrica originada en Europa, pero que actualmente se usa en todo el mundo. DECT

es una tecnología de radio para aplicaciones de voz. DECT utiliza TDMA (time division multiple access) para transmitir señales de radio a teléfonos y funciona mejor en pequeñas áreas con un gran número de usuarios. Ha sido diseñado y especificado para trabajar con muchos otros tipos de redes como PSTN, ISDN, GSM y otras.

DHTML: Dynamic HTML. HTML dinámico. Una extensión de HTML que permite, entre otras cosas, de la inclusión de pequeñas animaciones y menús dinámicos en páginas web. El código DHTML usa estilos y JavaScript. Permite que las páginas web puedan “reaccionar” dependiendo del usuario y su sistema, por ejemplo una página web que cambie su estilo dependiendo del navegador o incluso tipo de computadora de los usuarios finales.

DRAE: Diccionario de la Real Academia de la Lengua Española.

ETSI: European Telecommunications Standards Institute.

FDD: Frequency Division Duplex: canales separados en frecuencia. La división de frecuencia duplex y subcintas de downlink, como se dice, es separada por " la compensación de frecuencia ". La división de frecuencia duplex es mucho más eficiente en el caso de tráfico simétrico. En este caso TDD tiende a gastar la amplitud de banda durante el intercambio de transmisión para recibir, tiene la mayor latencia inherente, y puede requerir el trazado de circuito más complejo, más hambriente de poder.

FDMA: (*Frequency Division Multiple Access*) es una técnica de acceso múltiple en la que el recurso compartido es el ancho de banda.

Firewall: Combinación de hardware y software la cual separa una red de área local (LAN) en dos o mas partes con propósitos de seguridad. Su objetivo básico es asegurar que todas las comunicaciones entre dicha red e Internet se realicen conforme a las políticas de seguridad de la organización que lo instala. Además, estos sistemas suelen incorporar elementos de privacidad, autenticación, etc.

Firma Digital: Información cifrada que identifica al autor de un documento electrónico y autentifica su identidad.

GPL: La licencia GPL (General Public Licence)

GPRS : General Packet Radio Service (Servicio general de paquetes vía radio). Con GPRS, podrá estar en línea continuamente. Puede estar continuamente conectado a las redes de datos, y utilizar los servicios de información y ocio que desee. Además, GPRS permite una conectividad inmediata y mayor flujo de transferencia de datos que antes, de modo que la recuperación de información resulta más accesible y cómoda. GPRS se utiliza principalmente como portador para servicios WAP. En la tecnología GPRS, los datos se envían en "paquetes" lo cual se adecua perfectamente a la transmisión por ráfagas propias de las aplicaciones de datos. GPRS se puede utilizar también como portador para aplicaciones de PC de forma similar a cómo se utilizan CSD y HSCSD.

GSM (Global System for Mobile communication): Sistema compatible de telefonía móvil digital desarrollado en Europa con la colaboración de operadores, administraciones públicas y empresas. Permite la transmisión de voz y datos.

HDSL : High bit-rate Digital Subscriber Line. Línea Digital de Subscriptor de Alta Velocidad. Sistema de transmisión de datos de alta velocidad que utiliza dos pares de cables trenzados. Es simétrico, lo que quiere decir que tiene el mismo ancho de banda de subida que de bajada.

HTML: Lenguaje de Marcado de Hipertexto (HyperText Markup Language). Un lenguaje utilizado para crear páginas web.

IIS: Internet Information Server. Es el servidor Web de Microsoft que corre sobre plataformas Windows. De hecho, el IIS viene integrado con Windows NT 4.0. Dado que el IIS está tan íntimamente integrado con el sistema operativo, es relativamente fácil de administrar. Sin embargo, actualmente el IIS está disponible sólo para el uso en la plataforma Windows NT, mientras que los servidores Web de Netscape corren en la mayoría de las plataformas, incluyendo Windows NT, OS/2 y UNIX.

ISDN: Red Digital de Servicios Integrados o RDSI (Integrated Services Digital Network). Red digital de servicios integrados – es un tipo de sistema de telefonía en red de circuitos conmutados diseñados para permitir la transmisión digital (en contraposición a analógica) de voz y datos sobre los cables telefónicos de cobre comunes, lo que implica una mejor calidad y mayor velocidad que la disponible con los sistemas analógicos.

ISO: International Organization for Standardization. Fundada en 1946, es una federación internacional que unifica normas en unos cien países. Una de ellas es la norma OSI, modelo de referencia universal para protocolos de comunicación.

ISP: Internet Service Provider: Proveedor de acceso a Internet: Compañía que ofrece a los usuarios de Internet la posibilidad de conectarse a la red. Estas compañías de telecomunicaciones cuentan con nodos que facilitan la conexión del usuario a la red a través de la red de telecomunicaciones que prefiera: telefonía fija, cable, telefonía móvil, etc.

J2EE:Java 2 Platform Enterprise Edition. Es un ambiente de Sun enfocado en Java, independiente de la plataforma, para el desarrollo, construcción e implementación de aplicaciones online para empresas basadas en Web. La plataforma J2EE consiste de un conjunto de servicios, APIs, y protocolos que proveen la funcionalidad para desarrollar aplicaciones Web interrelacionadas.

J2ME : Java 2 Micro Edition. Versión Sun Microsystems de Java 2 destinada a dispositivos de recursos limitados como PDAs, teléfonos móviles, sistemas electrónicos para vehículos, hornos microondas de última generación con interfaz de interacción con el usuario y otros, requiriendo tan sólo un mínimo de 128 Kb de RAM. Así, esta plataforma Java está destinada a procesadores mucho menos potentes que los utilizados habitualmente como PCs.

JDBC:Java Database Connectivity. Es un API de Java que permite que programas en Java ejecuten comandos SQL, permitiendo esto que los programas puedan usar bases de datos en ese esquema. Dado que casi todos los sistemas de administración de bases de datos relacionales (DBMS) soportan SQL, y que Java corre en la mayoría de las plataformas, JDBC hace posible escribir una sólo aplicación de base de datos que pueda correr en diferentes plataformas e interactuar con distintos DBMS. JDBC es similar a

ODBC, pero está diseñado específicamente para programas Java, mientras que el ODBC es independiente del lenguaje. JDBC es un desarrollo de JavaSoft, una subsidiaria de Sun Microsystems.

MIDlet: Es una aplicación compatible con el estándar MIDP.

MIDP: Mobile Information Device Profile. MIDP es un conjunto de APIs para J2ME que definen cómo interactúan las aplicaciones de software con los teléfonos celulares y pagers de doble vía. Las aplicaciones compatibles con este standard se llaman MIDlets. Entre las compañías que trabajan con MIDP se incluyen Ericsson, NEC, Nokia, Palm Computing, Research In Motion (RIM), DoCoMo, LG TeleCom, Samsung y Motorota

Mbps : Megabits por Segundo. Unidad de medida de la capacidad de transmisión por una línea de telecomunicación donde cada megabit está formado por 1.048.576 bits.

ODBC: Open Database Connectivity. Estándar de acceso a Bases de Datos desarrollado por Microsoft cuyo objetivo es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS.

OTA : Over the Air. Se pueden enviar y recibir, por ejemplo, favoritos u opciones WAP Over-the-Air como Smart Messaging (con SMS como medio de transferencia)

PSTN: Red pública de telefonía conmutada - es la concentración de las redes públicas mundiales de circuitos conmutados, al igual que Internet es la concentración de redes públicas mundiales de paquetes conmutados basados en IP.

RDSI (Red Digital de Servicios Integrados) :Red digital sobre soporte físico de cable de cobre que permite alcanzar una mayor velocidad de transmisión de datos. Tecnología que combina servicios de voz y datos digitales a través de la red en un solo medio, haciendo posible ofrecer a los clientes servicios digitales de datos así como conexiones de voz a través de un sólo “cable”.

RFC :En inglés es Requests for Comments. Serie de documentos iniciada en 1967 la cual describe el conjunto de protocolos de Internet y experimentos similares. La serie de documentos RFC es inusual en cuanto los protocolos que describen son elaborados por la comunidad Internet que desarrolla e investiga, en contraste con los protocolos revisados y estandarizados formalmente que son promovidos por organizaciones como CCITT y ANSI.

RSA : Algoritmo criptográfico de clave pública y amplia utilización el cual está patentado por los autores que le dan nombre.

SDSL : Línea Digital Simétrica de Abonado. Sistema de transferencia de datos de alta velocidad en líneas telefónicas normales.

Servlet : Pequeña aplicación Java (applet) la cual se ejecuta en un servidor web y se envía al usuario junto a una página web con objeto de realizar determinadas funciones, tales como el acceso a bases de datos o la personalización de dicha páginas web.

SMS : Sistema de Mensajes Cortos, servicio de envío y recepción de mensajes escritos de pequeño tamaño a través del teclado y la pantalla de los celulares. Gracias a este servicio se puede enviar y recibir e-mails desde un teléfono GSM.

TDMA: (Time Division Multiple Access). TDMA es una tecnología inalámbrica de segunda generación, que distribuye las unidades de información en ranuras alternas de tiempo, dando acceso múltiple a un número reducido de frecuencias. TDMA permite dar servicios de alta calidad de voz y datos.

Tomcat: Es un servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

UMTS o 3GSM: Universal Mobile Telecommunications System :Sistema universal de telecomunicaciones móviles. Estándar de telefonía móvil celular de tercera generación en fase de desarrollo por **ETSI**. Se trata de un sistema de tercera generación más avanzado que el sistema **GSM** que operara a velocidades de transferencia de datos de 144 Kbps a 2 Mbps y proporcionara acceso a Internet multimedia.

URL: Un URL o Uniform Resource Locator (Localizador de recursos uniforme) es la dirección que se usa para acceder a una web.

WAP: Wireless Application Protocol. Norma internacional para aplicaciones que utilizan la comunicación inalámbrica, por ejemplo el acceso a Internet desde un teléfono celular.

WML : Wireless Markup Language. Lenguaje en el que se escriben las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDAs) dotados de tecnología WAP. Es una versión reducida del lenguaje HTML que facilita la conexión a Internet de dichos dispositivos.

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

WTLS :Wireless Transport Layer Security

XHTML :Siglas del ingles eXtensible HyperText Markup Language. XHTML es básicamente HTML expresado como XML valido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios, buscar errores, etc

XML : eXtensible Markup Language. Lenguaje Extensible de Marcado. Lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar

y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste.

ANEXO II

RFC Y ESTÁNDARES USADOS COMO BASE TEÓRICA

[ClassConform]

WAP Class Conformance Requirements,
WAP Forum.

[ECMAScript] Standard ECMA-262

"ECMAScript Language Specification",
ECMA, June 1997

[HTML4]

"HTML 4.0 Specification, W3C
Recommendation 18-December-1997,
REC-HTML40-971218", D.

[HTTPState]

"HTTP State Management", WAP-223-
HTTPSM, WAP Forum.

[JavaScript]

"JavaScript: The Definitive Guide", David Flanagan. O'Reilly & Associates, Inc. 1997

[MMSEncapsulation]

"WAP Multimedia Messaging Service Message Encapsulation", WAP Forum.

[ProvArch]

"WAP Provisioning Architecture Overview", WAP Forum.

[PushArchOverview]

"WAP Push Architectural Overview", WAP Forum.

[PushOTA]

"WAP Push OTA Protocol", WAP Forum.

[RFC2045]

"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed, et al., November 1996. URL: <http://www.rfc-editor.org/rfc/rfc2045.txt>

[RFC2048]

"Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", N. Freed, et al. November 1996. URL: <http://www.rfc-editor.org/rfc/rfc2048.txt>

[RFC2119]

"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. URL: <http://www.rfc-editor.org/rfc/rfc2119.txt>

[RFC2246]

“The TLS Protocol Version 1.0”, T.
Dierks, C. Allen, January 1999. URL:
<http://www.rfc-editor.org/rfc/rfc2246.txt>

[RFC2396]

“Uniform Resource Identifiers (URI):
Generic Syntax”, T. Berners-Lee, R.
Fielding, L. Masinter. August 1998. URL:
<http://www.rfc-editor.org/rfc/rfc2396.txt>

[RFC2401]

“Security Architecture for the Internet
Protocol”, S. Kent, R. Atkinson, November
1998. URL: [http://www.rfc-
editor.org/rfc/rfc2401.txt](http://www.rfc-editor.org/rfc/rfc2401.txt)

[RFC2616]

"Hypertext Transfer Protocol - HTTP/1.1",
R. Fielding, J. Gettys, J. Mogul, H. Frystyk,
L. Masinter, P. Leach, T. Berners-Lee,
June 1999. URL: [http://www.rfc-
editor.org/rfc/rfc2616.txt](http://www.rfc-editor.org/rfc/rfc2616.txt)

[RFC2617]

"HTTP Authentication: Basic and Digest
Access Authentication", J. Franks, P.
Hallam-Baker, J. Hostetler, S. Lawrence, P.
Leach, A. Luotonen, L. Stewart, June 1999.
URL: [http://www.rfc-
editor.org/rfc/rfc2617.txt](http://www.rfc-editor.org/rfc/rfc2617.txt)

[STD0006]

"User Datagram Protocol", J. Postel,
August 1980. URL: [http://www.rfc-
editor.org/rfc/std/std6.txt](http://www.rfc-editor.org/rfc/std/std6.txt)

[STD0007]

“Transmission Control Protocol”, J. Postel,
September 1981. URL: [http://www.rfc-
editor.org/rfc/std/std7.txt](http://www.rfc-editor.org/rfc/std/std7.txt)

[STD0013]

“Domain Name System”, P. Mockapetris,
November 1987. URL: [http://www.rfc-
editor.org/rfc/std/std13.txt](http://www.rfc-editor.org/rfc/std/std13.txt)

[TransportE2ESec]

“WAP Transport Layer End-to-End
Security Specification”, WAP Forum.

[UAProf]

“User Agent Profile Specification”, WAP
Forum.

[WAE]

"Wireless Application Environment Specification", Version 2, WAP-236-WAESpec, WAP Forum.

[WAPCert]

“WAP Certificate and CRL Profiles”, WAP Forum.

[WDP]

"Wireless Datagram Protocol Specification", WAP Forum.

[WIM]

“WAP Identity Module Specification”, WAP Forum.

[WML]

"Wireless Markup Language", WAP
Forum.

[WMLScriptCrypto]

"WMLScript Crypto Library", WAP
Forum.

[WPKI]

"WAP Public Key Infrastructure
Definition", WAP Forum.

[WSP]

"Wireless Session Protocol", WAP Forum.

[WTA]

"Wireless Telephony Application
Specification", WAP Forum.

[WTAI]

"Wireless Telephony Application
Interface", WAP Forum.

[WP-TCP]

"Wireless Profiled TCP Specification",
WAP Forum.

[WTLS]

"Wireless Transport Layer Security
Protocol", WAP Forum.

[WTP]

"Wireless Transaction Protocol
Specification", WAP Forum.

[XHTML]

"XHTML 1.1 – Module Based XHTML",
World Wide Web Consortium. URL:
<http://www.w3.org/TR/xhtml11/>

[XML]

“ Extensible Markup Language (XML)

1.0”, World Wide Web Consortium. URL:

<http://www.w3.org/TR/REC-xml/>

NOTA: Al ser el protocolo patentado por la WAP-FORUM no existe un rfc específico para el WAP.

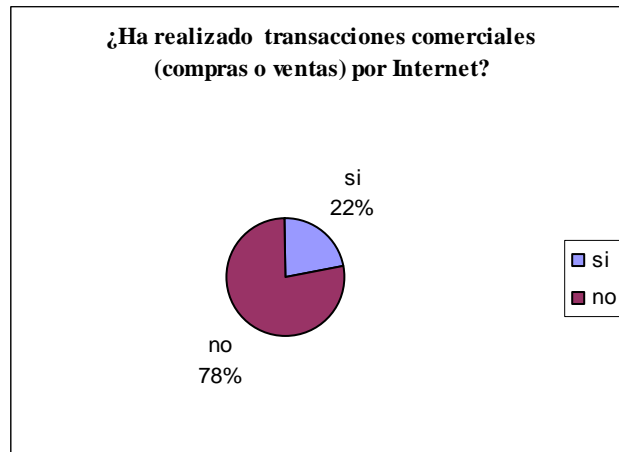
ANEXO III

ESTUDIO DE MERCADO

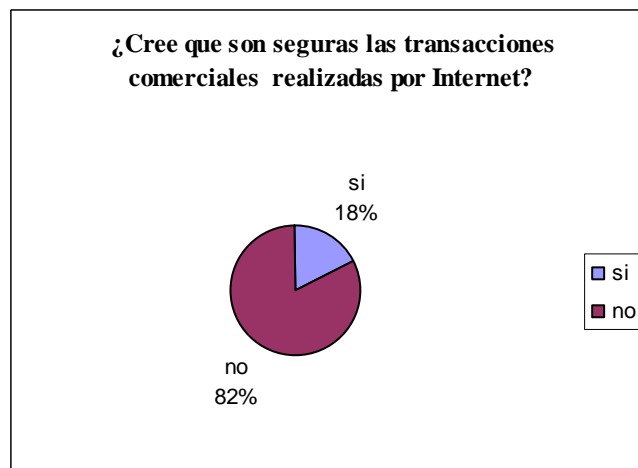
RESULTADOS DE LAS ENCUESTAS ORIENTADAS A LOS CLIENTES

Esta fue una encuesta realizada a una muestra de personas representativas al medio de Ibarra, dando como resultado los siguientes reportes tabulados (100 encuestas).

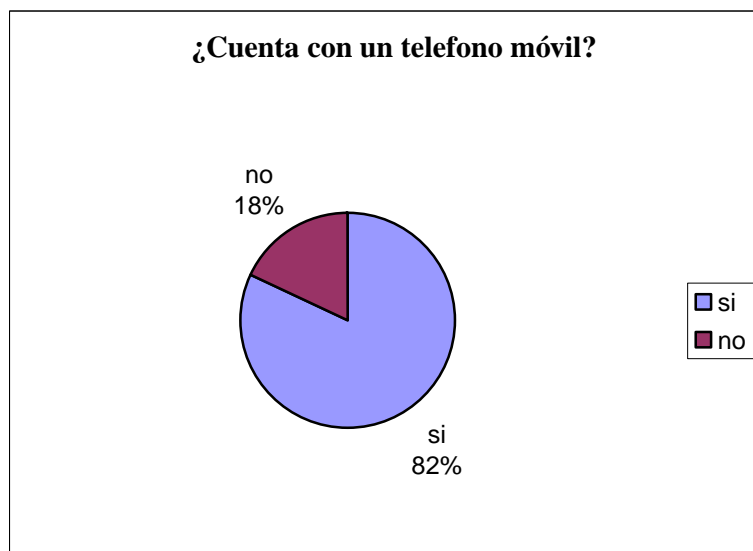
1.



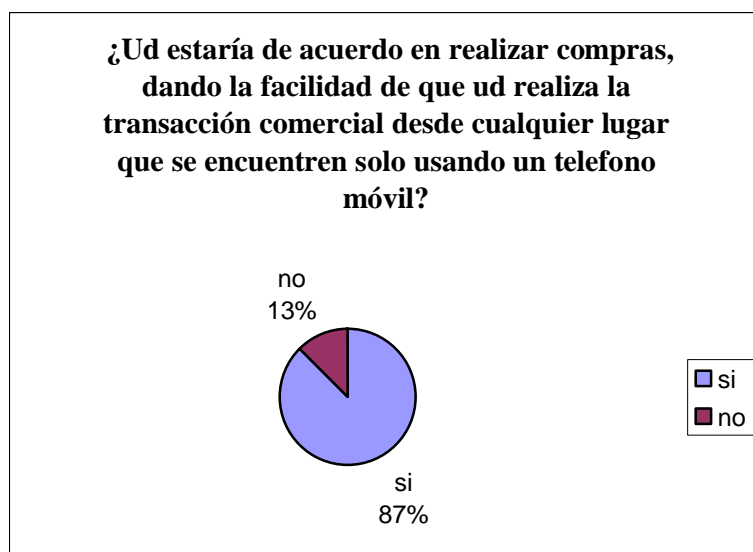
2.



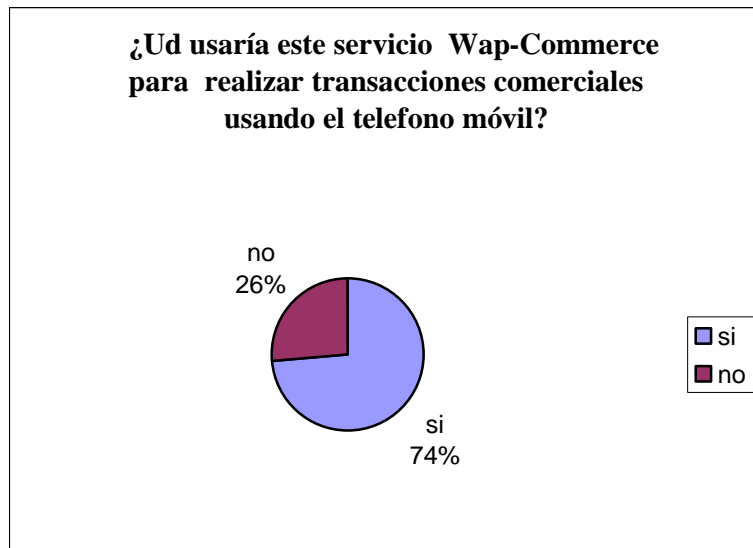
3.



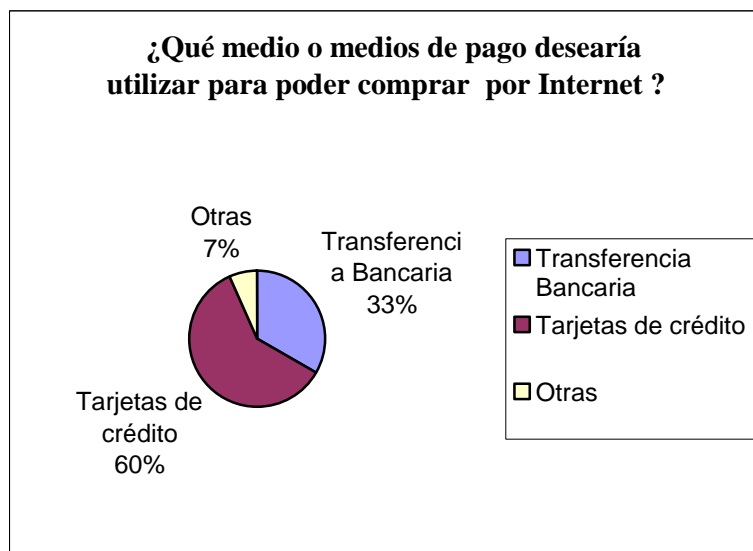
4.



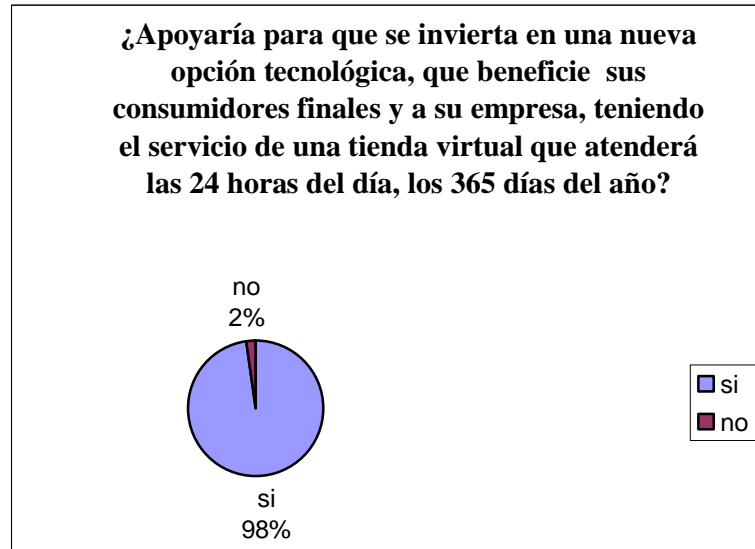
5.



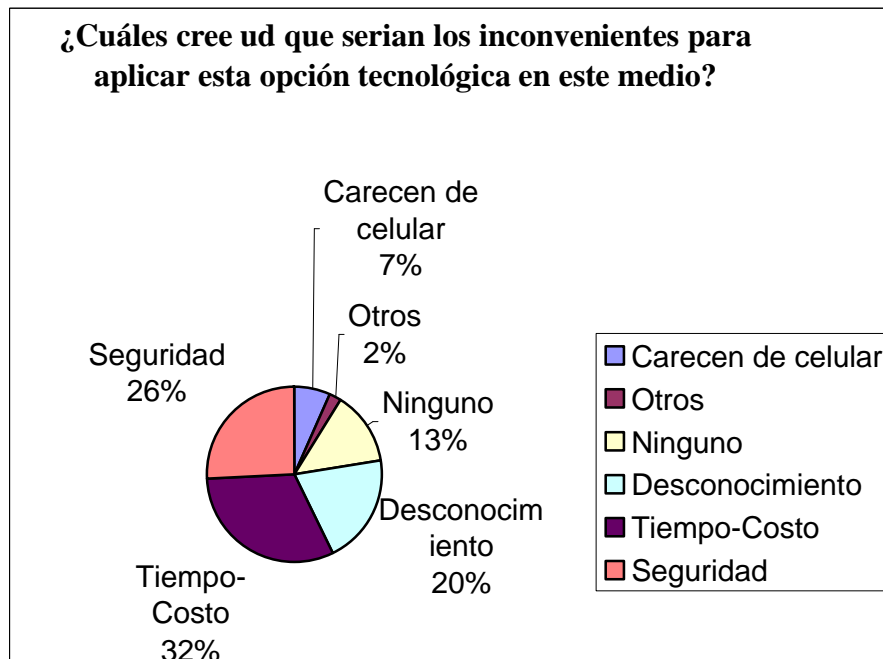
6.



7.



8.



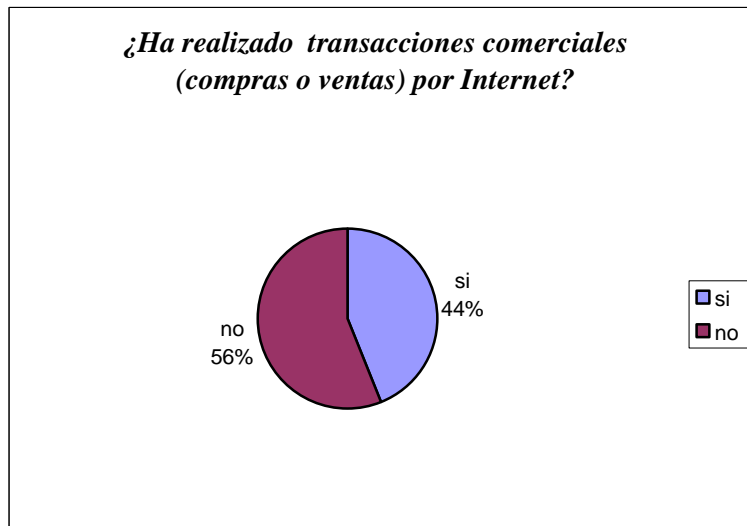
ANEXO IV

ESTUDIO DE MERCADO

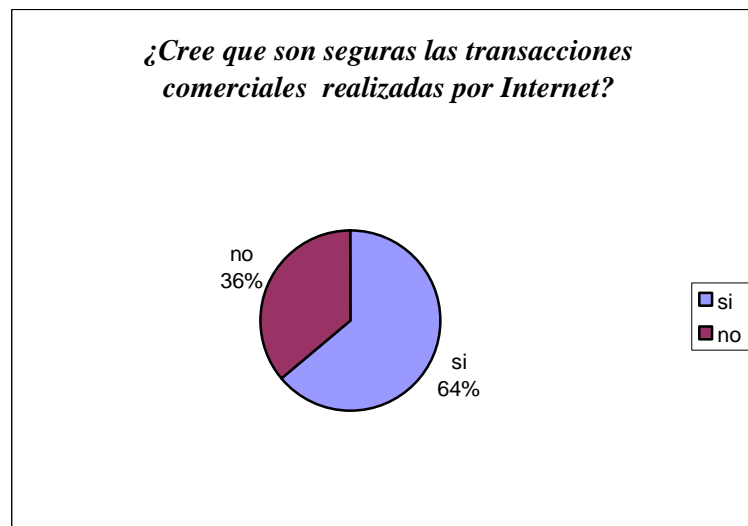
RESULTADOS DE LAS ENCUESTAS ORIENTADAS A LAS EMPRESAS

Esta fue una encuesta realizada a la mayoría de las empresas más representativas de Ibarra, dando como resultado los siguientes reportes tabulados (25 encuestas).

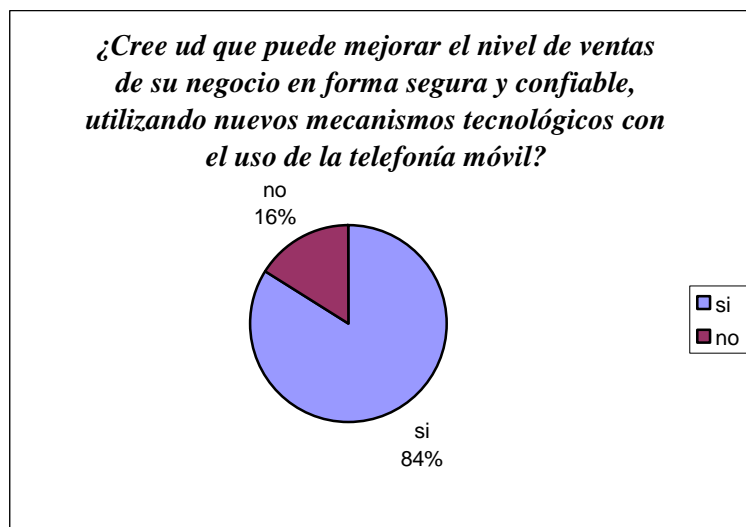
1.



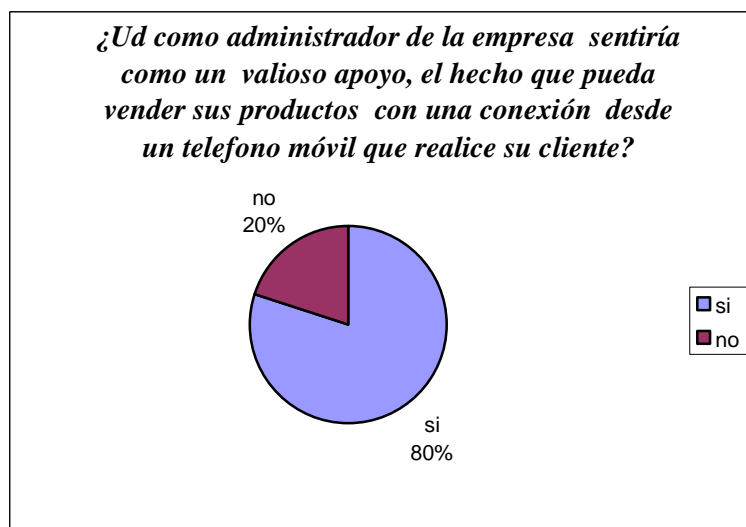
2.



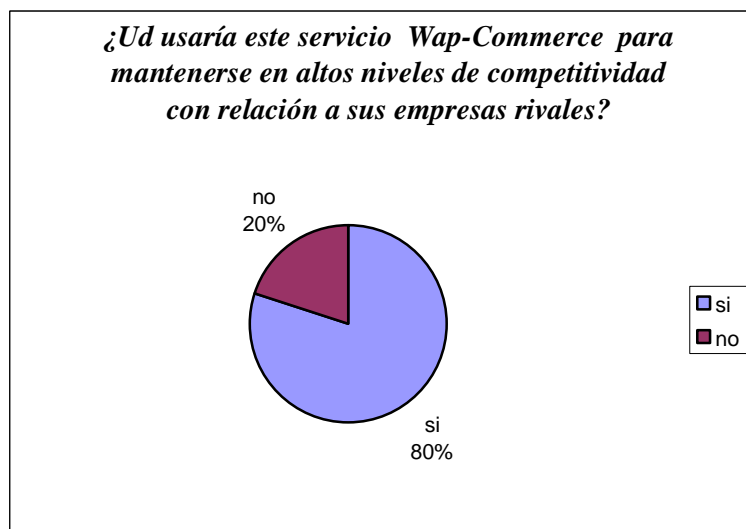
3.



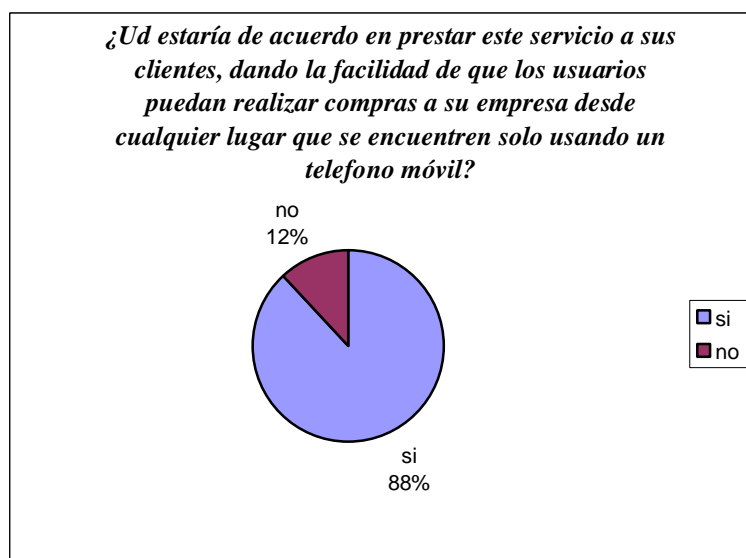
4.



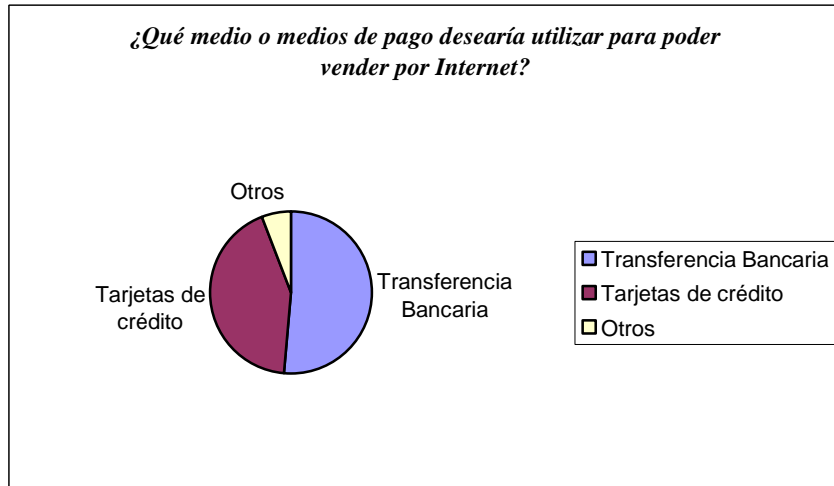
5.



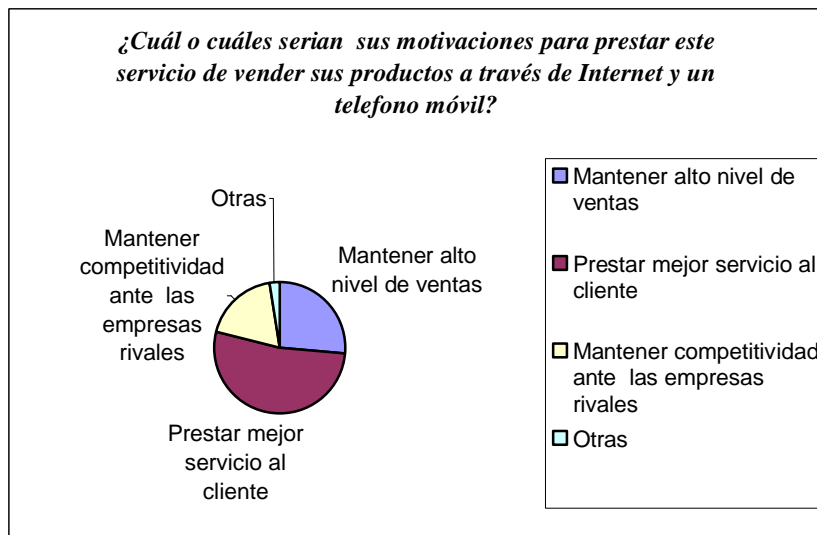
6.



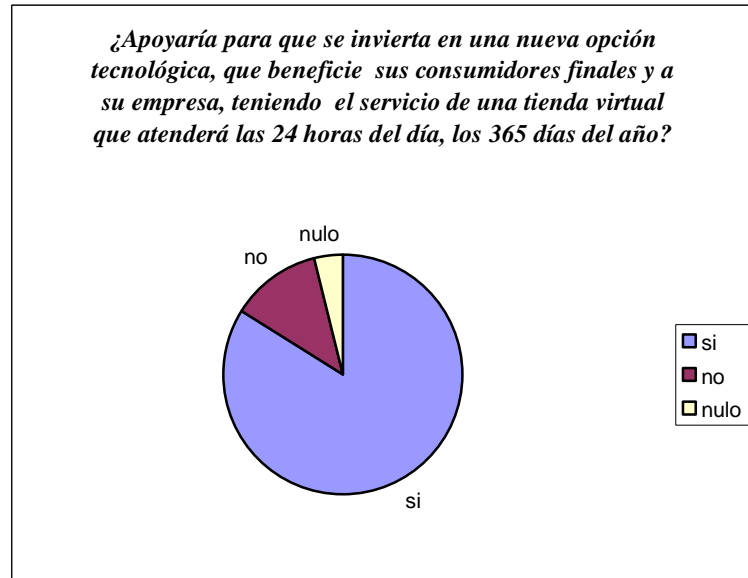
7.



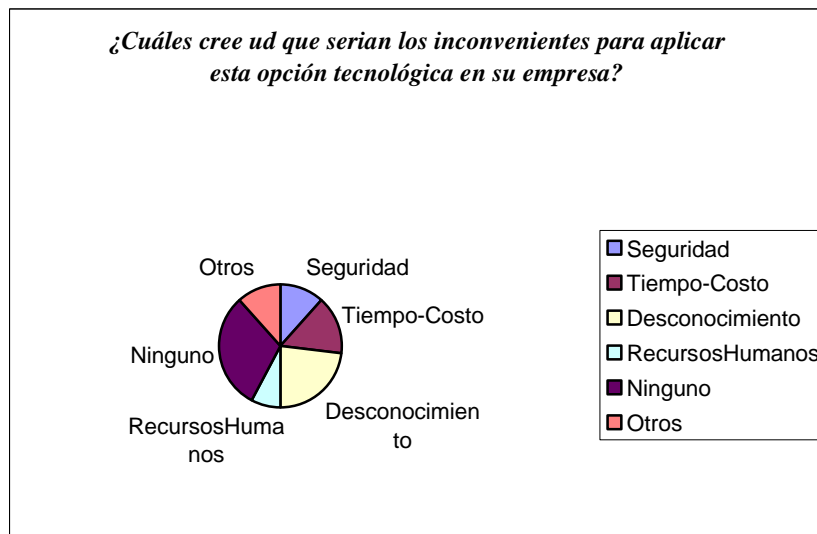
8.



9.



10.



ANEXO V

MANUAL TÉCNICO - - APLICATIVO EMAPAIWAP - -

A continuación presentamos el manual técnico del sistema Emapai-WAP.

El manual esta descrito según el estándar de JDK para usar los comentarios de sus modelos de objetos, en una manera deductiva desde la definición y explicación de clase hasta llegar a sus métodos y atributos.

Packages	
nu.ec.emapai	Contiene clases de acceso a datos.
nu.ec.emapai.servlet	Contiene servlets.
nu.ec.emapai.util	Contiene clases utilitarias.

Package nu.ec.emapai

Resumen de Clases	
CData	Maneja conexión y acceso a los datos.

Class CData

java.lang.Object

└ nu.ec.emapai.CData

```
public class CData
```

```
extends java.lang.Object
```

```
Maneja conexión y acceso a los datos.
```

Constructor

CData()

Crea una nueva instancia de CData

Métodos

java.lang.String	conecta() Inicializa el connection configurando los parámetros con el respectivo ODBC
boolean	getExisteCuenta (java.lang.String cue_id) Método que valida existencia de cuenta
java.sql.ResultSet	getHistorical (java.lang.String cuenta) Método que devuelve dataset de un histórico de los últimos 3 meses.
java.sql.ResultSet	getInformationCount (java.lang.String cue_id) Método que devuelve dataset de la información de una

	cuenta (CUE_ID, CLIENTE, DIRECCION)
java.sql.ResultSet	getInformationCountCI (java.lang.String cle_ci_ruc) Método que devuelve dataset de la información de un cliente (CUE_ID, CLIENTE, DIRECCION)
java.sql.ResultSet	getInformationCountUser (java.lang.String cle_nombre) Método que devuelve dataset de la información de un cliente (CUE_ID, CLIENTE, DIRECCION)
java.sql.ResultSet	getValues (java.lang.String cue_id) Método que devuelve dataset de la información de una cuenta valores apagar (CUE_ID, CLIENTE, VALOR_A_PAGAR, MESES_DE_DEUDA)
java.sql.ResultSet	getValuesC (java.lang.String cue_id) Método que devuelve dataset de la información de una cuenta valores apagar (CUE_ID, CLIENTE, VALOR_A_PAGAR, MESES_DE_DEUDA)
int	getVisit () Método que devuelve el numero de visitas realizadas al sitio
void	registerAction (java.lang.String cuenta, java.lang.String observación) Método que permite ingresar la operación realizada usando aplicación móvil

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalles

CData

```
public CData()
```

Crea una nueva instancia de CData

Métodos Detalles

conecta

```
public java.lang.String conecta()
```

Inicializa el connection configurando los parámetros con el respectivo ODBC

Returns:

String de mensaje especificando Exito o error.

getInformationCount

```
public java.sql.ResultSet getInformationCount(java.lang.String cue_id)
```

Método que devuelve dataset de la información de una cuenta (CUE_ID, CLIENTE, DIRECCION)

Parameters:

cue_id - id de la cuenta

Returns:

Retorna un ResultSet con información de la cuenta

getInformationCountUser

```
public java.sql.ResultSet
```

```
getInformationCountUser(java.lang.String cle_nombre)
```

Método que devuelve dataset de la información de un cliente (CUE_ID, CLIENTE, DIRECCION)

Parameters:

cle_nombre - Información por nombre de cliente

Returns:

Retorna un ResultSet con información de la cuenta

getInformationCountCI

```
public java.sql.ResultSet
```

```
getInformationCountCI(java.lang.String cle_ci_ruc)
```

Método que devuelve dataset de la información de un cliente (CUE_ID, CLIENTE, DIRECCION)

Parameters:

cle_ci_ruc - CI/RUC de cliente.

Returns:

Retorna un ResultSet con las cuentas pertenecientes a esa CI/RUC

getValues

```
public java.sql.ResultSet getValues(java.lang.String cue_id)
```

Método que devuelve dataset de la información de una cuenta valores apagar

(CUE_ID, CLIENTE, VALOR_A_PAGAR, MESES_DE_DEUDA)

Parameters:

cue_id - Cuenta a consultar.

Returns:

Retorna un ResultSet con valores de la cuenta

getValuesC

```
public java.sql.ResultSet getValuesC(java.lang.String cue_id)
```

Método que devuelve dataset de la información de una cuenta valores apagar

(CUE_ID, CLIENTE, VALOR_A_PAGAR, MESES_DE_DEUDA)

Parameters:

cue_id - Cuenta a consultar.

Returns:

Retorna un ResultSet con valores de la cuenta.

getHistorical

```
public java.sql.ResultSet getHistorical(java.lang.String cuenta)
```

```
throws java.sql.SQLException
```

Método que devuelve dataset de un histórico de los últimos 3 meses.

Parameters:

cuenta - Cuenta a consultar.

Returns:

Retorna un ResultSet con valores de la cuenta historial de los últimos 3 meses.

Throws:

java.sql.SQLException - Por consulta.

registerAction

```
public void registerAction(java.lang.String cuenta,  
                           java.lang.String observación)  
    throws java.sql.SQLException
```

Método que permite ingresar la operacion realizada usando aplicación movil

Parameters:

cuenta - Valor de la Cuenta.

observación - Texto que ingresa el cliente como observación.

Throws:

java.sql.SQLException – Excepciones del sql por inserta en la base.

getVisit

```
public int getVisit()
```

Método que devuelve el número de visitas realizadas al sitio

Returns:

Int numero de visitas al sitio.

getExisteCuenta

```
public boolean getExisteCuenta(java.lang.String cue_id)
```

Método que valida existencia de cuenta

Parameters:

cue_id - Cuenta a consultar, Existencia.

Returns:

boolean true/false; true si existe false si no existe.

Package nu.ec.emapai.servlet

Resumen de Clases	
s_Consulta	Clase que realiza consulta dependiendo de los criterios.
s_DatosCuenta	Clase que devuelve datos de la cuenta.
s_DatosCuentaC	Clase para consultar datos de una cuenta de las mostradas por la cédula.
s_Graph	Clase que permite generar gráficos dinámicamente.
s_InsertaObservación	Clase que inserta Observaciones - Validando que la cuenta o CI/RUC existan.
s_VerHistorial	Clase que consulta el top 3 del consumo de los últimos meses.

Class s_Consulta

java.lang.Object

└ javax.servlet.GenericServlet

└ javax.servlet.http.HttpServlet

└ nu.ec.emapai.servlet.s_Consulta

Interfaces implementadas:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

```
public class s_Consulta
```

```
extends javax.servlet.http.HttpServlet
```

Clase que realiza consulta de dependiendo de los criterios. Puede ser CI/RIC o cuenta.

Escribe el resultado como una pagina xhtml-mp

Class s_DatosCuenta

java.lang.Object

└ javax.servlet.GenericServlet

└ javax.servlet.http.HttpServlet

└ nu.ec.emapai.servlet.s_DatosCuenta

Interfaces implementadas:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

```
public class s_DatosCuenta
```

```
extends javax.servlet.http.HttpServlet
```

Clase que devuelve datos de la cuenta. Acepta parámetro cuenta

Class s_DatosCuentaC

java.lang.Object

└ javax.servlet.GenericServlet

└ javax.servlet.http.HttpServlet

└ **nu.ec.emapai.servlet.s_DatosCuentaC**

Interfaces implementadas:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

```
public class s_DatosCuentaC
```

```
extends javax.servlet.http.HttpServlet
```

Clase para consultar datos de una cuenta de las mostradas por la cedula.

Class s_Graph

java.lang.Object

└ javax.servlet.GenericServlet

└ javax.servlet.http.HttpServlet

└ **nu.ec.emapai.servlet.s_Graph**

Interfaces implementadas:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

```
public class s_Graph  
extends javax.servlet.http.HttpServlet
```

Class s_InsertaObservación

```
java.lang.Object  
└─ javax.servlet.GenericServlet  
    └─ javax.servlet.http.HttpServlet  
        └─ nu.ec.emapai.servlet.s_InsertaObservación
```

Interfaces implementadas:

```
java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig
```

```
public class s_InsertaObservación  
extends javax.servlet.http.HttpServlet
```

Clase que inserta Observaciones - Validando que la cuenta o CI/RUC existan.

Class s_VerHistorial

```
java.lang.Object  
└─ javax.servlet.GenericServlet  
    └─ javax.servlet.http.HttpServlet  
        └─ nu.ec.emapai.servlet.s_VerHistorial
```

Interfaces implementadas:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

```
public class s_VerHistorial
extends javax.servlet.http.HttpServlet
```

Clase que consulta el Top 3 del consumo de los últimos meses.

Resumen de Clases

EmapaUtil	Clase con Métodos utilitarios.
------------------	--------------------------------

Class EmapaUtil

```
java.lang.Object
└─nu.ec.emapai.util.EmapaUtil
```

```
public class EmapaUtil
extends java.lang.Object
```

Clase con Métodos utilitarios.

Constructor

`EmapaUtil()`

Crea una nueva instancia de EmapaUtil

Métodos

`boolean validaNumero(java.lang.String cuenta)`

Método que recorre el string caracter por caracter y valida que todos sean digitos.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detalles

EmapaUtil

`public EmapaUtil()`

Crea una nueva instancia de EmapaUtil

Métodos Detalles

validaNumero

```
public boolean validaNumero(java.lang.String cuenta)
```

Método que recorre el string caracter por caracter y valida que todos sean dígitos.

Parameters:

cuenta - String a validar.

Returns:

Devuelve V/F dependiendo respectivamente.

ANEXO VI

MANUAL TÉCNICO - - APLICATIVO VPC - -

A continuación presentamos el manual técnico del sistema VPC.

El manual esta descrito según el estándar de JDK para usar los comentarios de sus modelos de objetos, en una manera deductiva desde la definición y explicación de clase hasta llegar a sus métodos y atributos.

Package nu.ec.vpc

Resumen de Clases	
CDatos	Clase donde realizamos todas las Conexiones a la base de datos
ServidorVPC	Servelet donde se define métodos y se llama a la clase CDatos

Class CDatos

```
java.lang.Object
```

```
└ nu.ec.vpc.CDatos
```

```
public class CDatos
extends java.lang.Object
```

Clase donde realizamos todas las Conexiones a la base de datos

Constructor

CDatos()

Crea una nueva instancia de CDatos

Métodos

java.lang.String	conecta() Permite conectar a la base de datos define el string y driver de conexión
java.sql.ResultSet	getDatos(int partido) Método que devuelve los datos de un partido específico
boolean	getExisteTarjeta (java.lang.String tarjeta, java.lang.String pin) Método que valida existencia de tarjeta y pin
java.sql.ResultSet	getPartido() Método que devuelve todos los partidos o eventos
boolean	getTipoEntradas (int partido, int tipo) Método que permite actualizar la cantidad de entradas de determinado partido y tipo de entrada
int	setCompra (java.lang.String tipoBoleto, int partido, java.lang.String tarjeta, java.lang.String pin, java.lang.String beneficiario)

	Método que permite actualizar la cantidad de entradas de determinado partido y tipo de entrada
int	UpdateCantidad (int partido, java.lang.String tipo) Método que permite actualizar la cantidad de entradas de determinado partido y tipo de entrada

Métodos que heredan de class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor

CDatos

```
public CDatos()
```

Crea una nueva instancia de CDatos

Métodos

conecta

```
public java.lang.String conecta()
```

Permite conectar a la base de datos define el string y driver de conexión

Returns:

String dependiendo si los parámetros enviados son correctos o erróneos

getTipoEntradas

```
public boolean getTipoEntradas(int partido,  
                                int tipo)  
    throws java.sql.SQLException
```

Método que permite actualizar la cantidad de entradas de determinado partido y tipo de entrada

Parameters:

partido - tipo int id del partido seleccionado

tipo - tipo int id del tipo de entrada

Returns:

boolean dependiendo de si existen entradas para ese partido

Throws:

java.sql.SQLException - Controla errores de la base de datos

getPartido

```
public java.sql.ResultSet getPartido()  
    throws java.sql.SQLException
```

devuelve todos los partidos o eventos

Returns:

REtorna un RS con la consulta

Throws:

java.sql.SQLException - Lanzada por conexión a la base

getDatos

```
public java.sql.ResultSet getDatos(int partido)
```

throws java.sql.SQLException

Método que devuelve los datos de un partido específico

Parameters:

partido - id del partido seleccionado

Returns:

resultset la consulta realizada

Throws:

java.sql.SQLException - los errores de sql

setCompra

```
public int setCompra (java.lang.String tipoBoleto,  
                      int partido,  
                      java.lang.String tarjeta,  
                      java.lang.String pin,  
                      java.lang.String beneficiario)  
    throws java.sql.SQLException
```

Método que permite actualizar la cantidad de entradas de determinado partido y

tipo de entrada

Parameters:

tipoBoleto - tipo del boleto seleccionado

partido - id de partido seleccionado

tarjeta - número de la tarjeta que va a realizar la compra

pin - número del pin de la tarjeta

beneficiario - cedula de identidad del beneficiario

Returns:

Número de filas insertadas

Throws:

`java.sql.SQLException` - controla errores de sql

UpdateCantidad

```
public int UpdateCantidad (int partido,  
                             java.lang.String tipo)  
    throws java.sql.SQLException
```

Método que permite actualizar la cantidad de entradas de determinado partido y tipo de entrada

Parameters:

`partido` - id del partido seleccionado

`tipo` - tipo de boleto que se desea comprar

Returns:

número de filas actualizadas en la base de datos

Throws:

`java.sql.SQLException` - controla errores sql

getExisteTarjeta

```
public boolean getExisteTarjeta (java.lang.String tarjeta,  
                                   java.lang.String pin)
```

Método que valida existencia de tarjeta y pin

Parameters:

`tarjeta` - número de tarjeta valida

`pin` - número del pin de la tarjeta

Returns:

boolean dependiendo de si existe o no esos datos ingresados

Class ServidorVPC

java.lang.Object

└ javax.servlet.GenericServlet

└ javax.servlet.http.HttpServlet

└ nu.ec.vpc.ServidorVPC

Interfaces implementadas:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

```
public class ServidorVPC
```

```
extends javax.servlet.http.HttpServlet
```

Servelet donde se define métodos y se llama a la clase CDatos

Detalles de Constructor

```
ServidorVPC()
```

Detalles de Métodos

Protected void	doGet (javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response)
----------------	---

	Handles the HTTP GET method.
Protected void	doPost (javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Handles the HTTP POST method.
java.lang.String	getServletInfo () Returns a short description of the servlet.
protected void	processRequest (javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Processes requests for both HTTP GET and POST methods.

Métodos que heredan de class javax.servlet.http.HttpServlet

doDelete, doHead, doOptions, doPut, doTrace, getLastModified, service, service

Métodos que heredan de class javax.servlet.GenericServlet

destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletName, init, init, log, log

Métodos que heredan de class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor

ServidorVPC

```
public ServidorVPC()
```

Métodos

processRequest

```
protected void
```

```
processRequest(javax.servlet.http.HttpServletRequest request,
```

```
javax.servlet.http.HttpServletResponse response)
```

```
throws javax.servlet.ServletException,
```

```
java.io.IOException
```

Processes requests for both HTTP GET and POST methods.

Parameters:

request - servlet request

response - servlet response

Throws:

javax.servlet.ServletException

java.io.IOException

doGet

```
protected void doGet(javax.servlet.http.HttpServletRequest request,
```

```
javax.servlet.http.HttpServletResponse response)
```

```
throws javax.servlet.ServletException,
```

```
java.io.IOException
```


Handles the HTTP `GET` method.

Overrides:

`doGet` in class `javax.servlet.http.HttpServlet`

Parameters:

`request` - servlet request

`response` - servlet response

Throws:

`javax.servlet.ServletException`

`java.io.IOException`

doPost

```
protected void doPost(javax.servlet.http.HttpServletRequest request,  
                       javax.servlet.http.HttpServletResponse response)  
    throws javax.servlet.ServletException,  
           java.io.IOException
```

Handles the HTTP `POST` method.

Overrides:

`doPost` in class `javax.servlet.http.HttpServlet`

Parameters:

`request` - servlet request

`response` - servlet response

Throws:

`javax.servlet.ServletException`

`java.io.IOException`

getServletInfo

```
public java.lang.String getServletInfo()
```

Returns a short description of the servlet.

Specified by:

`getServletInfo` in interface `javax.servlet.Servlet`

Overrides:

`getServletInfo` in class `javax.servlet.GenericServlet`

Resumen de Class	
AcercaDe	Midlets visual AcecaDe
Ayuda	Midlets visual Ayuda
ConexionHilo	Clase que conecta al midlet y al servlet a través de streams y el protocolo http.
Inicio	Midlet visual Inicio
Util	Clase que sirve de colección de utilitarios para las tareas del midlet y el servlet.

Class AcercaDe

java.lang.Object

└ javax.microedition.midlet.MIDlet

└ ec.nu.vpc.AcercaDe

Interfaces implementadas:

java.lang.Runnable, javax.microedition.lcdui.CommandListener

```
public class AcercaDe
extends javax.microedition.midlet.MIDlet
implements javax.microedition.lcdui.CommandListener,
java.lang.Runnable
```

Midlets visual AcecaDe

Detalles de Constructor

AcercaDe()

Crea una nueva instancia de AcercaDe

Detalles de Métodos

void	commandAction (javax.microedition.lcdui.Command c, javax.microedition.lcdui.Displayable s) Este Método coge el evento de los botones
void	destroyApp (boolean unconditional)
void	exitMIDlet () Este Método sale del midlets
javax.microedition.lcdui. Command	get_exitCommand1 () Este método devuelve ejemplo para el componente exitCommand1 y debe ser llamado en lugar de acceder directamente al campo exitCommand1.
javax.microedition.lcdui. Form	get_PantallaAcercaDe ()
Javax.microedition.lcdui. StringItem	get_stringItem1 () Este método devuelve ejemplo para el componente stringItem1 y debe ser llamado en lugar de acceder directamente

	al campo campo stringItem1.
javax.microedition.lcdui. Display	getDisplay() Este método deberá devolver una instancia de la pantalla
void	pauseApp()
void	run() Este método se ejecuta en otro hilo y se comunica con gestores de eventos y proporciona la capacidad de procesamiento de eventos en diferentes hilo.
void	startApp()

Métodos que heredan de class javax.microedition.midlet.MIDlet

getAppProperty, notifyDestroyed, notifyPaused, resumeRequest

Métodos que heredan de class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

Constructor

AcercaDe

```
public AcercaDe()
```

Crea una nueva instancia de AcercaDe

Métodos

run

```
public void run()
```

Este método se ejecuta en otro hilo y se comunica con gestores de eventos y proporciona la capacidad de procesamiento de eventos en diferentes hilos.

Specified by:

run in interface `java.lang.Runnable`

commandAction

```
public void commandAction(javax.microedition.lcdui.Command c,  
                             javax.microedition.lcdui.Displayable s)
```

Este Método coje el evento de los botones

Specified by:

commandAction in interface `javax.microedition.lcdui.CommandListener`

getDisplay

```
public javax.microedition.lcdui.Display getDisplay()
```

Este método deberá devolver una instancia de la pantalla

Returns:

pantalla

exitMIDlet

```
public void exitMIDlet()
```

Este Método sale del midlets

get_PantallaAcercaDe

```
public javax.microedition.lcdui.Form get_PantallaAcercaDe()
```

get_exitCommand1

```
public javax.microedition.lcdui.Command get_exitCommand1()
```

Este método devuelve ejemplo para exitCommand1 componente y debe ser llamado en lugar de acceder directamente exitCommand1 campo.

Returns:

Instancia del componente exitCommand1

get_stringItem1

```
public javax.microedition.lcdui.StringItem get_stringItem1()
```

Este método devuelve ejemplo para stringItem1 componente y debe ser llamado en lugar de acceder directamente stringItem1 campo.

Returns:

Instancia del componente stringItem1

startApp

```
public void startApp()
```

Specified by:

```
startApp in class javax.microedition.midlet.MIDlet
```

pauseApp

```
public void pauseApp()
```

Specified by:

```
pauseApp in class javax.microedition.midlet.MIDlet
```

destroyApp

```
public void destroyApp(boolean unconditional)
```

Specified by:

```
destroyApp in class javax.microedition.midlet.MIDlet
```

Class Ayuda

```
java.lang.Object
```

```
└ javax.microedition.midlet.MIDlet
```

```
└ ec.nu.vpc.Ayuda
```

Interfaces implementadas:

```
java.lang.Runnable, javax.microedition.lcdui.CommandListener
```

```
public class Ayuda
```

```
extends javax.microedition.midlet.MIDlet
```


implements javax.microedition.lcdui.CommandListener,
 java.lang.Runnable

midlets visual Ayuda

Detalles de Constructor	
Ayuda()	Crea una nueva instancia de Ayuda

Detalles de Métodos	
void	commandAction (javax.microedition.lcdui. Command c, javax.microedition.lcdui.Displayable s)
void	destroyApp (boolean unconditional)
void	exitMIDlet () Este Método sale del midlets
javax.microedition.lcdui.Comman d	get_exit2 () Este método devuelve ejemplo para exit2 componente y debe ser llamado en lugar de acceder directamente exit2 campo.
javax.microedition.lcdui.Comman	get_exitCommand1 ()

<p>d</p>	<p>Este método devuelve ejemplo para exitCommand1 componente y debe ser llamado en lugar de acceder directamente exitCommand1 campo.</p>
<p>Javax.microedition.lcdui.ChoiceGroup</p>	<p>get_listaAyudas()</p> <p>Este método devuelve ejemplo para listaAyudas componente y debe ser llamado en lugar de acceder directamente listaAyudas campo.</p>
<p>javax.microedition.lcdui.Command</p> <p>d</p>	<p>get_ok()</p> <p>Este método devuelve ejemplo para ok componente y debe ser llamado en lugar de acceder directamente ok campo.</p>
<p>javax.microedition.lcdui.Form</p>	<p>get_PantallaListarTemasAyuda()</p>
<p>javax.microedition.lcdui.Form</p>	<p>get_PantallaOpcionCompra()</p> <p>This method returns instance for PantallaOpcionCompra component and should be called instead of accessing PantallaOpcionCompra field directly.</p>
<p>javax.microedition.lcdui.Form</p>	<p>get_PantallaPartidosListados()</p> <p>Este método devuelve ejemplo para PantallaPartidosListados componente y debe ser llamado en lugar de acceder directamente PantallaPartidosListados campo.</p>

<p>javax.microedition.lcdui.Form</p>	<p>get_PantallaSeguridad()</p> <p>Este método devuelve ejemplo para PantallaSeguridad componente y debe ser llamado en lugar de acceder directamente PantallaSeguridad campo.</p>
<p>javax.microedition.lcdui.Form</p>	<p>get_PantallaTipoEntradaA()</p> <p>Este método devuelve ejemplo para PantallaTipoEntradaA componente y debe ser llamado en lugar de acceder directamente PantallaTipoEntradaA campo.</p>
<p>javax.microedition.lcdui.String Item</p>	<p>get_txtCompra()</p> <p>This method returns instance for txtCompra component and should be called instead of accessing txtCompra field directly.</p>
<p>javax.microedition.lcdui.String Item</p>	<p>get_txtEntradas()</p> <p>Este método devuelve ejemplo para txtEntradas componente y debe ser llamado en lugar de acceder directamente txtEntradas campo.</p>
<p>javax.microedition.lcdui.String Item</p>	<p>get_txtListado()</p> <p>Este método devuelve ejemplo para txtListado componente y debe ser llamado en lugar de acceder directamente txtListado campo.</p>
<p>javax.microedition.lcdui.String Item</p>	<p>get_txtSeguridad()</p> <p>Este método devuelve ejemplo para</p>

	txtSeguridad componente y debe ser llamado en lugar de acceder directamente txtSeguridad campo.
javax.microedition.lcdui.Display	getDisplay() Este método deberá devolver una instancia de la pantalla
void	pauseApp()
void	run() Este método se ejecuta en otro hilo y se comunica con gestores de eventos y proporciona la capacidad de procesamiento de eventos en diferentes hilo.
void	startApp()

Métodos que heredan de class javax.microedition.midlet.MIDlet
getAppProperty, notifyDestroyed, notifyPaused, resumeRequest

Métodos que heredan de class java.lang.Object
Equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor

Ayuda

```
public Ayuda()
```

Crea una nueva instancia de Ayuda

Métodos

run

```
public void run()
```

Este método se ejecuta en otro hilo y se comunica con gestores de eventos y proporciona la capacidad de procesamiento de eventos en diferentes hilo.

Specified by:

run in interface `java.lang.Runnable`

commandAction

```
public void commandAction(javax.microedition.lcdui.Command c,  
                             javax.microedition.lcdui.Displayable s)
```

Specified by:

commandAction in interface `javax.microedition.lcdui.CommandListener`

getDisplay

```
public javax.microedition.lcdui.Display getDisplay()
```

Este método deberá devolver una instancia de la pantalla

Returns:

this

exitMIDlet

```
public void exitMIDlet()
```

Este Método sale del midlets

get_PantallaListarTemasAyuda

```
public javax.microedition.lcdui.Form get_PantallaListarTemasAyuda()
```

get_listaAyudas

```
public javax.microedition.lcdui.ChoiceGroup get_listaAyudas()
```

Este método devuelve ejemplo para listaAyudas componente y debe ser llamado en lugar de acceder directamente listaAyudas campo.

Returns:

Instancia del componente listaAyudas

get_exitCommand1

```
public javax.microedition.lcdui.Command get_exitCommand1()
```

Este método devuelve ejemplo para exitCommand1 componente y debe ser llamado en lugar de acceder directamente exitCommand1 campo.

Returns:

Instancia del componente exitCommand1

get_ok

```
public javax.microedition.lcdui.Command get_ok()
```

Este método devuelve ejemplo para ok componente y debe ser llamado en lugar de acceder directamente ok campo.

Returns:

Instancia del componente Ok

get_exit2

```
public javax.microedition.lcdui.Command get_exit2()
```

Este método devuelve ejemplo para exit2 componente y debe ser llamado en lugar de acceder directamente exit2 campo.

Returns:

Instancia del componente exit2

get_PantallaPartidosListados

```
public javax.microedition.lcdui.Form get_PantallaPartidosListados()
```

Este método devuelve ejemplo para PantallaPartidosListados componente y debe ser llamado en lugar de acceder directamente PantallaPartidosListados campo.

Returns:

Instancia del componente PantallaPartidosListados

get_txtListado

```
public javax.microedition.lcdui.StringItem get_txtListado()
```

Este método devuelve ejemplo para txtListado componente y debe ser llamado en lugar de acceder directamente txtListado campo.

Returns:

Instancia del componente txtListado

get_PantallaTipoEntradaA

```
public javax.microedition.lcdui.Form get_PantallaTipoEntradaA()
```

Este método devuelve ejemplo para PantallaTipoEntradaA componente y debe ser llamado en lugar de acceder directamente PantallaTipoEntradaA campo.

Returns:

Instancia del componente PantallaTipoEntradaA

get_txtEntradas

```
public javax.microedition.lcdui.StringItem get_txtEntradas()
```

Este método devuelve ejemplo para txtEntradas componente y debe ser llamado en lugar de acceder directamente txtEntradas campo.

Returns:

Instancia del componente txtEntradas

get_PantallaOpcionCompra

```
public javax.microedition.lcdui.Form get_PantallaOpcionCompra()
```

Este método devuelve la instancia del componente PantallaOpcionCompra y debería llamarse cuando se instancia el campo de PantallaOpcionCompra

Returns:

Instancia del componente PantallaOpcionCompra

get_txtCompra

```
public javax.microedition.lcdui.StringItem get_txtCompra()
```

Este método retorna la instancia del componente txtCompra component y debería llamarse cuando se accede al campo txtCompra field

Returns:

Instancia del componente txtCompra

get_PantallaSeguridad

```
public javax.microedition.lcdui.Form get_PantallaSeguridad()
```

Este método devuelve ejemplo para PantallaSeguridad componente y debe ser llamado en lugar de acceder directamente PantallaSeguridad campo.

Returns:

Instancia del componente PantallaSeguridad

get_txtSeguridad

```
public javax.microedition.lcdui.StringItem get_txtSeguridad()
```

Este método devuelve ejemplo para txtSeguridad componente y debe ser llamado en lugar de acceder directamente txtSeguridad campo.

Returns:

Instancia del componente txtSeguridad

startApp

```
public void startApp()
```

Specified by:

```
startApp in class javax.microedition.midlet.MIDlet
```

pauseApp

```
public void pauseApp()
```

Specified by:

```
pauseApp in class javax.microedition.midlet.MIDlet
```

destroyApp

```
public void destroyApp(boolean unconditional)
```

Specified by:

```
destroyApp in class javax.microedition.midlet.MIDlet
```

Class ConexionHilo

```
java.lang.Object
```

```
└ ec.nu.vpc.ConexionHilo
```

Interfaces implementadas:

```
java.lang.Runnable
```

```
public class ConexionHilo
```

extends java.lang.Object

implements java.lang.Runnable

Clase que conecta al midlet y al servlet a través de streams y el protocolo http.

Resumen de campos	
Javax.microedition.io.HttpConnection	conexion
Inicio	midlet
java.lang.String	op
java.lang.String	parámetros
java.lang.String	respuesta
java.lang.Thread	t
Util	util

Detalles de Constructor
ConexionHilo()

Crea una nueva instancia de ConexionHilo.

Detalles de Métodos

void	conectar (Inicio m, java.lang.String parámetros, java.lang.String op) Método Conectar - inicializa atributos de la clase y lanza el hilo.
void	run () Lanza el hilo de conexión

Métodos que heredan de class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait,
 wait

Detalle de campos

t

public java.lang.Thread **t**

respuesta

public java.lang.String **respuesta**

conexion

```
public javax.microedition.io.HttpConnection conexion
```

midlet

```
public Inicio midlet
```

parámetros

```
public java.lang.String parámetros
```

op

```
public java.lang.String op
```

util

```
public Util util
```

Constructor

ConexionHilo

```
public ConexionHilo()
```

 Crea una nueva instancia de ConexionHilo.

Métodos

conectar

```
public void conectar(Inicio m,
```

```
java.lang.String parámetros,  
java.lang.String op)
```

Método Conectar - inicializa atributos de la clase y lanza el hilo.

Parameters:

m - Inicializa el midlet a escribir en el cliente

parámetros - Los parámetros que enviamos para conectar en el servlet.

op - opcion a realizar.

run

```
public void run()
```

Lanza el hilo de conexion

Specified by:

run in interface java.lang.Runnable

Detalles de Métodos

void	commandAction (javax.microedition.lcdui.Command c, javax.microedition.lcdui.Displayable s)
void	destroyApp (boolean unconditional)
void	exitMIDlet () Este método debe salir del midlets
javax.microedition.lcdui.Command	get_exitEventos () Este método retorna una instancia del componente exitEventos y debe llamarse al acceder al campo exitEventos
javax.microedition.lcdui.ChoiceGroup	get_listaEventos () Este método retorna una instancia del componente listaEventos y debe llamarse al acceder al campo listaEventos
javax.microedition.lcdui.ChoiceGroup	get_listaTiposEntrada () Este método retorna una instancia del componente listaTiposEntrada y debe llamarse al acceder al campo listaTiposEntrada.
javax.microedition.lcdui.Command	get_okCompraTipo () Este método retorna una instancia del componente okCompraTipo y debe llamarse al

<p>javax.microedition.lcdui.Form</p>	<p>get_PantallaExito()</p> <p>Este método retorna una instancia del componente okDescripcionEventos y debe llamarse al acceder al campo okDescripcionEventos .</p>
<p>javax.microedition.lcdui.Form</p>	<p>get_PantallaTiposEntradas()</p> <p>Este método retorna una instancia del componente PantallaTiposEntradas y debe llamarse al acceder al campo PantallaTiposEntradas.</p>
<p>javax.microedition.lcdui.StringItem</p>	<p>get_stringItem1()</p> <p>Este método retorna una instancia del componente stringItem1 y debe llamarse al acceder al campo stringItem1 .</p>
<p>javax.microedition.lcdui.TextField</p>	<p>get_txtBeneficiario()</p> <p>Este método retorna una instancia del componente txtBeneficiario y debe llamarse al acceder al campo txtBeneficiario.</p>
<p>javax.microedition.lcdui.TextField</p>	<p>get_txtNumTarjeta()</p> <p>Este método retorna una instancia del componente txtNumTarjeta y debe llamarse al acceder al campo txtNumTarjeta.</p>
<p>javax.microedition.lcdui.TextField</p>	<p>get_txtPin()</p> <p>Este método retorna una instancia del componente txtPin y debe llamarse al acceder al campo txtPin.</p>

javax.microedition.lcdui.TextBox	get_txtResultado() Este método retorna una instancia del componente txtResultado y debe llamarse al acceder al campo txtResultado.
javax.microedition.lcdui.Display	getDisplay() Este método deberá devolver una instancia de la pantalla
java.lang.String	getResultado()
void	pauseApp()
void	setResultado(java.lang.String res)
void	startApp()

Class Inicio

java.lang.Object

└ javax.microedition.midlet.MIDlet

└ ec.nu.vpc.Inicio

Interfaces implementadas:

javax.microedition.lcdui.CommandListener

```
public class Inicio
```

```
extends javax.microedition.midlet.MIDlet
```

```
implements javax.microedition.lcdui.CommandListener
```

Midlet visual Inicio

Detalles de Constructor

Inicio()

Crea una nueva instancia de Inicio

Métodos que heredan de class javax.microedition.midlet.MIDlet

`getAppProperty, notifyDestroyed, notifyPaused, resumeRequest`

Métodos que heredan de class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait`

Constructor

Inicio

```
public Inicio()
```

Crea una nueva instancia de Inicio

Métodos

getDisplay

```
public javax.microedition.lcdui.Display getDisplay()
```

Este método deberá devolver una instancia de la pantalla

Returns:

este componente pantalla

exitMIDlet

```
public void exitMIDlet()
```

Este método sale del midlets.

get_PantallaEventos

```
public javax.microedition.lcdui.Form get_PantallaEventos()
```

Este método retorna una instancia del componente PantallaEventos y debe

llamarse al acceder al campo PantallaEventos

Returns:

Instancia del componente PantallaEventos

commandAction

```
public void commandAction(javax.microedition.lcdui.Command c,  
                             javax.microedition.lcdui.Displayable s)
```

Specified by:

`commandAction` in interface `javax.microedition.lcdui.CommandListener`

get_exitEventos

```
public javax.microedition.lcdui.Command get_exitEventos()
```

Este método retorna una instancia del componente exitEventos y debe llamarse al acceder al campo exitEventos

Returns:

Instancia del componente exitEventos

get_okTiposEventos

```
public javax.microedition.lcdui.Command get_okTiposEventos()
```

Este método retorna una instancia del componente okTiposEventos y debe llamarse al acceder al campo okTiposEventos

Returns:

Instancia del componente okTiposEventos

get_okDescripcionEventos

```
public javax.microedition.lcdui.Command get_okDescripcionEventos()
```

Este método retorna una instancia del componente okDescripcionEventos y debe llamarse al acceder al campo okDescripcionEventos

Returns:

Instancia del componente okDescripcionEventos

get_PantallaTiposEntradas

```
public javax.microedition.lcdui.Form get_PantallaTiposEntradas()
```

Este método retorna una instancia del componente PantallaTiposEntradas y debe llamarse al acceder al campo PantallaTiposEntradas

Returns:

Instancia del componente PantallaTiposEntradas

get_PantallaDescripcion

```
public javax.microedition.lcdui.Form get_PantallaDescripcion()
```

Este método retorna una instancia del componente PantallaDescripcion y debe llamarse al acceder al campo PantallaDescripcion

Returns:

Instancia del componente PantallaDescripcion

get_txtResultado

```
public javax.microedition.lcdui.TextBox get_txtResultado()
```

Este método retorna una instancia del componente txtResultado y debe llamarse al acceder al campo txtResultado

Returns:

Instancia del componente txtResultado

get_listaEventos

```
public javax.microedition.lcdui.ChoiceGroup get_listaEventos()
```

Este método retorna una instancia del componente listaEventos y debe llamarse al acceder al campo listaEventos

Returns:

Instancia del componente listaEventos

get_listaTiposEntrada

```
public javax.microedition.lcdui.ChoiceGroup get_listaTiposEntrada()
```

Este método retorna una instancia del componente listaTiposEntrada y debe llamarse al acceder al campo listaTiposEntrada

Returns:

Instancia del componente listaTiposEntrada

get_okCompraTipo

```
public javax.microedition.lcdui.Command get_okCompraTipo()
```

Este método retorna una instancia del componente okCompraTipo y debe llamarse al acceder a okCompraTipo

Returns:

Instancia del componente okCompraTipo

get_PantallaCompra

```
public javax.microedition.lcdui.Form get_PantallaCompra()
```

Este método retorna una instancia del componente PantallaCompra y debe llamarse al acceder a PantallaCompra

Returns:

Instancia del componente PantallaCompra

get_txtNumTarjeta

```
public javax.microedition.lcdui.TextField get_txtNumTarjeta()
```

Este método retorna una instancia del componente txtNumTarjeta y debe

llamarse al acceder al campo txtNumTarjeta

Returns:

Instancia del componente txtNumTarjeta

get_txtPin

```
public javax.microedition.lcdui.TextField get_txtPin()
```

Este método retorna una instancia del componente txtPin y debe llamarse al

acceder al campo txtPin

Returns:

Instancia del componente txtPin

get_txtBeneficiario

```
public javax.microedition.lcdui.TextField get_txtBeneficiario()
```

Este método retorna una instancia del componente txtBeneficiario y debe

llamarse al acceder al campo txtBeneficiario

Returns:

Instancia del componente txtBeneficiario

get_okRealizaCompra

```
public javax.microedition.lcdui.Command get_okRealizaCompra()
```

Este método retorna una instancia del componente okRealizaCompra y debe llamarse al acceder a okRealizaCompra

Returns:

Instancia del componente okRealizaCompra

get_PantallaExito

```
public javax.microedition.lcdui.Form get_PantallaExito()
```

Este método retorna una instancia del componente PantallaExito y debe llamarse al acceder a PantallaExito

Returns:

Instancia del componente PantallaExito

get_stringItem1

```
public javax.microedition.lcdui.StringItem get_stringItem1()
```

Este método retorna una instancia del componente stringItem1 y debe llamarse al acceder a stringItem1

Returns:

Instancia del componente stringItem1

startApp

```
public void startApp()
```

Specified by:


```
startApp in class javax.microedition.midlet.MIDlet
```

pauseApp

```
public void pauseApp()
```

Specified by:

```
pauseApp in class javax.microedition.midlet.MIDlet
```

destroyApp

```
public void destroyApp(boolean unconditional)
```

Specified by:

```
destroyApp in class javax.microedition.midlet.MIDlet
```

getResultado

```
public java.lang.String getResultado()
```

setResultado

```
public void setResultado(java.lang.String res)
```

Class Util

```
java.lang.Object
```

```
└ ec.nu.vpc.Util
```

```
public class Util
extends java.lang.Object
```

Clase que sirve de coleccion de utilitarios para las tareas del midlet y el servlet.

Detalles de Constructor

Util()

Crea nueva instancia de Util

Detalles de Métodos

java.lang.String[]	split (java.lang.String cadena, java.lang.String caracter)
--------------------	--

Método para devolver un vector de cadenas cortadas de acuerdo a un caracter de criterio.

Métodos que heredan de class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

Constructor

Util

```
public Util()
```

Crea nueva instancia de Util

Métodos

split

```
public java.lang.String[] split(java.lang.String cadena,  
                                java.lang.String character)
```

Método para devolver un vector de cadenas cortadas de acuerdo a un caracter de criterio.

Parameters:

`cadena` - Cadeba base

`character` - Por el cual se basa para tomar las subcadenas.

Returns:

Un vector de subcadenas

ANEXO VII

MANUAL TÉCNICO - - APLICATIVO WAP-COMMERCE - -

A continuación presentamos el manual técnico del sistema WAP-COMMERCE.

El manual esta descrito según el estándar de JDK para usar los comentarios de sus modelos de objetos, en una manera deductiva desde la definición y explicación de clase hasta llegar a sus métodos y atributos.

Resumen de Clases	
CData	Clase donde realizamos todas las Conexiones a la base de datos
CompanyDTO	Clase que mapea la tabla tbl_company para acceder a métodos que accionan sobre esa tabla.
OptarDTO	Clase que mapea la tabla tbl_Optar para acceder a métodos que accionan sobre esa tabla.
SQLHelper	Clase que permite accesos a la base de datos, implementa de la librería Microsoft.ApplicationBlocks.Data
TransactionDetail	Clase que mapea la tabla tbl_Detail_Transaction para acceder a métodos que accionan sobre esa tabla.
TransactionDTO	Clase que mapea la tabla tbl_Transaction para acceder a métodos que accionan sobre esa tabla.

ProductDTO	Clase que mapea la tabla tbl_Product para acceder a métodos que accionan sobre esa tabla.
CategoryDTO	Clase que mapea la tabla tbl_Category para acceder a métodos que accionan sobre esa tabla.

Class CData

```
public class CData
```

Clase donde realizamos todas las Conexiones a la base de datos

Constructor	
CData()	Crea una nueva instancia de CData

Métodos	
DataSet	spgetCategory() Método que devuelve las categorías que existen
DataSet	getCompanyByType(int idcategory) Método que devuelve las empresas de acuerdo a la categoría

Dataset	productByCompany(int idCompany) Método que devuelve los productos de una empresa determinada
DataSet	ProductByDateCompany(DateTime date, int idCompany) Método que devuelve los productos de una empresa disponibles en una fecha
DataSet	getProduct_company_criterio(int idCategory, string criterio) Método que devuelve los productos de la empresa y dando un criterio de búsqueda
void	registerTransaction(TransactionDTO transation) Método que permite ingresar la transacción en la tabla transacción
int	getTransaction() Método que devuelve la última transacción realizada
bool	getValideNumber(string Company) Método que devuelve la expresión regular de la entidad financiera para q en el frontal usar el control RegularExpressionValidator
bool	updateProductEnable(int codProduct, int codCompany, int state) Método que actualiza el estado del producto y devuelve false si no encuentra un producto con el id recibido.
DataSet	getCompanyFinanciamiento() Método que devuelve la las entidades financieras de OPTAR
int	getSecuencial() Método que devuelve un id de la transacción para agregar el registro detalle.
void	registerRecord(int idTransaction, bool successful)

	Método que permite ingresar la transacción en la tabla historial
--	--

Class CategoryDTO

```
public class CategoryDTO
```

Detalles de Constructor

CategoryDTO ()

Inicializa un nueva instancia de CategoryDTO.

Detalles de Métodos

DataSet	getCategorys ()
	Devuelve un DataSet con las categorías existentes.

Class CompanyDTO

```
public class CompanyDTO
```

Detalles de Constructor

CompanyDTO ()

Inicializa un nueva instancia de CompanyDTO.

Detalles de Métodos

DataSet	<code>getCompanyByType(int idcategory)</code>
	Devuelve un DataSet con las compañías por un id de categoría.

Class OptarDTO

```
public class OptarDTO
```

Detalles de Constructor

OptarDTO ()

Inicializa un nueva instancia de OptarDTO.

Detalles de Métodos

DataSet	getCompanyF () Devuelve un DataSet con todas las entidades de crédito.
string	getExpressionValide(string nameCompany) Devuelve un string con la expresión.

Class ProductDTO

```
public class ProductDTO
```

Detalles de Constructor

ProductDTO ()

Inicializa un nueva instancia de ProductDTO.

Detalles de Métodos

DataSet	productByCompany(int idCompany) Devuelve un DataSet con todos los productos de una compañía específica.
DataSet	getProduct_company_criterio(int idCategory, string criterio) Devuelve un DataSet con todos los productos de una compañía

	específica y por un criterio.
bool	<code>updateProduct(int codProduct, int codCompany, int state)</code> Actualiza el estado.
DataSet	<code>ProductByDateCompany(DateTime date, int idCompany)</code> Devuelve un DataSet con todos los productos de una compañía específica y en una fecha.

Class TransactionDetail

```
public class TransactionDetail
```

Detalles de Constructor

TransactionDetail ()

Inicializa un nueva instancia de TransactionDetail.

Class TransactionDTO

```
public class TransactionDTO
```

Detalles de Constructor

TransactionDTO ()

Inicializa un nueva instancia de TransactionDTO.

Detalles de Métodos

<code>void</code>	<code>registerTransaction(TransactionDTO transaction)</code> Registra un objeto transactionDTO.
<code>int</code>	<code>getTransaction()</code> Devuelve el id de la instancia actual de transaccion.

Resumen de Páginas ASP.NET	
index.aspx	Form1: Pagina inicial donde escogemos la categoría. Form2: Muestra las compañías pertenecientes a la categoría seleccionada.
Products.aspx	Muestra todos los productos disponibles de determinada compañía.
Transaction.aspx	Form1: Muestra compañías de crédito y permite ingresar datos de cuenta para compra. Form2: Muestra información de éxito o error.
Dates.aspx	Pagina que muestra un calendario y permite seleccionar una fecha.
Rooms.aspx	Pagina que despliega todas las habitaciones en el día señalado.
empty.aspx	Muestra un error personalizado en caso de no haber habitaciones disponibles.