

**ANÁLISIS DEL RENDIMIENTO EN SERVICIOS WEB MÓVILES**

**GUILLERMO SÁNCHEZ JAIMES**

**JESÚS EVELIO ORTEGA ARÉVALO**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - COLOMBIA  
INSTITUTO TECNOLÓGICO DE MONTERREY - MÉXICO  
MAESTRÍA EN CIENCIAS COMPUTACIONALES  
2005**

**ANÁLISIS DEL RENDIMIENTO EN SERVICIOS WEB MÓVILES**

**GUILLERMO SÁNCHEZ JAIMES  
MCC961644**

**JESÚS EVELIO ORTEGA ARÉVALO  
MCC950754**

**Línea de Investigación  
Telecomunicaciones y Tecnologías Web**

**Director  
Doctor EDUARDO CARRILLO ZAMBRANO**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - COLOMBIA  
INSTITUTO TECNOLÓGICO DE MONTERREY - MÉXICO  
MAESTRÍA EN CIENCIAS COMPUTACIONALES  
2005**

*A Evelyn Dayanna, Cinthya  
Melissa y Diana Ruth; son mi  
fuerza; por su apoyo, amor  
incondicional y comprensión  
permanente.*

*A mi madre por sus detalles,  
cariño y apoyo; a mi padre por su  
mano amiga en todo momento; a  
mis hermanas por su cariño y  
apoyo desde siempre y en todo  
momento.*

*A mi compañero de tesis porque  
con su ánimo y aportes hizo  
posible este logro mutuo.*

Evelio

*A Dios por darme la vida, la fortaleza y la paciencia necesaria para alcanzar esta meta.*

*A Sonia Patricia, Eybar Arley, Guillermo Andrés, y Carlos Fernando por ser mi compañía permanente en la vida y por su comprensión ante las largas jornadas de estudio.*

*A mi hijo Heller Guillermo por el aporte inicial que me brindó.*

*A mi compañero de tesis Jesús Evelio Ortega Arévalo quien con su dedicación y capacidad hizo posible este logro.*

*Guillermo*

## AGRADECIMIENTOS

Los autores expresan sus más sinceros agradecimientos a:

Doctor Eduardo Carrillo Zambrano, Director del Proyecto. Por haber gestado y modelado la idea inicial y por su acompañamiento permanente, generación de ideas, constante actualización de información y guía permanente en los derroteros óptimos del trabajo.

Al Doctor Hong Ong por haber sugerido una idea de proyecto de investigación que al final se formalizó y por sus valiosos aportes. El Doctor Ong está afiliado a Distributed Systems Group (DSG) University of Portsmouth, UK.

## CONTENIDO

	<b>pág.</b>
INTRODUCCIÓN	
1. GENERALIDADES .....	1
1.1 DEFINICIÓN DE SERVICIOS WEB MÓVILES .....	1
1.2 PRESENTACIÓN DEL PROBLEMA.....	2
1.3 OBJETIVOS .....	3
1.3.1 General .....	3
1.3.2 Específicos .....	3
1.4 METODOLOGÍA DE TRABAJO .....	4
1.5 ORGANIZACIÓN DEL DOCUMENTO .....	8
2. ESTADO DEL ARTE.....	10
2.1 INTRODUCCIÓN.....	10
2.1.1 Esquema general de los servicios web móviles .....	11
2.2 ORGANIZACIONES INTERNACIONES QUE ACOMPAÑAN LOS SERVICIOS MÓVILES.....	16
2.2.1 OMA (Open Mobile Alliance) [9] .....	16
2.2.2 W3C (World Wide Web Consortium) [6] .....	17
2.2.3 WS-I (Web Services Interoperability Organization) [10].....	17
2.2.4 OASIS (Organization for the Advancement of Structured Information Standards) [11].....	18
2.2.5 Resumen de las organizaciones que acompañan los Servicios Web Móviles.....	18
2.3 DISPOSITIVOS MÓVILES EN LA ACTUALIDAD.....	19
2.3.1 Teléfonos móviles o teléfonos celulares .....	20
2.3.2 SmartPhones .....	22
2.3.3 PDAs .....	25

	<b>pág.</b>
2.4 FORMAS DE CONECTIVIDAD .....	29
2.4.1 Infrarrojo [22].....	30
2.4.2 WLAN 802.11b [23].....	31
2.4.3 Bluetooth [24].....	32
2.4.4 GPRS [25].....	37
2.5 AMBIENTES DE DESARROLLO .....	39
2.5.1 Sistemas operativos.....	39
2.5.2 Herramientas de desarrollar .....	41
3. DISEÑO DE LAS PRUEBAS EN SERVICIOS WEB MÓVILES .....	46
3.1 VISTA GRÁFICA DEL BANCO DE PRUEBAS.....	46
3.2 DETALLES DEL BANCO DE PRUEBAS.....	46
3.2.1 Hardware y Software de servicios .....	46
3.2.2 Elementos clientes .....	48
3.2.3 Elementos para el desarrollo .....	49
3.3 DESARROLLO DEL BANCO DE PRUEBAS.....	50
3.3.1 Creación de escenarios de conectividad .....	50
3.3.2 Ideas para la medición de uso de conectividad.....	50
3.3.3 Compresión de la información desde los extremos.....	50
3.3.4 Publicadores de Servicios Web usados.....	50
3.3.5 Simuladores de concurrencia .....	51
3.4 NARRACIÓN DEL DETALLE DE PRUEBAS .....	51
3.4.1 La primera prueba, tamaño en bytes de la información enviada y recibida.....	51
3.5 Dificultades .....	52
3.5.1 Desarrollo de clientes de servicios web para palm .....	52

	<b>pág.</b>
4. PARÁMETROS DE MEDICIÓN .....	53
4.1 INTRODUCCIÓN .....	53
4.2 PARÁMETROS DE MEDICIÓN .....	53
4.2.1 El procesamiento.....	53
4.2.2 Consumo de ancho de banda.....	54
4.2.3 Memoria y almacenamiento.....	55
5. TUTORIAL DE PROGRAMACIÓN EN SERVICIOS WEB MÓVILES .....	59
5.1 INTRODUCCIÓN .....	59
5.2 PALM [12].....	59
5.2.1 prc-tools [14] .....	59
5.2.2 PODS (Palm OS Developer Suite) [12].....	68
5.3 Pocket PC [13].....	75
5.3.1 Introducción a los Web Services con PocketSOAP, Apache SOAP y Axis [34].....	75
5.4 SERVICIOS WEB EN EL LADO DEL SERVIDOR.....	89
5.4.1 Axis Java [7].....	89
5.4.2 Axis C++ [7].....	89
5.4.3 .NET Framework [8] .....	89
6. DISCUSIÓN Y ANÁLISIS DE RESULTADOS .....	90
6.1 INTRODUCCIÓN .....	90
6.2 CANTIDAD DE BYTES TRANSMITIDOS EN AMBOS SENTIDOS (ANCHO DE BANDA)90	
6.2.1 .NET Framework en Windows .....	91
6.2.2 Linux .....	95
6.3 LA COMPRESIÓN Y EL USO DEL PROCESAMIENTO .....	100

	<b>pág.</b>
6.3.1 Microsoft .NET Framework .....	102
6.3.2 Linux .....	111
6.4 RESUMEN.....	114
6.4.1 Según resultados, vale la pena comprimir .....	114
6.4.2 Y qué se transporta en los Servicios Web .....	115
6.4.3 ¿Quién sufre más, el cliente o el servidor? .....	115
6.4.4 Situaciones ideales para habilitar la compresión HTTP: .....	116
6.4.5 Situaciones no ideales para habilitar la compresión HTTP .....	117
6.4.6 Otros recursos como la memoria RAM y el almacenamiento .....	118
6.5 ESCANEADO DE PAQUETES .....	120
7. CONCLUSIONES .....	121
8. TRABAJO FUTURO.....	123
BIBLIOGRAFÍA.....	124
ANEXOS.....	127

## LISTA DE ANEXOS

	<b>pág.</b>
Anexo A. Consideraciones de servicios web con microsoft [42] .....	128
Anexo B. Descripción básica de global assembly cache (GAC) .....	129
Anexo C. Ethereal - Captura de paquetes [36] .....	131
Anexo D. Habilitar compresión http en .NET [44] .....	132
Anexo E. Publicaciones .....	140

## LISTA DE TABLAS

	<b>pág.</b>
Tabla 1. Características del celular Motorola C332 .....	21
Tabla 2. Características actuales en un Pocket PC .....	29
Tabla 3. Principales estándares WLAN.....	31
Tabla 4. Características de los ejemplos de Clientes de Servicios Web para PDAs .....	51
Tabla 5. Capacidad en memoria RAM en dispositivos Palm actuales .....	57
Tabla 6. Capacidad en memoria RAM en dispositivos Pocket PC actuales.....	57
Tabla 7. Enlaces hacia las especificaciones SOAP 1.1 y SOAP 1.2.....	76
Tabla 8. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web MathService .....	95
Tabla 9. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web Factorial ServiceC.....	95
Tabla 10. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web Calculator en Linux con Axis .....	100
Tabla 11. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web Factorial en Linux con Axis .....	100
Tabla 12. Tareas en la compresión de HTTP y de Archivos y Carpetas.....	101
Tabla 13. Resultados de comprimir XML en el servidor Windows con gzip con una sola petición.....	103
Tabla 14. Resultados de comprimir SOAP XML en el servidor Windows con gzip y 10 peticiones.....	104
Tabla 15. Resultados de comprimir SOAP XML en el servidor Windows con gzip y 100 peticiones .....	104
Tabla 16. Resultados de la prueba de carga con ANTS al Servicio Web MathService en Microsoft Windows.....	106
Tabla 17. Tiempo para conectarse el cliente al Servicio Web Math Service en prueba ANT .....	107

	<b>pág.</b>
Tabla 18. Tiempo para que el cliente reciba el primer byte del Servicio Web Math Service en prueba ANT.....	108
Tabla 19. Tiempo para que el cliente reciba el último byte del Servicio Web Math Service en prueba ANT.....	109
Tabla 20. Número de bytes enviados al cliente desde el Servicio Web Math Service en prueba ANT .....	110
Tabla 21. Resultados de comprimir XML en el servidor con XMLPPM con una sólo petición en Linux Axis .....	111
Tabla 22. Resultados de comprimir XML en el servidor con XMLPPM con una 10 peticiones en Linux Axis .....	112
Tabla 23. Resultados de comprimir XML en el servidor con XMLPPM con 100 peticiones en Linux Axis .....	112
Tabla 24. Comparación entre gzip y XMLPPM en Linux con respecto a compresión .....	113
Tabla 25. Comparación del consumo de procesamiento entre XMLPPM y gzip en Linux para un archivo de 10000K.....	114
Tabla 26. Recursos y demandas con la compresión HTTP habilitada .....	118

## LISTA DE FIGURAS

	<b>pág.</b>
Figura 1. Los estándares de Servicios Web en acción .....	12
Figura 2. Arquitectura de conectividad general de los Servicios Web Móviles.....	13
Figura 3. Transportes utilizables en Servicios Web Móviles.....	13
Figura 4. Diferentes modelos alternativos a Services Web Móviles en soluciones IT móviles .....	15
Figura 5. Conexiones tradicionales con Bluetooth (Ámbito de trabajo).....	33
Figura 6. Definición de canal Bluetooth.....	35
Figura 7. Sincronización del equipo máster con un cliente. ....	36
Figura 8. Definición del paquete Bluetooth.....	36
Figura 9. Evolución de los estándares Wireless globales.....	39
Figura 10. Vista gráfica del Banco de Pruebas del proyecto .....	46
Figura 11. Arquitecta de los Servicios Web Móviles.....	55
Figura 12. Esquema de trabajo para programación en Palm.....	64
Figura 13. La ventana de tipo de nuevo proyecto en PODS.....	69
Figura 14. Ventana para definición del tipo de salida en un nuevo proyecto PODS.....	70
Figura 15. Ventana para la selección de plantilla en un nuevo proyecto PODS.....	71
Figura 16. La ventana de administración del proyecto en PODS.....	72
Figura 17. Esquema simplificado de los Servicios Web .....	75
Figura 18. Modo de transmitir mensajes en SOAP.....	77
Figura 19. Envelope SOAP .....	77
Figura 20. Propiedad de compartir carpetas entre PC y Pocket PC .....	81
Figura 21. Opción de ver todos los archivos en Pocket PC .....	82
Figura 22. Opción copiar archivos en Pocket PC .....	82

	<b>pág.</b>
Figura 23. Opción de pegar archivos en Pocket PC .....	83
Figura 24. Opción Menú para ejecutar comandos en Pocket PC .....	83
Figura 25. Ventana para ejecutar comando en Pocket PC .....	84
Figura 26. Error cuando no se han registrado de forma correcta archivos dll en Pocket PC .....	84
Figura 27. Ejecución de regsvrce en Pocket PC .....	85
Figura 28. Mensaje de instalación correcta del archivo dll .....	85
Figura 29. Ventana de License Agreement en PocketSOAP .....	86
Figura 30. Creación de una nueva configuración para uso del Emulador Pocket PC 2003 en Microsoft eMbedded Visual C++ 4.0 .....	87
Figura 31. Adición de una configuración a un proyecto en Microsoft eMbedded Visual C++ 4.0 .....	88
Figura 32. Barra de resumen de configuración en Microsoft eMbedded Visual C++ 4.0 .....	88
Figura 33. Configuración de proyectos en Microsoft eMbedded Visual C++ 4.0 .....	88
Figura 34. Ejemplo getQuote en Pocket PC .....	89
Figura 35. Funcionamiento básico de los Servicios Web usando SOAP .....	91
Figura 36. Tiempo para conectarse el cliente al Servicio Web Math Service en prueba ANT .....	107
Figura 37. Tiempo para que el cliente reciba el primer byte del Servicio Web Math Service en prueba ANT .....	108
Figura 38. Tiempo para que el cliente reciba el último byte del Servicio Web Math Service en prueba ANT .....	109
Figura 39. Número de bytes enviados al cliente desde el Servicio Web Math Service en prueba ANT .....	110
Figura 40. Verificación del uso de HTTP 1.1 en Internet Explorer .....	132
Figura 41. La hoja de propiedades principales del servicio WWW .....	134

## LISTA DE IMÁGENES

	<b>pág.</b>
Imagen 1. Celular Motorola C332 .....	20
Imagen 2. Smartphone Nokia 7650.....	23
Imagen 3. Smartphone Nokia 3650.....	24
Imagen 4. Smartphone Sony-Ericsson P800.....	24
Imagen 5. Smartphone Nokia 9210.....	25
Imagen 6. PDA Palm .....	25
Imagen 7. Una Tungsten actual .....	27
Imagen 8. PDA Pocket PC.....	28

## LISTA DE CÓDIGOS

	<b>pág.</b>
Código 1. Un archivo Makefile ejemplo para prc-tools con cygwin .....	67
Código 2. Ejemplo de referencia a sección en prc-tools para segmentación de código ...	73
Código 3. Datos enviados para una solicitud a un Servicio Web en el ejemplo de getQuote.....	78
Código 4. Datos recibidos para una solicitud a un Servicio Web en el ejemplo de getQuote.....	78
Código 5. Código en ASP para el ejemplo de MathService - Servicio Web de calculadora sencilla .....	92
Código 6. Código en ASP para el ejemplo de Factorial - Servicio Web de cálculo de factorial.....	92
Código 7. Contenido SOAP para el ejemplo básico de la calculadora en la operación Multiplicación en ASP.NET .....	93
Código 8. Contenido SOAP para el ejemplo básico de FactorialServiceC en ASP.NET ...	94
Código 9. Código en C para el ejemplo de Calculator - Servicio Web de calculadora sencilla en Linux .....	96
Código 10. Código en C para el ejemplo de Factorial - Servicio Web de cálculo de factorial en Linux.....	97
Código 11. Contenido SOAP para el ejemplo básico de la calculadora en la operación Multiplicación en Linux con Axis .....	98
Código 12. Contenido SOAP para el ejemplo básico de Factorial en Linux con Axis.....	99

## 1. GENERALIDADES

### 1.1 DEFINICIÓN DE SERVICIOS WEB MÓVILES

En la actualidad hay una tendencia hacia usar los denominados dispositivos móviles para acceder a información en la red. Estos dispositivos móviles están representados principalmente por los denominados PDAs (Personal Digital Assistant).

A los Servicios Web que por medio de “aplicaciones móviles” se accede desde un dispositivo móvil se les denominan Servicios Web Móviles. Esta nueva tendencia basa su popularidad y su creciente demanda en algo muy sencillo y que siempre busca el usuario, la disponibilidad de la información desde cualquier sitio en cualquier momento.

Se trata entonces de un nuevo escenario, dos pasos adelante de las aplicaciones web iniciales que por medio de un navegador estándar obtienen la información. Entre otros usos, los Servicios Web Móviles se pueden implementar en aplicaciones empresariales que permiten a los trabajadores móviles hacer uso de información corporativa. El esquema facilita que debido a no tener los mismos recursos en conectividad ni en computación que los equipos que convencionalmente se conectan a la red LAN de la corporación, los empleados “móviles” que se conectan por medio de dispositivos móviles como PDAs, usen sólo los servicios (la información) que les interesa de las bases de datos corporativas. Entran a jugar papel preponderante las aplicaciones móviles que son desarrolladas para un sistema operativo específico, para un dispositivo específico, para obtener servicios de un Servicio Web específico. Sería impensable instalar un cliente normal de la aplicación corporativa en estos dispositivos, por eso lo importante de la oportunidad de que al implementar clientes de Servicios Web Móviles sólo se obtenga lo necesario.

En cuanto a conectividad la tendencia para estos dispositivos son las WWAN, las WAN inalámbricas de las cuales la más representativa es la GPRS [1] que se trata de conmutación de paquetes sobre la ya implementada GSM [2]. De esta forma, aprovechando la conectividad existente para telefonía móvil, y que se usa para servicios de proveedores de este tipo y en dispositivos convencionales de telefonía móvil, se logra implementar principalmente el Internet y ofrecer nuevas posibilidades a usuarios y proveedores. Actualmente un usuario en un dispositivo móvil puede adquirir anchos de banda similares a las tradicionales líneas fijas conmutadas y con 3G (EDGE) [2] un usuario puede lograr en su dispositivo móvil un ancho de banda alrededor de los 340 Kbps.

Aunque es una forma de aprovechar mejor los recursos, esto representa nuevos retos en dos grandes áreas:

1. La disponibilidad de recursos
2. La seguridad

La gran mayoría de administradores de sistemas de empresas en el mundo tiene un gran reto con respecto a la seguridad. Ya no se trata de asegurar la información dentro de un ámbito local (la LAN), tema en lo que ya se posee experiencia y en el cual existen estándares comprobados para un buen esquema, sino que se trata de asegurar la información no sólo en ambientes locales inalámbricos como Bluetooth o WLAN, sino en ambientes donde el área de cobertura es mucho mayor, como en las denominadas WWAN representadas principalmente por la tecnología GPRS, datos sobre GSM.

Además de la seguridad típica que requieren los Servicios Web, este nuevo esquema requiere características adicionales no sólo tecnológica, sino administrativamente por parte de los que manejan los sistemas de una empresa. Al mencionar la seguridad se desea recalcar que para lograr un óptimo nivel de seguridad se hace un mayor uso de los recursos críticos en materia de Servicios Web Móviles, tales como la conectividad y las capacidades del dispositivo servidor y del dispositivo cliente. En cuanto a conectividad es necesario transportar de extremo a extremo del servicio una cantidad de bytes que representan datos concernientes a seguridad, mientras que estos datos de seguridad en los que se basan los conceptos de Autenticación y Autorización, Confidencialidad e Integridad, representan costos en recursos de los equipos extremos como procesamiento y memoria RAM (principalmente procesamiento). Vale aclarar que el análisis de este consumo por parte de la implementación de la seguridad no está dentro del alcance del presente trabajo, aunque se plantea como trabajo futuro.

## **1.2 PRESENTACIÓN DEL PROBLEMA**

En los Servicios Web, que se pueden considerar como una tendencia relativamente nueva, los recursos de computación en ambos extremos de la comunicación no son tan críticos pues se desarrollan sobre la tendencia anterior, aquella en la cual el usuario accede a la información usando un navegador estándar, usando servidores de buena capacidad y clientes PCs como los que se conocen en la actualidad.

Con respecto a la conectividad, se usa un esquema muy similar al tradicional, aquel en el que el cliente navegador solicita servicios y recibe servicios de un servidor Web, sólo que en este caso, los extremos varían, siendo el consumidor un cliente de un Servicio Web específico y el servidor, un Servicio Web. Es decir, se usan las tecnologías tradicionales de conectividad lo que permite en cierta forma un amplio rango de posibilidades de conexión y de buen ancho de banda.

Ahora, los Servicios Web para dispositivos móviles (generalmente PDAs) denominados Servicios Web Móviles, no tienen las mismas posibilidades que se tienen en la implementación de Servicios Web tradicionales y las limitaciones en cuanto a recursos computacionales así como en conectividad generan un caso interesante de estudio sobre la definición de parámetros de rendimiento generales a medir en estos servicios, a tendencias actuales y sugeridas de mejoramiento en el rendimiento y al aprovechamiento óptimo del ancho de banda tan escaso en los dispositivos móviles.

Con respecto a trabajos anteriores, al menos en la Web en Colombia no existe un estudio de este tipo. Esto se puede deber a que es una tendencia bastante nueva pero que tal como se plantea y a la cobertura que ya tiene en otros países en donde el acceso a la tecnología es mayor, se ve como definitiva en los próximos años. Con respecto al ambiente internacional no hay un documento organizado en el cual se de un ESTADO

DEL ARTE apropiado de la tendencia, o que ofrezca mediciones y recomendaciones que aclaren algunas posibilidades para aprovechar el ancho de banda en los dispositivos móviles; el presente estudio lo abarca, explicando adicionalmente las posibilidades de desarrollo de Servicios Web y de clientes de Servicios Web.

Hasta donde se pudo llegar con la revisión bibliográfica no se encontraron trabajos que busquen los objetivos específicos del presente estudio TESIS DE MAESTRÍA.

Un trabajo encontrado [3], basa sus pruebas en un dispositivo Pocket PC, un tipo de PDA muy usado que ejecuta sistema operativo propietario de Microsoft, el WindowsCE. El trabajo en mención no detalla el desarrollo en el lado del cliente ni en el lado del servidor, no genera tutoriales, no genera un documento de estado del arte. Adicionalmente, la principal diferencia con el presente trabajo es que se centra en hacer mediciones de los tiempos de respuesta o de la cantidad de información que viaja por la red. Una similitud es que usa un algoritmo, o más bien una librería para comprimir información en Microsoft IIS, SharpZipLib [4]. El SharpZipLib logra implementar algoritmos estándar como gzip. En el presente trabajo se prefiere usar las características de compresión que por defecto trae .NET Framework, que usa directamente el algoritmo estándar gzip sin recurrir a ninguna librería. Adicionalmente en el presente trabajo se usa un algoritmo reconocido de compresión de XML, el XMLPPM [5], en el caso del sistema operativo tipo UNIX, aunque se puede implementar en Windows. El trabajo relacionado encontrado se centra en tecnologías tipo Windows, el presente trabajo tiene en cuenta tanto Microsoft Windows como tipos UNIX. Adicionalmente en el presente trabajo se usan los emuladores para dispositivos físicos Palm en las pruebas durante el desarrollo de clientes, igualmente se usa un dispositivo Palm IIIc. Para los mismos desarrollos y pruebas se usa un dispositivo real Pocket PC. Igualmente, se busca en el presente trabajo exponer un tutorial claro y conciso en algunas posibilidades de desarrollo de clientes de Servicios Web Móviles.

### 1.3 OBJETIVOS

**1.3.1 General.** Realizar un estudio de las consideraciones de rendimiento que se deben tener en cuenta en el desarrollo de Servicios Web Móviles, soportado en una investigación del estado del arte de dicha tecnología, en pruebas realizadas sobre Servicios Web Móviles existentes de dominio público y en la implementación de algunos Servicios Web Móviles, usando tecnologías muy difundidas tanto en el desarrollo como en el esquema de pruebas.

#### 1.3.2 Específicos.

- Obtener un documento del estado del arte en Servicios Web Móviles que permita obtener una visión clara de la actualidad, las diferentes tecnologías disponibles, las posibilidades de implementación y la posición de los Servicios Web Móviles frente a la arquitectura general de la Web.
- Obtener resultados cuantificados sobre los parámetros primordiales de rendimiento en Servicios Web Móviles de acuerdo a una serie de pruebas realizadas en un esquema de servicios conformado por dispositivos móviles simulados y reales y de ambientes de conexión creados.

- Desarrollar algunos Servicios Web y clientes móviles de esos Servicios Web usando las herramientas más apropiadas y generalizadas para presentarlos como modelos de prueba haciendo uso de emuladores y de dispositivos reales.
- Generar un tutorial muy claro acerca del desarrollo de Servicio Web Móviles usando las herramientas más apropiadas, tutorial que además será explicativo de los desarrollos hechos para realizar las pruebas.

#### 1.4 METODOLOGÍA DE TRABAJO

El trabajo fue netamente investigativo apoyado en abundante documentación encontrada en la Web.

Inicialmente se realizó una búsqueda de información que lograra dar claridad en la conceptualización de Servicios Web. Esta búsqueda se apoyó en muchos sitios en Internet, principalmente aquellos que generan estándares internacionales para la Web. Entre los más destacados se tiene [www.w3c.org](http://www.w3c.org) [6] y otros que se mencionan en la Bibliografía del trabajo y a los cuales se hace referencia a lo largo del presente documento.

El avance en las investigaciones permitió ir dando forma al documento ESTADO DEL ARTE, documento que forma parte de este trabajo final. Luego de obtener una adecuada base teórica en Servicios Web, se procedió a investigar acerca de las posibilidades de desarrollo de Servicios Web en el lado del servidor. Vale aclarar que la publicación de un Servicio Web se hace bajo el estándar WSDL, que apoyado en XML permite ofrecer el servicio con independencia de las características del cliente. Con esto, la investigación no debía entonces dirigirse a cómo desarrollar Servicios Web para dispositivos móviles, entendiéndose Servicios Web como el lado del servidor y no del cliente.

Durante este proceso, inicialmente se detallaron las posibilidades que se ofrecen para desarrollar Servicios Web inicialmente en Linux y posteriormente en Windows. En Linux, la organización Apache ofrece su proyecto AXIS [7], en Windows, Microsoft ofrece .NET Framework [8].

Estas dos posibilidades dan la opción de obtener mediciones en el lado del servidor, en dos ambientes diferentes, tipos UNIX (Linux) con Apache AXIS y Microsoft Windows con .NET Framework.

Luego de la implementación de unos Servicios Web de prueba, se procede a diseñar los parámetros de medición que se quieren probar en el lado del servidor. Se definieron cuatro: procesamiento, memoria RAM, lecturas/escrituras en el dispositivo de almacenamiento del equipo que actúa como servidor y la emisión y recepción de paquetes en las interfaces de red. Las conclusiones dejan ver que según los objetivos, conceptualización técnica, experiencia de campo de los autores del presente trabajo y las pruebas en sí, sólo el procesamiento se torna crítico.

A continuación se procedió a investigar sobre dispositivos móviles. Durante este proceso se descubrió que los dispositivos más usados eran los denominados PDAs (Personal Digital Assistant), pero que existía variedad y que se encuentran dispositivos adicionales

que entran dentro del concepto de dispositivos móviles como celulares, smartphones y tablets.

Se decidió entonces realizar el trabajo sobre los dispositivos más importantes, los PDAs que de alguna forma representan el mejor elemento en la actualidad para un trabajador móvil, ofreciendo funciones ofimáticas, de conectividad y de obtención de servicios en Internet de manera muy adecuada.

Hasta este momento la dificultad se encontraba en la selección, depuración y análisis de la abundante información. Esta dificultad se fue solventando con el tiempo, al ir aclarando las dudas con la copiosa y repetida lectura de los documentos encontrados.

El siguiente paso constituyó la formalización de la definición de Servicios Web Móviles, encontrando con sorpresa que no hay una formal, pero que las organizaciones internacionales involucradas en el proceso definen de manera tácita. Por eso, con apoyo de las organizaciones OMA: Open Mobile Alliance [9], W3C: World Wide Web Consortium [6], WS-I: Web Services Interoperability Organization [10] y OASIS: Organization for the Advancement of Structured Information Standards [11] se establece como "SERVICIOS WEB QUE SE USAN DESDE DISPOSITIVOS MÓVILES.

Al conocer que los PDAs más usados son las Palm y los Pocket PC, fue claro que en el sitio de cada proveedor se encontrarían los derroteros para conocerlos en detalle y conseguir las herramientas para desarrollar o probar. Efectivamente, en el sitio de Palm [12], se encuentran diversas herramientas, una de las cuales son los Emuladores que se consiguen tanto para las Palm como para las Pocket PC en el sitio de Microsoft para estos últimos [13].

Al conocer aspectos característicos de las PDAs se determinó que son dispositivos que presentan alguna escasez en recursos tales como procesamiento, memoria RAM, almacenamiento y conectividad. Con esto se logró de antemano determinar que la conectividad sería el factor predominante y el parámetro hacia el que básicamente se buscarían las recomendaciones de aprovechamiento en el uso del ancho de banda. Esto sin dejar de lado el procesamiento y el almacenamiento, este último una limitante bastante fuerte en el tema de desarrollo de aplicaciones o de clientes de Servicios Web como se explicará en el capítulo del tutorial.

Iniciando con cada PDA se estableció como forma de trabajo, primero el desarrollo de aplicaciones normales apoyado en tutoriales tanto para Palm como para Pocket PC. En ninguno de los dos casos se encontró dificultad, habida cuenta de la investigación preliminar que se había realizado con respecto a las formas de desarrollo y a la documentación en tutoriales.

Las posibilidades de desarrollo principalmente son tres: Java para los dos tipos PDA, desarrollos típicos Windows adaptados para Windows CE para las Pocket PC y C/C++ para las Palm.

A pesar de que el desarrollo en C/C++ en las Palm para aplicaciones sencillas no presenta mayor problema, en el desarrollo de aplicaciones que consuman Servicios Web en estos dispositivos es bastante dispendioso debido a que no existen documentaciones

muy claras ni tampoco muy abundantes que referencien dificultades. Queda en el desarrollador la responsabilidad de conseguir y absorber diversas recomendaciones.

Existía una herramienta denominada WSTKMD (Web Service Toolkit for Mobile Device), que hasta el año 2004 se conseguía en el sitio de Alphaworks, una filial de software de IBM. Esta herramienta, según documentación, permitía generar los denominados Stubs que son los headers para el cliente y también los Skeletons que son el encabezado que conforma el Servicio Web. Mediante la ejecución del programa `wsdlcpp2` teniendo como parámetro el archivo WSDL del Servicio Web, se obtenía un archivo con el mismo nombre del `wsdl` con la extensión `.h` que sería entonces el parámetro de entrada de la aplicación `soapcpp2` que finalmente generaba todos los archivos que deberían empaquetarse junto con la aplicación normal de Palm para formar finalmente un archivo de base de datos para Palm OS con extensión `prc`.

Realmente es en este punto en el que se establece el esquema en que desarrolla para Palm. Se definen entonces dos grandes opciones, una gratuita y una no gratuita. La gratuita es la opción `prc-tools` [14], la no gratuita es el CodeWarrior [15] que se afirma es la IDE para desarrollar en C/C++ más usada.

Existe una IDE gratuita, de excelente calidad, que ofrece la empresa Palm. Esta IDE se denomina Palm OS Developer Suite y se apoya en el paquete `prc-tools`. `prc-tools` es de tipo GNU.

Al desaparecer WSTKMD que se apoyaba totalmente en gSOAP, es precisamente gSOAP quien asume el soporte para el desarrollo de aplicaciones que consumen Servicios Web en C/C++. Pero, realmente existen grandes dificultades en el desarrollo de aplicaciones en C para Palm que consuman Servicios Web, debido a la escasa documentación y ejemplos existentes, al menos en el ambiente Internet.

Por citar un punto definitivo en lo que se menciona, sólo existen dos ejemplos en todo Internet, proveído por gSOAP para la implementación de una aplicación para Palm o Pocket PC que sea cliente de un Servicio Web. Los dispositivos Palm tienen una opción de desarrollo que se denomina 68K. Esto quiere decir que para archivos de mayor tamaño que 68K comienzan las dificultades. En el ejemplo mencionado existe un README en el que se explica que se entregan dos archivos para simular la subdivisión de un archivo grande, pero al examinar los archivos se concluye que son el mismo archivo original con líneas adicionales que subdividen el código objeto al momento de compilar. No existen explicaciones adicionales de cómo definir y unir los archivos que se subdividen.

Bien, estas dificultades en el desarrollo se deben subsanar con mucha investigación en la Web, y en foros de desarrollo en donde personas con dificultades similares tratan de conseguir información importante.

A continuación se procede a la consecución de dispositivos reales logrando obtener una Pocket PC y una PDA para la realización de las pruebas. La metodología más adecuada es implementar sobre los Emuladores para finalmente instalar en los PDAs reales.

Se determina una forma en que se puede medir el uso de procesamiento y de consumo de ancho de banda en ambos extremos. Para las Palm se obtienen aplicaciones que logran

medir el nivel de procesamiento, aunque con alguna dificultad por las propiedades de unitarea de los PDA, es decir que no se pueden ejecutar aplicaciones mientras otras estén ejecutándose u ocupando el screen, al menos para lograr la visualización. La opción en la Palm fue crear varias instancias del Emulador y ejecutar en una instancia las aplicaciones y en otra los medidores de procesamiento a los cuales se debería enviar la orden de ejecución por lo que se hizo necesario realizar numerosas pruebas. En los servidores se usan las herramientas estándar del sistema operativo. Por ejemplo, en Windows se utiliza la Herramienta del sistema Rendimiento, mientras que para la máquina con sistema operativo Linux, se usan comandos de monitoreo de rendimiento tales como vmstat, top, netstat y ps.

Uno de los recursos que se ha definido como críticos además del procesamiento en la Palm es la conectividad. Se establece que la conectividad a las WWAN es costosa y no abundante. Puede llegar a ser el elemento de mayor criticidad en este ambiente. Por tal motivo, se concluye que muchos de los análisis y recomendaciones irán dirigidos hacia esta parte.

Todo esto dio las bases para el diseño del banco de pruebas.

Con el avance en las mediciones se generaron bitácoras que permiten entonces ofrecer unos resultados de análisis sobre todo en tamaños de información a transportar entre los extremos y unas recomendaciones para aprovechamiento de recursos de conectividad.

Se procedió entonces a formalizar el documento final de tesis y a la organización de los elementos de software que formaron parte del desarrollo de la investigación.

A continuación una lista descriptiva de los pasos seguidos en el desarrollo del proyecto:

1. Concepción de la idea
2. Presentación del anteproyecto
3. Recepción de recomendaciones de evaluadores
4. Anteproyecto corregido
5. Investigación (Conceptualización, tecnologías en PDAs, elementos de desarrollo de Servicios Web, elementos de desarrollo de clientes de Servicios Web para PDAs)
  - 5.1 Estado del arte preliminar
  - 5.2 Conceptualización en Servicios Web Móviles
    - 5.2.1 Instalación de Apache Axis en Linux
    - 5.2.2 Instalación de .NET Framework en Windows
    - 5.2.3 Instalación de POSE (Palm Operation System Emulator) en Windows
    - 5.2.4 Instalación de Palm Desktop (Contiene el elemento HotSync)
    - 5.2.5 Instalación de WSTKMD
    - 5.2.6 Instalación de CYGWIN
    - 5.2.7 Instalación de PRC-TOOLS
    - 5.2.8 Instalación de PIRLC

- 5.2.9 Instalación de ANTS OPTIMIZ en Linux para simulación de concurrencia
- 5.2.10 Instalación de GATE de ERICSSON y ANITE SAT+ SYSTEM para simular ambientes de conectividad desde equipo Windows
- 5.3 ESTADO DEL ARTE mejorado
- 5.4 Prueba de clientes de Servicios Web en POSE
  - 5.4.1 Instalación de ejemplos WSTKMD (gSOAP) en POSE
  - 5.4.2 Prueba de clientes de Servicios Web en POSE
- 5.5 Instalación y desarrollo de aplicaciones para Palm
  - 5.5.1 Instalación y aprendizaje de aplicaciones elaboradas en C++ para Palm: hello.prc, forms.prc, menu.prc, prefs.prc, db.prc, ui.prc, i18n.prc, beam.prc, adicional sample code from Palm.com)
  - 5.5.2 Instalación y aprendizaje de clientes de Servicios Web para Palm en ejemplos de WSTKMD (babelfish, stockquote, interop)
- 6. Desarrollo de clientes de Servicios Web para Palm
  - 6.1 Aprendizaje y desarrollo con cygwin, prc-tools y pirlc
  - 6.2 Aprendizaje de gSOAP
  - 6.3 Aprendizaje y desarrollo de gSOAP on Palm
    - 6.3.1 Uso de Strikelron y otras herramientas para el descubrimiento de Servicios Web (wsdl)
    - 6.3.2 Uso de los componentes wsdlcpp y soapcpp2 para crear los stubs clients
  - 6.4 Desarrollo de clientes Servicios Web en C++ para Palm apoyados en ejemplos de wstkmd (gsoap2)
  - 6.5 Desarrollo de clientes para Pocket PC
- 7. Establecimiento de parámetros de medición
- 8. Monitoreo en ambos lados del servicio de los parámetros de medición
- 9. Elaboración de conclusiones
- 10. Elaboración de recomendaciones
- 11. Elaboración de documento preliminar de tesis y envío a Director
- 12. Afinamiento de documento de tesis según recomendaciones de Director

## 1.5 ORGANIZACIÓN DEL DOCUMENTO

En el CAPÍTULO UNO se describe la metodología usada desde el inicio hasta el final de la investigación. Se usa una forma narrativa para describir de la mejor manera posible los descubrimientos, dificultades y soluciones a las mismas.

El CAPÍTULO DOS está conformado por el documento de ESTADO DEL ARTE. Esta es una descripción en detalle de conceptos, tecnologías, fabricantes, esquemas de comunicación. Es un buen apoyo para la conceptualización en Servicios Web Móviles.

El CAPÍTULO TRES detalla la forma en que se realizaron las pruebas. El diseño del banco de pruebas, los elementos de hardware y software usados y los mecanismos para conectividad son parte de este ítem.

El CAPÍTULO CUATRO aclara los parámetros de medición definidos luego de diferentes conclusiones obtenidas durante la investigación acerca de los dispositivos que forman parte del trabajo, buscando el logro de los objetivos planteados para el trabajo de tesis.

El CAPÍTULO CINCO es el tutorial. Un tutorial para el desarrollo de Servicios Web Móviles. Es bastante interesante habida cuenta de las dificultades en el desarrollo y de la escasez de documentación concisa. El análisis de rendimiento es el elemento central, pero el tutorial es muy útil.

El CAPÍTULO SEIS es la discusión y análisis de resultados. Presenta entre otros temas abundantes tablas de resultados para comparaciones y conclusiones. Contiene análisis de resultados obtenidos.

El CAPÍTULO SIETE contiene la formalización definitiva de conclusiones tomadas de la experiencia obtenida en las pruebas y de la documentación general utilizada.

El CAPÍTULO OCHO presenta el trabajo futuro relacionando qué temas afines podrían aportar nuevas recomendaciones y conclusiones.

Posteriormente se presenta la Bibliografía.

Finalmente, se incluyen todos los anexos que complementan el trabajo desarrollado.

## 2. ESTADO DEL ARTE

### 2.1 INTRODUCCIÓN

En la actualidad hay una tendencia hacia usar los denominados dispositivos móviles para acceder a información en la red. Estos dispositivos móviles están representados principalmente por los denominados PDAs (Personal Digital Assistant).

A los Servicios Web que por medio de “aplicaciones móviles” se accede desde un dispositivo móvil se les denominan Servicios Web Móviles. Esta nueva tendencia basa su popularidad y su creciente demanda en algo muy sencillo y que siempre busca el usuario, la disponibilidad de la información desde cualquier sitio en cualquier momento.

Se trata entonces de un nuevo escenario, dos pasos adelante de las aplicaciones web iniciales que por medio de un navegador estándar obtienen la información. Entre otros usos, los Servicios Web Móviles se pueden implementar en aplicaciones empresariales que permiten a los trabajadores móviles hacer uso de información corporativa. El esquema facilita que debido a no tener los mismos recursos en conectividad ni en computación que los equipos que convencionalmente se conectan a la red LAN de la corporación, los empleados “móviles” que se conectan por medio de dispositivos móviles como PDAs, usen sólo los servicios (la información) que les interesa de las bases de datos corporativas. Entran a jugar papel preponderante las aplicaciones móviles que son desarrolladas para un sistema operativo específico, para un dispositivo específico, para obtener servicios de un Servicio Web específico. Sería impensable instalar un cliente normal de la aplicación corporativa en estos dispositivos, por eso lo importante de la oportunidad de que al implementar clientes de Servicios Web Móviles sólo se obtenga lo necesario.

En cuanto a conectividad la tendencia para estos dispositivos son las WWAN, las WAN inalámbricas de las cuales la más representativa es la GPRS [1] que se trata de conmutación de paquetes sobre la ya implementada GSM [2]. De esta forma, aprovechando la conectividad existente para telefonía móvil, y que se usa para servicios de proveedores de este tipo y en dispositivos convencionales de telefonía móvil, se logra implementar principalmente el Internet y ofrecer nuevas posibilidades a usuarios y proveedores. Actualmente un usuario en un dispositivo móvil puede adquirir anchos de banda similares a las tradicionales líneas fijas conmutadas y con 3G (EDGE) [2] un usuario puede lograr en su dispositivo móvil un ancho de banda alrededor de los 340 Kbps. En la sección **2.1.1 Esquema general de los servicios web móviles** se extiende el concepto y se trata la evolución y las diferentes redes inalámbricas de área amplia que existen actualmente.

Aunque es una forma de aprovechar mejor los recursos, esto representa nuevos retos en dos grandes áreas:

1. La disponibilidad de recursos
2. La seguridad [16]

La gran mayoría de administradores de sistemas de empresas en el mundo tiene un gran reto con respecto a la seguridad.

Ya no se trata de asegurar la información dentro de un ámbito local (la LAN), tema en lo que ya se posee experiencia y en el cual existen estándares comprobados para un buen esquema, sino que se trata de asegurar la información no sólo en ambientes locales inalámbricos como Bluetooth o WLAN, sino en ambientes donde el área de cobertura es mucho mayor, como en las denominadas WWAN representadas principalmente por la tecnología GPRS, datos sobre GMS.

Además de la seguridad típica que requieren los Servicios Web, este nuevo esquema requiere características adicionales no sólo tecnológica, sino administrativamente por parte de los que manejan los sistemas de una empresa. Al mencionar la seguridad se desea recalcar que para lograr un óptimo nivel de seguridad se hace un mayor uso de los recursos críticos en materia de Servicios Web Móviles, tales como la conectividad y las capacidades del dispositivo servidor y del dispositivo cliente.

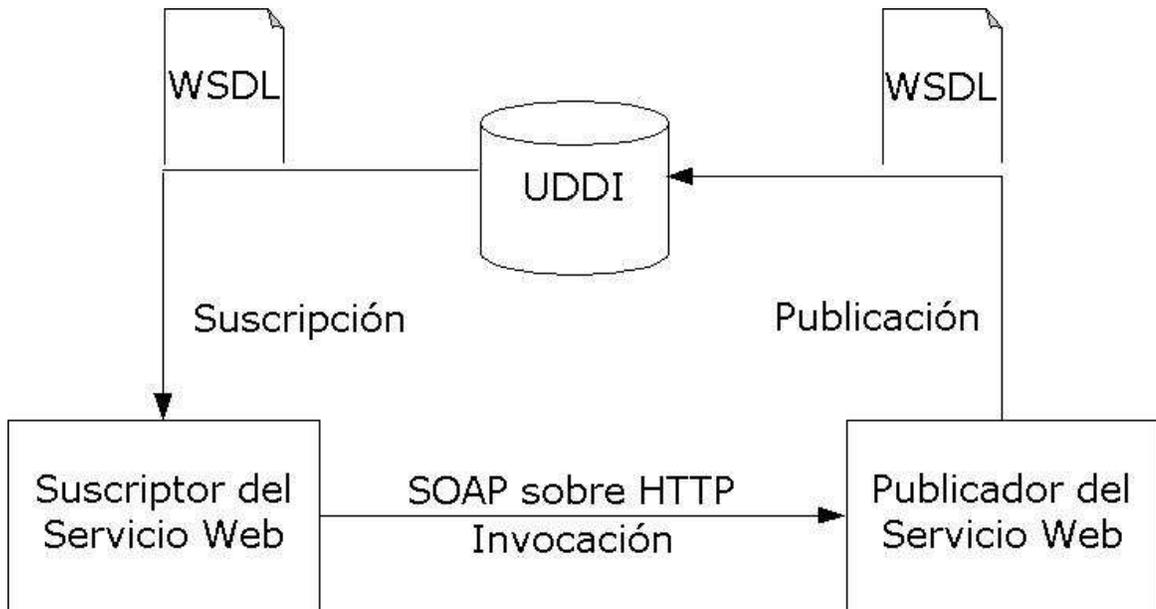
En cuanto a conectividad es necesario transportar de extremo a extremo del servicio una cantidad de bytes que representan datos concernientes a seguridad, mientras que estos datos de seguridad en los que se basan los conceptos de Autenticación y Autorización, Confidencialidad e Integridad, representan costos en recursos de los equipos extremos como procesamiento y memoria RAM (principalmente procesamiento). Vale aclarar que el análisis de este consumo por parte de la implementación de la seguridad no está dentro del alcance del presente trabajo, aunque se plantea como trabajo futuro.

Es algo que no se puede evitar. La seguridad debe existir pero teniendo en cuenta las limitaciones de conectividad y de recursos que posee el dispositivo que solicita el servicio, recursos que se tratan a fondo en el apartado **2.3 DISPOSITIVOS MÓVILES EN LA ACTUALIDAD**, es necesario tener una idea de qué tanto consume de estos recursos.

El documento ahora pasa a tratar el esquema general de los Servicios Web Móviles para tener una idea de los elementos que la componen y de la arquitectura del servicio.

**2.1.1 Esquema general de los servicios web móviles.** Inicialmente, la Figura 1 presenta el esquema de Servicios Web en general:

Figura 1. Los estándares de Servicios Web en acción



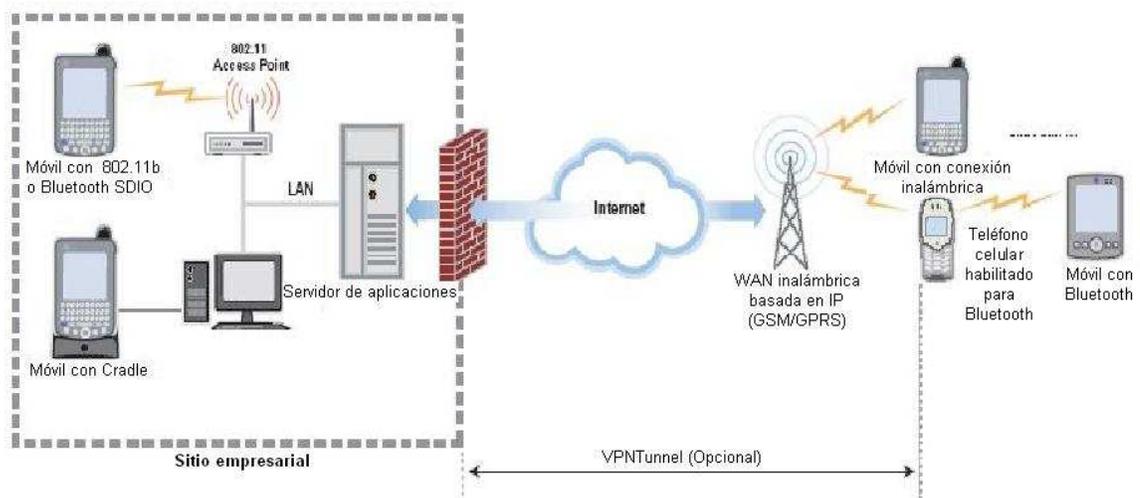
¿Cómo funciona esto?

1. El publicador del Servicio Web crea el Servicio Web
2. El publicador crea un documento WSDL y lo publica en un servidor UDDI [17]
3. El suscriptor descubre el servicio en UDDI
4. El suscriptor invoca el servicio usando SOAP
5. El suscriptor recibe la respuesta por medio de SOAP

Como se sabe WSDL [18][19] y SOAP [20] están todos basados en XML para permitir interoperabilidad, es decir, no importa hardware ni software en los extremos.

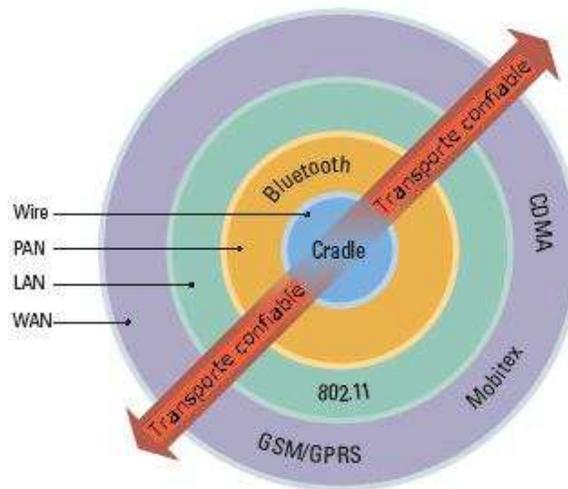
Ahora, en la siguiente gráfica se puede observar el esquema general de los Servicios Web Móviles, que implican las nuevas tecnologías inalámbricas de conexión en redes de área amplia y los dispositivos móviles con capacidad para conectarse a ellas y obtener servicios.

Figura 2. Arquitectura de conectividad general de los Servicios Web Móviles<sup>1</sup>



También el siguiente gráfico apoya el entendimiento de las formas de conexión posibles en el esquema de Servicios Web Móviles:

Figura 3. Transportes utilizables en Servicios Web Móviles<sup>2</sup>



<sup>1</sup> Tomado de [21]

<sup>2</sup> Ibidem

Aunque los Servicios Web Móviles es la arquitectura que mejor se adapta a las necesidades y a los esquemas tradicionales, existen otros tres modelos que se podrían utilizar. Los tres esquemas adicionales entonces para una empresa con sistemas de información móvil serían:

### **El modelo basado en navegador**

Se trata del tradicional esquema de tres niveles en el que un servidor http ofrece contenido ya sea dinámico o estático a un cliente con un navegador estándar como Internet Explorer o Netscape por mencionar dos muy conocidos. La gran dificultad en este esquema es que este modelo fue pensado para ambiente Desktop con un buen tamaño de pantalla, con teclado y con mouse. Además estilos como el HTML, JavaScript y otros dificultan también las posibilidades de trabajar este esquema sin contar los inconvenientes que por baja conectividad puede encontrar un cliente móvil.

### **Modelo basado en WAP**

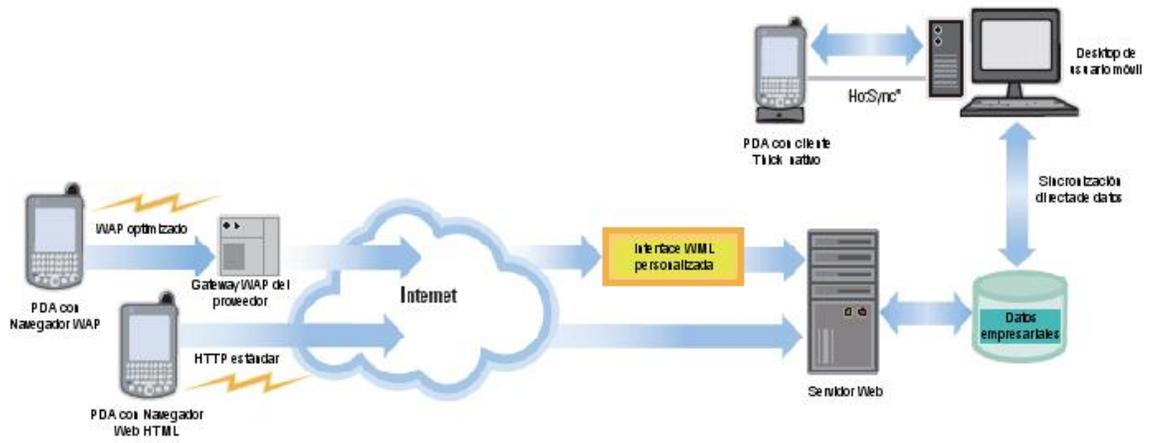
Este modelo por medio del Wireless Access Protocol aprovecha mucho mejor los ambientes inalámbricos pero como se limitan a menús textuales limitan bastante las posibilidades de crear una experiencia enriquecedora para el usuario, pues aunque usando el WML (Wireless Markup Language) se desaprovecharían características de los dispositivos móviles actuales.

### **Modelo Thick Client**

El Think Client es un cliente de aplicación corporativa que a menudo genera altos consumos en la PDA desaprovechando el dispositivo pues sólo usaría algunas de las funcionalidades de la aplicación. Es un modelo impuesto por los denominados Pocket PC que son los PDAs que contienen el sistema operativo propietario de Microsoft, el Windows CE. Como se puede entender, se trata de un sistema operativo que tiene muchas de las características del sistema operativo Windows para un Desktop, y aunque más liviano requiere de recursos altos para poder procesar los altos volúmenes de datos de este cliente. Es realmente una experiencia más agradable para el usuario pero esa experiencia puede tornarse desagradable al bajar severamente el performance de la máquina. Si vamos a mirar los dispositivos PDAs que en este momento se pueden conseguir, podemos ver que la memoria RAM es de 64 MB y el procesador puede llegar a Celeron de 400 Mhz. Estas configuraciones son en extremo pequeñas para un sistema operativo de Desktop, aunque el Windows CE es mucho más liviano y en estos puede haber un rendimiento mucho mejor, son dispositivos que no estarían generalizados y la mayoría de Pocket PCs tendrían configuraciones menores, por ejemplo procesamientos por menos de los 100 Mhz y memoria RAM por debajo de los 16 MB.

La siguiente figura ilustra los modelos descritos anteriormente:

**Figura 4** Diferentes modelos alternativos a Services Web Móviles en soluciones IT móviles<sup>3</sup>



Los siguientes párrafos ilustran las ventajas que los Servicios Web Móviles tiene sobre los modelos descritos anteriormente.

El middleware basado en Servicios Web ofrece una forma más abierta, más escalable de construir lógica de negocio a nivel de área amplia, permite aprovechar las características de los dispositivos móviles entre estas las escasas posibilidades de conexión, aunque se debe optimizar el acceso y aprovechamiento de los recursos como procesamiento, RAM y conectividad de los dispositivos cliente.

El primer protocolo de Servicios Web es el SOAP que se transporta sobre XML y que permite invocar procedimientos remotos, transportando código sofisticado. El XML se puede comprimir bastante y ser llevado a binario, y aunque cause consumo de recursos como el procesamiento en los extremos de la comunicación al ser comprimido, permite un aprovechamiento más óptimo de los recursos de conectividad que al fin y al cabo son escasos y costosos.

Existió un intento inicial parecido a Servicios Web (CORBA, JINI, DCOM) pero por la complejidad de desarrollo y estar ligado a soluciones propietarias de sistema operativos y otras funciones no tuvo mucho éxito.

Debido a que el servidor envía al cliente las funciones y la lógica del negocio y no la capa de presentación, esto aprovecha mucho mejor los recursos en los extremos, evitando además el manejo de volúmenes grandes de información en el cliente.

---

<sup>3</sup> Tomado de [21]

## 2.2 ORGANIZACIONES INTERNACIONES QUE ACOMPAÑAN LOS SERVICIOS MÓVILES

Existen varias organizaciones que trabajan en conjunto para garantizar características tan importantes como la interoperabilidad y la seguridad cuando de servicios móviles se trata. A continuación se describen las más importantes.

**2.2.1 OMA (Open Mobile Alliance) [9].** Es propicio mencionar que existe una organización que genera especificaciones por medio de grupos de trabajo para garantizar que la comunicación móvil de aplicaciones usando una red empresarial o el Internet sea posible. Esta organización se denomina OMA por Open Mobile Alliance.

Fue formada en Junio de 2002 por cerca de 200 compañías incluyendo los operadores móviles líderes a nivel mundial, proveedores de dispositivos y redes, compañías de información tecnológica y proveedores de contenidos y servicios.

Tiene 4 metas principales:

- Emitir especificaciones técnicas abiertas y de gran calidad basado en requerimientos del mercado que lleven a la modularidad, la extensibilidad, y la consistencia entre proveedores para reducir los esfuerzos de implementación de la industria.
- Asegurar que las especificaciones emitidas provean interoperabilidad entre diferentes dispositivos, geografías, proveedores de servicio, operadores y redes; facilitar interoperabilidad de las implementaciones de productos resultantes.
- Ser catalizador para la consolidación de actividades de estándares dentro de la industria de servicios de datos móviles; trabajando en conjunto con otras organizaciones de estándares existentes y de la industria para mejorar la interoperabilidad y decrementar los costos operacionales por todo lo que esto involucra.
- Proveer valor y beneficios a los miembros de OMA desde toda la cadena de valores incluyendo proveedores de contenido y de servicios, proveedores de tecnologías de información, operadores móviles y proveedores de wireless que ellos elijan para participar activamente en la organización.

Y muy importante conocer los principios de esta organización:

- Los productos y servicios se basan en protocolos e interfaces estándar abiertos y globales y no son dirigidos a tecnologías propietarias.
- La capa de aplicación es independiente de las capas inferiores (ejemplos: GSM, GPRS, EDGE, CDMA, UMTS)
- La arquitectura del framework y los habilitadores de servicio son independientes del Sistema Operativo (OS)
- Las aplicaciones y plataformas son interoperables, proveyendo geografía sin fronteras y garantizando ser intergeneracional.

**2.2.2 W3C (World Wide Web Consortium) [6].** Fue creado en Octubre de 1994 para llevar la World Wide Web a su máximo potencial desarrollando protocolos comunes que promuevan su evolución y aseguren su interoperabilidad. W3C tiene cerca de 350 organizaciones miembros de todo el mundo y ha tenido reconocimiento internacional por su contribución al crecimiento de la Web.

### **Metas del W3C**

- **Acceso Universal:** Hacer la Web accesible a todos promoviendo tecnologías que tomen en cuenta las vastas diferencias en cultura, lenguajes, educación, habilidad, recursos materiales, dispositivos de acceso y limitaciones físicas de los usuarios en todos los continentes;
- **Semántica Web:** Para desarrollar un ambiente de software que permita a cada usuario hacer el mejor uso de los recursos disponibles en la Web;
- **Web de Confianza:** Guiar el desarrollo de la Web con consideración cuidadosa de las nuevas y nacientes características legales, comerciales y sociales para esta tecnología.

### **Principios de diseño de la Web**

- **Interoperabilidad:** Las especificaciones para los lenguajes y los protocolos de la Web deben ser compatibles con otros y permitir cualquier hardware y software usado para acceder a la Web para trabajar juntos.
- **Evolución:** La Web debe ser capaz de acomodarse a tecnologías futuras. Principios de diseño tales como simplicidad, modularidad y extensibilidad incrementarán las oportunidades de que la Web trabaje con tecnologías emergentes tales como Servicios Web Móviles y televisión digital, así como otras que vengan.
- **Descentralización:** Es el principio más nuevo y el más difícil de aplicar. Para permitir que la Web “escale” a proporciones mundiales mientras resiste errores y caídas, la arquitectura (como el Internet) debe limitarse o eliminar dependencias de registros centrales.

**2.2.3 WS-I (Web Services Interoperability Organization) [10].** La Organización para la Interoperabilidad de Web Services es un esfuerzo legal de la industria abierta para promover la interoperabilidad de los Servicios Web a través de plataformas, aplicaciones y lenguajes de programación. La organización une una comunidad diversa para responder a clientes por necesidades proveyendo guías, prácticas recomendadas y recursos de soporte para desarrollar Servicios Web interoperables.

## **Sociedades**

Existe una relación estrecha y complementaria de WS-I con otras organizaciones que buscan el crecimiento de los Servicios Web, como W3C, IETF (Internet Engineering Task Force) y otros.

El perfil básico de WS-I define cuatro especificaciones que han sido desarrolladas o que desarrollan otras organizaciones. El perfil incluye XML Schema, SOAP, WSDL, y UDDI - adoptando trabajos realizados por W3C y por UDDI.org. A medida que se vayan desarrollando perfiles se irán tomando especificaciones de otras organizaciones a conveniencia.

Esta organización provee guías para soporte de implementación y recursos de testeo para cada uno de los perfiles que desarrolla. El W3C sirve como la principal fuente de especificaciones formando el núcleo de los perfiles WS-I, y su membresía es el consumidor de los materiales que WS-I produce.

WS-I está comprometido en crear fuertes relaciones y adoptar especificaciones desarrolladas por una amplia gama de organizaciones tales como la EITF, OAGI, OASIS, OMG, UDDI, W3C y muchas otras. Estas organizaciones cubren las necesidades del vasto rango base de comunidades de clientes. El plan de WS-I es involucrar estos grupos y trabajar juntos para suplir de la mejor forma las necesidades de los clientes.

**2.2.4 OASIS (Organization for the Advancement of Structured Information Standards) [11].** OASIS (Organización para el progreso de los estándares de información estructurada) es un consorcio internacional sin ánimo de lucro que busca el desarrollo, convergencia y adopción de estándares e-business. Según el About de su página produce más estándares de Servicios Web que cualquier otra organización dentro de los que se incluyen seguridad, e-business y esfuerzos de estandarización en el sector público y para mercados de aplicación específica. Se fundó en 1993 y tiene más de 3.000 miembros que representan más de 600 organizaciones y miembros individuales en 100 países.

El consorcio aloja dos de los portales más ampliamente respetados en estándares de XML y Servicios Web, CoverPages y XML.org.

## **SGML Open**

OASIS se fundó en 1993 bajo el nombre de SGML Open como un consorcio de vendedores y usuarios dedicados a desarrollar guías para la interoperabilidad entre productos que soportaran el Standard Generalized Markup Language (SGML). En 1998 cambió su nombre para reflejar el alcance del trabajo técnico, que incluye Extensible Markup Language (XML) y otros estándares relacionados.

**2.2.5 Resumen de las organizaciones que acompañan los Servicios Web Móviles.** Todas estas organizaciones tiene características especiales que buscan objetivos comunes como la interoperabilidad, la seguridad y la extensibilidad de los servicios en la Web.

Todos están bastante involucrados en las nuevas tendencias hacia los Servicios Web Móviles; por ejemplo el W3C tiene especificaciones con respecto a que los contenidos y

los servicios deben ser accedidos desde cualquier dispositivo en cualquier lugar por medio de un esfuerzo que se denomina Independencia de Dispositivo, lo cual sería muy extenso para tratar en este proyecto, además de no ser el objetivo del mismo, pero es muy interesante que se lea sobre esto en la página del W3C (<http://www.w3c.org>) en donde se habla principalmente del esfuerzo de la organización por lograr que el contenido y el estilo que se exponga debe ir de acuerdo a las capacidades del dispositivo y aclaran específicamente que esto se hace para permitir que esa nueva gama de equipos como los PDAs, Smartphones, televisión, en fin logren una experiencia satisfactoria en el acceso a la información.

Realmente no es posible decir que W3C trata directamente el tema de las aplicaciones móviles por tratarse de una iniciativa más que todo de fabricantes de hardware y software, pero ellos dan las bases para que la interoperabilidad, la seguridad y la extensibilidad sea un hecho por medio de sus grupos de trabajo, además de sus abundantes recomendaciones para la independencia de dispositivo.

Estas empresas que se dedican al desarrollo e implementación de las aplicaciones móviles basan sus desarrollos y sus herramientas en las especificaciones de estas organizaciones.

Es muy importante volver a mencionar que WS-I, tal y como se habló en el apartado propio de esta organización, se apoya para la creación de su perfil en organizaciones como W3C y UDDI para esquematizar idealmente una infraestructura que garantice básicamente la comunicación de los servicios a través de cualquier red, de cualquier hardware y de cualquier software independizando además las capas de aplicación y las inferiores. Esto es muy importante porque el que un dispositivo como un Pocket PC o un Smartphone o una Palm que poseen cada uno su sistema operativo y forma muy diferente de tratar las interfaces para el usuario no tengan que depender de esto para conectarse a una de las tecnologías Wireless WAN emergentes como GPRS o EDGE.

Es propicio entonces que este documento hable ya de los diferentes tipos de dispositivos móviles que se destacan en la actualidad, sus características, sus sistemas operativos, sus capacidades y limitaciones para hacer un preámbulo a lo que realmente interesa en el proyecto, establecer unos parámetros de medición de rendimiento de Servicios Web Móviles.

### **2.3 DISPOSITIVOS MÓVILES EN LA ACTUALIDAD**

Existen en la actualidad varios grandes grupos de dispositivos móviles. Acá los listamos:

- Los teléfonos móviles o más conocidos como teléfonos celulares
- Los Smartphones
- Los PDAs (Personal Digital Assitant)

Cada uno de ellos posee características bastante diferentes y aplicaciones o funciones muy propias. Pasemos a examinar ahora cada uno de ellos y se debe prestar especial atención en los dos últimos que son los que realmente pueden poseer las denominadas

aplicaciones móviles para hacer uso de los Servicios Web Móviles. Se les denomina en ocasiones handhelds por la propiedad que tienen de caber en la mano.

Los handhelds se usan mucho en sincronizar información entre dispositivos convencionales de computación más que como reemplazo de ellos. Mantienen información de listas de contactos, correos electrónicos y agenda y en la actualidad con el advenimiento de las aplicaciones móviles pueden también permitir que el trabajador haga su trabajo conectándose a las aplicaciones empresariales o a Servicios Web en cualquier momento en cualquier lugar.

Algunos sofisticados tienen procesadores de palabra y hojas de cálculo y las ya mencionadas aplicaciones móviles y los más actuales tiene un procesamiento que llega a un Celeron de 400 Mhz y a una memoria RAM de 64 MB.

**2.3.1 Teléfonos móviles o teléfonos celulares.** Estos son los tradicionales teléfonos que permiten llamar, enviar mensajes por medio del servicio SMS y por supuesto cumplen en algunas ocasiones con la capacidad de usar WAP por medio de lo cual desde hace algunos años permiten acceder a aplicaciones WAP por medio de Wireless Access Protocol que está optimizado para aprovechar las grandes limitantes de estos teléfonos pero que no permiten aprovechar al máximo las nuevas tendencias de aplicaciones móviles.

A continuación y como ejemplo la Imagen 1 y la Tabla 1 describen un dispositivo de este tipo con el máximo de funciones que puede soportar uno de estos teléfonos en la actualidad, el motorola C332:

**Imagen 1. Celular Motorola C332**



**Tabla 1. Características del celular Motorola C332**

<b>Características</b>	
Batería	En tiempo de conversación hasta 3 h. En tiempo de espera hasta 5 días.
Agenda	Con calendario y recordatorio para que no olvide sus citas.
Alerta vibratoria	Con esta modalidad su teléfono vibrará cuando reciba una llamada o un mensaje.
Calculadora	Con convertidor de divisas. Realiza rápidamente operaciones matemáticas. También podrá convertir fácilmente cantidades monetarias desde su teléfono.
Pantalla	96 x 64 pixels con 4 niveles de grises.
Texto predictivo	ITAP - Anticipa la palabra que está escribiendo, por lo que simplifica la escritura de mensajes 2 vías o de otros modos de edición.
Kit de herramientas de aplicaciones SIM	Menú de Servicios de Mensajes de 2 Vías que vienen incluidos en la Sim Card (chip).
Mensajes COMCEL 2 vías	Comuníquese a través de mensajes enviados directamente desde su teléfono. Además podrá disfrutar de todos los servicios de 2 vías.
<b>Características de la red</b>	
Banda	850 y 1900 MHz.
<b>Servicios de datos</b>	
CSD	'Circuit Switched Data'. Le permite transmitir datos por medio de una conexión móvil a una velocidad máxima de 9.6 Kbps. Utiliza canales de voz.
GPRS	"General Packet Radio Service". Le permite transmitir datos por medio de una conexión móvil. Reciba y realice llamadas sin perder la conexión.
HSCSD	'High Speed Circuit Switched Data. Esta tecnología le permite conexión en línea a Servicios de Internet a mayor velocidad que CSD para servicios de Conexión Móvil.
WAP	Este equipo cuenta con un micro navegador por lo que podrá utilizar el servicio de Conexión Móvil.

La limitante es por las capacidades del equipo ya que su función primordial es como máximo conectarse a una red de Internet y permitir la consulta del correo electrónico, pero además dependiendo del servicio WAP que esté ofreciendo una empresa también es posible realizar navegación no muy avanzada claro está.

Se habla mucho más bien de usar este dispositivo de teléfono celular como módem convencional para poder conectar el portátil o PC al Internet. Claro, si se tiene uno de los dos dispositivos que se mencionan a continuación, los Smartphones y los PDAs se tiene la posibilidad de realizar muchas de las labores que se realizarían con el portátil o PC sin necesidad de ellos. Para cumplir la función de módem el teléfono celular debe tener una conexión inalámbrica (IRDA) y la característica GPRS.

Para hacerse una idea de tarifas en nuestro medio, en Colombia, existen planes que oscilan en los \$45.000 + IVA, aclarando que los cobros por este tipo de conexión se realizan por volumen de información y no por tiempo de conexión. Esta tarifa es equivalente a trabajar un volumen de información de 10 MB lo cual se antoja poco para alguien que use bastante el servicio. Acá retomamos el inconveniente que hay con la limitante de recursos, por lo cual es importante retomar el tema de la compresión de datos.

Realmente si se desea más información acerca de esto se pueden visitar las páginas de los proveedores de telefonía celular en Colombia, a saber, <http://www.comcel.com>, <http://www.bellsouth.com.co> y <http://www.ola.com.co>.

**2.3.2 SmartPhones.** Son un nuevo estilo de handhelds que combina las posibilidades de un PDA y un teléfono móvil.

Aunque no posee en extremo las características de procesamiento, de memoria RAM o de aplicaciones que una PDA tiene la gran ventaja de que por defecto ofrece la posibilidad de conectarse al Internet, posibilidad que en la mayoría de las PDAs hasta el momento no se tiene pues ofrecen conectividad serial, por infrarrojos o por USB. Aunque algunos modelos en la actualidad ofrecen WLAN 802.11b, Bluetooth e incluso la posibilidad de usar funciones convencionales de telefonía celular incluyendo conexión por medio de GPRS. Por eso el auge de los Smartphone.

En los Smartphone predomina un sistema operativo en particular que se denomina Symbian OS, pero también el Windows CE por medio de unas personalizaciones a su shell original tiene su parte en este tipo de dispositivos.

Según una encuesta de mercado en EMEA (Europa, Medio Oriente y Asia) estos dispositivos están creciendo en popularidad y están incluso sobrepasando los márgenes de venta de los tradicionales PDAs (Palms y Pocket PCs).

Este sistema operativo posee las siguientes características importantes:

- **Telefonía multimodo integrada** - para voz y datos móviles 2G, 2.5G y 3G.
- **Mensajería** - con soporte para SMS, EMS, MMS, e-mail y fax.
- **Estándares abiertos** - IPv4 e IPv6, MMS, Bluetooth, GPRS, GSM, CDMA, Java, WAP y SyncML lo que asegura la interoperabilidad con una amplia gama de dispositivos y permite a los operadores producir excelentes servicios de datos.
- **Programable** - Estos teléfonos son abiertos y programables, lo que permite desarrollar aplicaciones y servicios. APIs estándares proveen acceso a funciones avanzadas como multimedia, comunicaciones y administración de información.
- **Robusto** - Permite uso bajo de recursos a programadores y garantiza integridad de datos aún con intermitencia de conectividad.
- **Internacionalización** - texto Unicode está integrado y tiene soporte para desarrollo en múltiples lenguajes.

Posee tres plataformas que se describen a continuación y que soportan un 80% de las características iniciales y difieren en un 20% en características en cada plataforma ofreciendo posibilidades de elección:

- Serie 60

**Imagen 2. Smartphone Nokia 7650**



**Imagen 3. Smartphone Nokia 3650**



Diseñado para el mercado en masa. Millones de usuarios desean aplicaciones simples. Navegación con un joystick, teclado pequeño numérico, teclas rápidas. Incluye Nokia 3650/7650/6600/N-Gage, Samsung SGH-D700 y Siemens SX1. Las Imágenes 2 y 3 ilustran esta serie.

- UIQ

**Imagen 4. Smartphone Sony-Ericsson P800**



Con abundante multimedia, telefonía y navegación multimedia. Interface para usuario basada en un bolígrafo. Incluye Sony-Ericsson P800 y BenQ P30. La Imagen 4 nos da un ejemplo gráfico.

- Serie 80

### Imagen 5. Smartphone Nokia 9210



Teclados completos y pantallas grandes. Usuarios corporativos que necesitan editar información y revisar datos de negocios mientras están fuera de la oficina. Incluye Nokia Communicator 9200 Series. En la Imagen 5 se puede observar uno de estos dispositivos.

Ahora se analiza el tercer tipo de dispositivo, los PDAs.

**2.3.3 PDAs.** Aunque existen varias marcas hay dos que se destacan por su gran calidad y por el mercado que tienen.

Inicialmente se analizan las Palm que son las PDAs que tiene el sistema operativo PalmOS de la empresa Palm, luego se sigue con los Pocket PC, otro tipo de PDA bastante difundido; para ver una comparación entre características y usabilidad se puede remitir a <http://myalpha.dls.net/palm/pda.htm>.

#### 2.3.3.1 Palm [12]

### Imagen 6. PDA Palm



Son los PDAs basados en el sistema operativo Palm OS, desarrollado por Palm Computing Inc. (filial de la prestigiosa 3Com). Desde hace varios años, son los líderes en el mercado de la informática de bolsillo, gracias a la extraordinaria calidad y versatilidad de sus productos. La Imagen6 es un ejemplo de una Palm.

La mayoría de los dispositivos son fabricados por la propia empresa Palm, aunque hay otros fabricantes (como Handspring con sus modelos Visor) basados en Palm OS, que también son llamados popularmente "Palms". Otros fabricantes como Sony también han desarrollado dispositivos Palm.

Para aprovechar información actualizada se toma de la página de la compañía.

Desde este sitio se pueden observar tres grandes líneas, haciendo resaltar el que esta empresa ya presenta los dispositivos Smartphone también y no se especializa sólo en las PDAs debido al gran auge de estos dispositivos.

Las tres grandes líneas son Zire, Tungsten y Treo, siendo el último la línea de Smartphone, pero también los más costosos encontrando que los más económicos son los Zire. En fin, todo depende de las características que se busquen en el dispositivo. Los que más cumplen con las necesidades para Servicios Web Móviles usando WWANs son los Treo, pues posee conectividad por medio de GSM/GPRS o CDMA.

Es propicio mencionar que en Colombia la máxima velocidad de conexión que se ofrece por medio del proveedor en la tecnología GSM/GPRS es de 170 Kbps, aunque a nivel mundial se habla ya de 384Kbps. Y si se usan tecnologías como CDMA, siendo su última versión la WCDMA se puede llegar a velocidad de 2 Mbps.

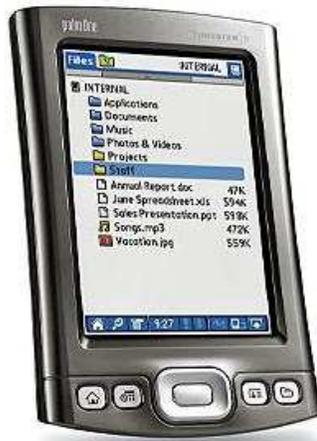
Los otros dispositivos dependiendo del modelo pueden ofrecer conectividad tipo Bluetooth y WLAN 802.11b, temas que se tratan en la siguiente sección para dar una idea clara de las posibilidades inalámbricas que se ofrecen actualmente.

En cuanto a características de hardware y software para estos dispositivos, podemos resumir lo siguiente:

La mejor de las Zire vale cerca de 299 dólares americanos y ofrece en procesamiento 312Mhz Intel® PXA270 processor (ARM-based), memoria RAM de 16 MB, puerto USB y conectividad Bluetooth, muchas aplicaciones personales y multimedia. El sistema operativo es el PalmOS v5.2.8.

La mejor de las Tungsten teniendo en cuenta que ofrece conectividad WLAN 802.11b es la Tungsten C con procesamiento 400MHz Intel® PXA255 processor with Intel® XScale technology, memoria RAM de 64 MB, PALMOS v5.2.1 y excelentes aplicaciones de oficina y de conectividad. Vale unos 399 dólares americanos. Una Tungsten se puede observar en la Imagen 7.

**Imagen 7. Una Tungsten actual**



Finalmente la Treo, la más costosa pues es prácticamente la unión de un PDA con un teléfono celular (un Smartphone); en algún detalle están son sus propiedades:

La mejor es la Treo 600 y tiene dos modelos, la que se conecta con GSM/GPRS y la que se conecta con CDMA. Tiene un procesador 144 Mhz ARM, memoria RAM a 32 MB, PalmOS v5.2.1H, comunicación Infrarrojo y numerosas aplicaciones funcionales y de Internet. Posee un pequeño teclado muy útil. Vale cerca de 500 dólares americanos.

Ahora se analizan los segundos PDAs muy difundidos, los Pocket PC.

**2.3.3.2 Pocket PC [13].** Es un PDA basado en el sistema operativo Windows CE con un grupo de aplicaciones personalizadas escritas por Microsoft pero vendidos por empresas como Hewlett-Packard, Toshiba, ViewSonic y otros.

#### **Es un teléfono o un PDA o ambos?**

Un Pocket PC hace más pequeña la línea divisoria entre un Personal Digital Assistant (PDA) y un teléfono celular. Hay dispositivos que sólo son PDAs, otros que son como PDAs con comunicación celular integrada, y teléfonos celulares que corren Windows CE. Entonces cuál es cuál y qué tanto tiene en común?

El dispositivo Pocket PC 2003 es una evolución del dispositivo Pocket PC realizado en Abril de 2000. El sistema operativo es ahora Windows CE 4.2, en vez de Windows CE 3.0, y han sido adheridas varias funciones para soportar características de usuario como interfaces shell de notificaciones, pero con todo esto es el mismo sistema a pesar de que tiene un sistema operativo mucho más robusto. Lo que más se nota es el nuevo y potente hardware que se ha implementado en los últimos años. Los dispositivos actuales pueden tener integrada conectividad inalámbrica o redes de área personal Bluetooth.

Otro avance ha sido la integración de tecnología celular dentro de algunos dispositivos Pocket PC. Microsoft lanzó Pocket PC Phone Edition que adiciona soporte para llamadas

de voz, SMS y otras características específicas para poder integrar un teléfono en el dispositivo.

La interfaz de usuario y los requerimientos del Phone Edition son casi las mismas que para el Pocket PC. Tiene la misma forma de pantalla y carece de teclado. Difiere del dispositivo estándar en que se pueden realizar llamadas telefónicas teniendo micrófono y parlante integrado.

Unos dispositivos recién introducidos son los Smartphones. Es un teléfono celular que se basa en Windows CE con un shell personalizado y aplicaciones personalizadas. Aparte del shell, que es muy diferente al del Pocket PC debido a las limitaciones y tipo de uso de un teléfono celular, el Smartphone es sorprendentemente similar al Pocket PC en el fondo.

Aparte de la economía que puede surgir de una solución común, el mismo grupo dentro de Microsoft es responsable por ambas plataformas, lo que provee un esfuerzo burocrático por unir las dos plataformas. La siguiente Imagen presenta un Pocket PC típico y la Tabla 2 detalla características estándares actuales de uno de estos dispositivos..

**Imagen 8. PDA Pocket PC**



**Tabla 2. Características actuales en un Pocket PC**

<p style="text-align: center;"><b>Operating System</b> Windows Mobile 2003 Second Edition</p>
<p style="text-align: center;"><b>Processor Type</b> Intel XScale Processor</p>
<p style="text-align: center;"><b>Processor Speed</b> WMMX up to 624MHz (also available in 312 MHz version)</p>
<p style="text-align: center;"><b>Memory</b> up to 64MB SD RAM up to 64MB</p>
<p style="text-align: center;"><b>Display</b> Type: 3.5" TFT Color 16-bit, Touch Sensitive, Transflective LCD, Number of Colors: 65,536 colors, Display Resolution: 320 x 240</p>
<p style="text-align: center;"><b>Dimensions</b> 117mm (L) x 77.2mm (W) x 14.9mm (H)</p>
<p style="text-align: center;"><b>Weight</b> 134.4g</p>
<p style="text-align: center;"><b>Battery</b> N/A, Type: 950 mAh Lithium-Ion Removable, Rechargeable battery</p>
<p style="text-align: center;"><b>Expansion</b> Built In: Integrated 802.11b and Bluetooth wireless technologies; Integrated Secure Digital / SDIO Now! / MMC card slot provides flexible expansion</p>
<p style="text-align: center;"><b>Synchronization Options</b> Included in Box: USB Cradle or USB Travel Sync Cable (depending on your order)</p>

## 2.4 FORMAS DE CONECTIVIDAD

Como se ha visto, los medio para conectarse que posee un sistema móvil son cuatro en la actualidad. Tres de estos, IrDA, WLAN y Bluetooth permiten que dispositivos diferentes a los HANDLEDS (PCs o portátiles) se tomen dentro de los que serían dispositivos móviles, claro, teniendo en cuenta que no tendrían la universalidad de conexión que se pretende con tecnología como GPRS que permite que dispositivos como PDAs y Smartphones se conecten desde cualquier sitio a servicios de red principalmente basados en TCP/IP.

### 2.4.1 Infrarrojo [22]

#### IRDA

La asociación de datos por infrarrojos (*Infrared Data Association, IrDA*) es una organización patrocinada por la industria y establecida en 1993 para crear estándares internacionales para el equipo y programas usados en los enlaces de comunicación por infrarrojos.

En esta forma especial de transmisión de radio, un haz enfocado de luz en el espectro de frecuencia infrarrojo, medido en terahertz o billones de hertzios (ciclos por segundo) se modula con información y se envía de un transmisor a un receptor a una distancia relativamente corta. La radiación infrarroja (IR) es la misma tecnología usada para controlar un televisor con un mando a distancia.

La comunicación de datos por infrarrojos está ahora jugando un importante papel en las comunicaciones de datos inalámbricas debido a la popularidad de ordenadores portátiles, asistentes digitales personales (personal digital assistants, PDAs), cámaras digitales, teléfonos móviles, buscapersonas y otros dispositivos. Entre los usos existentes o posibilidades razonables están:

- Enviar un documento de nuestro ordenador portátil a una impresora.
- Intercambiar tarjetas de visita entre PCs manuales.
- Coordinar agendas y libretas telefónicas entre nuestros ordenadores de escritorio y portátiles.
- Enviar faxes desde nuestro ordenador portátil a una máquina de fax distante usando un teléfono público.
- Cámaras digitales que pueden enviar las imágenes a nuestro ordenador.

La comunicación infrarroja involucra un transcéptor (una combinación de transmisor y receptor) en los dos dispositivos que se comunican. Hay microchips especiales que proporcionan esta capacidad. Adicionalmente, uno o ambos dispositivos pueden requerir software especial para que la comunicación pueda sincronizarse. Un ejemplo es el soporte especial para IR en el sistema operativo Windows 95 de Microsoft. En el estándar IrDA-1.1, el máximo tamaño de datos que se pueden transmitir es de 2048 bytes y la tasa máxima de transmisión es de 4 Mbps.

La IR también puede usarse para interconexiones un tanto más largas y es una posibilidad para las interconexiones en redes de área local (LAN). La distancia efectiva máxima es algo menor a los ocho kilómetros y el máximo ancho de banda proyectado es de 16 megabits por segundo. Dado que la IR es transmisión en línea visual (ambos dispositivos deben poder "verse" entre sí), es sensible a la niebla y otras condiciones atmosféricas.

**2.4.2 WLAN 802.11b [23].** Los estándares son desarrollados por organismos reconocidos internacionalmente, tal es el caso de la IEEE (Institute of Electrical and Electronics Engineers) y la ETSI (European Telecommunications Standards Institute). Una vez desarrollados se convierten en la base de los fabricantes para desarrollar sus productos.

Entre los principales estándares se encuentran:

- **IEEE 802.11:** El estándar original de WLANs que soporta velocidades entre 1 y 2 Mbps.
- **IEEE 802.11a:** El estándar de alta velocidad que soporta velocidades de hasta 54 Mbps en la banda de 5 GHz.
- **IEEE 802.11b:** El estándar dominante de WLAN (conocido también como Wi-Fi) que soporta velocidades de hasta 11 Mbps en la banda de 2.4 GHz.
- **HiperLAN2:** Estándar que compite con IEEE 802.11a al soportar velocidades de hasta 54 Mbps en la banda de 5 GHz.
- **HomeRF:** Estándar que compite con el IEEE 802.11b que soporta velocidades de hasta 10 Mbps en la banda de 2.4 GHz.

La siguiente tabla ofrece algo más de detalle con respecto a los estándares WLAN.

**Tabla 3. Principales estándares WLAN**

Estándar	Velocidad máxima	Interface de aire	Ancho de banda de canal	Frecuencia
802.11b	11 Mbps	DSSS	25 MHz	2.4 GHz
802.11a	54 Mbps	OFDM	25 MHz	5.0 GHz
802.11g	54 Mbps	OFDM/DSSS	25 MHz	2.4 GHz
HomeRF2	10 Mbps	FHSS	5 MHz	2.4 GHz
HiperLAN2	54 Mbps	OFDM	25 MHz	5.0 GHz
5-UP	108 Mbps	OFDM	50 MHz	5.0 GHz

DSSS: Direct Sequence Spread Spectrum

OFDM: Orthogonal Frequency Division Multiplexing

FHSS: Frequency Hopping Spread Spectrum

5-UP: 5-GHz Unified Protocol (5-UP), Protocolo Unificado de 5 GHz propuesto por Atheros Communications

El gran éxito de las WLANs es que utilizan frecuencias de uso libre, es decir no es necesario pedir autorización o algún permiso para utilizarlas. Aunque hay que tener en mente, que la normatividad acerca de la administración del espectro varía de país a país.

La desventaja de utilizar este tipo de bandas de frecuencias es que las comunicaciones son propensas a interferencias y errores de transmisión. Estos errores ocasionan que sean reenviados una y otra vez los paquetes de información. Una razón de error del 50% ocasiona que se reduzca el caudal eficaz real (throughput) dos terceras partes aproximadamente. Por eso la velocidad máxima especificada teóricamente no es tal en la realidad. Si la especificación IEEE 802.11b nos dice que la velocidad máxima es 11 Mbps, entonces el máximo caudal eficaz será aproximadamente 6 Mbps y menos.

Para reducir errores, el 802.11a y el 802.11b automáticamente reducen la velocidad de información de la capa física. Así por ejemplo, el 802.11b tiene tres velocidades de información (5.5, 2 y 1 Mbps) y el 802.11a tiene 7 (48, 36, 24, 18, 12, 9 y 6 Mbps). La velocidad máxima permisible [ver tabla 3] sólo es disponible en un ambiente libre de interferencia y a muy corta distancia.

La transmisión a mayor velocidad del 802.11a no es la única ventaja con respecto al 802.11b. También utiliza un intervalo de frecuencia más alto de 5 GHz. Esta banda es más ancha y menos atestada que la banda de 2.4 GHz que el 802.11b comparte con teléfonos inalámbricos, hornos de microondas, dispositivos Bluetooth, etc. Una banda más ancha significa que más canales de radio pueden coexistir sin interferencia.

Sin bien, la banda de 5 GHz tiene muchas ventajas, también tiene sus problemas. Las diferentes frecuencias que utilizan ambos sistemas significa que los productos basados en 802.11a son no interoperables con los 802.11b. Esto significa que aunque no se interfieran entre sí, por estar en diferentes bandas de frecuencias, los dispositivos no pueden comunicarse entre ellos. Para evitar esto, la IEEE desarrolló un nuevo estándar conocido como 802.11g, el cual extenderá la velocidad y el intervalo de frecuencias del 802.11b para así hacerlo totalmente compatible con los sistemas anteriores. Sin embargo, no será más rápido que el estándar 802.11a. En 802.11a existe una gran variedad de fabricantes de chips [circuitos integrados] tales como Atheros, National Semiconductor, Resonext, Envara, inclusive Cisco Systems quien adquirió a Radiata, la primer compañía en desarrollar un prototipo en 802.11a en el 2000.

Como otro intento de permitir la interoperabilidad entre los dispositivos de bajas y altas velocidades, la compañía Atheros Communications, Inc. (<http://www.atheros.com/>) propuso unas mejoras a los estándares de WLANs de la IEEE y la ETSI. Este nuevo estándar conocido como 5-UP (5 GHz Unified Protocol) permite la comunicación entre dispositivos mediante un protocolo unificado a velocidades de hasta 108 Mbps.

Ambas especificaciones, la 802.11a (IEEE) y la HiperLAN2 (ETSI) son para WLANs de alta velocidad que operan en el intervalo de frecuencias de 5.15 a 5.35 GHz. El radio espectro asignado para el 802.11a y el HiperLAN2 es dividido en 8 segmentos o canales de 20 Mhz cada uno. Cada canal soporta un cierto número de dispositivos; dispositivos individuales pueden transitar a través de segmentos de red como si fueran teléfonos móviles de una estación a otra. Este espectro de 20 Mhz para un segmento de red soporta 54 Mbps de caudal eficaz compartido entre los dispositivos en el segmento en un tiempo dado.

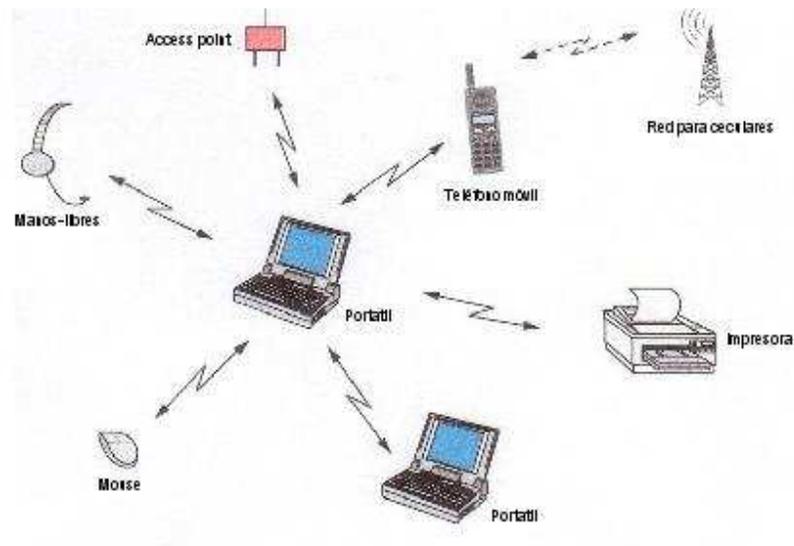
**2.4.3 Bluetooth [24].** Es la norma que define un estándar global de comunicación inalámbrica, que posibilita la transmisión de voz y datos entre diferentes equipos mediante

un enlace por radiofrecuencia. Los principales objetivos que se pretende conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales

La tecnología Bluetooth comprende hardware, software y requerimientos de interoperabilidad, por lo que para su desarrollo ha sido necesaria la participación de los principales fabricantes de los sectores de las telecomunicaciones y la informática, tales como: Ericsson, Nokia, Toshiba, IBM, Intel y otros. Posteriormente se han ido incorporando muchas más compañías, y se prevé que próximamente los hagan también empresas de sectores tan variados como: automatización industrial, maquinaria, ocio y entretenimiento, fabricantes de juguetes, electrodomésticos, etc., con lo que en poco tiempo se nos presentará un panorama de total conectividad de nuestros aparatos tanto en casa como en el trabajo. La Figura 5 nos presenta un panorama de las conexiones tradicionales con el Bluetooth simbolizando su ámbito de trabajo.

**Figura 5. Conexiones tradicionales con Bluetooth (Ámbito de trabajo)**



## **Antecedentes**

En 1994 Ericsson inició un estudio para investigar la viabilidad de una interfase vía radio, de bajo coste y bajo consumo, para la interconexión entre teléfonos móviles y otros accesorios con la intención de eliminar cables entre aparatos. El estudio partía de un largo proyecto que investigaba sobre unos multicomunicadores conectados a una red celular, hasta que se llegó a un enlace de radio de corto alcance, llamado *MC link*. Conforme éste proyecto avanzaba se fue viendo claro que éste tipo de enlace podía ser utilizado ampliamente en un gran número de aplicaciones, ya que tenía como principal virtud el que se basaba en un chip de radio relativamente económico.

## **El SIG**

A comienzos de 1997, según avanzaba el proyecto MC link, Ericsson fue despertando el interés de otros fabricantes de equipos portátiles. En seguida se vio claramente que para que el sistema tuviera éxito, un gran número de equipos deberían estar equipados con ésta tecnología. Esto fue lo que originó a principios de 1998, la creación de un grupo de interés especial (SIG), formado por 5 promotores que fueron: Ericsson, Nokia, IBM, Toshiba e Intel. La idea era lograr un conjunto adecuado de áreas de negocio, dos líderes del mercado de las telecomunicaciones, dos líderes del mercado de los PCS portátiles y un líder de la fabricación de chips. El propósito principal del consorcio fue y es, el establecer un estándar para la *interfase* aérea junto con su software de control, con el fin de asegurar la interoperabilidad de los equipos entre los diversos fabricantes.

## **La interfase aérea Bluetooth**

El primer objetivo para los productos Bluetooth de primera generación eran los entornos de la gente de negocios que viaja frecuentemente. Por lo que se debería pensar en integrar el chip de radio Bluetooth en equipos como: PCS portátiles, teléfonos móviles, PDAs y auriculares. Esto originaba una serie de cuestiones previas que deberían solucionarse tales como:

- El sistema debería operar en todo el mundo.
- El emisor de radio deberá consumir poca energía, ya que debe integrarse en equipos alimentados por baterías.
- La conexión deberá soportar voz y datos, y por lo tanto aplicaciones multimedia.

## **Banda de frecuencia libre**

Para poder operar en todo el mundo es necesaria una banda de frecuencia abierta a cualquier sistema de radio independientemente del lugar del planeta donde nos encontremos. Sólo la banda ISM (médico-científica internacional) de 2,45 Ghz cumple con éste requisito, con rangos que van de los 2.400 Mhz a los 2.500 Mhz, y solo con algunas restricciones en países como Francia, España y Japón.

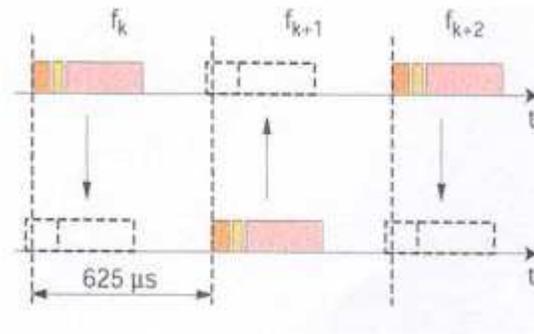
## Salto de frecuencia

Debido a que la banda ISM está abierta a cualquiera, el sistema de radio Bluetooth deberá estar preparado para evitar las múltiples interferencias que se pudieran producir. Éstas pueden ser evitadas utilizando un sistema que busque una parte no utilizada del espectro o un sistema de salto de frecuencia. En los sistemas de radio Bluetooth se suele utilizar el método de salto de frecuencia debido a que ésta tecnología puede ser integrada en equipos de baja potencia y bajo coste. Éste sistema divide la banda de frecuencia en varios canales de salto, donde, los transceptores, durante la conexión van cambiando de uno a otro canal de salto de manera pseudo-aleatoria. Con esto se consigue que el ancho de banda instantáneo sea muy pequeño y también una propagación efectiva sobre el total de ancho de banda. En conclusión, con el sistema FH (Salto de frecuencia), se pueden conseguir transceptores de banda estrecha con una gran inmunidad a las interferencias.

## Definición de canal

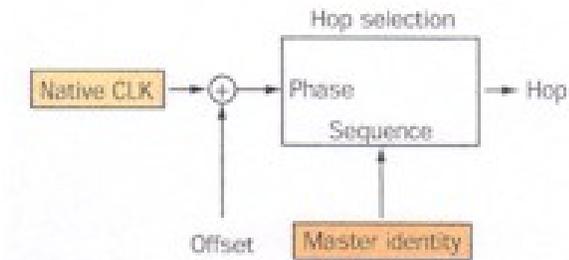
Como hemos comentado, Bluetooth utiliza un sistema FH/TDD (salto de frecuencia/división de tiempo dúplex), en el que el canal queda dividido en intervalos de  $625 \mu\text{s}$ , llamados slots, donde cada salto de frecuencia es ocupado por un slot. Esto da lugar a una frecuencia de salto de 1600 veces por segundo, en la que un paquete de datos ocupa un slot para la emisión y otro para la recepción y que pueden ser usados alternativamente, dando lugar a un esquema de tipo TDD. La Figura 6 ilustra la definición de canal en Bluetooth.

Figura 6. Definición de canal Bluetooth



Dos o más unidades Bluetooth pueden compartir el mismo canal dentro de una piconet, donde una unidad actúa como maestra, controlando el tráfico de datos en la piconet que se genera entre las demás unidades que deben actuar como esclavas, enviando y recibiendo señales hacia el maestro. El salto de frecuencia del canal está determinado por la secuencia de la señal, es decir, el orden en que llegan los saltos y por la fase de ésta secuencia. En Bluetooth, la secuencia queda fijada por la identidad de la unidad maestra de la piconet (un código único para cada equipo), y por su frecuencia de reloj. Por lo que, para que una unidad esclava pueda sincronizarse con una unidad maestra, ésta primera debe añadir un ajuste a su propio reloj nativo y así poder compartir la misma portadora de salto. La Figura 7 presenta la sincronización entre equipos esclavos y maestros.

Figura 7. Sincronización del equipo máster con un cliente.

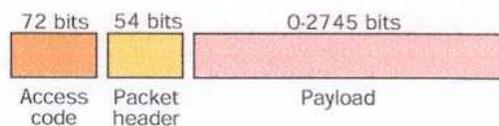


En países donde la banda está abierta a 80 canales o más, espaciados todos ellos a 1 Mhz., se han definido 79 saltos de portadora, y en aquellos donde la banda es más estrecha se han definido 23 saltos.

### Definición de paquete

La información que se intercambia entre dos unidades Bluetooth se realiza mediante un conjunto de slots que forman un paquete de datos. Cada paquete comienza con un código de acceso de 72 bits, que se deriva de la identidad maestra, seguido de un paquete de datos de cabecera de 54 bits. Éste contiene importante información de control, como tres bits de acceso de dirección, tipo de paquete, bits de control de flujo, bits para la retransmisión automática de la pregunta, y chequeo de errores de campos de cabeza. Finalmente, el paquete que contiene la información, que puede seguir al de cabeza, tiene una longitud de 0 a 2745 bits. En cualquier caso, cada paquete que se intercambia en el canal está precedido por el código de acceso. La Figura 8 ilustra la definición del paquete Bluetooth.

Figura 8. Definición del paquete Bluetooth.



Los receptores de la piconet comparan las señales que reciben con el código de acceso, si éstas no coinciden, el paquete recibido no es considerado como válido en el canal y el resto de su contenido es ignorado.

## Definición de enlace físico

En la especificación Bluetooth se han definido dos tipos de enlace que permitan soportar incluso aplicaciones multimedia:

- Enlace de sincronización de conexión orientada (SCO)
- Enlace asíncrono de baja conexión (ACL)

Los enlaces SCO soportan conexiones asimétricas, punto a punto, usadas normalmente en conexiones de voz, éstos enlaces están definidos en el canal, reservándose dos slots consecutivos (envío y retorno) en intervalos fijos. Los enlaces ACL soportan conmutaciones punto a punto simétricas o asimétricas, típicamente usadas en la transmisión de datos.

Un conjunto de paquetes se han definido para cada tipo de enlace físico:

- Para los enlaces SCO, existen tres tipos de slot simple, cada uno con una portadora a una velocidad de 64 kbit/s. La transmisión de voz se realiza sin ningún mecanismo de protección, pero si el intervalo de las señales en el enlace SCO disminuye, se puede seleccionar una velocidad de corrección de envío de 1/3 o 2/3.
- Para los enlaces ACL, se han definido el slot-1, slot-3, slot-5. Cualquiera de los datos pueden ser enviados protegidos o sin proteger con una velocidad de corrección de 2/3. La máxima velocidad de envío es de 721 kbit/s en una dirección y 57.6 kbit/s en la otra.

## Inmunidad a las interferencias

Como se mencionó anteriormente Bluetooth opera en una banda de frecuencia que está sujeta a considerables interferencias, por lo que el sistema ha sido optimizado para evitar éstas interferencias. En este caso, la técnica de salto de frecuencia es aplicada a una alta velocidad y una corta longitud de los paquetes (1600 saltos/segundo, para slots-simples). Los paquetes de datos están protegidos por un esquema ARQ (repetición automática de consulta), en el cual los paquetes perdidos son automáticamente retransmitidos, aun así, con este sistema, si un paquete de datos no llegase a su destino, sólo una pequeña parte de la información se perdería. La voz no se retransmite nunca, sin embargo, se utiliza un esquema de codificación muy robusto. Éste esquema, que está basado en una modulación variable de declive delta (CSVD), sigue la forma de la onda de audio y es muy resistente a los errores de bits. Estos errores son percibidos como ruido de fondo, que se intensifica si los errores aumentan.

**2.4.4 GPRS [25].** Las tradicionales redes GSM no se adaptan adecuadamente a las necesidades de transmisión de datos con terminales móviles. Por ello surge una nueva tecnología portadora denominada GPRS ( General Packet Radio Service) que unifica el

mundo IP con el mundo de la telefonía móvil, creándose toda una red paralela a la red GSM y orientada exclusivamente a la transmisión de datos.

### **Qué es GPRS**

GPRS es una nueva tecnología que comparte el rango de frecuencias de la red GSM utilizando una transmisión de datos por medio de 'paquetes'. En GSM, cuando se realiza una llamada se asigna un canal de comunicación al usuario, que permanecerá asignado aunque no se envíen datos. En GPRS los canales de comunicación se comparten entre los distintos usuarios dinámicamente, de modo que un usuario sólo tiene asignado un canal cuando se está realmente transmitiendo datos. Para utilizar GPRS se precisa un teléfono que soporte esta tecnología. La mayoría de estos terminales soportarán también GSM, por lo que podrá realizar sus llamadas de voz utilizando la red GSM de modo habitual y sus llamadas de datos (conexión a Internet, WAP,...) tanto con GSM como con GPRS.

#### **Ventajas para el usuario**

- Característica de "Always connected": un usuario GPRS puede estar conectado todo el tiempo que desee, puesto que no hace uso de recursos de red (y por tanto no paga) mientras no esté recibiendo ni transmitiendo datos.
- Tarificación por volumen de datos transferidos, en lugar de por tiempo.
- Coste nulo de establecimiento de conexión a la red GPRS, frente a los quantum de conexión existentes actualmente en GSM.
- Mayor velocidad de transmisión. En GSM sólo se puede tener un canal asignado (un "timeslot"), sin embargo, en GPRS, se pueden tener varios canales asignados, tanto en el sentido de transmisión del móvil a la estación base como de la estación base al móvil. La velocidad de transmisión aumentará con el número de canales asignados. Además, GPRS permite el uso de esquemas de codificación de datos que permiten una velocidad de transferencia de datos mayor que en GSM.

#### **Servicios para el usuario**

- Acceso a Internet desde un PC o PDA conectado al teléfono móvil GPRS.
- Acceso a WAP desde el teléfono GPRS.
- Acceso a la intranet de su empresa de forma idéntica a cómo lo haría en caso de estar utilizando su PC de sobremesa conectado directamente a la red LAN de la empresa.

#### **Ventajas de GPRS para la operadora**

Uso eficiente de la red: los usuarios sólo ocupan los recursos de la red en el momento en que están transmitiendo o recibiendo datos, y además se pueden compartir los canales de comunicación entre distintos usuarios.

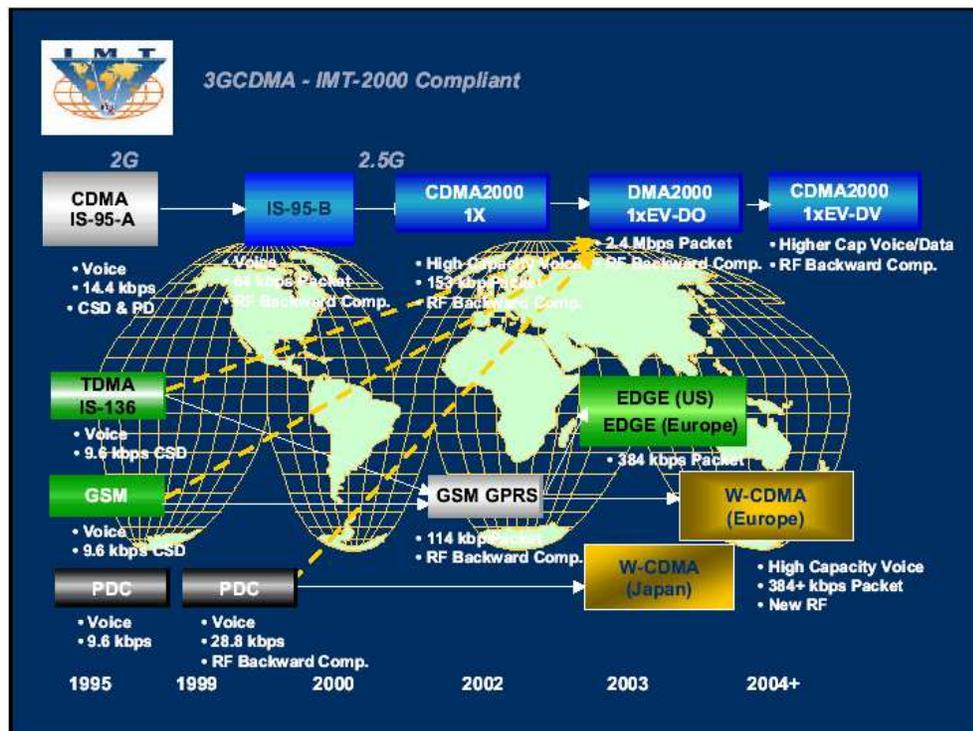
## Evolución de la red

GPRS precisa nuevos elementos de red que permitirán una conexión orientada a Internet.

Esta es la herramienta que garantiza que los dispositivos móviles realmente accedan a los Servicios Web y son parte esencial del concepto Servicios Web Móviles.

Es importante citar la siguiente figura que da una idea clara de la evolución de las denominadas WWANs desde sus inicios hasta el presente.

Figura 9. Evolución de los estándares Wireless globales



## 2.5 AMBIENTES DE DESARROLLO

### 2.5.1 Sistemas operativos.

2.5.1.1 PalmOS [12]. Es el sistema operativo propio de los dispositivos denominados Palm.

Desde la introducción de la primera Palm Pilot en 1996, la plataforma Palm OS ha definido los derroteros y expectativas de la computación móvil - como ejemplo la gente está usando handhelds como organizadores personales y como herramientas esenciales de negocio, apoyado también en la posibilidad de acceder al Internet o a una base de datos central corporativa por medio de conexiones inalámbricas.

Palm OS 5, que ha estado disponible desde el 2003, soporta procesadores ARM®-compliant. Palm OS Garnet es una versión ampliada de Palm OS 5 y provee características como soporte para un amplio rango de resoluciones de pantalla incluyendo QVGA, áreas de ingreso de datos dinámicas, comunicaciones en red mejoradas y Bluetooth.

Palm OS Cobalt es la nueva generación de Palm OS. Posibilita la creación de nuevas categorías de dispositivos para las comunicaciones, empresas, educación y mercados de entretenimiento.

Como la versión previa de Palm OS, Palm OS Garnet y Palm OS Cobalt mantienen compatibilidad a nivel de aplicación con aplicaciones existentes 68K-based.

**2.5.1.2 Microsoft Windows Mobile Platform [13].** Familia del sistema operativo Microsoft Windows que funciona en dispositivos móviles principalmente Pocket PC.

Está conformado por Windows CE .NET 4.2, Pocket PC 2002, y Windows Mobile 2003 software para Pocket PC. Pocket PC 2002 es el sistema propio de los Pocket PC. Los otros dos pueden adicionar características de desarrollo y de habilidades adicionales a los dispositivos móviles.

**2.5.1.3 Symbian OS [26].** Symbian OS es un sistema operativo con librerías asociadas, frameworks de interfaces de usuario e implementaciones de referencia a herramientas comunes, producido por Symbian y que fue iniciativa de Motorola, Ericsson, Nokia y Psion que lideró. Actualmente Nokia lidera el esfuerzo.

Hay varias interfaces de usuario que usa Symbian OS. Entre estas tenemos UIQ y la Serie 60 de Nokia. La adaptabilidad de las interfaces de usuario habilita el uso de Symbian OS en varios tipos de formas externas de dispositivos handhelds: clam-shell o tablet, teclado y/o estiletes de escritura, PDAs o teléfonos móviles y otros.

El actual Symbian OS se deriva del sistema operativo EPOC32 , que fue usado en los PDAs Serie 5 de Psion y originalmente desarrollado por Psion Software. Symbian OS ha pasado por varias versiones.

Dispositivos que usan Symbian OS:

- Nokia 9210 Communicator phone/PDA (2001)
- Sony Ericsson P800 (2002)
- Nokia Series 60 (2002): Nokia Series 60 se usa en varios teléfonos, de los cuales el primero fue el Nokia 7650. También se usa en otros teléfonos como Siemens SX1 y Sendo X.
- Sony Ericsson P900 (2003)

## 2.5.2 Herramientas de desarrollar.

### 2.5.2.1 Palm OS.

#### **Palm OS Developer Suite [12]**

Palm OS Developer Suite es una serie de herramientas integradas basadas en Eclipse que provee a los desarrolladores de software Palm OS con una forma más fácil y rápida para la creación de aplicaciones. Permite el desarrollo tanto de aplicaciones nativas ARM como de aplicaciones 68K, todo dentro de una sola IDE. Esta Suite de desarrollo se encuentra en: [http://www.palmos.com/dev/tools/dev\\_suite.html](http://www.palmos.com/dev/tools/dev_suite.html).

#### **CodeWarrior [15]**

CodeWarrior para la plataforma Palm OS es una de las herramientas de programación más completa de la industria. Está disponible para las plataformas Windows y MAC OS. Se puede encontrar como CodeWarrior Development Studio for Palm OS Platform, Version 9.2.

#### **PRC-Tools [14]**

PRC-Tools es un grupo de herramientas de compilación basadas en GCC muy completo para la creación de aplicaciones Palm OS bajo C o C++. Es software libre. El paquete PRC-Tools incluye versiones parchadas de paquetes GNU GCC, binutils, GDB y varias herramientas de linkeo posterior. Las partes específicas para Palm OS de PRC-Tools fueron originalmente escritas por D. Jeff Dionne, Kresten Krab Thourp, y los mantiene actualmente John Marshall, y muchas otras personas han hecho contribuciones a lo largo de los años. Para conseguir la última versión y documentación sobre instalación se debe visitar <http://prc-tools.sourceforge.net/>.

Para poder desarrollar aplicaciones Palm OS con PRC-Tools hay un par de cosas que se necesitan. Algunas de estas están disponibles en la página de SourceForge y otras están disponibles en el sitio web Palm OS Developer.

Para Windows se necesita:

Cygwin, PRC-Tools, PiIRC para Win32, Palm OS SDK y Palm OS Emulator.

Para Unix/Linux se requiere: PRC-Tools, PiIRC, Palm OS SDK y Palm OS Emulator.

#### **PiIRC**

PiIRC genera archivos de recurso binarios desde archivos textuales de definición. Es un editor de UI estándar para PRC-Tools y se integra con CodeWarrior usando un plug-in extra. Se encuentra en <http://www.ardiri.com/index.php?redir=palm&cat=pilrc>.

### **Bear River's PAF**

Un framework para aplicaciones empresariales, especialmente para aplicaciones grandes para negocios y el gobierno. Provee soporte para varias características. Se encuentra en: <http://www.bearriver.com/paf.html>.

### **Teenee**

Es un software libre. Es un framework para aplicaciones para programadores de C++ que se enfoca en la facilidad de uso y en la administración segura de la memoria.

### **Visual Form Designer**

Un entorno de desarrollo integrado (IDE) para el desarrollo de aplicaciones basadas en formas.

Se encuentra en <http://www.vfdide.com/>.

### **Microsoft Windows Mobile Platform [13]**

Existen una gama de herramientas para desarrollar para Pocket PC en plataforma Windows:

- Microsoft® eMbedded Visual Tools 3.0, 2002 Edition
- Microsoft eMbedded Visual C++® 4.0 SP3
- Microsoft Visual Studio® .NET 2003
- Microsoft .NET Compact Framework 1.0, 1.0 SP1, and 1.0 SP2

Igualmente se pueden tener las siguientes versiones de sistemas operativos específicos según el dispositivo móvil:

- Windows Mobile™-based Pocket PC 2002
- Windows Mobile™-based Pocket PC 2003
- Windows Mobile™-based Pocket PC 2003 Second Edition
- Windows Mobile™-based Smartphone 2002
- Windows Mobile™-based Smartphone 2003
- Windows Mobile™-based Smartphone 2003 Second Edition

Desde el lanzamiento del sistema operativo Microsoft Windows® CE en 1996 y el Handheld PC versión 1.0, Microsoft ha ofrecido herramientas que van direccionadas a solución de necesidades específicas.

Hay un documento web que ofrece a los desarrolladores interesados un entendimiento amplio sobre las herramientas de desarrollo para Windows Mobile en <http://msdn.microsoft.com/mobility/understanding/gettingstarted/windowsmobile/default.aspx?pull=/library/en-us/dnppcgen/html/devtoolsmobileapps.asp>.

### **Herramientas para el desarrollo de aplicaciones móviles:**

#### **eMbedded Visual Tools 3.0 [27]**

Microsoft eMbedded Visual Tools 3.0 provee un entorno de desarrollo integrado de nivel de entrada para la construcción de aplicaciones móviles, incluyendo los compiladores necesarios, facilidades de depuración y documentación de la plataforma. Esta herramienta incluye Microsoft eMbedded Visual C++® 3.0 y eMbedded Visual Basic® 3.0. Ambas herramientas son ambientes de desarrollo separados y no requieren ningún otro ambiente de desarrollo, como Microsoft Visual Studio o Visual Studio .NET para ejecutarse.

#### **eMbedded Visual C++ 3.0 [27]**

El sistema de desarrollo Microsoft eMbedded Visual C++ 3.0 ofrece compilador de código C++ nativo para el desarrollo basado en Windows Mobile. Puede dirigirse a todos los dispositivos basados en Microsoft Windows CE 2.11 y 3.0 si se instala el SDK correcto. El depurador eMbedded Visual C+ trabaja por medio de Microsoft ActiveSync®, a través de una LAN cableada, WLAN o conexión cradle (del handheld al PC).

Se debe usar para:

- Escribir código nativo que corra sobre Pocket PC 2002, Pocket PC 2003 y Pocket PC 2003 Second Edition, y código nativo que corra en Smartphone 2002, Smartphone 2003 y Smartphone 2003 Second Edition
- Drivers para Pocket PC y Smartphone
- Aplicaciones que corren nativamente en el dispositivo
- Juegos usando gráficos de alta velocidad a través del Game API (GAPI)
- Aplicación con bastante procesamiento
- Servidores COM o controles Microsoft ActiveX®

### **eMbedded Visual Basic 3.0 [27]**

eMbedded Visual Basic no puede ser usado para desarrollar aplicaciones para Smartphones basados en Windows Mobile 2002. Debido a esto y otros factores, incluyendo el gran interés en la comunidad de desarrolladores en Visual Basic .NET, la tecnología eMbedded Visual Basic está siendo reemplazada por el más poderoso lenguaje, Visual Basic .NET. Microsoft recomienda que los desarrolladores usen Visual Basic .NET para crear todas las nuevas aplicaciones Pocket PC y usen eMbedded Visual Basic para mantener aplicaciones viejas.

### **eMbedded Visual C++ 4.0 [27]**

Microsoft eMbedded Visual C++ 4.0 soporta desarrollos para dispositivos basados en Windows CE .NET 4.2 y por medio de service pack 3, soporta software Windows Mobile 2003 Second Edition, software para Pocket PCs y Smartphones. eMbedded Visual C++ 4.0 es ideal para dispositivos móviles o embebidos basados en Windows CE .NET 4.2 usando código nativo

Se usa para:

- Escribir código nativo que tome ventajas de las nuevas características introducidas en el software Windows Mobile 2003 para Pocket PC y Smartphone
- Escribir aplicaciones que tengan en cuenta Orientación y Resolución para software Windows Mobile 2003 Second Edition software para Pocket PC y Smartphone

**2.5.2.2 Java 2 Platform, Micro Edition (J2ME) [28].** Java 2 Platform, Micro Edition es la edición de la plataforma Java 2 dirigida a electrodomésticos y dispositivos embebidos. La tecnología consiste de una máquina virtual y un grupo de APIs adaptables para proveer entornos de ejecución a la medida para electrónicos y dispositivos embebidos. Tiene dos clases principales de componentes: configuraciones y perfiles.

#### **Configuración**

La tecnología J2ME tiene dos centros de diseño - objetos que se tiene en la mano y objetos que se conectan a la pared. Estos diseños tienen diferentes calidades que son optimizadas en la máquina virtual y en librerías de bajo nivel. Las configuraciones se componen de dos APIs de bajo nivel y máquinas virtuales optimizadas dirigidas a dos categorías principales de dispositivos: (1) dispositivos con 128 a 512 K de memoria disponible para el entorno de la tecnología Java (y aplicaciones) y (2) dispositivos con más de 512 K de memoria disponible para la tecnología Java y aplicaciones.

#### **Perfil**

Un perfil es una especificación que detalla los APIs de la tecnología Java, construida encima y que utiliza la Configuración subyacente, necesaria para proveer un entorno de ejecución completo para una clase de dispositivo específico. Los perfiles especifican tanto los APIs como la Configuración (si se ofrece). Un perfil debe ser completo en el sentido de que una aplicación escrita para la especificación se puede ejecutar en el entorno de tecnología Java especificado sin adicionar otras clases Java (contrario a paquetes

opcionales (Optional Packages)). Los perfiles pueden ser pensados como selección de clases desde APIs para formar un entorno completo. Los perfiles son diseñados e integrados para solucionar las necesidades de un segmento específico de la industria.

Para ver la información completa de esta opción de desarrollo se puede ver <http://java.sun.com/j2me>

Existen dos herramientas de desarrollo dirigidas a J2ME:

- J2ME Wireless Toolkit
- Sun Java Studio Mobility

### **J2ME Wireless Toolkit**

Java 2 Platform, Micro Edition (J2ME) Wireless Toolkit es un grupo de herramientas para desarrollar aplicaciones inalámbricas que están basadas en J2ME's Connected Limited Device Configuration (CLDC) y Mobile Information Device Profile (MIDP), y diseñadas para ejecutarse en teléfonos celulares, PDAs y otros dispositivos móviles pequeños. El toolkit incluye la emulación de entornos, optimización de rendimiento y características de afinación, documentación y ejemplos que los desarrolladores necesitan para lograr aplicaciones inalámbricas eficientes y exitosas al mercado rápidamente.

### **Sun Java Studio Mobility**

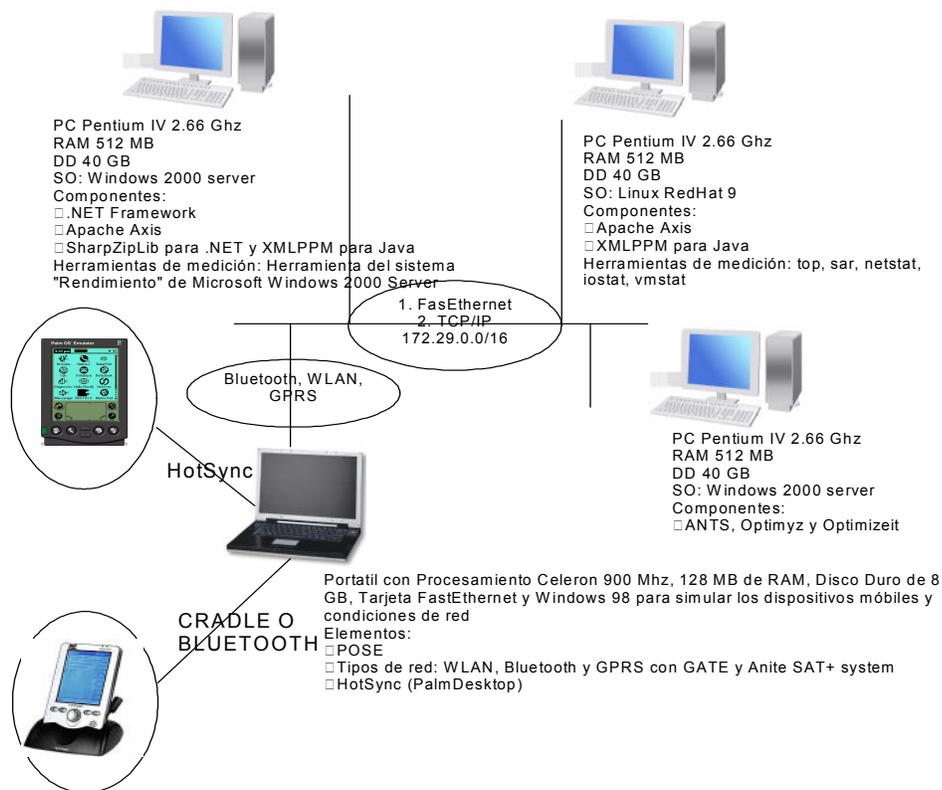
Sun Java Studio Mobility 6 2004Q3 es un entorno integrado de desarrollo (IDE) para el desarrollo de aplicaciones que pueden ser implementadas en dispositivos móviles habilitados para tecnología Java. Con características optimizadas para desarrollo con Mobile Information Device Profile (MIDP)/Connected Limited Device Configuration (CLDC), los desarrolladores de aplicaciones que usan la tecnología Java 2 Platform, Micro Edition (J2ME) se beneficiarán de las capacidades de desarrollo rápido de aplicaciones de Sun Java Studio Mobility - tanto de desarrollo standalone de clientes J2ME MIDlets o MIDlet que se conecten a servicios disponibles en una red.

### 3. DISEÑO DE LAS PRUEBAS EN SERVICIOS WEB MÓVILES

#### 3.1 VISTA GRÁFICA DEL BANCO DE PRUEBAS

La siguiente figura ilustra el “Banco de pruebas” del proyecto.

Figura 10. Vista gráfica del Banco de Pruebas del proyecto



#### 3.2 DETALLES DEL BANCO DE PRUEBAS

##### 3.2.1 Hardware y Software de servicios.

###### 3.2.1.1 Publicación de Servicios Web bajo Windows.

- Descripción: Generación de Servicios Web en el ambiente Windows para ser consumidos por PDAs en el banco de pruebas
- Arquitectura: PC x86

- Procesamiento: Pentium IV 2.66 Ghz
- Memoria RAM: 512 MB
- Almacenamiento interno: IDE 40 GB
- Sistema operativo: Microsoft Windows 2000 Server con Service Pack 4
- Antivirus: Norton

Componentes:

- .NET Framework [8]: La IDE de Windows para desarrollo y publicación de servicios en la Web
- Apache Axis [7]: El componente de la organización Apache para publicar Servicios Web
- SharpZipLib [29] para .NET y XMLPPM [5] para Java: Compresores para el XML y HTML producido en el extremo del servicio
- Herramientas de medición: Herramienta del sistema "Rendimiento" de Microsoft Windows 2000 Server

### **3.2.1.2 Publicación de Servicios Web bajo Linux.**

- Descripción: Generación de Servicios Web en el ambiente Linux (Un tipo UNIX) para ser consumidos por PDAs en el banco de pruebas
- Arquitectura: PC x86
- Procesamiento: Pentium IV 2.66 Ghz
- Memoria RAM: 512 MB
- Almacenamiento interno: IDE 40 GB
- Sistema operativo: Microsoft Windows 2000 Server con Service Pack 4
- Antivirus: Norton

Componentes:

- ANTS [30], Optimyz y Optimizeit [31]

### **3.2.1.3 Generación de carga.**

- Descripción: Simulación de carga en el consumo de Servicios Web. Se hace desde el sistema operativo Windows hacia los dos publicadores de Servicios Web en los ambientes Windows y Linux.
- Arquitectura: PC x86
- Procesamiento: Pentium IV 2.66 Ghz
- Memoria RAM: 512 MB
- Almacenamiento interno: IDE 40 GB
- Sistema operativo: Linux RedHat 9

Componentes:

- Apache Axis [7]: El componente de la organización Apache para publicar Servicios Web
- XMLPPM [5] para C: Compresores para el XML y HTML producido en el extremo del servicio
- Herramientas de medición: top, sar, netstat, iostat, vmstat

#### **3.2.1.4 Conectividad para los PDAs y simulación de ambientes de conectividad.**

- Descripción: Tres objetivos: ejecutar los Emuladores de PDAs tanto Pocket PC como Palm, ofrecer conectividad a los PDAs y simular ambientes de conectividad Bluetooth, WLAN y GPRS
- Portátil con Procesamiento Celeron 900 Mhz, 128 MB de RAM, Disco Duro de 8 GB, Tarjeta FastEthernet y Windows 98 para simular los dispositivos móviles y condiciones de red
- Arquitectura: Portátil x86
- Procesamiento: Intel Celeron a 900 Mhz
- Memoria RAM: 128 MB
- Almacenamiento interno: IDE 8 GB
- Sistema operativo: Windows 98

Componentes:

- POSE y Emulador Pocket PC: Emuladores de PDAs
- Tipos de red: WLAN, Bluetooth y GPRS con GATE y Anite SAT+ system
- HotSync (PalmDesktop): Sincronizar la instalación de aplicaciones para Palm

#### **3.2.2 Elementos clientes.**

##### **3.2.2.1 POSE [12].**

El Emulador para PALM

##### **3.2.2.2 Emulador Pocket PC [13].**

Emulador para Pocket PC

##### **3.2.2.3 Equipo real Palm III.**

- Modelo: PalmIIIc

##### **3.2.2.4 Equipo real Pocket PC.**

- Modelo: iPAQ

### **3.2.3 Elementos para el desarrollo.**

#### **3.2.3.1 Hardware/software para el desarrollo.**

##### Hardware

- Procesador: 1.5 Ghz
- Memoria RAM: 1.5 GB
- Disco duro: 36 GB
- Interfaz de red: FastEthernet

##### Software

- Sistema Operativo Microsoft Windows 2000 Server Service Pack 4

#### **3.2.3.2 Herramientas para aplicaciones sobre Palm.**

- Palm OS Developer Suite [12] (La suite de desarrollo para Palm)
- Cygwin [32] (Simulador de ambiente Unix para poder usar PRC-Tools en Windows)
- PRC-Tools [14] (compiladores, enlazadores y otros para GCC)
- PilRC para Win32 (generador de formularios para Palm)
- Palm OS Emulator [12] (simuladores de dispositivos Palm de acuerdo al ROM que se use)

#### **3.2.3.3 Herramientas para la creación de clientes de Servicios Web sobre Palm.**

En el sitio de Alphaworks, de IBM se conseguía una herramienta para el desarrollo de clientes de Servicios Web para dispositivos móviles. Esta herramienta se denominaba Web Service Toolkit for Mobile Devices (WSTKMD). Se apoyaba en una herramienta de sourceforge.net denominada gSoap, la cual al desaparecer WSTKMD tomo la vocería en cuanto el soporte a este tipo de aplicaciones. El gSoap es una herramienta que genera bindings entre desarrollos hechos en C y C++ hacia Servicios Web del tipo SOAP/XML.

**Soap** (sourceforge.net) [33]

#### **3.2.3.4 Ejemplos usados en Palm.**

##### Aplicaciones:

- GlobalCity Matriz: Navegador web para Palm, Código objeto: GlobalCity20.zip - 134 K
- Inventory 123: Software de inventario para Palm, Código objeto: d93ccf0d.zip - 39 K
- KRemote: Generación de órdenes de compra fuera de línea para posterior actualización en línea, Código: kremote1\_2\_1.zip - 566 K
- Diagnostic: Medición de uso de conexión entre el handheld y la estación wireless

Clientes de Servicios Web:

- BabelFish: Traductor de texto desde y hacia varios idiomas, Código objeto: babelFish.zip - 120 K
- Amazon: Búsqueda de productos en el sitio Amazon.zom, Código objeto: Amazon.pqa - 13 K

### 3.2.3.5 ROMs Palm usadas [12].

- 35dr2
- 35-en-ez-8mb-wc
- 412
- PalmOS351-ROMs (muy adecuado, el más estándar, el más usado)
- Treo GPRS
- Treo
- VisorPrismPlatinum

## 3.3 DESARROLLO DEL BANCO DE PRUEBAS

**3.3.1 Creación de escenarios de conectividad.** Se crean varios ambientes de conectividad con la herramienta GATE de Ericsson y la herramienta Anite SAT+ system. Son ideales para cuando realmente no se posee físicamente la conexión, pues personaliza a fondo los parámetros y simula características de la conexión agregando situaciones por las que pasa generalmente una señal de este tipo. Esto permite medir el tiempo de servicio sobre Bluetooth, WLAN y GPRS considerando que el tiempo de servicio es desde la solicitud hasta el recibo del “último byte”, con compresión y sin compresión.

**3.3.2 Ideas para la medición de uso de conectividad.** Se crean varios escenarios en los cuales como punto de partida se hace una medición en cuanto a bytes de un Servicio Web de los desarrollados como ejemplo. La medición también se le hace a los datos comprimidos. Se usan algunos “emulators” de dispositivos encontrados en la web, para permitir la creación completa del ambiente end-to-end.

**3.3.3 Compresión de la información desde los extremos.** Para la compresión se usa gzip en el caso de .NET y XMLPPM para C en Linux. No se pretende hacer un análisis exhaustivo de las herramientas, sólo se usan, aunque se complementa indicando las tasas de compresión que generan y consideramos que al usarlas en el estudio siempre es adecuado ofrecer tablas de comparación y descripciones en términos adecuados para dar ideas de lo que se tratan.

**3.3.4 Publicadores de Servicios Web usados.** Para tener una idea adecuada de estos escenarios se trabajan dos publicadores de Servicios Web, uno en .NET y otro en Apache. Esto para medir uso de recursos de servidores en los dos ambientes.

**3.3.5 Simuladores de concurrencia.** Para simular concurrencia y lograr umbrales de servicio, así como pruebas de stress y monitoreo se usa ANTS.

### 3.4 NARRACIÓN DEL DETALLE DE PRUEBAS

El banco de pruebas descrito obtiene los resultados tabulados y detallados en el capítulo 6. Los parámetros a medir se detallan en el capítulo 4.

#### 3.4.1 La primera prueba, tamaño en bytes de la información enviada y recibida.

**3.4.1.1 Tamaño original de archivos cliente en HTTP 1.1.** Esta prueba inicialmente ofrece tablas de la cantidad de bytes que transmite y recibe cada uno de los servicios. Existen dos ambientes, Windows y Linux. En cada ambiente existen dos ejemplos, una calculadora y un cálculo de factorial. En el ambiente Windows el ejemplo de calculadora se llama MathService y se desarrolla en VB (Visual Basic), el segundo ejemplo que se denomina FactorialServiceC está desarrollado en C#. En el ambiente Linux se desarrollan en C los dos ejemplos. Se usan en Windows el bloc de notas y en Linux el editor estándar vi para algunas modificaciones. Los ejemplos son obtenidos o modificados de ejemplos estándar en Internet.

La siguiente tabla resume las características de los ejemplos:

**Tabla 4. Características de los ejemplos de Clientes de Servicios Web para PDAs**

Ambiente	Servidor	Lenguaje de desarrollo para cliente Calculadora	Lenguaje de desarrollo para cliente Calculadora	Nombres de ejemplo de calculadora	Nombres del ejemplo factorial
Windows	.NET Framework	Visual Basic	C#	MathService	FactorialServiceC
Linux	Apache Axis	C	C	Calculator	Factorial

Las tablas en el capítulo 6 detallan la cantidad de bytes para cada uno de los ejemplos. Estos bytes consideran ya todos los agregados por el parser XML y por el protocolo HTTP 1.1.

**3.4.1.2 La información HTTP 1.1 comprimida.** Luego de esto, y de acuerdo a la importancia definida para el recurso de ancho de banda en el capítulo 4 como parámetro fundamental de las mediciones, se concluye que un procedimiento adecuado es comprimir la información a enviar. Por tal motivo se usan dos algoritmos estándar para compresión en Internet, uno muy difundido para cualquier tipo de información, el gzip y otro para XML, el XMLPPM.

Las tablas de resultados inicialmente detallan el tamaño de cada uno de los ejemplos comprimidos.

En Windows se usa cygwin para tener la posibilidad de usar el comando gzip directamente sobre los archivos. En Linux se consigue la librería XMLPPM, que sirve para comprimir archivos XML.

**3.4.1.3 El uso del procesamiento.** Realmente, el uso de procesamiento para la compresión o descompresión de los ejemplos en los dos servidores Windows y Linux es tendiente a cero, por lo cual se decide pasar a otra etapa de pruebas. Una etapa que permite determinar cuál es el uso del procesamiento para rangos de tamaños de archivos y rangos de cantidad de procesos paralelos de compresión. Es decir, se toman como base archivos de 1K, 10K, 100K y 1000K. Ya pensar en un archivo generado en XML de 1000K es una exageración, pues según lo investigado y probado, como máximo un archivo XML para un Servicio Web podría llegar a 10K. Esto se explica precisamente por una de las características más importantes y llamativas de los Servicios Web, el envío y recepción de sólo la información que se requiere. Para cada uno de estos archivos entonces se tabulan los usos de procesamiento para 1, 10 y 100 procesos en paralelo en el servidor.

Como se mencionó en el apartado anterior, se usa gzip en Windows y XMLPPM en Linux. Vale aclarar que tanto gzip como XMLPPM se pueden implementar en ambos ambientes.

Para poder generar las pruebas de uso de procesamiento por medio de procesos paralelos, se crearon inicialmente scripts que generan 100 archivos de cada uno de los tamaños con el prefijo númeroarchivo\_tamañoenKbytes.txt. Entonces, existe un archivo para cada uno de los tamaños, así: 1K.txt, 10K.txt, 100K.txt y 1000K.txt. Y se generaron 100 archivos para cada tamaño así: 1\_1K, 2\_1K,...,100\_1K, para el archivo de 1K, 1\_10K, 2\_10K,...,100\_10K, para el archivo de 10K, 1\_100K, 2\_100K,...,100\_100K, para el archivo de 100K y 1\_1000K, 2\_1000K,...,100\_1000K, para el archivo de 1000K. En el caso particular de XMLPPM fue necesario cuidar la estructura del formato XML para que el analizador (parser) XMLPPM pudiera actuar sobre los archivos. Finalmente se crearon scripts para ejecutar en paralelo el comando gzip sobre los 100 archivos de cada uno de los tamaños en Windows y para generar la compresión paralela de los 100 archivos de cada uno de los tamaños con XMLPPM en Linux.

El capítulo 6 detalla todos estos cuadros.

Como complemento se detallan los resultados obtenidos para calidad del servicio en los ejemplos en Windows obtenidos para una configuración de rata de 100 y 5 usuarios en ANTS. Esto para definir calidad del servicio independientemente del uso del procesador sin compresión.

## **3.5 DIFICULTADES**

**3.5.1 Desarrollo de clientes de servicios Web para PALM.** No existen herramientas ni sitios ejemplos bien fundamentados para el desarrollo de estas aplicaciones.

La solución a la dificultad fue la investigación en Internet y prueba y error de manera abundante.

## 4. PARÁMETROS DE MEDICIÓN

### 4.1 INTRODUCCIÓN

Los PDAs son dispositivos con escaso poder computacional (comparado con un PC convencional). El procesamiento de una PDA de excelentes características en la actualidad puede estar en el orden de la séptima parte de un PC actual. Esto también incluye la memoria RAM. Adicionalmente, la conectividad desde un PDA, aquella que realmente complementa el concepto de movilidad está representada por las denominadas WWAN de las cuales la más difundida es GPRS (GSM para envío de paquetes). Este tipo de conectividad es muchas veces escasa y por lo general costosa. Generalmente se configuran los pagos de servicio por la cantidad de bytes transmitidos hacia la PDA y viceversa.

De acuerdo a esto, existen dos parámetros que se deben medir en cuanto a consumo de recursos cuando se ejecuta un cliente de un Servicio Web Móvil desde un PDA. Estos recursos son el procesamiento y la conectividad y lo que se debe buscar es garantizar un aprovechamiento óptimo del escaso recurso de conectividad sin afectar severamente el procesamiento y garantizando que los parámetros de calidad del servicio (QoS) no se deterioren.

### 4.2 PARÁMETROS DE MEDICIÓN

**4.2.1 El procesamiento.** Realmente, los procesos que involucran encriptación/descriptación o compresión/descompresión consumen procesamiento cualquiera que sea el equipo que se use para tal fin.

Ahora, por qué la descompresión?. En el CAPÍTULO TRES - DISEÑO DE LAS PRUEBAS EN SERVICIOS WEB MÓVILES, se detalla claramente que se van a usar elementos compresores para el envío de información. La compresión forma parte de las recomendaciones habida cuenta de la necesidad de optimizar el uso del ancho de banda que es el recurso más escaso en el consumo de Servicios Web Móviles en el lado del cliente, el dispositivo PDA. En el capítulo CAPÍTULO DOS - ESTADO DEL ARTE, se detallan las formas en que un PDA se puede conectar a una red. Tenemos entonces Bluetooth, WLAN, Cradle y WWAN (GPRS). Este último (GPRS) es el que al fin de cuentas ofrece el concepto de movilidad en toda su extensión, porque en los otros casos el dispositivo móvil estaría sujeto a conexiones dentro de un área geográfica pequeña. Adicionalmente, en cualquiera de estos escenarios, es posible que se presente el consumo de un Servicio Web por parte del PDA, pero se estaría dando plenamente el concepto de movilidad. Debido a que este recurso de conectividad se considera muy escaso, la compresión de la información es una recomendación definitiva y para poder establecer el grado de uso del procesador en el servidor, este será uno de los parámetros a medir.

**4.2.2 Consumo de ancho de banda.** Inicialmente, con Bluetooth, el PDA se puede conectar con un equipo que ofrece un Servicio Web, pero que también posea la forma de conectividad Bluetooth. Con WLAN se puede conectar en una LAN inalámbrica y consumir Servicios Web desde un equipo servidor que posea la forma de conectividad WLAN. Con el Cradle, se conecta directamente a un equipo que puede Servidor de Servicios Web. Y la cuarta forma es la forma más representativa de lo que sería al ambiente móvil del dispositivo.

¿Por qué el ambiente WWAN sería el único escenario para los Servicios Web Móviles?. Esto se debe a que en los otros tres ambientes el dispositivo estaría “atado” a un ambiente LAN. Depende de dispositivos que están instalados dentro de un Campus, un edificio o en el mejor de los casos a una comunicación radial en un ambiente metropolitano.

La WWAN, de los cuales el caso más representativo es GPRS sobre GSM, permite garantizar la característica móvil de poder conectarse desde cualquier parte del mundo en cualquier momento, directamente con la PDA. Vale anotar en este punto que Bluetooth es muy usado para conectar una PDA al carrier GPS.

Si bien es cierto que usando las tres primeras opciones, Bluetooth, WLAN o Cradle, es posible conectarse a la red de redes, y consumir Servicios Web, el ancho de banda ya no sería un parámetro crítico por cuanto este parámetro ya dependería de un proxy, o de un firewall, y al final del ancho de banda de una institución sobre un ISP. Aunque en algunos casos Bluetooth es una opción para conectarse a GPS.

Por tal motivo, y teniendo en cuenta que el ambiente característico de los Servicios Web Móviles es la WWAN, y que la conectividad WWAN no es un recurso muy disponible, ni muy económico, se tiene que este es un segundo parámetro a medir.

Una estrategia para el máximo aprovechamiento de este escaso recurso es la compresión. Aunque se debe buscar un equilibrio y una de las recomendaciones finales del trabajo es precisamente este equilibrio que debe lograr el punto óptimo entre compresión, consumo de ancho de banda y procesamiento puesto que la descompresión en el lado del PDA causaría consumo de procesamiento que es el otro recurso crítico de la máquina.

En esta caso, para establecer cuál es consumo de ancho de banda, se medirán entonces en bytes las solicitudes y las respuestas tanto en la forma comprimida como en la no comprimida.

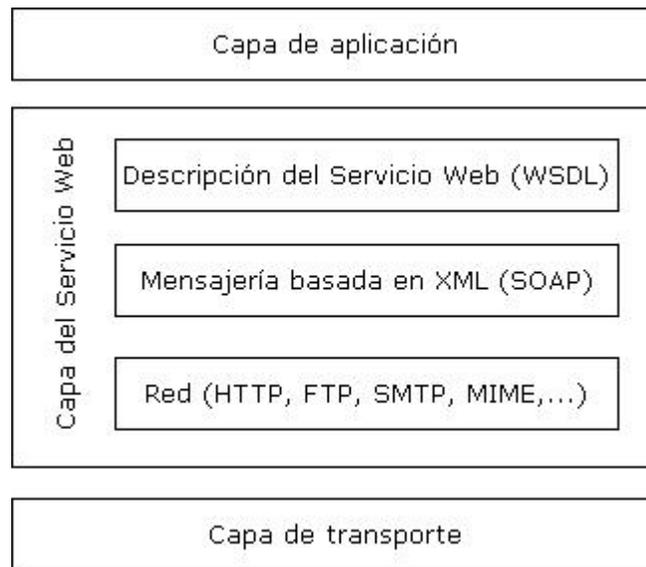
Parámetro de medición: Tamaño en bytes de la solicitud y de la respuesta tanto comprimidas como sin descomprimir. Se debe tener en cuenta que los Servicios Web por defecto no están comprimidos, así que se deben poner de acuerdo los extremos en si van a usar compresión. HTML es más liviano que XML y por eso XML consume mucho más ancho de banda que HTML. Los objetivos son medir el tamaño de SOAP. Todo esto se debe considerar pues GPRS se cobra por consumo. Se mide en el servidor también, al igual que el procesamiento pues si se comprime para beneficiar en ancho de banda al cliente, entonces, esto puede generar deterioro de QoS. Medir cada uno de las capas según la Figura 11, Arquitectura de los Servicios Web Móviles: Transporte, Servicio Web (red: http, smtp, ftp; XML: SOAP y Descripción del Servicios Web: WSDL) y Aplicación. Sirve para dos casos: cuando es pobre la conexión y cuando es costosa y por consumo de

bytes o tiempo. Se debe tener en cuenta seguridad (encriptación y firmas digitales) - y esta seguridad puede generar mediciones de tamaños y procesamientos.

Parámetro de medición: En el servidor, paquetes IP de entrada y de salida de la interfaz de red.

Parámetro de medición: Tiempo de respuesta del servidor en ofrecer el servicio y del cliente en obtenerlo y mostrarlo en pantalla.

**Figura 11. Arquitectura de los Servicios Web Móviles<sup>4</sup>**



#### 4.2.3 Memoria y almacenamiento.

**La memoria RAM y el almacenamiento.** Aunque todo proceso consume memoria RAM, entre ellos los Servicios Web o los clientes de Servicios Web en el lado del dispositivo móvil, en la compresión y descompresión de información la memoria RAM no constituye un factor crítico.

Los PDAs tienen una característica especial, tanto almacenamiento como memoria RAM forman parte de un recurso unido, pero cuando se habla de consumo de este recurso se habla más bien de las etapas de desarrollo que se contemplan en detalle en el tutorial incluido en este documento.

---

<sup>4</sup> Tomado de [3]

El tutorial ofrece las consideraciones y recomendaciones a tener en cuenta cuando se desarrolla para dispositivos con escaso poder de memoria RAM o almacenamiento, y no sólo eso, sino que detalla las técnicas para poder sobrepasar los inconvenientes que genera la arquitectura de hardware de los PDAs.

Es importante mencionar que existen básicamente dos limitantes en este campo, el primero tiene que ver con el máximo tamaño de fracción de código compilado que un dispositivo puede soportar. Un PDA puede alojar fracciones de código compilado no mayor a 64Kbytes lo que dificulta sobremanera la programación de aplicaciones sobre todo aquellas cuyo objetivo es consumir un Servicio Web en Internet. La dificultad está representada en el hecho de que el único proyecto dirigido a crear clientes de Servicios Web desde dispositivos PDA usando el lenguaje de programación C es el proyecto gSOAP [33], el cual contiene dentro de los archivos necesarios para llevar a cabo el consumo, un fuente de 333 Kbytes y no existen dentro de la documentación de gSOAP recomendaciones puntuales hacia cómo solventar la dificultad.

gSOAP al instalarse genera un directorio de ejemplos para Palm. En este directorio es posible encontrar dos ejemplos, uno de consumo de un Servicio Web para cuotas de stock de empresas denominado stockquote y otro dirigido a consumir un Servicio Web que determina la validez de parámetros de interoperabilidad denominado interop.

En ambos casos es necesario incluir dentro del proyecto dirigido a crear un archivo final (base de datos) de tipo prc para Palm el archivo stdsoap2 de 333Kbytes.

gSOAP ofrece dos archivos palmsoap1.c y palmsoap2.c que buscan solventar el inconveniente de RAM y almacenamiento de la PDA. Estos archivos buscan dividir el código compilado pero realmente según las pruebas realizadas no lo logra por lo cual queda de parte del desarrollador solventar la situación. En el capítulo que contiene el tutorial se dan recomendaciones importantes para desarrollar estos clientes de Servicios Web.

La arquitectura de las PDAs también ofrece otro obstáculo; no pueden haber llamados a funciones, variables o elementos dentro del código compilado que sobrepase los 32Kbytes de distancia dentro de la memoria en ejecución. Por tal motivo el desarrollador debe solventar la situación y dentro del capítulo tutorial en busca de dar una respuesta a estas dos barreras, se generan recomendaciones de subdivisión de códigos usando una técnica de prc-tools denominada secciones.

Las siguientes tablas presentan las características en cuanto a memoria RAM de los PDA actuales.

**4.2.3.1 Palm [12].** La siguiente tabla resume la capacidad en memoria y almacenamiento de los dispositivos PDAs tipo Palm:

**Tabla 5. Capacidad en memoria RAM en dispositivos Palm actuales**

Tipo de PDA	Modelo	Memoria RAM
Palm	Zire 21	8MB
Palm	Zire 31	16MB
Palm	Zire 72	8MB de masked ROM y 32MB
Palm	Zire 72 GPS	8MB de masked ROM y 32MB + 64MB memory card
Palm	Tungsten C	64MB (51MB de real capacidad de almacenamiento)
Palm	Tungsten E	32MB (28.3MB de real capacidad de almacenamiento)
Palm	Tungsten T3	64MB (52MB de real capacidad de almacenamiento)
Palm	Tungsten T5	256MB (215MB de real capacidad de almacenamiento: drive de flash interno de 160MB, 55MB de memoria de programas para aplicaciones y datos.)

La siguiente tabla resume la capacidad en memoria y almacenamiento de los dispositivos PDAs tipo Pocket PC:

#### 4.2.3.2 Pocket PC [13]

**Tabla 6. Capacidad en memoria RAM en dispositivos Pocket PC actuales**

Tipo de PDA	Modelo	Memoria RAM
Pocket PC	ASUS My Pal A620	64 MB SDRAM (55 MB user accessible) 32 MB Flash ROM
Pocket PC	ASUS My Pal A620BT	64 MB SDRAM (55 MB user accessible) 64 MB Flash ROM (32 MB available for storage)
Pocket PC	ASUS My Pal A716	64 MB SDRAM (55 MB user accessible) 64 MB Flash ROM
Pocket PC	Dell Axim X30 624MHz	up to 64MB SD RAM up to 64MB
Pocket PC	Dell Axim X30 312 MHz (non-wireless)	up to 32MB SD RAM up to 32MB SD ROM
Pocket PC	Dell Axim X50	64MB SD RAM 64MB or 128 MB Intel StrataFlash ROM
Pocket PC	Dell Axim X50v	64MB SD RAM 128MB StrataFlash ROM
Pocket PC	Garmin iQue M5	64 MB SDRAM 64 MB ROM

Continuación Tabla 6. Capacidad en memoria RAM en dispositivos Pocket PC actuales

<b>Tipo de PDA</b>	<b>Modelo</b>	<b>Memoria RAM</b>
Pocekt PC	HP iPAQ h2210/h2215	64 MB (56 MB main memory)
Pocekt PC	HP iPAQ h4150/h4155	64 MB SDRAM (55 MB user accessible) 32 MB Flash ROM
Pocekt PC	HP iPAQ h4350/h4355	64 MB SDRAM (55 MB user accessible) 32 MB Flash ROM
Pocekt PC	HP iPAQ h5150	64 MB SDRAM 32 MB Flash ROM Memory
Pocekt PC	HP iPAQ h5455	64 MB SDRAM (46 MB user-accessible); 128 MB (using optional 64-MB SD Memory Card) 16 MB CMOS Flash ROM
Pocekt PC	HP iPAQ h5550	128 MB SDRAM 48 MB Flash ROM Memory
Pocekt PC	HP iPAQ h5555	128 MB SDRAM 48 MB Flash ROM Memory
Pocekt PC	HP iPAQ hx2100/2110	64 MB SDRAM 64 ROM
Pocekt PC	HP iPAQ hx2410/2415	64 MB SDRAM 64 MB ROM
Pocekt PC	HP iPAQ hx2700/2750	128 MB SDRAM 128 MB ROM
Pocekt PC	HP iPAQ hx4700/4705	64 MB SDRAM 128 MB Strata Flash ROM
Pocekt PC	HP iPAQ rx3115 Mobile Media Companion	56MB user available memory
Pocekt PC	HP iPAQ rx3715 Mobile Media Companion	152MB user available memory (includes 96MB of iPAQ File Store)
Pocekt PC	HP iPAQ rz1710/1715 Mobile Media Companion	25 MB user available memory (includes 10 MB of iPAQ File Store)
Pocekt PC	Mio 168 GPS Pocket PC	64MB SDRAM 32MB ROM
Pocekt PC	Symbol Technologies MC50	64MB ROM 64MB RAM

## 5. TUTORIAL DE PROGRAMACIÓN EN SERVICIOS WEB MÓVILES

### 5.1 INTRODUCCIÓN

Los dos dispositivos PDAs más usados son las Palm y los Pocket PC. Estos dos dispositivos marcan la pauta en cuanto a movilidad, entendiéndose movilidad como la posibilidad de poder realizar actividades similares a las que se realizan en un PC normal desde cualquier ubicación. Dentro de estas actividades están también aquellas en la cual es posible acceder a una red desde cualquier ubicación.

El concepto de movilidad y el concepto de Servicios Web se unen para formar los denominados servicios web móviles, es decir, usar las PDAs principalmente para acceder a Servicios Web. Se dice principalmente porque existen dispositivos como los Smartphones que cumple algunas de las funciones de un PDA pero dando también funciones de un teléfono móvil.

Existe entonces gran demanda por las aplicaciones que puedan acceder a estos Servicios Web.

En cuanto a aplicaciones normales para PDAs, aquellas que no acceden a Servicios Web, los desarrolladores tienen estas opciones:

- Para Palm: PODS (Palm OS Developer Suite - prc-tools), Palmphi (prc-tools), Codewarrior y Java.
- Para Pocket PC: Mobile Application Development Toolkit y Java

Ahora, con respecto a aplicaciones que acceden a Servicios Web Móviles, está la opción de gSOAP para desarrollar en C para Palm, de Java para ambos dispositivos y de Mobile Application Development Toolkit para Pocket PC.

En los siguientes apartados se describen cada una de estas posibilidades.

### 5.2 PALM [12]

**5.2.1 prc-tools [14].** El paquete prc-tools soporta programación en C y C++ para Palm OS. Contiene:

- La colección del compilador GNU, el assembler, el linker y el depurador simbólico.
- Modificaciones a esta herramienta genérica para soportar funcionalidades peculiares a Palm OS.
- Herramientas para manejar archivos. prc, versiones de Palm OS para aplicaciones ejecutables.

- Documentación convenientemente empaquetada para todos los elementos mencionados.

El sitio oficial de la herramienta es <http://prc-tools.sourceforge.net> y los siguientes párrafos se basan en la documentación del sitio oficial y generan adicionalmente recomendaciones de acuerdo a la experiencia de los desarrolladores del presente proyecto. Bien, si se instala PODS no es necesario instalar y configurar el paquete prc-tools pues el PODS lo hace por defecto debido a que se basa en el. Cuando no se usa PODS se debe usar un entorno a modo de ventana de comandos. Si se usa para desarrollar el sistema operativo tipo UNIX, simplemente se bajan los paquetes y se instalan según las indicaciones siguientes, pero si se desarrolla en ambiente Windows se hace necesario usar cygwin, un emulador de la consola de tipos UNIX. A continuación entonces la descripción para instalar y configurar prc-tools.

**5.2.1.1 Instalar y configurar con prc-tools cuando no se usa PODS.** La programación en Palm OS requiere (como mínimo) de un compilador C o C++, un SDK de Palm OS (Standar Development Kit) y de un compilador de recursos UI (compilador de recursos de interfaces de usuario). La mayoría de los programadores también usan el Emulador para Palm OS y otros usarán varios SDKs para Palm OS y tendrán otros encabezados (headers), librerías y SDKs instalados.

Estas instrucciones explican cómo instalar prc-tools y dónde instalar SDKs y otro material de desarrollo para Palm OS de manera que el compilador los pueda ver. Para detalles completos de la instalación de otros paquetes diferentes a prc-tools, se puede consultar la documentación respectiva.

**5.2.1.1.1 Paquetes binarios.** Se proveen binarios precompilados de prc-tools para Linux x86 (en formato RPM).

Si se usan paquetes Debian, se podrían instalar paquetes RPM usando alien. También el proyecto Debian mantiene paquetes prc-tools para Debian.

Si usa FreeBSD, también se podría estar interesado en prc-tools.

En este trabajo de tesis se han realizado los desarrollos en un PC con sistema operativo Windows 2000 Server. Como se explicará más adelante, se usa PODS. Los desarrollos se hacen para Palm y Pocket PC.

**5.2.1.1.2 Compilando e instalando de los fuentes.** Si no se encuentran paquetes binarios para un sistema, o si se prefiere, es posible compilar prc-tools desde los códigos fuentes. Los tarballs de códigos fuente de versiones formales están disponibles en la página de download. El repositorio CVS contiene todas las versiones del código, incluyendo cambios hechos desde la versión más reciente. El código fuente obtenido en cualquiera de estas formas contiene instrucciones para compilar en el archivo BUILDING.html.

PRC-Tools contiene parches para versiones específicas de GCC, GDB, y binutils, como se detalla en BUILDING.html. Se necesita código fuente para estas versiones de estos paquetes, lo cual puede ser encontrado en varios espejos GNU o en el directorio misceláneo de archivos.

Las varias versiones del Palm OS SDK están disponibles como archivos que contienen sólo los archivos relevantes en varios formatos, así como paquetes con instaladores para plataformas específicas. (Estos archivos son listados en la página SDK y titulados Palm OS SDK Core Components).

Ni prc-tools ni GCC tienen conocimiento de qué Palm OS SDKs u otros archivos de encabezados o librerías se van a instalar. Se puede usar palmdev-prep para proveer tal conocimiento: esto analiza la estructura de directorios recursivamente hacia adentro, por defecto, el directorio que Usted especifique por medio de prc-tools; con la opción de configuración palmdev-prefix, y proveyendo las opciones -I y -L de manera que GCC Palm OS pueda encontrar automáticamente los encabezados y librerías relacionados.

Los SDKs Palm OS y otros materiales de desarrollo deberían ser instalados de tal forma que palmdev-prep pueda detectarlos. Si se configura prc-tools con -with-palmdev-prefix=palmdev-prefix, la versión N de SDK debe estar instalada de tal manera que la estructura de directorio sea:

- palmdev-prefix/sdk-N/include
- Los archivos de encabezado, acomodados en directorios arbitrarios bajo include
- palmdev-prefix/sdk-N/lib/m68k-palmos-coff
- Librerías estáticas (tiempo de enlace) para m68k-palmos

**5.2.1.1.3 Compilando otras herramientas.** PilRC es un compilador de recursos UI para Palm OS ampliamente usado que es fácil de configurar. Especialmente si no se está ejecutando GTK+, se puede configurar con -disable-pilrcui.

El emulador para Palm OS está escrito en un estilo moderno de C++ y en FLTK de UNIX como su interfase de usuario de configuración liviana. C++ moderno quiere decir que es necesario implementaciones de plantillas y STL que son recientes; por ejemplo, el uso de una versión antigua pre-2.95 de GCC no se recomienda. En el momento de la escritura (versión 3.5), el Emulador es más fácil de configurar con una versión 1.0.x de FLTK.

**5.2.1.1.4 Trabajar con cygwin [32].** Cuando se desarrolla en el sistema operativo Windows con prc-tools se deben usar el cygwin, un emulador de la consola tipo UNIX. El sitio oficial es: <http://www.cygwin.com>.

**5.2.1.1.4.1 Instalando Cygwin y prc-tools.**

- Se ejecuta setup.exe desde el sitio oficial de Cygwin: <http://www.cygwin.com>

- Aparece un asistente, inicialmente pregunta por los sitios desde los cuales bajar la información. Se debe seleccionar uno de los del listado para paquetes normales, pero para lo concerniente a Palm OS se debe indicar: <http://prc-tools.sourceforge.net/install>.
- Al pasar a la siguiente ventana del asistente aparece la lista de paquetes que por defecto son los necesarios, pero para el desarrollo para Palm se necesitan:
  - prc-tools contiene las herramientas de compilación (m68k) Palm OS.
  - prc-tools-arm, contiene las herramientas adicionales necesarias para construir programas ARM.
  - prc-tools-htmldocs provee copias locales de los manuales que también están en línea en <http://prc-tools.sourceforge.net/doc/>.
  - pilrc es la UI para generar recursos (formas y sus objetos ), estos paquetes proveen PilRC UI resource compile.r
  - make, a menudo usado para construir proyectos para Palm OS.

Por defecto sólo prc-tools se instala, por lo tanto los otros cuatro se deben seleccionar.

**5.2.1.1.4.2 Montando el árbol de directorios PalmDev.** Prc-tools necesita el árbol de directorios PalmDev para estar disponible al camino tipo POSIX /PalmDev. La mejor forma de lograrlo es crear un punto de montura apuntando a la ubicación que se ha seleccionado:

- Se crea *C:\PalmDev*, o cualquier otro que se quiera; preferiblemente el que se recomienda, esto por estandarización
- Abrir una ventana shell desde el menu inicio, en el ícono de Cygwin, luego se tipea:
  - `mount -tf "C:\PalmDev" /PalmDev`
  - Se reemplaza *C:\PalmDev* por el sitio en el cual se instaló SDK

Esto se hace una sola vez.

**5.2.1.1.4.3 Instalando SDKs.** Algunos Palm OS SDKs están disponibles como instaladores para Windows, otros son zips que se deben desempaquetar. Se recomienda instalar estos SDKs en subdirectorio de /PalmDev.

**5.2.1.1.4.4 Configuración Post-instalación.** Se debe usar palmdev-prep para indicar dónde están los SDKs y otros materiales de desarrollo. Con palmdev-prep m68k-palmos-gcc podrá encontrar encabezados y librerías relacionados con Palm Os automáticamente, sin usar opciones -I (include) y -L (linking) adicionales, y tendrá una nueva opción -palmosN para seleccionar qué SDK usar.

Usando la opción `-d SDK` se puede cambiar el SDK, cuando la opción `-palmosN` no se use.

Palmdev-prep busca *C:\PalmDev* (por simplicidad se usa este en la documentación, pero debe ser el que se escoja en la configuración anterior) y generalmente se debe encontrar algo como:

- *C:\PalmDev\sdk-Mininclude*
- Los archivos de encabezado, bajo subdirectorios llamados *include*, así:
- *C:\PalmDev\sdk-Mlib\m68k-palms-coff*
- Static (link-time) libraries for m68k-palms

Por compatibilidad con algunos SDKs, palmdev-prep detecta estos archivos bajo los subdirectorios *Incs* en vez de *include* y/o librerías bajo *Libraries* en vez de *lib*.

**5.2.1.1.4.5 Usar prc-tools en una ventana de comandos de Windows.** Para trabajar en una consola Windows, se debe adicionar *C:\cygwin\bin* al PATH de Windows. (Puede ser otro diferente a *c:\cygwin*, eso depende de lo que se halla indicado en la instalación).

Dependiendo de la versión de Windows, depende la forma en que se configure esta variable del sistema.

**5.2.1.1.4.6 prc-tools y gSOAP [33].** Hasta el momento se ha tocado el tema de prc-tools. prc-tools en resumen es una herramienta libre que sirve para crear aplicaciones para Palm en lenguaje C o C++.

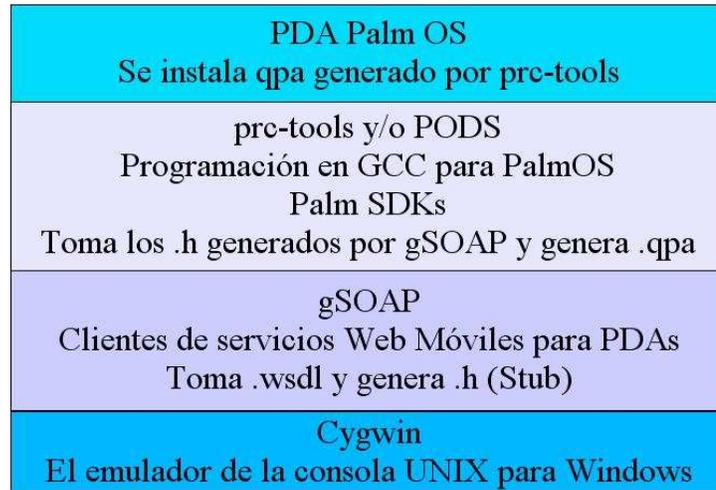
Ahora bien, para poder crear servicios web para estos dispositivos móviles existe una herramienta también de carácter gratuito llamada gSOAP que tiene su sitio oficial en <http://gsoap2.sourceforge.net/>.

GSOAP es una herramienta para la generación de Servicios Web SOAP/XML en C y C++.

Ofrece un lenguaje para la transformación de XML a C/C++ para facilitar el desarrollo de Servicios Web SOAP/XML en C y C++. La mayoría de herramientas para Servicios Web en C++ adoptan una vista centrada en SOAP y ofrecen APIs que requieren el uso de clases de librerías para estructuras de datos específicos de SOAP. Esto a menudo obliga a un usuario a adoptar la lógica de aplicación para estas librerías. Por el contrario, gSOAP provee un API SOAP transparente por medio del uso de tecnologías de compilación probadas. Estas tecnologías apoyan fuertes codificaciones para mapear esquemas XML a definiciones C/C++. Esta codificación fuerte provee mucha seguridad en la validación del contenido tanto de esquemas WSDL como de mensajes SOAP/XML. El compilador gSOAP genera serializadores XML eficientes para tipos de datos en C y C++ nativos y definidos por el usuario. Como resultado, se logra la interoperabilidad de SOAP/XML con una simple API evitándole al usuario el peso de detalles en WSDL y SOAP, y permitiéndole concentrarse en la lógica esencial de la aplicación. El compilador posibilita la (herencia) integración de código C/C++ y Fortran (a través de una interfase Fortran a C), sistemas embebidos y software para aplicaciones en tiempo real en aplicaciones SOAP que comparten recursos computacionales e información con otros servicios Web, posiblemente a través de diferentes plataformas, lenguajes y organizaciones separadas protegidas por cortafuegos.

Ahora, ya se puede esquematizar la forma de trabajo para el uso de estas herramientas en la generación de Servicios Web Móviles, lo cual se detalla en la siguiente figura:

**Figura 12. Esquema de trabajo para programación en Palm**



El esquema básicamente presenta en la parte más inferior el cygwin. Cygwin es el emulador para la consola tipo UNIX en el sistema operativo Windows. Esto es necesario para poder usar las herramientas prc-tools y gSOAP pues están creadas para sistemas operativos tipo UNIX.

En el siguiente nivel se tiene el gSOAP. Esta herramienta toma un archivo .wsdl, que es el archivo en codificación XML que describe el servicio web. Todo Servicio Web se describe mediante un archivo wsdl que generalmente es de uso público o que puede obtenerse bajo condiciones comerciales de uso.

El siguiente nivel comprende las herramientas de desarrollo para aplicaciones móviles y finalmente se tienen los qpa, las bases de datos entendidas como aplicaciones para PDAs de tipo Palm.

Los siguientes párrafos se apoyan en la documentación de gSOAP for Palm OS, específicamente de su archivo gSOAP\_on\_PALM.html y describen claramente la forma de generar los “Stubs” para que prc-tools, luego de la creación del aplicativo en C++ para Palm pueda llamarlas y usar servicios web.

La guía de programación completa para gSOAP se encuentra en el sitio de gSOAP: <http://www.cs.fsu.edu/~engelen/soap.html>, aunque a continuación se hace un resumen de la programación.

El software se puede bajar de la siguiente dirección:  
[http://sourceforge.net/project/showfiles.php?group\\_id=52781](http://sourceforge.net/project/showfiles.php?group_id=52781)

Pasos en el uso de gSOAP:

a. Instalación

Para instalar, se descomprime el archivo bajado y se consiguen los siguientes archivos:

- README.txt (se debe leer primero)
- NOTES.txt (instrucciones de instalación y notas del release)
- changelog.html (cambios)
- license.html (archivo HTML con licencia pública gSOAP)
- license.pdf (licencia en PDF)
- gpl.txt (licencia GPL)
- soapcpp2 (ejecutable, compilador gSOAP 2)
- wSDL2h (ejecutable, parser gSOAP WSDL)
- soapdoc2.html (documentación en HTML)
- soapdoc2.pdf (documentación PDF)
- stdsoap2.h (archivo para stdsoap2.c[pp])
- stdsoap2.c (librería y runtime gSOAP)
- stdsoap2.cpp (librería, idéntica a stdsoap2.c)
- dom.html (parser gSOAP DOM en HTML)
- dom.pdf (parser gSOAP DOM en PDF)
- dom.h (archivo header para dom.c)
- dom.c (parser C DOM)
- dom++.h (archivo header para dom++.cpp)
- dom++.cpp (parser C++ DOM)
- stl.h (extensiones STL (combina los archivos .h de abajo)
- stldeque.h (interface STL deque)
- stllist.h (interface STL list)
- stlvector.h (interface STL vector)
- stlset.h (interface STL set)
- typemap.dat (esquema XML mapeador de tipos de C/C++ para parser WSDL)
- xml.h (Archivo importado por archivos .h generados por el parser WSDL)

y los siguientes subdirectorios:

- src (fuentes del compilador gSOAP)
- samples (ejemplos)
- extras
- wSDL (fuentes de parsers WSDL)

Los ejemplos para Palm están en el directorio `gsoap-win32-2.7\gsoap-palm` y se llaman `interop` y `stockquote`.

## b. Configuración

El parser WSDL de gSOAP convierte WSDL en un archivo header de gSOAP para el procesamiento con los compiladores `stub` y `skeleton` de gSOAP para construir Servicios Web (Clientes y Servidores). Todo esto se genera hacia C o C++.

Un ejemplo:

```
$ wsd12h -o Amazon.h http://soap.amazon.com/schemas/AmazonWebServices.wsdl
```

```
$ soapcpp2 Amazon.h
```

Esto genera el archivo de encabezado (header) `Amazon.h`. Este archivo contiene una traducción de los servicios y esquemas XML hacia C/C++ y otra información útil copiada del WSDL. Luego el archivo header se procesa con el compilador `stub` y `skeleton` de gSOAP produciendo los siguientes archivos:

- `soapClient.cpp`                    client-side stub routines for service invocation
- `soapServer.cpp`                   server-side skeleton routines for server development
- `soapC.cpp`                         C/C++ parameter marshalling code

Para desarrollar una aplicación cliente C++, se pueden usar la clase `'soapAmazonSearchBindingProxy.h'` y la tabla de namespaces XML `'AmazonSearchBinding.nsmmap'` del Servicio Web Amazon. Ambos se deben incluir en los fuentes del programador. Luego se compilan y linquean los fuentes `soapC.cpp`, `soapClient.cpp`, y `stdsoap2.cpp` para completar la implementación.

En estos ejemplos no se dan archivos `Makefile` para Palm, pero a continuación se presenta un ejemplo que se debe acomodar de acuerdo a necesidades:

## Código 1. Un archivo Makefile ejemplo para prc-tools con cygwin

```
# This make file assumes that two environment variables are set
# PRC-TOOLS - the Location of the GNU prc-tools
# PALMDEV - the Location of the PALM SDK 4.0

CC = m68k-palmos-gcc
AS = m68k-palmos-as
LD = m68k-palmos-ld

VPATH = .././../lib .. .././../include

PILRCFLAGS = -l .. -q -allowEditID

PRC-TOOLS.USR=$(PRC-TOOLS)/prc/usr

CFLAGS=-O2 -DPALM_GCC -I.. -I.././../include -I$(PALMDEV)/sdk-4/include -I$(PRC-
TOOLS.USR)/m68k-palmos/include -I$(PRC-TOOLS.USR)/share/prc-tools/include -I$(PRC-
TOOLS.USR)/lib/gcc-lib/m68k-palmos/2.95.3-kgpd/include -I$(PALMDEV)/sdk-4/include/Core/System -
I$(PALMDEV)/sdk-4/include/Core/System/unix -I$(PALMDEV)/sdk-4/include/libraries -I$(PALMDEV)/sdk-
4/include/Core -I$(PALMDEV)/sdk-4/include/Dynamic -I$(PALMDEV)/sdk-4/include/Core/UI -
I$(PALMDEV)/sdk-4/include/Core/Hardware -I$(PALMDEV)/sdk-4/include/libraries/PalmOSGlue

LDFLAGS=-L.././../lib -L$(PRC-TOOLS.USR)/m68k-palmos/lib -L$(PRC-TOOLS.USR)/lib/gcc-lib/m68k-
palmos/2.95.3-kgpd -L$(PALMDEV)/sdk-4/lib/m68k-palmos-coff -lc
all: stockQuote.prc

stockQuote.prc: QuoteStarter stockQuote.ro
    build-prc -c STQG -n "stockQuote" QuoteStarter stockQuote.ro

stockQuote_objs = QuoteStarter.o soapC.o soapClient.o

QuoteStarter: $(stockQuote_objs) SoapLibGlue.a

QuoteStarter.o: QuoteStarter.c StockQuoteRsc.h soapH.h soapStub.h

soapC.o: soapC.c soapH.h soapStub.h

soapClient.o: soapClient.c soapH.h soapStub.h

stockQuote.ro: stockQuote.rcp
    pilrc $(PILRCFLAGS) -ro -o $$@ $$<

clean:
    rm -f *.ro *.a *.o *.prc *.c *.h *.bin
```

Para esclarecer ideas:

1. Seguir las instrucciones de instalación de gSOAP
2. Generar el encabezado del servicio desde WSDL y stubs con los compiladores y parsers de gSOAP
3. Copiar los archivos generados (soapClient.c y soapC.c) y los archivos propios de la implementación de los clientes de gSOAP para Palm (palmFunctions.c, palmFunctions.h, stdsoap2.c, stdsoap.h, soapH.h, soapStub.h) en el directorio de fuentes del proyecto del programador.

4. Afinar el archivo Makefile del ejemplo, compilar y generar la base de datos prc.

Como nota especial vale decir que existen dos ambientes para programar esta opción de prc-tools y gSOAP, una se trata de la que se acaba de explicar, prc-tools con cygwin en la que se genera un entorno totalmente texto y de comandos, y la otra se detalla a continuación y es PODS.

**5.2.2 PODS (Palm OS Developer Suite) [12].** Es una Suite muy interesante para programar mediante la opción gratuita de prc-tools. Es un IDE muy apropiado para desarrollar bajo Windows aprovechando prc-tools. Permite más comodidad que simplemente desarrollar usando comandos en un tipo UNIX. También se apoya en cygwin, y cuando se instala por defecto se instalan los componentes prc-tools y cygwin.

Para bajar la suite se debe ir al sitio [http://www.palmos.com/dev/tools/dev\\_suite.html](http://www.palmos.com/dev/tools/dev_suite.html).

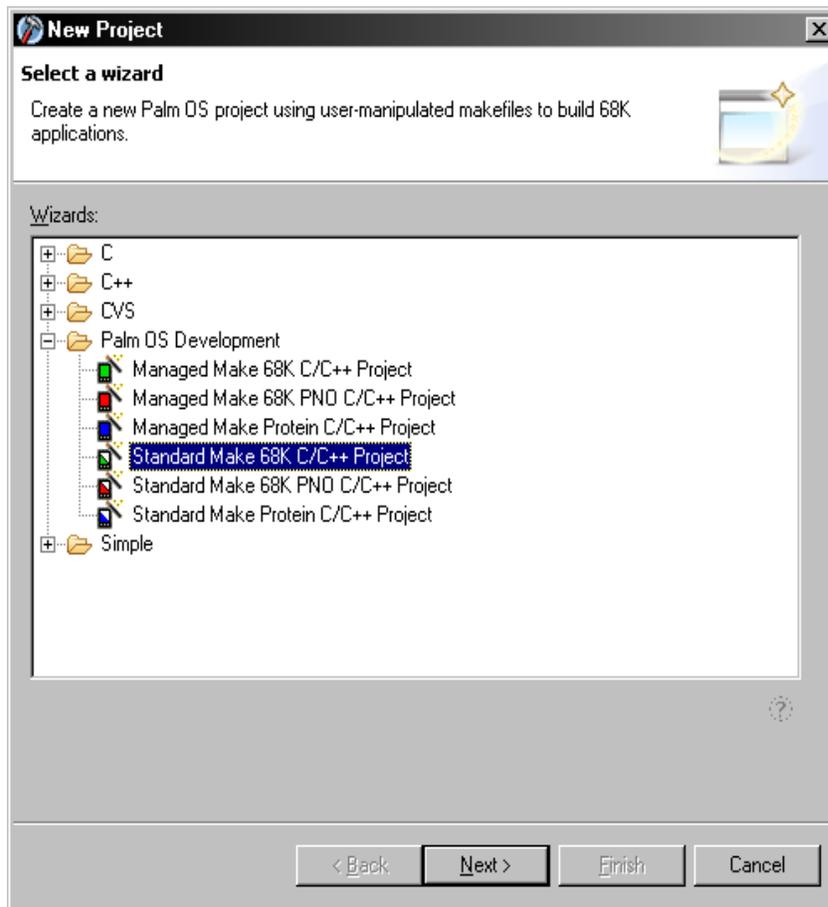
Bien, es un poco complicado el inicio para alguien que aún no ha programado en Palm. Por eso es muy recomendable inicialmente crear un nuevo proyecto usando un asistente.

Para ejecutar el PODS; luego de la instalación se encuentra en Inicio\_Programas\_PalmSource\_Palm OS Developer Suite.

Para crear un nuevo proyecto se usa la opción File\_ New\_ Project y se debe seleccionar el tipo de aplicación que se quiere crear. Las aplicaciones para Palm son bases de datos de extensión prc generalmente, y la más usada es de tipo Standard 68K. Para el efecto de este trabajo de tesis se usaron este tipo de aplicaciones debido a que el dispositivo real que se tiene es de este tipo y adicionalmente porque no es del alcance del proyecto generar aplicaciones para los diversos dispositivos. Básicamente, no es de importancia para el objetivo puesto que simplemente se trata de elegir en el asistente el tipo de proyecto de acuerdo al dispositivo y la programación es muy similar. Realmente es más conveniente de acuerdo a lo buscado en el trabajo actual realizar una aplicación Standard 68K puesto que este tipo de dispositivos presenta inconvenientes al no permitir fuentes mayores de 68K de tamaño en bytes y se genera una situación interesante para dividir los fuentes grandes necesarios en gSOAP (consumo de Servicios Web en C/C++).

La ventana para seleccionar el tipo de proyecto es:

Figura 13. La ventana de tipo de nuevo proyecto en PODS

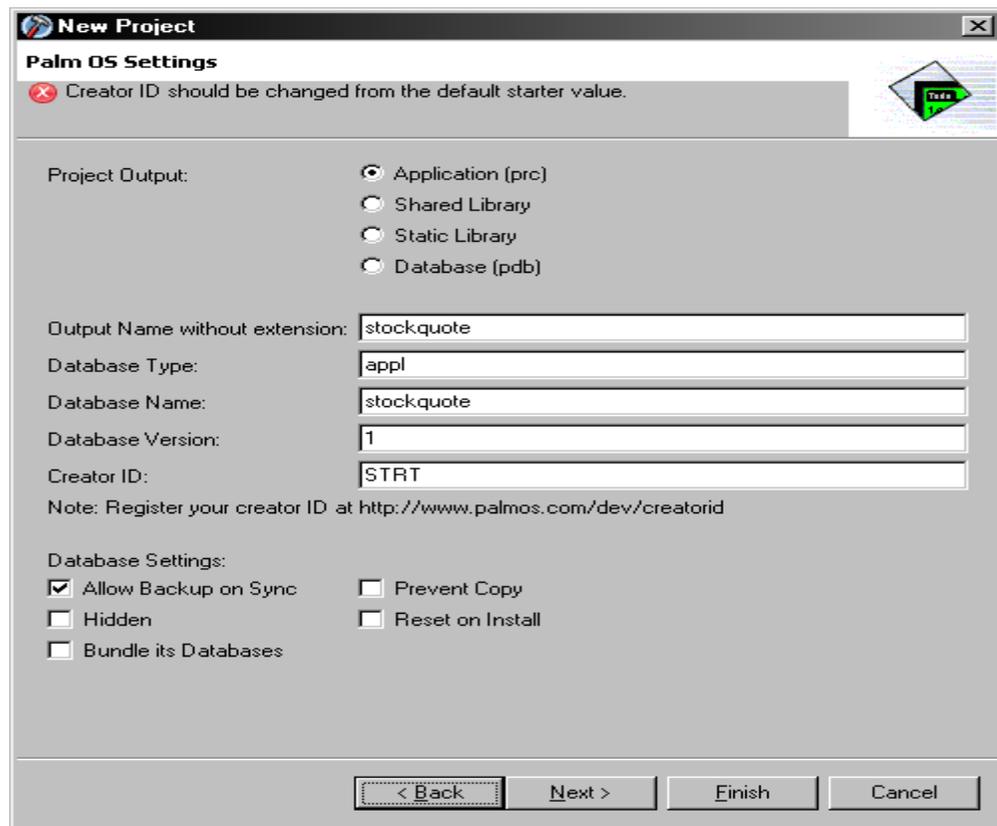


En la siguiente ventana dar el nombre del proyecto y dar clic en siguiente para continuar en el asistente.

En la siguiente ventana se coloca la salida del proyecto, en este caso el objetivo es generar una aplicación para Palm tipo prc (una base de datos para instalar en Palm). Por esto se deja la opción por defecto, Application (prc). Las casillas adicionales se pueden dejar por defecto pues van de acuerdo con el nombre del proyecto seleccionado en el paso 3 y con el tipo de salida seleccionado en este paso. La casilla Creator ID se refiere a un identificador único que asigna Palm a los desarrolladores, y que se puede obtener <http://www.palmos.com/dev/creatorid>; se puede dejar por defecto (STRT) o colocar uno y luego registrarlo. Este registro no es una condición para que la aplicación funcione, es una forma de identificar el creador de las aplicaciones para temas relacionados con autoría. Tiene unas casillas de verificación relacionadas con la forma de instalar, por ejemplo evitar que se copie la base de datos, activar un reseteo de tipo software en el dispositivo en el que se va a instalar (no es necesario en este caso), permitir backup cuando se instale con HotSync, si está oculta o no. HotSync es una aplicación que viene dentro de una aplicación denominada Palm Desktop que se instala en Windows y permite sincronizar el dispositivo Palm con el PC. Sirve principalmente para instalaciones y para

actualizaciones de agendas y aplicaciones desde el PC hacia la Palm y viceversa. Luego entonces se hace clic en Siguiente para ir a la siguiente ventana del asistente. La siguiente figura presenta las casillas descritas anteriormente.

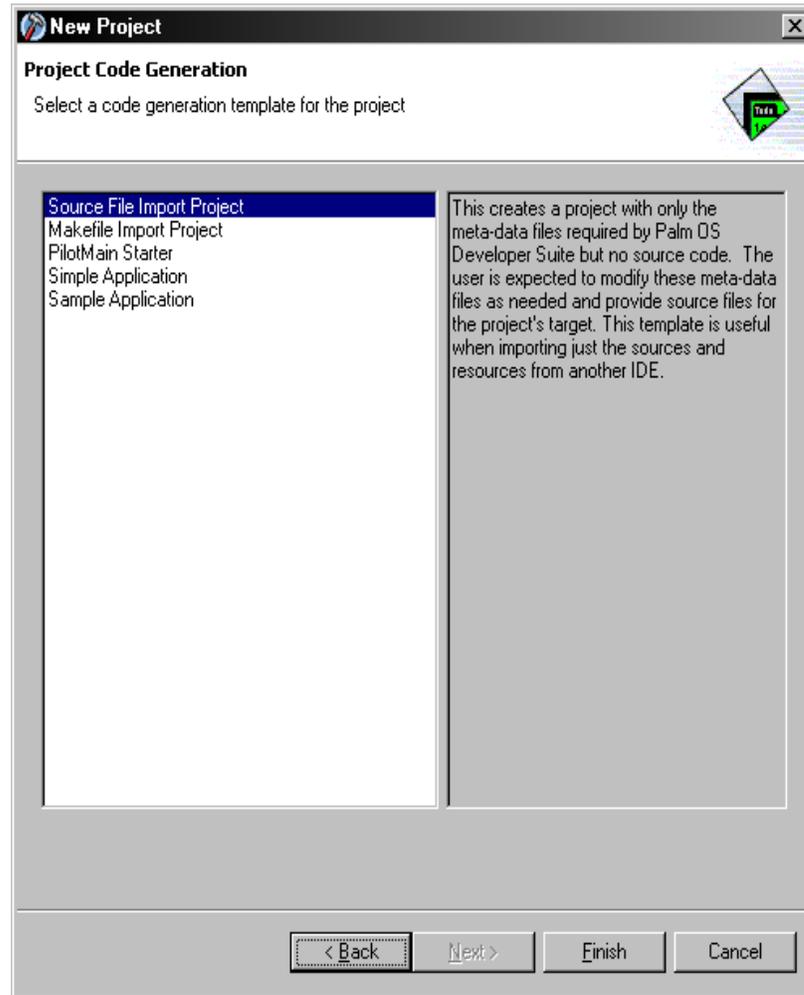
Figura 14. Ventana para definición del tipo de salida en un nuevo proyecto PODS



En esta ventana se selecciona la plantilla para generación de archivos de inicio de trabajo del proyecto. La mejor opción en este caso es Simple Application que genera los archivos makefile, genera una aplicación simple, unas referencias a recursos y una configuración inicial de secciones. Seleccionar entonces esta opción y hacer clic en Finish para terminar el asistente y permitir que PODS genere todos los archivos del proyecto. Cuando se termina el asistente y se generan los archivos del mismo, inmediatamente PODS genera por medio de make el prc. Crea entonces un archivo llamado stockquote.prc (si se dejaron los valores por defecto en la ventana del paso 4 del asistente). Este archivo es una base de datos tipo aplicación para Palm que se puede instalar y probar de inmediato en POSE (el emulador de dispositivos Palm) o en un dispositivo real. El archivo se ha probado y funciona adecuadamente; es un simple mensaje “Hello World” con una ventana “About” que se activa haciendo clic en el menú que muestra la pequeña aplicación. Esto es el inicio, el siguiente apartado muestra la forma de importar los fuentes del ejemplo

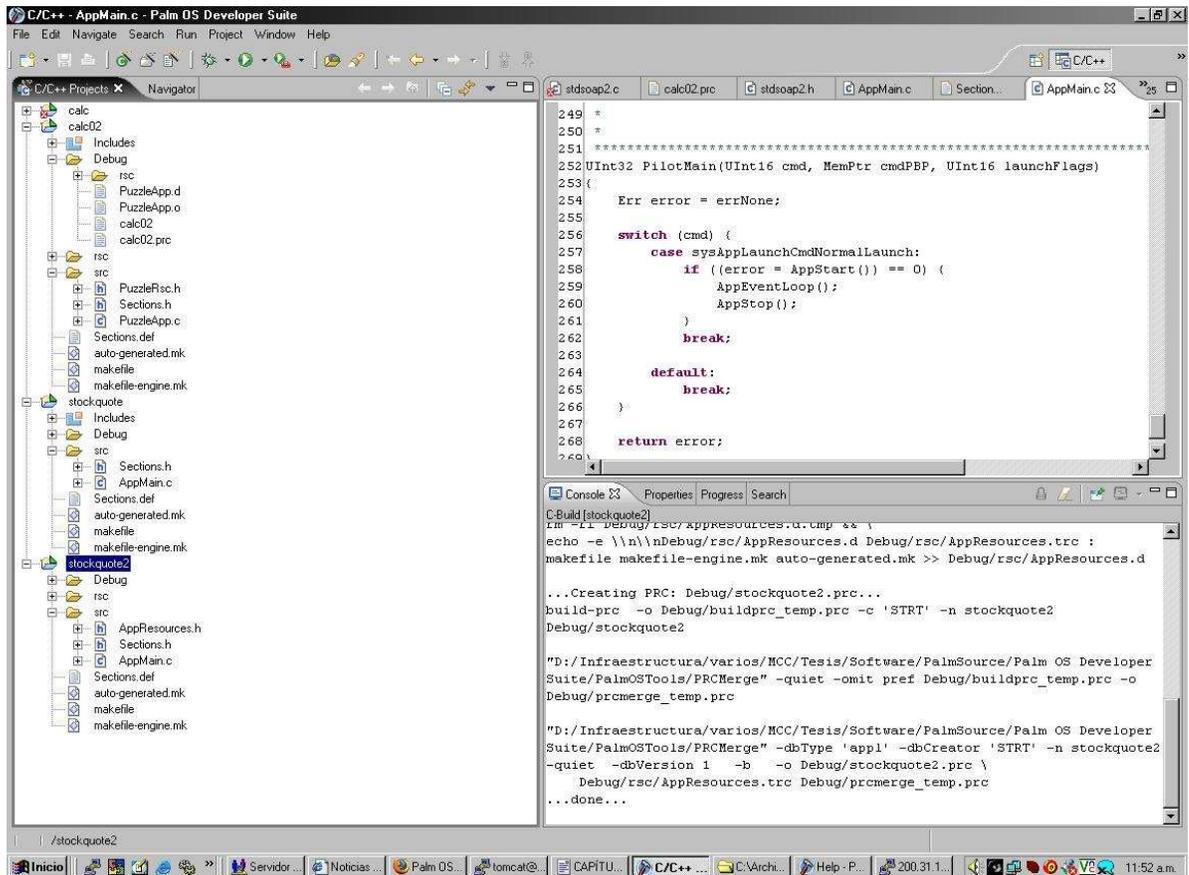
stockquote de gSOAP2. En la siguiente ventana se selecciona la plantilla para la generación de código. La siguiente figura ilustra las opciones descritas anteriormente.

**Figura 15. Ventana para la selección de plantilla en un nuevo proyecto PODS**



Aparece el árbol de directorios para trabajar el proyecto que se presenta en la siguiente figura:

Figura 16. La ventana de administración del proyecto en PODS



Aparecen igualmente ventanas de apoyo muy importantes. Se observa en la parte superior derecha la ventana de visualización del contenido de archivos. En la parte inferior derecha la pestaña de consola, que es la salida del avance de una compilación del proyecto, esta pestaña presenta paso a paso el avance de una compilación de proyecto, la pestaña de progreso, que sirve para visualizar cuál proceso en particular está en ejecución cuando se ha comenzado a crear el prc, la pestaña de búsqueda que sirve para visualizar los resultados de una búsqueda realizada mediante la opción del menú principal Search, la pestaña de propiedades generales del proyecto y la pestaña de problemas, que es un historial de problemas durante cada compilación.

Como nota adicional vale mencionar que PODS tiene activada la opción de auto-compilación por defecto, esto es, cada vez que se haga una modificación, compila de nuevo el proyecto y trata de generar el prc. Esta opción se puede desactivar en el la opción principal del menú "Project" y desactivando la opción Build Automatically.

Adicionalmente, la aplicación es muy pequeña y no necesita secciones. En caso de que fuera mayor de 68K, se debe configurar inicialmente el makefile para que tome secciones y luego configurar el fuente para que las tome. Las secciones se toman en el momento de realizar la creación del prc que es la última etapa. Ocurre luego del "Linking".

Para ilustrar mejor el proceso, se va a tomar como base el ejemplo particular de stockquote de gSOAP2, y es necesario entonces copiar en el directorio src los archivos soapClient.c, soapC.c y los archivos propios de la implementación de los clientes de gSOAP para Palm, estos son palmFunctions.c, palmFunctions.h, stdsoap2.c, stdsoap.h, soapH.h, soapStub.h igualmente en el directorio src del proyecto. Finalmente también copiar los archivos QuoteStarter.c y quote.h. Como se explica en la documentación de gSOAP2, es necesario que todos los .h sean llamados desde fuentes .c, y que todos los archivos fuentes .c sean compilados en el Makefile. El PODS genera un archivo Makefile, que puede ser modificado, y precisamente se debe modificar llamando entonces en el apartado Sources todos los archivos fuentes (con extensión .c o .cpp). Igualmente en el apartado se deben incluir con la sintaxis -I '.h a incluir' todos los archivos que automáticamente no se incluyan en el proyecto y que generen errores en la compilación de tipo 'undefined'. La ventana de Console del PODS explica el archivo y la línea con el error, y generalmente son errores de este tipo en los cuales no se encuentra la definición de alguna variable llamada desde un fuente o un encabezado. h.

Muy importante lo siguiente: como se mencionó anteriormente durante el documento, los dispositivos Palm tienen dos limitantes: una para archivos mayores a 64K en tamaño y otra con referencias que estén separadas por más de 32K en los código objeto. Realmente la programación para Palm es complicada según los autores de este trabajo de tesis, pues se dan algunos ejemplos que realmente no funcionan tal y como se explica, o no dan la totalidad de pautas a seguir para implementarlas correctamente. Un caso muy específico es el del archivo stdsoap2.c que tiene un tamaño de más de 280K lo cual impide usarlo tal cual en un proyecto para Palm. gSOAP en su ejemplo da una pequeña ayuda, ofreciendo dos archivos que separan el código; estos archivos son palmsoap1.c y palmsoap2.c. Estos archivos resuelven en algo el problema del tamaño pero no totalmente porque al final para el archivo palmsoap2.c se generan errores de compilación iguales a los de stdsoap2.c aunque en menor cantidad. Se decidió entonces por los autores de este trabajo separar no en dos si no en 3 el código lo que implicó hacer un análisis detallado del archivo original y ver la forma en que tomara para cada archivo partes de código determinadas y llamados particulares de funciones.

Bien, esto resuelve el inconveniente de tamaño de archivo pero no resuelve el inconveniente de referencias a más de 32K de distancia para la arquitectura de memoria de Palm. Por tal motivo se desechó esta solución y se tomó una sugerida en prc-tools con respecto a segmentación de código, esto se encuentra en la documentación en línea en: [http://prc-tools.sourceforge.net/doc/prc-tools\\_3.html#SEC17](http://prc-tools.sourceforge.net/doc/prc-tools_3.html#SEC17). Esto permitió eliminar los dos tipos de errores. Es una tarea bastante dispendiosa por cuanto:

1. Se usó un comando tipo UNIX: el split, para dividir stdsoap2.c en archivos de 28K de tamaño
2. Se hizo una tabla en una hoja de cálculo para referenciar la línea inicial y final de cada subarchivo de stdsoap2.c
3. En el archivo stdsoap2.c se crearon las referencias según la documentación de prc-tools para segmentar el código. Un ejemplo es:

## **Código 2. Ejemplo de referencia a sección en prc-tools para segmentación de código**

```

#ifndef PALM_1
SOAP_FMAC1
int
SOAP_FMAC2
soap_flush_raw(struct soap *soap, const char *s, size_t n)
EXTRA_SECTION_ONE;
SOAP_FMAC1
int
SOAP_FMAC2
soap_flush_raw(struct soap *soap, const char *s, size_t n)
{ if ((soap->mode & SOAP_IO) == SOAP_IO_STORE)
  { register char *t;
    if (!(t = (char*)soap push block(soap, n)))
      return soap->error = SOAP_EOM;
    memcpy(t, s, n);
#ifndef WITH_LEANER
    if (soap->fpreparesend)
      return soap->fpreparesend(soap, s, n);
#endif
    return SOAP_OK;
  }
#ifndef WITH_LEANER
  if ((soap->mode & SOAP_IO) == SOAP_IO_CHUNK)
  { char t[16];
    sprintf(t, "\r\n%lx\r\n" + (soap->chunksize ? 0 : 2),
(unsigned long)n);
    DBGMSG(SENT, t, strlen(t));
    if ((soap->error = soap->fsend(soap, t, strlen(t))))
      return soap->error;
    soap->chunksize += n;
  }
  DBGMSG(SENT, s, n);
#endif
  return soap->error = soap->fsend(soap, s, n);
}
#endif

```

Esta sección: EXTRA\_SECTION\_ONE, debe estar definida en el archivo Sections.h así:  
**#define** EXTRA\_SECTION\_ONE \_\_attribute\_\_((section("code1")))

Se usaron entonces trece secciones, que se definieron en los dos archivos, Sections.def y Sections.h del proyecto en PODS.

### 5.3 Pocket PC [13]

Para la programación en Pocket PC, los autores de este trabajo se han apoyado en el tutorial del enlace <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=soap>, un tutorial de apoyo muy claro para comenzar a utilizar el denominado pocketSoap para consumir Servicios Web desde un Pocket PC.

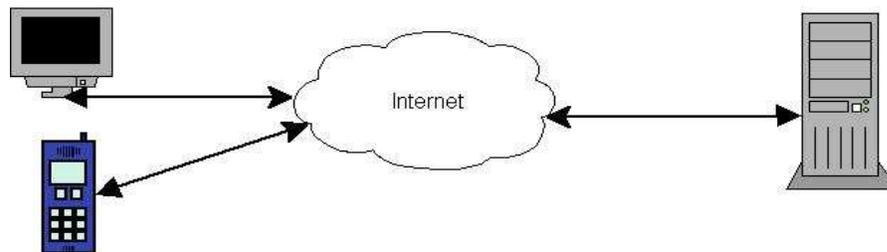
Es muy importante antes de relacionar el tutorial mencionar que existe una herramienta gratuita que ofrece Microsoft para la elaboración de aplicaciones para Pocket PC denominada Microsoft eMbedded Visual Tools que se consigue en el enlace <http://www.microsoft.com/downloads/details.aspx?FamilyID=F663BF48-31EE-4CBE-AAC5-0AFFD5FB27DD&displaylang=en>. Adicionalmente, es necesario apoyarse en las herramientas pocketSoap que son de uso libre para poder consumir Servicios Web desde Pocket PC y que se encuentran en el enlace <http://www.pocketsoap.com/pocketsoap/>.

A continuación el tutorial mencionado.

**5.3.1 Introducción a los Web Services con PocketSOAP, Apache SOAP y Axis [34].** Los sistemas distribuidos y el acceso a servicios remotos han centrado gran parte de los esfuerzos tecnológicos de los últimos años habiéndose creado distintas tecnologías y estándares distintos como por ejemplo: Java RMI, CORBA, COM/DCOM, las clásicas RPCs o los sistemas MOM basados en el paso de mensajes.

Con el crecimiento de la WWW surgió la idea de realizar estas tareas en sistemas distribuidos usando un protocolo estandarizado como HTTP:

**Figura 17. Esquema simplificado de los Servicios Web**



Los Servicios Web son básicamente funciones que se ejecutan en un servidor y son accedidas por los clientes mediante Internet.

Para realizar este tipo de comunicación es necesario definir un formato en el que se transmitirán esos datos. Aquí es donde entra en juego SOAP, WSDL y UDDI:

Servicios Web = SOAP + UDDI + WSDL

## SOAP , WSDL y UDDI

Simple Object Access Protocol (SOAP) (a veces también referido como Service Oriented Access Protocol) actualmente es un protocolo de especificación que define una forma uniforme de transmitir datos codificados en XML. A su vez también define una forma de realizar llamadas a procedimientos remotos (similar a RPCs).

La definición oficial que da el consorcio W3C: es la siguiente:

*SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.*

La siguiente tabla presenta enlaces importantes a las especificaciones SOAP.

**Tabla 7. Enlaces hacia las especificaciones SOAP 1.1 y SOAP 1.2**

<b>SOAP 1.1</b>	
Namespace Name	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
Spec Location	<a href="http://www.w3.org/TR/SOAP/">http://www.w3.org/TR/SOAP/</a>
<b>SOAP 1.2</b>	
Namespace Name	<a href="http://www.w3.org/2002/12/soap-envelope">http://www.w3.org/2002/12/soap-envelope</a>
Spec Location	<a href="http://www.w3.org/TR/soap12-part0/">http://www.w3.org/TR/soap12-part0/</a> (Primer) <a href="http://www.w3.org/TR/soap12-part1/">http://www.w3.org/TR/soap12-part1/</a> <a href="http://www.w3.org/TR/soap12-part2/">http://www.w3.org/TR/soap12-part2/</a>

Originalmente los autores de SOAP confiesan que se centraron en el "acceso a objetos" de ahí la O de las siglas, pero con el tiempo SOAP se centró en una forma de RPCs vía web.

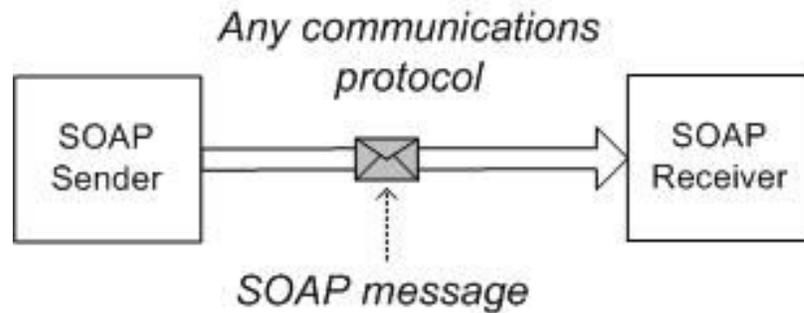
SOAP define el modo de transmitir mensajes XML de un punto a otro. Lo hace proporcionando un protocolo de mensajes que es:

1. Extensible (se basa en XML).
2. Usable por una gran variedad de protocolos de transporte de red. Una gran ventaja de SOAP es que puede usar cualquier protocolo de transporte como TCP, HTTP, SMTP o incluso MQs (colas de mensajes)

### 3. Independencia de las plataformas o lenguajes de programación

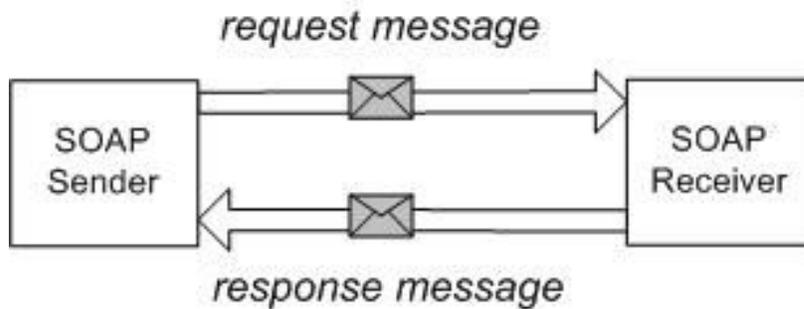
La siguiente figura ilustra muestra el modo de transferir mensajes en SOAP.

**Figura 18. Modo de transmitir mensajes en SOAP**



SOAP engloba cada mensaje dentro de un *sobre* (envelope), como lo presenta la siguiente figura:

**Figura 19. Envelope SOAP**



Por ejemplo si se quiere acceder a una función llamada `getQuote` al que se le pasa el nombre de una empresa y devuelve un *float* con el valor su *Stock Quote*.

Si se tratara de una función normal por ejemplo se realizaría esta llamada `float q = getQuote("XXX");` devolviendo el valor 97.8.

Si se realizara esta operación pidiendo el resultado a un Web Service los datos enviados y recibidos serían los siguientes.

Mensaje de Petición:

### Código 3. Datos enviados para una solicitud a un Servicio Web en el ejemplo de getQuote

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:n="urn:xmethods-delayed-quotes"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<n:getQuote>
<symbol xsi:type="xs:string">XXX</symbol>
</n:getQuote>
</soap:Body>
</soap:Envelope>
```

Mensaje de Respuesta:

### Código 4. Datos recibidos para una solicitud a un Servicio Web en el ejemplo de getQuote

```
<soap:Envelope soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
<soap:Body>
<n:getQuoteResponse xmlns:n="urn:xmethods-delayed-quotes">
<Result xsi:type="xsd:float">97.8</Result>
</n:getQuoteResponse>
</soap:Body>
</soap:Envelope>
```

Se observa que el formato del mensaje consiste en un *Envelope* donde se incluyen datos en XML. También se observa que se definen tipos predefinidos: **xsd:float**

La codificación del envelope puede estar codificado según diferentes filosofías:

- **rpc encoded:**  
Aquí se codifican los datos siguiendo el schema <http://schemas.xmlsoap.org/soap/envelope/>
- **doc /literal xml**  
Esta filosofía es la elegida por .NET donde el contenido del envelope se codifica como XML literal.
- **mensaje**  
Filosofía en la que el contenido del cuerpo del envelope es un mensaje con el formato XML

Para no tener que generar manualmente estos mensajes XML diferentes implementaciones de Servicios Web proporcionan APIs y frameworks tanto para programar clientes como Servicios Web en distintos lenguajes.

Aquí se puede encontrar una lista con algunas implementaciones para consumir y producir Servicios Web:

<http://www.xmethods.com/ve2/ViewImplementations.po>

Como se observa existen varias posibles implementaciones dependiendo de la plataforma.

En <http://www.xmethods.com> también se pueden encontrar una serie de Servicios Web consumibles por los clientes que se programarán para comprobar su funcionamiento y su interoperabilidad.

Una de las grandes posibilidades de los Servicios Web es proporcionar una gran interoperabilidad entre diferentes clientes y servicios. Para ello es necesario que el cliente sea capaz de obtener una descripción del servicio con el que se va a comunicar. Para esto existe el **Web Service Definition Language**. **WSDL** es un formato que describe las funciones de un Servicio Web así como los parámetros de entrada y salida que éste tiene. Se podría decir que WSDL es el IDL (Interface Definition Language) de los Servicios Web.

Si además se quiere proporcionar transparencia de localización al Servicio Web se usa **UDDI**, Universal Description, Discovery and Integration protocol. UDDI es un protocolo que crea una plataforma estándar e interoperable que permite a las compañías y aplicaciones, encontrar servicios dinámicamente en Internet (<http://www.uddi.org/>).

Se usará como cliente una PDA con Windows Pocket PC 2003 usando Embedded Visual C++ 4.0. Se elige la implementación PocketSoap (<http://www.pocketsoap.com/pocketsoap/>) para la parte cliente y para crear Servicios Web Axis (<http://ws.apache.org/axis/>) aunque también se explicara Apache SOAP (<http://ws.apache.org/soap/>). Los Servicios Web correrán sobre Tomcat.

Para el desarrollo del cliente SOAP para Windows Pocket PC 2003 se usará el entorno Embedded Visual C++ (eVC++) y el emulador de Pocket PC 2003.

PocketSOAP:

PocketSOAP es un componente COM cliente Open Source para la familia Windows, que se proporciona con la [MPL](#) (Mozilla Public License). Originalmente se diseñó para PocketPC, pero también existe una versión que funciona sobre Windows 95/98/Me/NT4/2000/XP:

<http://www.pocketsoap.com/pocketsoap/>

Este objeto COM puede ser usado desde aplicaciones escritas en distintos lenguajes, como por ejemplo Visual Basic y Visual C++.

## Instalación de las librerías de PocketSoap en el emulador Pocket PC 2003

Para instalar las librerías de PocketSoap en el emulador se baja el paquete para el emulador de:

<http://www.pocketsoap.com/pocketsoap/>

Se usará esta versión:

binaries for PocketPC 2002 emulator:

[http://www.pocketsoap.com/pocketsoap/pocketsoap\\_pp2e\\_1.4.3.zip](http://www.pocketsoap.com/pocketsoap/pocketsoap_pp2e_1.4.3.zip)

A pesar de indicar que son para el emulador Pocket PC 2002 también funcionan para el emulador Pocket PC 2003, que será el que se usará.

PocketSoap lo componen una serie de controles ActiveX. Estos controles tienen varios objetos COM que ofrecen interfaces para comunicarse con Servicios Web.

Para que las aplicaciones puedan usar estos controles ActiveX y puedan instanciarlos como objeto COM se deben copiar las DLLs y luego registrarlas en el sistema operativo. En el caso de Win32 se usará el programa regsvr32.exe, y en el caso de Pocket Windows 2003, regsvrce.exe.

En el caso de Win32 la aplicación regsvr32.exe se encuentra incluida con Windows. En cambio tanto en el emulador como en la pda con Pocket PC 2003 la aplicación regsvrce.exe no viene incluida. Esta aplicación se encuentra dentro del Microsoft Embedded C++ 4.0:

Para el caso del emulador:

<C:\bin\MicrosoftEmbeddedC++4.0\EVCWCE400\TARGET\X86>

Para el caso de PDAs con procesadores ARM y X-Scale

<C:\bin\MicrosoftEmbeddedC++4.0\EVCWCE400\TARGET\ARMV4>

Para poder usar desde el programa las DLLs proporcionadas por PocketSoap se deben copiar al directorio \Windows y registrarlas con regsvrce.exe.

Para realizar esta tarea se usa la capacidad del emulador para compartir directorios de nuestro PC. Este directorio se mostrará dentro del emulador como una tarjeta de memoria. Por lo tanto se copian las DLLs:

- PocketHTTP.dll
- psDime.dll

- pSOAP.dll
- xmlparse.dll
- xmltok.dll
- zlib.dll

**Figura 20. Propiedad de compartir carpetas entre PC y Pocket PC**



Al directorio que se comparte:

Se abre el explorador de ficheros (Inicio>Programas>Explorador de ficheros) y se va a la "Tarjeta de Almacenamiento" donde se mostrarán los ficheros del directorio del pc que se compartió.

Se observa que las DLLS no aparecen, por lo que se debe seleccionar la opción "Ver todos los ficheros":

Figura 21. Opción de ver todos los archivos en Pocket PC

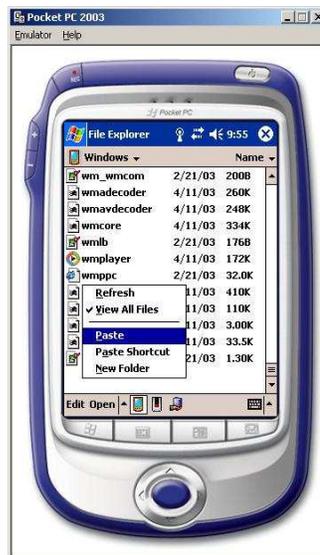


Figura 22. Opción copiar archivos en Pocket PC



Se copian los ficheros y se pegan en \Windows

Figura 23. Opción de pegar archivos en Pocket PC



Windows Pocket PC 2003 no tiene una Consola de comandos desde la cual poder ejecutar programas y pasarle parámetros. En cambio si tiene un menú para ejecutar comandos. Para que aparezca este menú es necesario situarse encima del reloj con el cursor y hacer click mientras se pulsa la tecla Control hasta que aparezca el siguiente menú:

Figura 24. Opción Menú para ejecutar comandos en Pocket PC



En el caso de una PDA real en vez de pulsar la tecla control se pulsa el botón principal de la PDA y se obtiene el mismo resultado:

Figura 25. Ventana para ejecutar comando en Pocket PC



En estos momentos se tienen las DLLs que se necesitan en el directorio \Windows así como el programa regsvrce.exe para registrarlas en el sistema.

En caso de no registrar correctamente las DLLs el programa producirá el siguiente error:

Figura 26. Error cuando no se han registrado de forma correcta archivos dll en Pocket PC



Para que los programas puedan usar PocketSOAP correctamente se debe ejecutar regsvrce con las siguientes DLLs:

- pocketHTTP.dll
- psoap.dll
- psdime.dll

Figura 27. Ejecución de regsvrce en Pocket PC



Figura 28. Mensaje de instalación correcta del archivo dll



Desgraciadamente esta tarea se tiene que realizar cada vez que se apague y se reinicie el emulador.

### Instalación de las librerías de PocketSoap en la PDA (iPaq 2210)

Para instalar las librerías de PocketSoap en la PDA se debe bajar el paquete para Pocket PC de:

<http://www.pocketsoap.com/pocketsoap/>

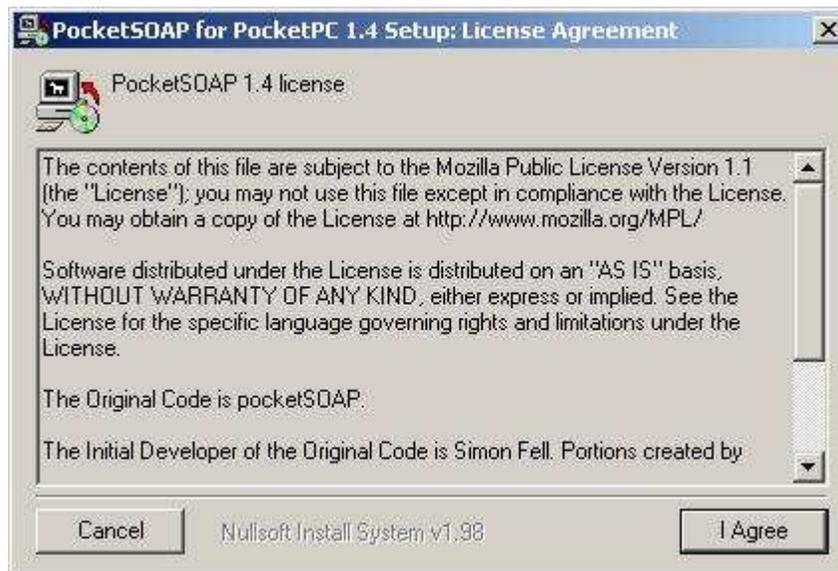
En el ejemplo se usará esta versión:

[PocketPC Version v1.4.3 Packaged Install](#)

que es compatible con procesadores ARM y X-Scale como nuestra iPaq 2210:

Para instalar correctamente es necesario tener la PDA conectada al PC y ejecutar desde el PC el programa de instalación:

### Figura 29. Ventana de License Agreement en PocketSOAP



Este instalador creado con [NSIS](#) copiará un ejemplo ejecutable, las dlls correspondientes y las registrará en el sistema Windows Pocket PC 2003 por nosotros.

Ejemplo de cliente SOAP con PocketSoap:

En la página de PocketSOAP existe una sección de ejemplos de clientes SOAP: [Samples](#)

En la sección de [eVC](#) se encuentra el ejemplo getQuote:

## [PocketPC Stock Quote](#)

This is the eVC source code for the Stock Quote demo that installs with PocketSOAP on PocketPC's. This is calling the [XMethods stock quote](#) service.

Como lo indica esta página este programa se conecta con un Servicio Web de la página de XMethods.

Una vez descargado el ejemplo y descomprimido es necesario estar seguro de que los ficheros tienen permisos de escritura.

El entorno de desarrollo será Embedded Visual C++ 4.0 con el emulador de Pocket PC 2003. Para que el ejemplo funcione con este entorno se deben realizar varios cambios:

- pSOAP.h: El ejemplo getQuote en el fichero psQuote2Dlg.cpp hace referencia al fichero pSOAP.h, este fichero se encuentra dentro del código fuente de PocketSOAP: [v1.4.3 Source Code](#), por lo que se tiene que añadir este fichero al proyecto e incluirlo: #include "pSOAP.h".
- Para poder usar este ejemplo con el emulador de Pocket PC 2003 se debe crear una nueva configuración. Para ello se selecciona en el menú Build > Configurations

**Figura 30. Creación de una nueva configuración para uso del Emulador Pocket PC 2003 en Microsoft eMbedded Visual C++ 4.0**

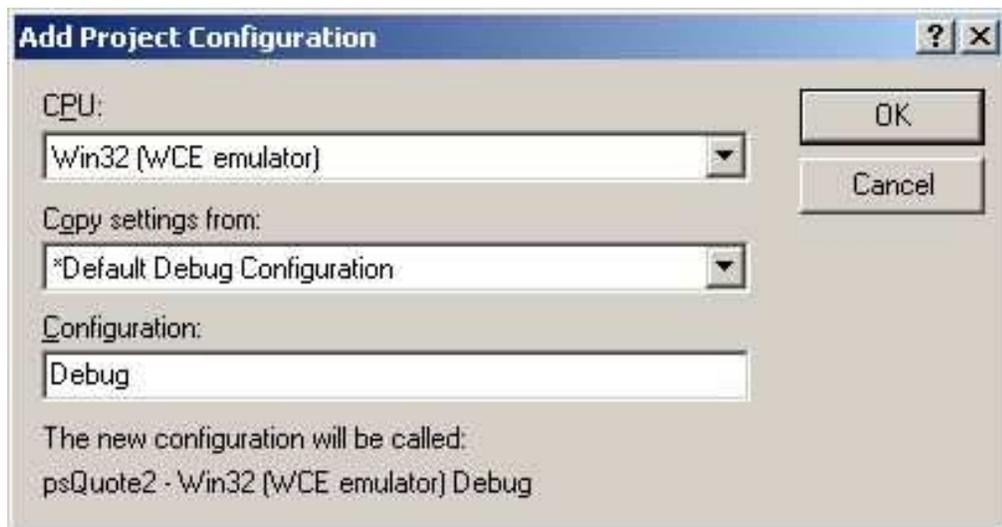


Figura 31. Adición de una configuración a un proyecto en Microsoft eMbedded Visual C++ 4.0

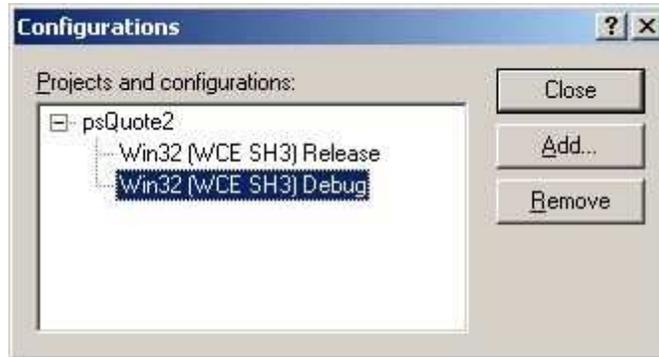
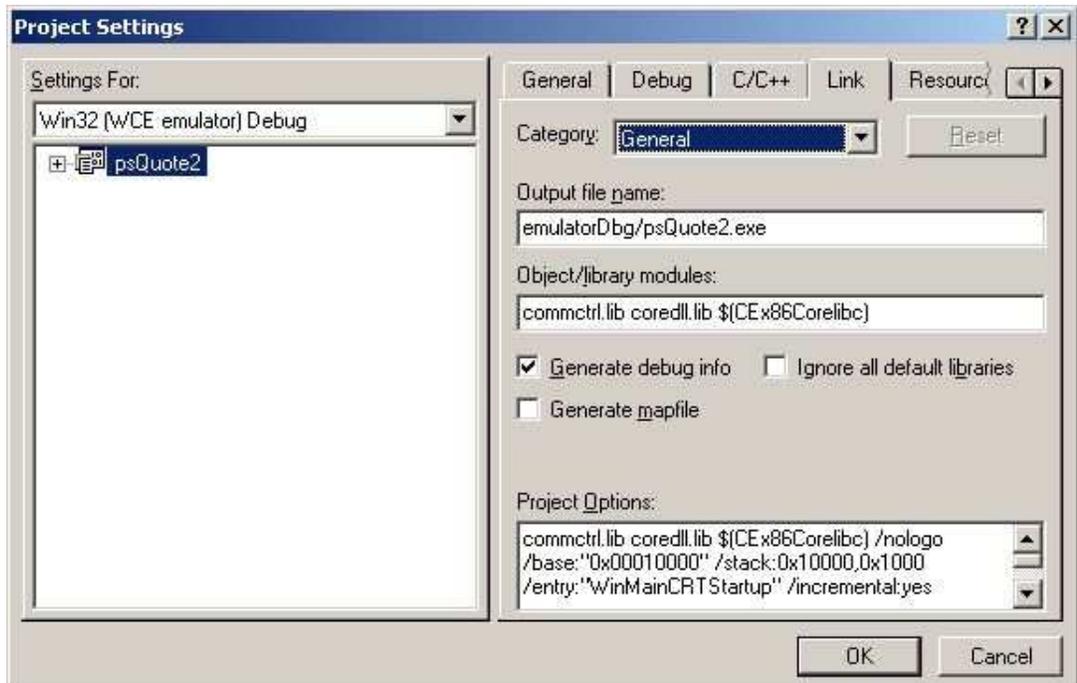


Figura 32. Barra de resumen de configuración en Microsoft eMbedded Visual C++ 4.0



- Finalmente se debe seleccionar en el menú Project > Settings y en la pestaña Link:

Figura 33. Configuración de proyectos en Microsoft eMbedded Visual C++ 4.0



Debemos cambiar /entry: "WinMainCRTStartup" por /entry: "wWinMainCRTStartup"

Finalmente se tendrá el ejemplo de PocketSOAP funcionando:

**Figura 34. Ejemplo getQuote en Pocket PC**



En <http://www.xmethods.com> se pueden encontrar varios Servicios Web con los que se puede probar el cliente.

Para dudas del API de PocketSoap, solución de dudas o errores en Yahoo está el siguiente grupo que sin duda será de gran ayuda:

<http://groups.yahoo.com/group/pocketsoap/>

## 5.4 SERVICIOS WEB EN EL LADO DEL SERVIDOR

### 5.4.1 Axis Java [7].

<http://ws.apache.org/axis/java/index.html>

### 5.4.2 Axis C++ [7].

<http://ws.apache.org/axis/cpp/documentation.html>,  
[http://www.programacion.com/java/tutorial/servic\\_web/1/](http://www.programacion.com/java/tutorial/servic_web/1/)

### 5.4.3 .NET Framework [8]

<http://www.oreilly.com/catalog/dotnetfrmess/chapter/ch06.html>, <http://www.c-sharpcorner.com/WebForms/WebServicesP1RSR.asp>

## **6. DISCUSIÓN Y ANÁLISIS DE RESULTADOS**

### **6.1 INTRODUCCIÓN**

Este capítulo contiene los resultados obtenidos en la ejecución de pruebas utilizando el Banco de Pruebas esquematizado en el CAPÍTULO TRES - DISEÑO DE LAS PRUEBAS EN SERVICIOS WEB MÓVILES. Los resultados que se describen contienen principalmente usos de ancho de banda determinados por el tamaño y cantidad de los paquetes emitidos por los Servicios Web de prueba, de acuerdo a las determinaciones del CAPÍTULO IV - PARÁMETROS DE MEDICIÓN. Realmente el parámetro en el que se centra la investigación es el uso del ancho de banda, pues, como se concluye según los análisis que siguen, el procesamiento realmente no se afectaría en gran medida cuando usamos compresión de datos para aprovechar el ancho de banda disponible.

Debido a que el recurso que se considera más crítico es la conectividad, el análisis del tamaño de esos mensajes enviados y recibidos desde y hacia la Palm es factor primordial para determinar recomendaciones y conclusiones en el presente trabajo.

Bien, inicialmente se hace un análisis del tamaño de los mensajes SOAP (Lo estándar de Servicios Web usando formatos XML) transmitidos por HTTP 1.1 (el protocolo estándar actual para el protocolo HTTP), para dos ejemplo específicos en cada tipo de plataforma, es decir, tanto en Windows como en Linux.

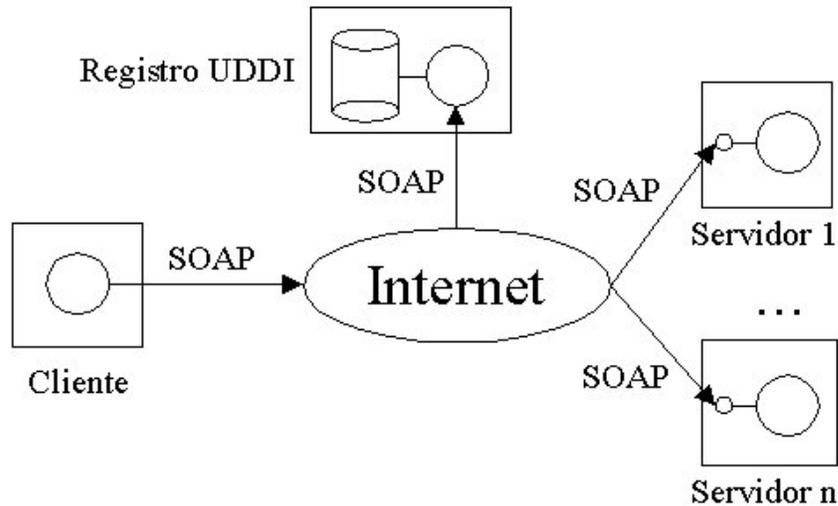
Luego de esto, y teniendo en cuenta que lo más crítico es la optimización del uso del ancho de banda, se hacen análisis de compresión de información usando el algoritmo más común para compresión de información, el gzip. Este análisis se hace tanto en Microsoft Windows como en Linux, para medir uso del procesador en ambos casos.

Finalmente, se hace el mismo análisis para cada PDA, Palm y Pocket.

### **6.2 CANTIDAD DE BYTES TRANSMITIDOS EN AMBOS SENTIDOS (ANCHO DE BANDA)**

SOAP es el protocolo por excelencia en el envío de mensajes de tipo XML para los Servicios Web, como se observa en la siguiente figura:

Figura 35. Funcionamiento básico de los Servicios Web usando SOAP



El análisis del tamaño de mensajes se debe hacer con respecto a inicialmente archivos escritos para una plataforma específica, luego en archivos XML que representan mensajes SOAP generados por cada plataforma, y finalmente a cómo en HTTP 1.1 se generarían esos mensajes que al final son los que se envían por la red para usar el ancho de banda.

A continuación se presentan los casos de dos ejemplos específicos en cada uno de los sistemas, Windows con .NET Framework y Linux con Apache AXIS.

En Microsoft .NET Framework se generan dos ejemplos, uno en Visual Basic y otro en C#. Para el caso de Linux se generan en C.

**6.2.1 .NET Framework en Windows.** El análisis del tamaño de los mensajes se ha hecho en base a dos ejemplos, un ejemplo básico de Servicio Web MathService.asmx escrito en Visual Basic y otro ejemplo básico FactorialServiceC.asmx de cálculo de raíz cuadrada y escrito en C; los Servicio Web están en ASP.NET y emitidos con IIS y .NET Framework 1.1. El primer servicio es una simple calculadora a la que un cliente le pregunta cuál es el resultado de las operaciones básicas, multiplicación, división, resta y suma; el segundo servicio recibe como parámetro un número y devuelve el factorial.

Los ejemplos se describen a continuación:

### Código 5. Código en ASP para el ejemplo de MathService - Servicio Web de calculadora sencilla

```
<%@ WebService Language="VB" Class="MathService" %>

Imports System
Imports System.Web.Services

Public Class MathService : Inherits WebService

    <WebMethod()> Public Function Add(A As System.Single, B As System.Single) As System.Single

        Return A + B
    End Function

    <WebMethod()> Public Function Subtract(A As System.Single, B As System.Single) As System.Single

        Return A - B
    End Function

    <WebMethod()> Public Function Multiply(A As System.Single, B As System.Single) As System.Single

        Return A * B
    End Function

    <WebMethod()> Public Function Divide(A As System.Single, B As System.Single) As System.Single

        If B = 0
            Return -1
        End If
        Return Convert.ToSingle(A / B)
    End Function

End Class
```

### Código 6. Código en ASP para el ejemplo de Factorial - Servicio Web de cálculo de factorial

```
<%@ WebService Language="C#" class="MyClass" %>

using System.Web.Services ;

public class MyClass
{
    [WebMethod()]
    public int Factorial ( int a)
    {
        int cont = 1 ;
        int Fact = 1 ;
        while ( cont <= a )
        {
            Fact = Fact * cont ;
            cont = cont + 1 ;
        }
        return Fact ;
    }
}
```

Basta con copiar y pegar cada ejemplo en un procesador de texto sencillo y ejecutarlo en un servidor que tenga instalado el servicio Microsoft IIS y Microsoft .NET Framework 1.1. Con detalle puede ver cómo implementar estos servicios en el ANEXO 1. CONSIDERACIONES DE SERVICIOS WEB CON MICROSOFT.

Para los ejemplos de los Servicios Web MathService y FactorialService, se genera la siguiente información de transporte de mensajes:

### Código 7. Contenido SOAP para el ejemplo básico de la calculadora en la operación Multiplicación en ASP.NET

```
POST /varios/Desarrollo/MathService.asmx HTTP/1.1
Host: xxx.xxx.xxx.xxx
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Multiply"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Multiply xmlns="http://tempuri.org/">
      <A>float</A>
      <B>float</B>
    </Multiply>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <MultiplyResponse xmlns="http://tempuri.org/">
      <MultiplyResult>float</MultiplyResult>
    </MultiplyResponse>
  </soap:Body>
</soap:Envelope>
```

## Código 8. Contenido SOAP para el ejemplo básico de Factorial ServiceC en ASP.NET

```
POST /varios/Desarrollo/FactorialServiceC.asmx HTTP/1.1
Host: xxx.xxx.xxx.xxx
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Factorial"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Factorial xmlns="http://tempuri.org/">
      <a>int</a>
    </Factorial>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FactorialResponse xmlns="http://tempuri.org/">
      <FactorialResult>int</FactorialResult>
    </FactorialResponse>
  </soap:Body>
</soap:Envelope>
```

Las tablas anteriores presentan absolutamente todo el contenido ya en formato del protocolo HTTP 1.1, que es en definitiva la cantidad de bytes que se deben considerar, pues HTTP 1.1 es lo que realmente se transporta por la red.

Bien, se omiten los códigos de las otras operaciones por ser casi idénticos en la forma, pero se presenta más adelante una tabla comparativa del tamaño de solicitudes y respuestas en cada operación.

La siguiente tabla presenta el tamaño en bytes para el ejemplo MathServices:

**Tabla 8. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web MathService**

Tipo de mensaje	Multiplicación	División	Resta	Suma
Solicitud SOAP	513	507	513	572
Respuesta SOAP	441	443	441	347

Esta tabla presenta para el ejemplo básico MathService el tamaño en bytes que se envía (la solicitud desde el PDA y la recepción de la solicitud en el servidor) en HTTP 1.1 y lo que se recibe (se recibe la respuesta SOAP en el PDA y es enviada desde el servidor) igualmente en HTTP 1.1 del ejemplo básico realizado en Visual Basic para ASP.NET en la plataforma Microsoft .NET Framework. En este caso particular el servicio tiene 4 métodos, uno para cada una de las operaciones básicas (multiplicación, división, resta y suma).

La siguiente tabla presenta el tamaño de lo emitido por el ejemplo FactorialServiceC:

**Tabla 9. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web FactorialServiceC**

Tipo de mensaje	Tamaño en bytes
Solicitud SOAP	502
Respuesta SOAP	414

Esta tabla presenta para el ejemplo básico FactorialServiceC el tamaño en bytes que se envía (la solicitud desde el PDA y la recepción de la solicitud en el servidor) en HTTP 1.1 y lo que se recibe (se recibe la respuesta SOAP en el PDA y es enviada desde el servidor) igualmente en HTTP 1.1 del ejemplo básico realizado en C# para ASP.NET en la plataforma Microsoft .NET Framework. Este ejemplo sólo tiene un método debido a que se trata de la solicitud única de factorial a partir de un argumento dado.

### **6.2.2 Linux.** Se usan dos ejemplos: Calculator y Factorial.

Para el desarrollo de Servicios Web con el lenguaje de programación C y Apache Axis bajo Linux, se puede encontrar información adicional en [35]. Los ejemplos se describen a continuación:

## Código 9. Código en C para el ejemplo de Calculator - Servicio Web de calculadora sencilla en Linux

```
Archivo Calculator.cpp:
/*
 *This is the Service implementation CPP file generated by theWSDL2Ws.
 *   Calculator.cpp: implementation for the Calculator.
 *
 */
#include "Calculator.h"

Calculator::Calculator()
{
}

Calculator::~Calculator()
{
}

int Calculator::add(int Value0, int Value1)
{
    return Value0+Value1;
}
int Calculator::sub(int Value0, int Value1)
{
    return Value0-Value1;
}
int Calculator::mul(int Value0, int Value1)
{
    return Value0*Value1;
}
int Calculator::div(int Value0, int Value1)
{
    if (Value1 == 0) return -1; /* :) */
    return Value0/Value1;
}

Archivo Calculator.h:
/*
 * This is the Service Class generated by the tool WSDL2Ws
 * Calculator.h: interface for the Calculatorclass.
 *
 */
#ifdef __CALCULATOR_SERVERSKELETON_H_OF_AXIS_INCLUDED_
#define __CALCULATOR_SERVERSKELETON_H_OF_AXIS_INCLUDED_

#include <axis/AxisUserAPI.hpp>

class Calculator
{
public:
    Calculator();
public:
    virtual ~Calculator();
public:
    int add(int Value0,int Value1);
    int sub(int Value0,int Value1);
    int mul(int Value0,int Value1);
    int div(int Value0,int Value1);
};
#endif /* !defined(__CALCULATOR_SERVERSKELETON_H_OF_AXIS_INCLUDED_)*/
```

## Código 10. Código en C para el ejemplo de Factorial - Servicio Web de cálculo de factorial en Linux

```
/*
 *This is the Service implementation CPP file generated by theWSDL2Ws.
 *   Factorial.cpp: implementation for the Factorial.
 *
 */
#include "Factorial.h"

Factorial::Factorial()
{
}

Factorial::~Factorial()
{
}

int Factorial::factor(int Value0)
{
    int fact=1;
    for (cont=1;cont<=Value0;cont++)
        fact=fact*cont;
    return fact;
}
[root@correo Factorial]# cat Factorial.h
/*
 * This is the Service Class generated by the tool WSDL2Ws
 * Factorial.h: interface for the Factorialclass.
 *
 */
#ifndef __FACTORIAL_SERVERSKELETON_H_OF_AXIS_INCLUDED__
#define __FACTORIAL_SERVERSKELETON_H_OF_AXIS_INCLUDED__

#include <axis/AxisUserAPI.hpp>

class Factorial
{
public:
    Factorial();
public:
    virtual ~Factorial();
public:
    int factor(int Value0);
};

#endif /* !defined(__FACTORIAL_SERVERSKELETON_H_OF_AXIS_INCLUDED__)*/
```

En este caso se hace necesario tener instalado el servidor web Apache y el emisor de Servicios Web Apache Axis. Para lograr obtener los denominados “ejecutables” a partir de estos fuentes y librerías, y para ver la forma en que se pueden desplegar desde el servidor u obtener los mismos con un cliente en la Internet, refiérase al apartado 5.4.2 Axis C++.

Para los ejemplos de los Servicios Web Calculator y Factorial, se genera la siguiente información de transporte de mensajes:

### **Código 11. Contenido SOAP para el ejemplo básico de la calculadora en la operación Multiplicación en Linux con Axis**

```
POST /axis/Calculator HTTP/1.1
Host: xxx.xxx.xxx.xxx
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/mul"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <mul xmlns="http://tempuri.org/">
      <A>float</A>
      <B>float</B>
    </mul>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <mulResponse xmlns="http://tempuri.org/">
      <mulResult>float</mulResult>
    </mulResponse>
  </soap:Body>
</soap:Envelope>
```

## Código 12. Contenido SOAP para el ejemplo básico de Factorial en Linux con Axis

```
POST /axis/Factorial HTTP/1.1
Host: xxx.xxx.xxx.xxx
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Factorial"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Factorial xmlns="http://tempuri.org/">
      <a>int</a>
    </Factorial>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FactorialResponse xmlns="http://tempuri.org/">
      <FactorialResult>int</FactorialResult>
    </FactorialResponse>
  </soap:Body>
</soap:Envelope>
```

Al igual que en el caso de .NET, las tablas anteriores presentan absolutamente todo el contenido ya en formato del protocolo HTTP 1.1, que es en definitiva la cantidad de bytes que se deben considerar, pues HTTP 1.1 es lo que realmente se transporta por la red.

Igualmente, se omiten los códigos de las otras operaciones por ser casi idénticos en la forma, pero se presenta más adelante una tabla comparativa del tamaño de solicitudes y respuestas en cada operación.

La siguiente tabla presenta el tamaño en bytes para el ejemplo Calculator.

**Tabla 10. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web Calculator en Linux con Axis**

Tipo de mensaje	Multiplicación	División	Resta	Suma
Solicitud SOAP	479	479	479	479
Respuesta SOAP	421	421	421	421

Esta tabla presenta para el ejemplo básico Calculator el tamaño en bytes que se envía (la solicitud desde el PDA y la recepción de la solicitud en el servidor) en HTTP 1.1 y lo que se recibe (se recibe la respuesta SOAP en el PDA y es enviada desde el servidor) igualmente en HTTP 1.1 del ejemplo básico realizado en C/C++ para Apache Axis en Linux. En este caso particular el servicio tiene 4 métodos, uno para cada una de las operaciones básicas (multiplicación, división, resta y suma).

La siguiente tabla presenta el tamaño de lo emitido por el ejemplo Factorial:

**Tabla 11. Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web Factorial en Linux con Axis**

Tipo de mensaje	Tamaño en bytes
Solicitud SOAP	476
Respuesta SOAP	443

Esta tabla presenta para el ejemplo básico Factorial el tamaño en bytes que se envía (la solicitud desde el PDA y la recepción de la solicitud en el servidor) en HTTP 1.1 y lo que se recibe (se recibe la respuesta SOAP en el PDA y es enviada desde el servidor) igualmente en HTTP 1.1 del ejemplo básico realizado en C/C++ para Apache Axis en Linux. Este ejemplo sólo tiene un método debido a que se trata de la solicitud única de factorial a partir de un argumento dado.

### **6.3 LA COMPRESIÓN Y EL USO DEL PROCESAMIENTO**

El protocolo de transporte de mensajes estándar y de mejor implementación en Servicios Web es SOAP. SOAP usa un formato XML para esquematizar sus mensajes. Pero el XML es más extenso que el HTTP puro. El protocolo estándar para transportar XML al final resulta siendo el protocolo HTTP, en lo cual lo convierte el servidor de Servicios Web para ser emitido por Internet. Cuando se trata de ahorrar ancho de banda así se use sólo HTTP para contenidos estáticos o dinámicos es muy conveniente comprimir HTTP, pues adicionalmente XML es más extenso que el lenguaje que se usa para transmitir contenido neto de HTTP.

El uso de la compresión HTTP reduce el ancho de banda de la red que se utiliza para transferir cada archivo desde el servidor al cliente. Además, ayuda a reducir el tiempo de descarga y a mejorar el aprovechamiento en los usuarios finales. La compresión HTTP utiliza generalmente el algoritmo estándar GZIP, y generalmente están integrados en los servidores y en los clientes en la actualidad. Los algoritmos de compresión y descompresión comprimen y almacenan en caché archivos estáticos y, opcionalmente, realizan compresión a petición de las respuestas generadas de forma dinámica antes de enviarlas a través de la red. Estos mismos algoritmos se utilizan de nuevo para descomprimir los archivos estáticos y las respuestas dinámicas en un cliente compatible con HTTP 1.1.

La primera vez que un cliente preparado para HTTP 1.1 solicita un archivo estático, se le envía el archivo sin comprimir al cliente. Se guarda entonces una copia comprimida del archivo estático en la carpeta temporal designada. La próxima vez que un cliente preparado para HTTP 1.1 solicita el archivo, se le envía la versión comprimida desde la carpeta temporal. El servidor también hace un seguimiento de las versiones originales sin comprimir de los archivos estáticos. Cuando se actualizan los archivos estáticos originales, se actualizan y vuelven a comprimir los archivos en la carpeta temporal. El resultado dinámico de las aplicaciones no se almacena en caché en la carpeta temporal, sino que el resultado de las secuencias de comandos se comprime cada vez que se envía a un cliente.

Compresión HTTP no es lo mismo que compresión de archivos y carpetas. Estos métodos de compresión realizan tareas muy diferentes, como se describe en la tabla siguiente.

**Tabla 12. Tareas en la compresión de HTTP y de Archivos y Carpetas**

<b>Compresión HTTP</b>	<b>Compresión de archivos y carpetas</b>
<ul style="list-style-type: none"> <li>• Almacena en caché una versión comprimida de los archivos estáticos en una carpeta temporal.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprime el archivo original para ahorrar espacio de disco.</li> </ul>
<ul style="list-style-type: none"> <li>• Los archivos leídos de la carpeta temporal se escriben en memoria en estado comprimido.</li> </ul>	<ul style="list-style-type: none"> <li>• Descomprime el archivo en memoria cuando se lee del disco.</li> </ul>
<ul style="list-style-type: none"> <li>• Envía archivos comprimidos a través de la red, lo que reduce el uso del ancho de banda de la red y mejora los tiempos de descarga.</li> </ul>	<ul style="list-style-type: none"> <li>• Envía archivos sin comprimir a través de la red, por lo que no reduce el uso del ancho de banda de la red ni mejora los tiempos de descarga.</li> </ul>

Estos dos métodos de compresión no pueden utilizarse simultáneamente en los mismos archivos.

Al final de este capítulo se concluyen algunas situaciones en las cuales se recomienda o no la compresión, pero es una definitiva que cuando el ancho de banda es limitado es muy necesaria la compresión pues de acuerdo a los resultados presentados en cada plataforma, es muy posible usarla.

A continuación se relacionan los análisis y resultados obtenidos usando gzip para comprimir y descomprimir las solicitudes y respuestas en cada una de las plataformas.

Se usa gzip por ser un estándar, además de ser utilizado en todos los servidores de HTTP cuando se habilita la compresión, adicionalmente es de tipo GPL y una de las que ofrece mejor compresión.

**6.3.1 Microsoft .NET Framework.** Como se ha mencionado en varias ocasiones en este trabajo, el ancho de banda es el recurso más crítico en la comunicación entre un dispositivo móvil de tipo PDA y un servidor en Internet dada la novedad de las tecnologías para la interconexión, la escasez de este y por consiguiente el costo de tal conexión, dándose el caso principalmente en las WWAN como GPRS en que se cobra no por tiempo sino por cantidad de información transmitida por el canal en un momento determinado.

Por tal motivo las pruebas se centraron principalmente en, además de hacer un cálculo de los bytes que consume un servicio muy sencillo como el de MathService (Servicio Web de calculadora para las operaciones multiplicación, división, resta y suma) y FactorialServiceC (Cálculo de un factorial de un número dado) en un formato totalmente estándar como el XML transmitido al último nivel con HTTP usando el protocolo estándar SOAP.

**6.3.1.1 Uso de procesamiento en servidor.** Ahora bien, con respecto a la compresión usando gzip, que es de los dos algoritmos que ofrece el servidor web el que más comprime los datos, se tiene que debe ser extremadamente grande el archivo a comprimir para que de una u otra forma se consuma el procesador de la máquina.

La siguiente tabla resume las pruebas y los resultados obtenidos al ejecutar compresiones de archivos con formato XML en el servidor, con un sólo usuario concurrente.

**Tabla 13. Resultados de comprimir XML en el servidor Windows con gzip con una sola petición**

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Consumo en procesamiento para descomprimir
Archivo de 1K	1058	348	3.04	Tiende a cero	Tiende a cero
Archivo de 10K	10580	426	24.84	Tiende a cero	Tiende a cero
Archivo de 100K	105800	985	107.41	Tiende a cero	Tiende a cero
Archivo de 1000K (1 MB)	1058000	6227	169.91	Tiende a cero	Tiende a cero
Archivo de 10000K (10 MB)	10580000	58541	180.73	Tiende a cero	Tiende a cero

Esto demuestra que para el ejemplo MathService, el consumo de procesamiento en el servidor, teniendo en cuenta que los archivos miden menos de 1K tiende a cero cuando hay una sola petición o respuesta.

Según se observa en las Tablas 10 y 11 referentes al tamaño de los mensajes de solicitud o envío SOAP, estos tamaños originales en HTTP 1.1 no sobrepasan 1K, lo cual quiere decir que comprimidos van a tener un tamaño aproximado del 33% del tamaño original. Según esto, todas las peticiones o solicitudes de ejemplos similares estarían en el rango de menos de 1K. Adicionalmente, es muy inusual encontrar tamaños de peticiones mayores de 10K, por lo cual las mediciones presentadas en la tabla anterior, pueden ofrecer datos importantes para concluir qué tanto consumo de procesamiento existe en un ambiente en el cual se realiza una sola petición al servidor en la Plataforma Windows usando gzip como algoritmo de compresión.

Como se observa en la tabla anterior, cuando hay un sólo proceso de compresión, sin importar qué tan grande es el archivo con texto del tipo HTTP 1.1 (hasta 10000 Kbytes, algo exageradamente grande para un ambiente de Servicios Web en el que las peticiones son muy pequeñas y en las que incluso más de 10Kbytes es supremamente inusual), se observa que el uso del procesamiento en el servidor de Servicios Web de pruebas con Plataforma Windows siempre tiende a cero.

Las siguientes tablas muestran los resultados de las pruebas usando 10 y 100 procesos.

**Tabla 14. Resultados de comprimir SOAP XML en el servidor Windows con gzip y 10 peticiones**

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Consumo en procesamiento para descomprimir
Archivo de 1K	1058	348	3.04	Tiende a cero	Tiende a cero
Archivo de 10K	10580	426	24.84	Tiende a cero	Tiende a cero
Archivo de 100K	105800	985	107.41	Tiende a cero	Tiende a cero
Archivo de 1000K (1 MB)	1058000	6227	169.91	Tiende a cero	Tiende a cero
Archivo de 10000K (10 MB)	10580000	58541	180.73	Tiende a cero	Tiende a cero

En la tabla anterior, se puede concluir que para 10 procesos concurrentes ejecutando una labor de compresión al máximo factor (gzip tiene factores de compresión del 1 al 9, consumiendo más procesamiento el 9, pero generando mayor compresión), el consumo de procesamiento del algoritmo gzip tiende a cero sin importar qué tan grande es el archivo conteniendo HTTP 1.1 en la plataforma Microsoft Windows.

**Tabla 15. Resultados de comprimir SOAP XML en el servidor Windows con gzip y 100 peticiones**

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Pico de procesamiento en compresión	Promedio en procesamiento para compresión
Archivo de 1K	1058	348	3.04	1.56	Tiende a cero
Archivo de 10K	10580	426	24.84	3.13	Tiende a cero
Archivo de 100K	105800	985	107.41	40.00%	5 segundos
Archivo de 1000K (1 MB)	1058000	6227	169.91	50.00%	10 segundos
Archivo de 10000K (10 MB)	10580000	58541	180.73	50.00%	60 segundos

En la tabla anterior, que presenta un ambiente más acorde a lo que se presentaría en un ambiente real de producción de Servicios Web en Internet, se puede concluir que para 100 procesos concurrentes ejecutando una labor de compresión al máximo factor, el consumo de procesamiento del algoritmo gzip tiende a cero en los casos de archivos del rango de 1Kbyte a 10Kbytes, lo normal en un ambiente de solicitud petición de Servicios Web transportando HTTP 1.1. Para archivos mayores de 10Kbytes, en el rango de hasta 10000Kbytes se denota un consumo de procesamiento que puede llegar a tener el procesador en un 50% en promedio durante un minuto. Se aclara de nuevo que un archivo de 10000Kbytes es supremamente inusual en un ambiente de solicitud respuesta de Servicios Web, sin embargo las pruebas determinan que el procesamiento no estaría por encima del 85%, umbral que se considera ya peligroso para el tema de QoS que se define generalmente como lo que percibiría el usuario. Ahora, 100 procesos concurrentes ejecutando una labor de compresión con el algoritmo gzip sobre un archivo de 100Kbytes con contenido HTTP 1.1, consumirían el procesador en un 40% durante 5 segundos. Teniendo en cuenta que ya un archivo mayor de 10Kbytes es exagerado en Servicios Web, se puede concluir que el procesamiento no sería consumido inapropiadamente en este caso.

Según estos resultados vemos que es muy factible usar gzip para comprimir el contenido XML que se lance desde el servidor. Estas pruebas fueron hechas en un PC de 1500 Mhz. Teniendo en cuenta que un servidor puede tener procesamiento de mejor arquitectura y velocidad, para Windows es muy propicio generar compresión para el contenido dinámico tipo XML que permita a los dispositivos móviles optimizar el uso del ancho de banda.

Como se ha mencionado y demostrado, un archivo de Servicio Web podría estar entre menos de 1K a 10K de tamaño, porque los Servicios Web permiten precisamente seleccionar desde el cliente del servicio el resultado que se quiere obtener. Igualmente el servidor además arroja sólo los resultados que se le piden. Se ve que el costo/beneficio de usar compresión con gzip con respecto a procesamiento en archivos menores a 10K es excelente. Se ve adicionalmente que se aprovecha casi tres veces más el ancho de banda para un pico de uso de procesamiento de 3.13 en el servidor. Pero este es el pico, el procesamiento en promedio realmente tiende a cero durante todo el proceso de generación de gzip.

**6.3.1.2 Resultados de la prueba con ANTS.** ANTS es un software que permite generar carga y medir rendimiento sobre un servicios en la web incluyendo un Servicio Web.

Los resultados de las pruebas presentan inicialmente consumo de procesamiento y luego los tiempos de respuesta y tamaño de los mensajes enviados.

Con un ratio de 100 y 5 usuarios virtuales en ANTS se obtiene que el uso máximo del procesamiento del proceso aspnet\_wp es de 4.6% del procesador estando desactivada la opción de compresión HTTP para el servidor IIS. Para tomar esta medida se usó la Herramienta del Sistema "Rendimiento" de Windows 2000 Server.

La siguiente tabla resume los resultados de las peticiones de cada método, incluyendo tiempos para el primer byte, tiempos para el último byte y cantidad de bytes recibidos por el cliente. Se debe tener en cuenta que la cantidad de bytes aumenta con respecto a los datos definidos en la Tabla 10, Comparación de tamaños de mensajes SOAP en el ejemplo básico del Servicio Web MathService, debido a que en este caso se cuentan los datos adicionales que tienen que ver con el transporte TCP/IP.

**Tabla 16. Resultados de la prueba de carga con ANTS al Servicio Web MathService en Microsoft Windows**

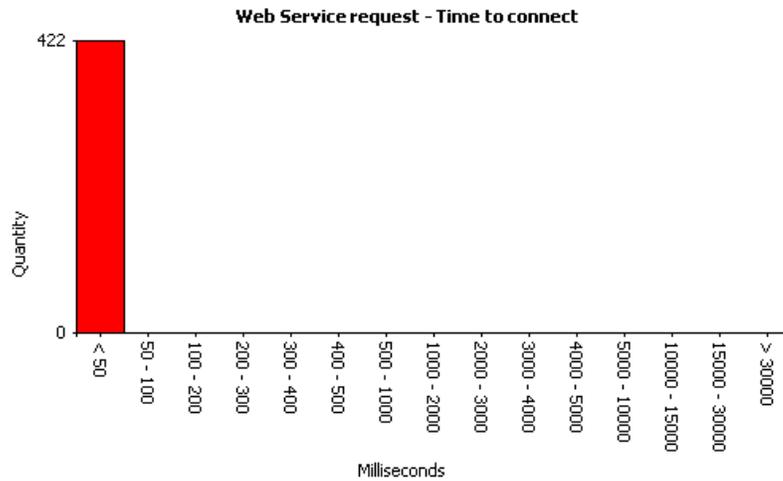
Método (Operación de la calculadora)	Tiempo para conectarse en milisegundos	Tiempo para recibir el primer byte en milisegundos	Tiempo para recibir el último byte en milisegundos	Bytes recibidos	Cantidad de respuestas del método
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Add	4	5	109	666	106
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Subtract	0	0	68	686	106
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Multiply	0	0	154	686	105
POST http://201.245.175.163:7070/vari0s/Desarrollo/MathService.asmx Divide	0	0	99	681	105

Esta tabla es un complemento para esclarecer las calidades que obtendría un usuario. De todas formas el consumo del procesamiento es uno de los factores fundamentales para QoS. En las tablas anteriores se demostró que la compresión no sería factor importante de consumo en archivos menores o iguales a 100Kbytes.

Esta tabla presenta para cada uno de los métodos del ServicioWeb MathService en Microsoft Windows, usando el simulador de carga ANTS, el tiempo en milisegundos que le toma a un cliente conectarse al Servicio Web, el tiempo en milisegundos para recibir el primer byte de respuesta a la solicitud, el tiempo en milisegundos para recibir el últimos byte (en este caso el byte 666, 686, 686 y 681 respectivamente para cada uno de los métodos suma, resta, multiplicación y división), y la cantidad de respuestas que generó la carga durante 60 segundos. Con esto se tiene que el Servicio Web fue atacado 422 veces en un minuto, algo así como 7 veces por segundo. Esta medición es independiente a la de consumo de procesamiento en la compresión. Estos datos obtenidos con ANTS representan información concerniente a QoS que se entiende como lo que percibe el usuario. Se estiman para ofrecer una perspectiva de lo que percibiría el usuario en un ambiente sin compresión, cuando el servidor recibe 7 peticiones en un segundo. Las pruebas de compresión representarían una carga mucho mayor sobre el servidor y son pruebas de compresiones netas de archivos tipo HTTP 1.1.

Los siguientes gráficos dejan ver el rendimiento en el ofrecimiento del servicio para el ejemplo de MathService, cuando se ataca 422 veces en un minuto, según los parámetros configurados para el ANTS en esta prueba:

**Figura 36. Tiempo para conectarse el cliente al Servicio Web Math Service en prueba ANT**



**Tabla 17. Tiempo para conectarse el cliente al Servicio Web Math Service en prueba ANT**

Tiempo en milisegundos	Cantidad de solicitudes
< 50	422
50 - 100	0
100 - 200	0
200 - 300	0
300 - 400	0
400 - 500	0
500 - 1000	0
1000 - 2000	0
2000 - 3000	0
3000 - 4000	0
4000 - 5000	0
5000 - 10000	0
10000 - 15000	0
15000 - 30000	0
> 30000	0

Figura 37. Tiempo para que el cliente reciba el primer byte del Servicio Web Math Service en prueba ANT

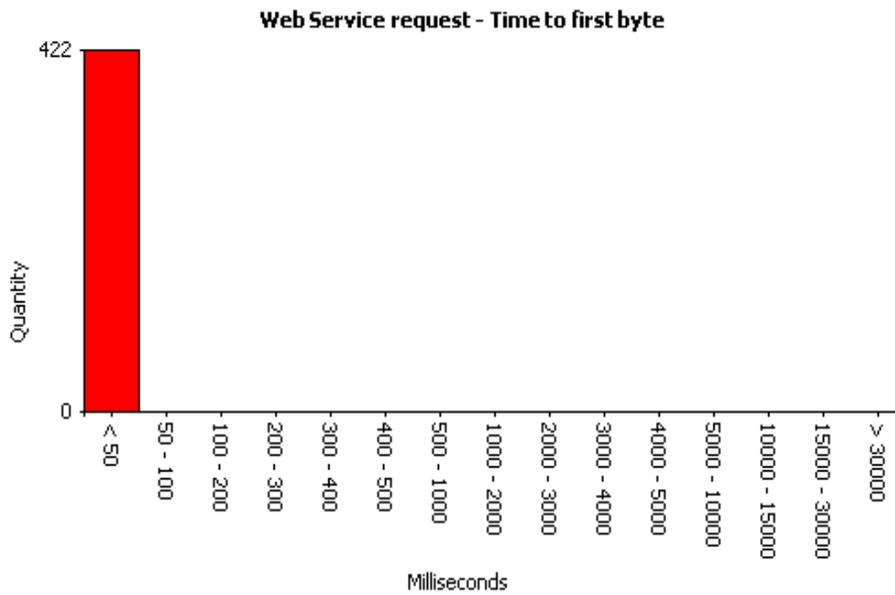
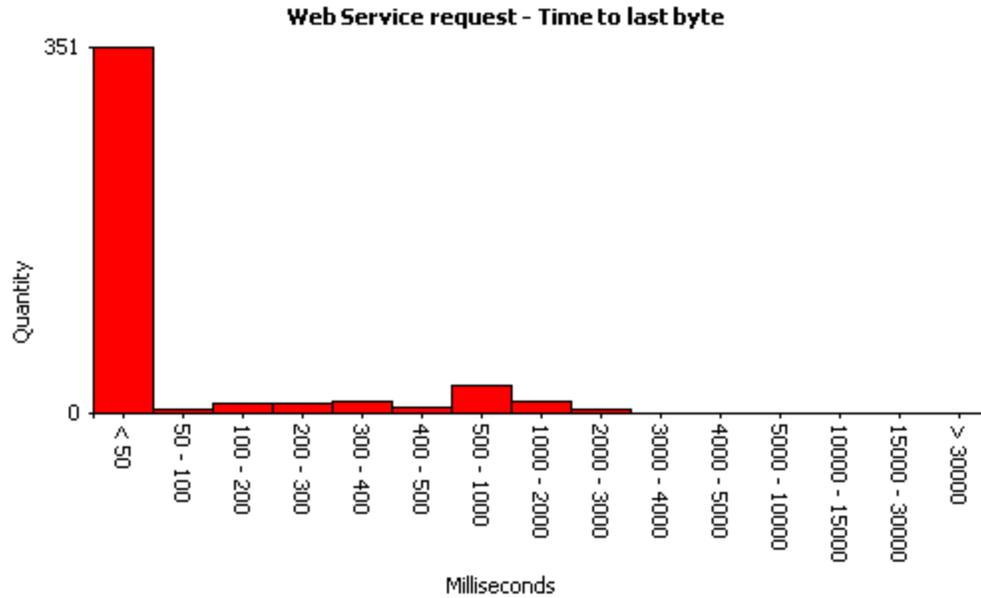


Tabla 18. Tiempo para que el cliente reciba el primer byte del Servicio Web Math Service en prueba ANT

Tiempo en milisegundos	Cantidad de respuestas
< 50	422
50 - 100	0
100 - 200	0
200 - 300	0
300 - 400	0
400 - 500	0
500 - 1000	0
1000 - 2000	0
2000 - 3000	0
3000 - 4000	0
4000 - 5000	0
5000 - 10000	0
10000 - 15000	0
15000 - 30000	0
> 30000	0

**Figura 38. Tiempo para que el cliente reciba el último byte del Servicio Web Math Service en prueba ANT**



**Tabla 19. Tiempo para que el cliente reciba el último byte del Servicio Web Math Service en prueba ANT**

Tiempo en milisegundos	Cantidad de respuestas
< 50	351
50 - 100	3
100 - 200	9
200 - 300	8
300 - 400	10
400 - 500	4
500 - 1000	25
1000 - 2000	10
2000 - 3000	2
3000 - 4000	0
4000 - 5000	0
5000 - 10000	0
10000 - 15000	0
15000 - 30000	0
> 30000	0

Figura 39. Número de bytes enviados al cliente desde el Servicio Web Math Service en prueba ANT

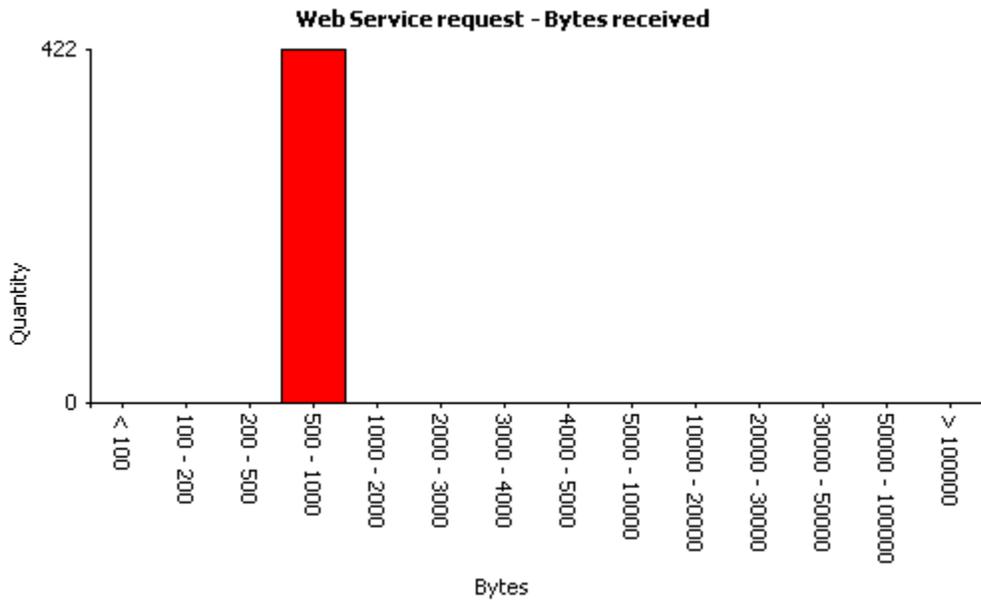


Tabla 20. Número de bytes enviados al cliente desde el Servicio Web Math Service en prueba ANT

Tamaño en bytes	Cantidad de respuestas
< 100	0
100 - 200	0
200 - 500	0
500 - 1000	422
1000 - 2000	0
2000 - 3000	0
3000 - 4000	0
4000 - 5000	0
5000 - 10000	0
10000 - 20000	0
20000 - 30000	0
30000 - 50000	0
50000 - 100000	0
> 100000	0

Para el servicio FactorialServiceC los resultados son realmente muy similares.

**6.3.2 Linux.** El ancho de banda es una consideración a tomar igualmente en la Plataforma Linux. Las pruebas se centraron de igual forma en el consumo de ancho de banda de una aplicación que inicialmente está en un lenguaje como Java o C, luego pasa a XML por medio del parser del servidor en este caso Apache Axis y finalmente el servidor HTTP de la plataforma lo convierte en HTTP 1.1. Es muy conveniente usar pocos caracteres en la especificación de los servicios y las respuestas en el código inicial sea cual sea el lenguaje (Visual Basic, C# o Java en Windows, o C o Java en Linux).

**6.3.2.1 Uso de procesamiento en servidor.** Se usa un análisis sobre la herramienta XMLPPM

La siguiente tabla resume las pruebas y los resultados obtenidos al ejecutar compresiones de archivos con formato XML en el servidor, con un sólo usuario concurrente.

**Tabla 21. Resultados de comprimir XML en el servidor con XMLPPM con una sólo petición en Linux Axis**

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Consumo en procesamiento para descomprimir
Archivo de 1K	1058	348	3.04	Tiende a cero	Tiende a cero
Archivo de 10K	10580	426	24.84	Tiende a cero	Tiende a cero
Archivo de 100K	105800	985	107.41	Tiende a cero	Tiende a cero
Archivo de 1000K (1 MB)	1058000	6227	169.91	Tiende a cero	Tiende a cero
Archivo de 10000K (10 MB)	10580000	58541	180.73	Tiende a cero	Tiende a cero

Esto demuestra que para el ejemplo Calculator, el consumo de procesamiento en el servidor, teniendo en cuenta que los archivos miden menos de 1K tiende a cero cuando hay una sola petición o respuesta.

Las siguientes tablas muestran los resultados de las pruebas usando 10 y 100 procesos.

**Tabla 22. Resultados de comprimir XML en el servidor con XMLPPM con una 10 peticiones en Linux Axis**

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Tiempo tomado para comprimir
Archivo de 1K	1058	348	3.04	8.00%	Tiende a cero
Archivo de 10K	10580	426	24.84	9.00%	Tiende a cero
Archivo de 100K	105800	985	107.41	20.00%	Tiende a cero
Archivo de 1000K (1 MB)	1058000	6227	169.91	30.00%	2 segundos
Archivo de 10000K (10 MB)	10580000	58541	180.73	100.00%	10 segundos

**Tabla 23. Resultados de comprimir XML en el servidor con XMLPPM con 100 peticiones en Linux Axis**

Archivo original	Tamaño del archivo sin comprimir en bytes	Tamaño de archivo comprimido en bytes usando la máxima compresión	Veces que se comprimió con respecto al tamaño original	Consumo en procesamiento para comprimir	Tiempo tomado para comprimir
Archivo de 1K	1058	348	3.04	60.00%	1 segundo
Archivo de 10K	10580	426	24.84	80.00%	1 segundo
Archivo de 100K	105800	985	107.41	80.00%	3 segundos
Archivo de 1000K (1 MB)	1058000	6227	169.91	100.00%	13 segundos
Archivo de 10000K (10 MB)	10580000	58541	180.73	100.00%	130 segundos

Aunque 10 MB es una exageración en cuanto a peticiones y respuestas en Servicios Web se refiere, se observa que XMLPPM es una opción que consume más procesamiento que gzip. Cualquiera de las dos opciones son usables en las dos plataforma, es decir, tanto en Windows como en Linux se pueden usar tanto gzip como XMLPPM, pero según estos resultados obtenidos, se genera una recomendación hacia usar gzip a pesar de que XMLPPM comprime más que gzip cuando de XML se trata pues XMLPPM es además de un compresor, un compresor que por medio del parser instalado en la máquina permite analizar el contenido del archivo XML y evitar errores de sintaxis. Claro está que la corrección de sintaxis erróneas corresponde mejor a otras etapas del desarrollo y no al nivel de compresión, pero para permitir una compresión específica para XML, XMLPPM debe realizar este análisis. El equipo en el cual se instaló Linux y se hicieron las pruebas

con XMLPPM tiene un procesamiento de 2800 Mhz, es decir un procesamiento mucho mejor que el equipo en el cual se hicieron las pruebas para Windows.

Vale dejar claro que para el archivo de 1K, rango en el cual se encuentran los ejemplos utilizados para las pruebas, el procesador se consumió en un 60% durante poco más de un segundo. De manera que tampoco se afirma que hay un consumo excesivo para un archivo pequeño, pero si se denota un consumo mucho mayor que en el caso de la utilización de gzip en Windows. Esto se debe a que el análisis que hace XMLPPM sobre el archivo XML genera un consumo de procesador mucho mayor, todo esto para lograr comprimir mucho mejor este tipo de archivos.

Bien, es muy conveniente hacer una comparación entre los niveles de compresión de gzip y XMLPPM en el equipo Linux:

**Tabla 24. Comparación entre gzip y XMLPPM en Linux con respecto a compresión**

Archivo	Gzip tamaño en bytes	XMLPPM tamaño en bytes
1K	257	227
10K	324	294
100K	774	706
1000K	5107	4707
10000K	48432	44615

Luego de este análisis se denota que la diferencia de compresión frente a el consumo de procesamiento de la máquina no es significativa, a pesar de que XMLPPM comprime un poco más, pero consume mucho más procesamiento por el análisis de sintaxis que por defecto trae la herramienta para XML.

Realmente el análisis de sintaxis debe recaer sobre otras etapas del desarrollo.

Ahora, la siguiente tabla compara consumo de procesador para 100 sesiones en un archivo de 10 MB:

**Tabla 25. Comparación del consumo de procesamiento entre XMLPPM y gzip en Linux para un archivo de 10000K**

Archivo	Consumo XMLPPM	Consumo gzip
10000K	100% durante 130 segundos	100% durante 71 segundos

Como se ha mencionado y demostrado, un archivo de Servicio Web podría estar entre menos de 1K a 10K de tamaño, porque los Servicios Web permiten precisamente seleccionar desde el cliente del servicio el resultado que se quiere obtener. Igualmente el servidor arroja sólo los resultados que se le piden. Vemos que el costo/beneficio de usar compresión con XMLPPM con respecto a procesamiento en archivos menores a 10K es excelente. Se ve adicionalmente que se aprovecha tres veces más el ancho de banda para un pico de uso de procesamiento de 3.13 en el servidor. Pero este es el pico, el procesamiento en promedio realmente tiende a cero durante todo el proceso de generación de XMLPPM.

## 6.4 RESUMEN

**6.4.1 Según resultados, vale la pena comprimir.** Constantemente el tema de la compresión preocupa por el uso del procesamiento en los extremos de la comunicación que esto conlleva.

Es una definitiva que la compresión consume procesamiento, pero esto depende del algoritmo que se use, del tamaño del archivo y si además de la compresión existe algún análisis sintáctico de la información como en el caso de XMLPPM.

El uso del procesamiento a más de 85% deteriora el tema de QoS, pero, este trabajo concluye que tendrían que cumplirse uno de dos factores para que el procesador se consuma a ese nivel: el primero, que exista una concurrencia muy alta de mucho más de 100 procesos en paralelo para que para archivos menores o iguales de 10K se consuma el procesador a este umbral. El otro factor es que el archivo tenga 1000K o más de tamaño, situación extremadamente inusual en Servicios Web.

Según esto una recomendación es comprimir el contenido HTTP 1.1 para enviarlo.

El trabajo también concluye que la compresión con gzip es la más adecuada por tres razones principales: la primera es que es un algoritmo estándar, la segunda es que consume un poco menos de procesamiento que XMLPPM, y la tercera es que XMLPPM comprime XML y analiza sintaxis, análisis que no es necesario pues esto queda a cargo de otros niveles de la implementación como el desarrollo de las aplicaciones o el runtime del servidor de Servicios Web, en este caso .NET Framework (ASP.NET) o Apache Axis. Tampoco es necesario el análisis sintáctico, porque la información final a enviar está formateada en HTTP 1.1 y gzip sigue todas reglas que garantizan que se comprima y se descomprima sin problemas en los extremos. En las recomendaciones para trabajo futuro se sugiere el tema del XML binario. XML binario no es estándar; el estudio estaría basado en qué tan estándar es, qué avances hay en el campo. Igualmente, otro algoritmo de compresión que se está imponiendo es el bzip2, que comprime más pero consume más

procesamiento. Un estudio complementario a este sería el generar las tablas comparativas entre consumos de gzip y bzip2 y de porcentajes de compresión con respecto a gzip.

Si se quieren más detalles de por qué se concluye que vale la pena comprimir, remitirse a la lectura completa del Capítulo 6 en los numerales 6.1, 6.2 y 6.3; los comentarios al final de los resultados obtenidos en cada tabla de uso de procesamiento en compresión son importantes.

**6.4.2 Y qué se transporta en los Servicios Web.** Bien, según los estudios realizados, lo que se transporta son netamente respuestas a solicitudes de datos en repositorios o datos generados por cálculos en memoria usando el procesamiento. Con esto se concluye que en los Servicios Web se transportan información definitiva, importante de una solicitud. Por ejemplo, ante una solicitud de temperatura actual en una ciudad, que lleva como parámetro el código de la ciudad, ofrece un número de tipo float para conocer la temperatura. Un Servicio Web no se preocupa de cómo en la pantalla se presenta el dato, sólo de enviar el dato solicitado.

Esto se explica para concluir que una función usual de los Servicios Web no es el envío de un documento de texto formateado en un procesador de texto avanzado, o una hoja de cálculo, o un video, o una imagen. Por tal motivo, la información transportada por un Servicio Web en ambos sentido no va a ser muy extensa, permitiendo complementar el hecho de que es muy factible y recomendable comprimir la información en los Servicios Web para aprovechar al máximo el recurso más crítico en los PDAs, el ancho de banda cuando de movilidad se trata.

**6.4.3 ¿Quién sufre más, el cliente o el servidor?** El servidor siente mucho más el peso de la compresión/descompresión.

Para aclarar este punto es necesario aclarar el procesos de solicitud respuesta.

1. El cliente del Servicio Web en el PDA genera HTTP 1.1 para enviar
2. El cliente comprime HTTP 1.1 para enviar (debe comprimir entre 500 bytes y 2Kbytes). Este HTTP 1.1 contiene entre otros elementos, el método del Servicio Web que necesita y los parámetros de entrada para este método a ser enviados al servidor
3. El cliente envía la solicitud
4. TCP/IP hace llegar la solicitud hasta el servidor
5. El servidor en escucha permanente recibe la solicitud que debe descomprimir
6. El servidor genera el procesamiento de la solicitud, genera un resultado emitido por medio de HTTP 1.1
7. El servidor comprime HTTP 1.1 (por lo general y que según las tablas relacionadas anteriormente en este capítulo son un 87% del tamaño de la solicitud)
8. El servidor envía HTTP 1.1 comprimido

9. TCP/IP hace llegar la respuesta al cliente PDA
10. El cliente PDA debe descomprimir la información

Según los resultados obtenidos en estas pruebas, la compresión hace más uso del procesador que la descompresión.

Si se habla de una sola comunicación el PDA debe comprimir más información (un 23% más) de la que debe comprimir el servidor. Claro está que el servidor tiene muchas comunicaciones en paralelo. Según esto y como se ha mencionado en varias ocasiones en el documento, la compresión de una solicitud para un Servicio Web podría estar a lo mucho en 10K, tamaño que realmente no exige sino una mínima cantidad de procesamiento así sea en el lado del PDA. El servidor tiene que contar los consumos de procesamiento del ofrecimiento de todas las respuestas en paralelo, además de la compresión de todas las salidas que a pesar de ser un 23% más livianas en peso que la del PDA, repercutirían en el rendimiento del procesamiento del servidor. En conclusión, el servidor sufre más, aunque tiene que presentarse una situación como también se ha mencionado en la cual la información a comprimir sea de 1000Kbytes o más en una respuesta de Servicio Web (situación por demás inusual) o que las solicitudes a un sólo Servicios Web sean mucho mayor a 100 concurrentes.

**6.4.4 Situaciones ideales para habilitar la compresión http.** Al final, todo lo que se envía es HTTP, específicamente HTTP 1.1. Según los estudios y pruebas realizadas, los siguientes son los escenarios ideales para usar compresión de HTTP.

- Ancho de banda de red estrecha: Los clientes que tienen acceso al servidor Web principalmente mediante módem o los PDAs por medio de una WWAN están utilizando ancho de banda estrecho. Esto significa que la cantidad de datos que se pueden transferir cada intervalo de tiempo es mínima en comparación con una red de área local. Si un cliente solicita datos comprimidos, se transferirán menos bytes a través de la red, lo que da como resultado tiempos de descarga menores.
- La mayoría del contenido alojado es estático: Si el servidor Web atiende principalmente archivos estáticos, la sobrecarga del procesador será mínima porque los archivos estáticos se almacenan en caché en una carpeta temporal. Los archivos estáticos sólo se vuelven a comprimir cuando cambia el archivo original, lo que reduce el tiempo que el procesador emplea en comprimir y volver a comprimir archivos. Bien, en el caso particular del presente proyecto, es necesario más que nada comprimir los datos, ya que como se ha dicho, lo más importante es el aprovechamiento al máximo del ancho de banda, posibilitando la compresión de los datos a enviar en todo caso en el servidor, y teniendo en cuenta que la descompresión generaría un consumo de procesamiento no excesivo en el lado del cliente. El servidor sería el que puede sufrir un consumo exagerado de recursos de procesamiento, pero esto se evita con un buen diseño de servidor y teniendo en cuenta las tablas 13, 14, 15, 23, 24 y 25 que presentan consumos de procesamientos para 1, 10 y 100 usuarios en los ejemplos MathService y Calculadora, es bastante posible.
- Espacio en disco adecuado: Si habilita la compresión HTTP, asegúrese de que tiene suficiente espacio de disco como para almacenar las versiones sin comprimir y las

versiones comprimidas de los archivos estáticos. Este es un recurso que realmente se ve algo abundante hoy en día. No sería una limitante grande.

**6.4.5 Situaciones no ideales para habilitar la compresión http.** Las siguientes serían las situaciones en las que inicialmente no se recomienda la compresión HTTP según los estudios y resultados obtenidos:

- Valor elevado del contador % de tiempo de procesador: Si el contador % de tiempo de procesador ya está al 80 % o más, no se recomienda utilizar la compresión HTTP a menos que pueda actualizar la CPU del servidor. Esto realmente se debe determinar con un buen análisis de carga. Las tablas 13, 14, 15, 23, 24 y 25 presentan este análisis en los ejemplos de los Servicios Webs MathServices y Calculadora.
- Contenido dinámico: Si en el servidor se genera un gran volumen de contenido dinámico, al habilitar la compresión HTTP para los archivos de aplicación se hará un mayor uso del procesador. Como el contenido dinámico no se almacena en caché, se debe volver a comprimir cada vez que lo solicita un cliente. Bien, los Servicios Web, generalmente ofrecen contenido dinámico, aunque como se ha dicho este consumo que se denotaría grande en procesamiento, se puede optimizar con un buen diseño de servidor y teniendo en cuenta resultados en las pruebas, es muy factible usarlo cuando como en el caso de los Servicios Web, el volumen de información que se envía es seleccionada de acuerdo a los parámetros que se detallan en el WSDL.
- Espacio en disco inadecuado: Si el servidor no tiene espacio de disco disponible para almacenar los archivos comprimidos, no habilite la compresión HTTP hasta que pueda agregar más espacio a la unidad de disco duro del servidor. Actualmente se puede considerar que este no es un factor crítico, aunque merece análisis en cada caso particular.

Como se mencionó anteriormente, la compresión HTTP conserva el ancho de banda de la red y reduce los tiempos de descarga al reducir el tamaño de un archivo antes de transferirlo a través de la red. Sin embargo, esto requiere más recursos del servidor para la compresión y más recursos del cliente para la descompresión (consulte la siguiente tabla).

**Tabla 26. Recursos y demandas con la compresión HTTP habilitada**

Recurso	Servidor	Cliente
Procesador	<ul style="list-style-type: none"> <li>• La compresión de resultados dinámicos requiere más ciclos de procesador (CPU) en el servidor.</li> <li>• Sólo es necesario volver a comprimir los archivos estáticos cuando se actualizan en el servidor.</li> <li>• Es necesario comprimir el resultado de las aplicaciones dinámicas cada vez que lo solicita un cliente.</li> <li>• Sólo es necesario volver a comprimir los archivo estáticos cuando se actualizan en el servidor.</li> </ul>	<ul style="list-style-type: none"> <li>• La descompresión de archivos en el cliente requiere más ciclos de procesador (CPU).</li> </ul>
Espacio en la unidad de disco duro	<ul style="list-style-type: none"> <li>• El servidor almacena en caché los archivos estáticos comprimidos en la unidad de disco duro, de manera que no haya que volver a comprimirlos cada vez que los solicite un cliente.</li> <li>• El almacenamiento en caché de la versión comprimida de un archivo estático, además de la versión sin comprimir del archivo, requiere espacio adicional en el disco duro.</li> <li>• El resultado de las aplicaciones dinámicas no se almacena en caché en el disco duro.</li> </ul>	<ul style="list-style-type: none"> <li>• No se necesita espacio adicional en el disco duro del equipo cliente.</li> </ul>

#### **6.4.6 Otros recursos como la memoria RAM y el almacenamiento**

**6.4.6.1 El lado del servidor.** El diseño de un óptimo esquema de servidores determina la QoS y el crecimiento a futuro.

Se deben prever las características de crecimiento y de calidad de los componentes para que la QoS obtenga resultados óptimos.

Bien, esto se menciona porque la memoria RAM y el almacenamiento influyen en la calidad de prestación de servicio y se concluye adicionalmente que en el caso de la compresión para Servicios Web no constituyen un factor crítico. Claro está que se deben hacer los análisis necesarios para que una memoria RAM o un almacenamiento no afecten los procesos que tienen que ver con la compresión de la información o con la emisión, análisis y generación de resultados de los métodos de los Servicios Web. En definitiva, la memoria RAM y el almacenamiento no son elementos de alto consumo en procesos de compresión, pero si pueden afectar la calidad del servicio si afectan la operación de las aplicaciones que se deben ejecutar para llevar a cabo el cumplimiento de los servicios.

**6.4.6.2 El cliente.** Los PDAs son dispositivos también con baja capacidad de memoria RAM y almacenamiento comparados con los PCs convencionales. Esto se menciona porque, además de que su arquitectura es diferente (memoria RAM y almacenamiento son un sólo elemento) con respecto a los equipos computacionales convencionales, este recurso realmente no afecta la calidad en la obtención del servicio pues igualmente lo que interviene en gran medida es el procesamiento y como se ha dicho en este mismo capítulo, la información que tendría que comprimir o descomprimir el PDA no sobrepasa los 10Kbytes de información.

La memoria RAM y el almacenamiento si constituyen un factor bastante limitante en el desarrollo de aplicaciones. Por ejemplo, no es posible desde ningún punto de vista, tratar de crear una aplicación para una Palm que contenga archivos objeto que sobrepasen los 64Kbytes de peso. Y eso no es todo, no pueden existir referencias internas o llamados a funciones que estén dentro del código ya compilado (objeto) a más de 32Kbytes de distancia.

Por tal motivo la programación en Palm no es sencilla y existen metodologías que según se concluye por lo investigado y lo experimentado en el desarrollo de este proyecto se pueden implementar. La primera es una metodología totalmente manual, esto es, separar los archivos para que en tamaño tengan menos de 64Kbytes, y hacer los llamados y linkeos en tiempos de compilación que se requieran. Manualmente también se deben generar llamados a funciones que cumplan con la característica de no estar a una distancia de más de 32Kbytes dentro del código ya compilado. Dentro de la forma manual estaría que en un archivo de más de 64Kbytes se hagan condicionales de acuerdo a variables definidas al inicio del archivo y se compile o no de acuerdo al resultado de la evaluación del condicional. Esto es, tener varias copias del archivo original y diferenciarlas con definiciones al inicio del archivo tendientes a definir cuáles partes de código se van a compilar en cada caso. Para este fin también es importante hacer los llamados a funciones de manera que no sobrepasen los 32Kbytes de distancia dentro del código ya compilado.

La mejor forma en definitiva es usar las recomendaciones de prc-tools con respecto a gcc para implementar Sections. Recordemos que prc-tools es la herramienta para generar archivos prc par Palm usando gcc. Bien, en el tutorial del Capítulo 5 se hace referencia a las formas de implementación que buscan evitar que el programador obtenga los constantes y molestos mensajes de 'undefined rereference to' o de 'more than 2 bytes', abundantes en Internet en foros y solicitudes de apoyo y que por lo general no tienen una respuesta clara. Los mensajes de 'undefined reference to' se pueden deber a inclusiones de archivos de encabezados faltantes, falta de compilación de archivos fuentes faltantes o errores de digitación, pero en este caso específico es muy importante definir que este mensaje es muy característico de las limitantes de recursos de memoria RAM/almacenamiento de las Palm.

## 6.5 ESCANEEO DE PAQUETES

Usando Ethereal, una herramienta gratuita para captura de paquetes se puede observar la diferencia en caracteres y en tamaño de la forma no comprimida a la forma comprimida de una respuesta HTTP de los Servicios Web en ASP.NET o en Apache Axis.

Para detalles de esta herramienta ver ANEXO 3. ETHEREAL [36] - Captura de paquetes.

## 7. CONCLUSIONES

Este trabajo permite determinar que usando algoritmos de compresión estándar como gzip es posible reducir el uso de ancho de banda para un dispositivo móvil en un 66% en el momento de acceder a un Servicio Web. El tema del procesamiento no es una preocupación en este caso según lo analizado pues el tamaño de los archivos que generalmente se generan en este ambiente de Servicios Web contienen tamaños que no exigen que los algoritmos de compresión requieran usos excesivos de procesamiento en los extremos.

Se determinó que el servidor es el que puede sufrir un poco más el uso de esta técnica, pero se concluyó luego de analizar el contenido de las peticiones y respuestas de un Servicio Web en dos ejemplos particulares, que lo que se transporta por la red (Internet) es contenido en formato HTTP 1.1, lo que permite usar los algoritmos de compresión estándar como el gzip para hacer que la información que viaje por el canal sea reducido. La preocupación del consumo de recursos queda resuelta con las pruebas realizadas usando scripts de simulación de 100 procesos concurrentes de compresión. Estos scripts a pesar de haberse ejecutado en PCs con 1.5 Ghz y 2.8 Ghz de velocidad en procesador no consumen más del 3.67% de procesamiento en un pico que dura mucho menos de un segundo. Esto garantiza que la compresión es una solución ideal para cuando el servidor tiene que enviar información a dispositivos con escasos recursos de conectividad como los PDAs, sin afectar la forma en que el usuario percibe el servicio (QoS).

El tamaño promedio de las solicitudes/peticiones de Servicios Web no sobrepasan los 10000 bytes, es decir, los tamaños de archivos que viajan por la red y que necesitan ser transportados en formato de protocolo HTTP 1.1 son pequeños y no llevarían el procesamiento en los extremos a un estado crítico que afecte el concepto de calidad del servicio (QoS). En el cliente como máximo el procesador se consumiría en un 5% durante menos de un segundo.

El uso de algoritmos de compresión para XML como XMLPPM a pesar de que comprime un 9% más que un algoritmo no específico para XML como gzip, consume el doble en procesamiento por el análisis que tiene que realizar. Se definió que a pesar de ocupar el doble de procesamiento que un algoritmo como gzip, no lleva a estado crítico por encima del 85% el uso del procesador, lo que no afecta negativamente el tema de QoS, pero se estableció que el análisis de sintaxis en el caso de los clientes de Servicios Web en los PDAs queda resuelto por las propias librerías que en los dos ambientes, Pocket PC y Palm, generan los Stubs (los clientes) de los Servicios Web. En el caso del servidor, los servicios generados por Microsoft .NET Framework y Apache Axis, incluyen analizadores de sintaxis (parsers) XML. Por otro lado, lo que se comprime al final es HTTP 1.1, y lo mejor es comprimir lo que al final va a viajar por la red. El comprimir antes el XML obligaría a una descompresión de lo generado por XMLPPM para poder emitir HTTP 1.1 por parte del servidor de Servicios Web. Con gzip, es en la última etapa, la etapa antes del envío de información, en la que se comprime el HTTP 1.1.

La creación de Servicios Web en los ambientes Microsoft Windows y Apache Axis, están suficientemente documentadas en la red. Para los dispositivos Pocket PC existen numerosos documentos y asistentes que permiten generar clientes de estos servicios. Caso contrario ocurre con los dispositivos PDA para los cuales aunque existen excelentes IDEs para la creación de aplicaciones, no existen IDEs para la creación de clientes de Servicios Web, siendo la única posibilidad (donde no se posee un IDE particular, sino una serie de archivos y librerías) el proyecto gSOAP creado originalmente para clientes convencionales como PCs trabajando en C/C++. El proyecto gSOAP no se centra en PDAs, sólo genera algunas recomendaciones bastante sencillas para el caso de los PDAs. Tal es el caso del archivo stdsoap2.c que sobrepasa todos los límites de un PDA en cuanto a recursos de manejo de archivos y de enlace hacia funciones compiladas, tema que se resuelve de manera bastante manual, siguiendo las recomendaciones de secciones de código en la documentación de prc-tools en la programación para Palm.

Se ha observado que la fabricación de PDAs con procesamiento que llegan a 400Mhz es ya común. Se determinó que igualmente la tendencia es ofrecer el concepto total de movilidad que se logra cuando el dispositivo móvil puede conectarse a la red (GPRS en la mayoría de los casos). Esta conectividad es el recurso crítico, y la compresión para ahorro de ancho de banda revierte bastante importancia en el concepto de movilidad.

## 8. TRABAJO FUTURO

- XML binario [37]. Un trabajo a realizar podría ser el uso de XML binario en PDAs. Aunado a esto un estudio sobre la estandarización de XML binario como forma de compresión; definir si es ya un estándar o en que nivel se encuentra este proceso y posiblemente generar recomendaciones. El estudio sobre el consumo de recursos sería importante.
- ¿Cómo usar bzip2 [38] en ASP.NET y Apache AXIS para mejor compresión?. Bzip2 comprime mejor que gzip. Se debe analizar entonces tablas comparativas de compresión entre gzip y bzip2 y tablas de consumo de procesamiento en ambos casos.
- ¿Cómo la implementación de seguridad (encriptación y firma digital entre otros conceptos) [39] afecta el rendimiento en Servicios Web para dispositivos móviles. Se tiene en cuenta para trabajo futuro porque la seguridad computacional debe ser tenida en este ambiente de Servicios Web y porque adiciona bytes en la información y procesamiento en los extremos al ser necesarias desencriptaciones. Se piensa igualmente que “seguridad en PDAs podría por sí solo ser un tema de estudio.
- El estudio a fondo de QoS en Servicios Web [40] y como el cumplimiento de las recomendaciones [41] puede influir en el rendimiento en los dispositivos móviles.

## BIBLIOGRAFÍA

- 1: Mobile Lifestreams Ltd, What is General Packet Radio Service?, 2000, <http://www.gsmworld.com/technology/gprs/intro.shtml>
- 2: GSM Association, GSM World - the world wide web site of the GSM Association, <http://www.gsmworld.com/index.shtml>
- 3: M. Tian; T. Voigt; T. Naumowicz; H. Ritter; J. Schiller, Performance Considerations for Mobile Web Services, 2003
- 4: IC#Code, SharpZipLib Home Page, <http://www.icsharpcode.net/OpenSource/SharpZipLib/Default.aspx>
- 5: Sourceforge, XMLPPM: XML-Conscious PPM Compression, <http://www.cs.cornell.edu/People/jcheney/xmlppm/xmlppm.html>
- 6: World Wide Web Consortium, World Wide Web Consortium Home Page, <http://www.w3c.org>
- 7: The Axis Development Team, Apache Axis Home Page - WebServices - Axis, , <http://ws.apache.org/axis/>
- 8: Microsoft Corporation, .NET Framework Developer Center, <http://msdn.microsoft.com/netframework/>
- 9: Open Mobile Alliance, Open Mobile Alliance Home Page, <http://www.openmobilealliance.org/>
- 10: Web Services Interoperability, Web Services Interoperability Home Page, <http://www.ws-i.org/>
- 11: Organization for the Advancement of Structured Information Standards, Organization for the Advancement of Structured Information Standards Home Page, , <http://www.oasis-open.org/home/index.php>
- 12: Palm, Inc., Palm Homa Page, , <http://www.palm.com>
- 13: Microsoft Corporation, Sitio Oficial de Pocket PC, <http://www.microsoft.com/windowsmobile/pocketpc/ppc/default.mspx>
- 14: Sourceforge, Sitio Web de prc-tools, , <http://prc-tools.sourceforge.net/>
- 15: Metrowerks, Sitio Web de Codewarrior, , <http://www.metrowerks.com/mw/default.htm>

- 16: Sun Microsystems, WSS - Web Services Security - Interop Scenarios, <http://java.sun.com/webservices/docs/1.4/tutorial/doc/XWS-Security6.html>
- 17: Organization for the Advancement of Structured Information Standards, <http://www.uddi.org/>, <http://www.uddi.org/>
- 18: World Wide Web Consortium, Web Services Description Language (WSDL) 1.1, 2001, <http://www.w3.org/TR/wsd/>
- 19: E, Cerami, Web Services Essentials - Distributed Applications with XML-RPC, SOAP, UDDI & WSDL - Chapter 6, WSDL Essentials, <http://www.oreilly.com/catalog/webseveress/chapter/ch06.html>
- 20: World Wide Web Consortium, SOAP Version 1.2, 2004, <http://www.w3.org/TR/soap/>
- 21: Palm, Inc., Integrating Mobile Data Services into Enterprise Infrastructures, 2003
- 22: , Definición de IRDA, 1999, <http://www2.terra.com/informatica/que-es/irda.cfm>
- 23: E. Martínez, Estándares WLAN, 2002, <http://www.eveliux.com/articulos/estandareswlan.html>
- 24: Alsitel, Definición de Bluetooth, , <http://www.alsitel.com/tecnico/bluetooth/quees.htm>
- 25: , SISTEMA GPRS, , <http://www.uv.es/~montanan/redes/trabajos/GPRS.doc>
- 26: Symbian, Sitio Web de Symbian, , <http://www.symbian.com/>
- 27: Microsoft Corporation, Sitio Web de Microsoft Mobile Developer Center, <http://msdn.microsoft.com/mobility/>
- 28: Sun Microsystems, Sitio Web de Java 2 Platform, Micro Edition (J2ME), <http://java.sun.com/j2me/index.jsp>
- 29: J, Gailly; M. Addler, gzip Home Page, , <http://www.gzip.org>
- 30: Red-gate Software, ANTS Load Home Page, [http://www.red-gate.com/dotnet/load\\_testing.htm](http://www.red-gate.com/dotnet/load_testing.htm)
- 31: Borland, Optimizeit Enterprise Suite, <http://www.borland.com/optimizeit/>
- 32: Cygwin, Sitio Web de Cygwin, <http://www.cygwin.com/>
- 33: gSOAP, Sitio Web de gSOAP2, <http://www.cs.fsu.edu/~engelen/soap.html>
- 34: C, Crespo, Introducción a los Web Services con PocketSOAP, Apache SOAP y Axis, 2005

- 35: D, Currie, Servicios Web en Linux con Apache Axis y lenguaje C, 2005, <http://programming.linux.com/programming/05/02/16/1947220.shtml?tid=63&tid=67&tid=68>
- 36: Ethereum.com, Sitio Web del escaneador de paquetes Ethereum, <http://www.ethereal.com/>
- 37: World Wide web Consortium, XML Binary Characterization Working Group Public Page, <http://www.w3.org/XML/Binary/>
- 38: RedHat, Sitio Web de gzip2, , <http://sources.redhat.com/bzip2/>
- 39: Microsoft Corporation, Security in a Web Services World: A Proposed Architecture and Roadmap, 2002, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwssecur/html/securitywhitepaper.asp>
- 40: R, Sumra; Arulazi D, Quality of Service for Web Services–Demystification, Limitations, and Best Practices, , <http://www.developer.com/java/web/article.php/2027911>
- 41: International Communication Union, Recommendation E.800 - Terms and definitions related to quality of service and network performance including dependability , 1994
- 42: Microsoft Corporation, Web Services Developer Center, <http://msdn.microsoft.com/webservices/>
- 43: Microsoft Corporation, Trabajar con la caché del ensamblado global, 2001, <http://es.gotdotnet.com/quickstart/howto/doc/gac.aspx>
- 44: Microsoft Corporation, Utilizar compresión HTTP en su sitio Web de IIS 5.0, <http://www.microsoft.com/latam/technet/articulos/200103/art03/default.asp>

## ANEXOS

## **Anexo A. Consideraciones de servicios web con microsoft [42]**

Existen varias formas de crear Servicios Web para servidores que tengan instalado un sistema operativo Microsoft Windows que soporte IIS.

Básicamente, se requieren dos componentes: Microsoft IIS, que es un componente que viene junto con el sistema operativo Windows en sus versiones 2000 Server, 2003 Server y XP Professional por mencionar los más recientes y utilizados.

Adicionalmente a ello se hace necesario instalar en el servidor Microsoft .NET Microsoft 1.1 como mínimo. Este software se puede conseguir en [http://msdn.microsoft.com/netframework/downloads/framework1\\_1/](http://msdn.microsoft.com/netframework/downloads/framework1_1/) junto con toda su documentación. Teniendo estos dos componentes el desarrollador ya posee en su servidor la tecnología ASP.NET, la tecnología de Microsoft para servir aplicaciones Web dinámicas de las cuales los Servicios Web forman parte.

Bien, luego de tener estos dos componentes existen varias formas de desarrollar Servicios Web para este servidor:

1. Visual Basic .NET (Visual Studio .NET)
2. C# (Visual Studio .NET)
3. J# (Visual Studio .NET)
4. Un procesador de texto

En [42] se encuentran los detalles, documentación técnica y tutoriales con respecto a la perspectiva Microsoft en Servicios Web.

## Anexo B. Descripción básica de global assembly cache (GAC)<sup>5</sup>

### Descripción:

La caché del ensamblado global es un almacén del equipo que se utiliza para contener ensamblados que van a compartir varias aplicaciones en el equipo. .NET Framework proporciona dos herramientas para trabajar con la caché. Una de ellas es una extensión Shell de Windows que permite trabajar con la caché mediante una Interfaz gráfica de usuario (Graphical User Interface, GUI). La otra es una herramienta de línea de comandos denominada Global Assembly (Gacutil.exe), que normalmente se utiliza en secuencias de comandos de generación y de prueba. La herramienta de línea de comandos se denomina Global Assembly Cache (Gacutil.exe).

Todos los ensamblados de la caché global deben tener nombres seguros.

### Ver el contenido de la caché del ensamblado:

Desplácese a %winnt%\assembly mediante el uso del Explorador de Windows para activar la extensión Shell. La vista predeterminada muestra el contenido de la caché del ensamblado. También se puede ver el contenido de la caché con la opción /l de la herramienta Global Assembly Cache:

```
gacutil /l
```

### Instalar ensamblados:

Para instalar un ensamblado mediante la extensión Shell, simplemente arrastre y coloque el archivo que contiene el manifiesto del ensamblado en el directorio de la caché del ensamblado global. Para instalar un ensamblado mediante el uso de la herramienta Global Assembly Cache, utilice la opción /i:

```
gacutil /i math.dll
```

### Desinstalar ensamblados:

Para eliminar un ensamblado con la extensión Shell, haga clic con el botón secundario del mouse (ratón) y seleccione Eliminar. También puede utilizarse la opción /u de la herramienta Global Assembly Cache:

```
gacutil /u math,ver=1.0.0.0
```

---

<sup>5</sup> Tomado de [43]

La versión de Windows Installer (1.5) que se incluirá en Visual Studio .NET la caché del ensamblado. Al crear un paquete Installer con la herramienta Visual Studio Deployment u otra herramienta de configuración, se pueden especificar los ensamblados que le gustaría que estuviesen instalados en la caché. El uso de Windows Installer para trabajar con la caché del ensamblado le proporciona las ventajas de Installer estándar, como la instalación bajo pedido, los anuncios, la edición, etc.

## **Anexo C. Ethereal - Captura de paquetes [36]**

Ethereal es usada por profesionales de redes alrededor del mundo para resolución de problemas, análisis, desarrollo de software e implementación de protocolos y educación. Tiene todas las características estándar que se pueden esperar de un analizador de protocolos y varias características no vistas en otro producto. Su tipo de licencia le permite a expertos hacer contribuciones. Corre en las plataformas más comunes: Unix, Linux y Windows.

Para documentación completa y bajar el software se puede remitir al enlace <http://www.ethereal.com/>.

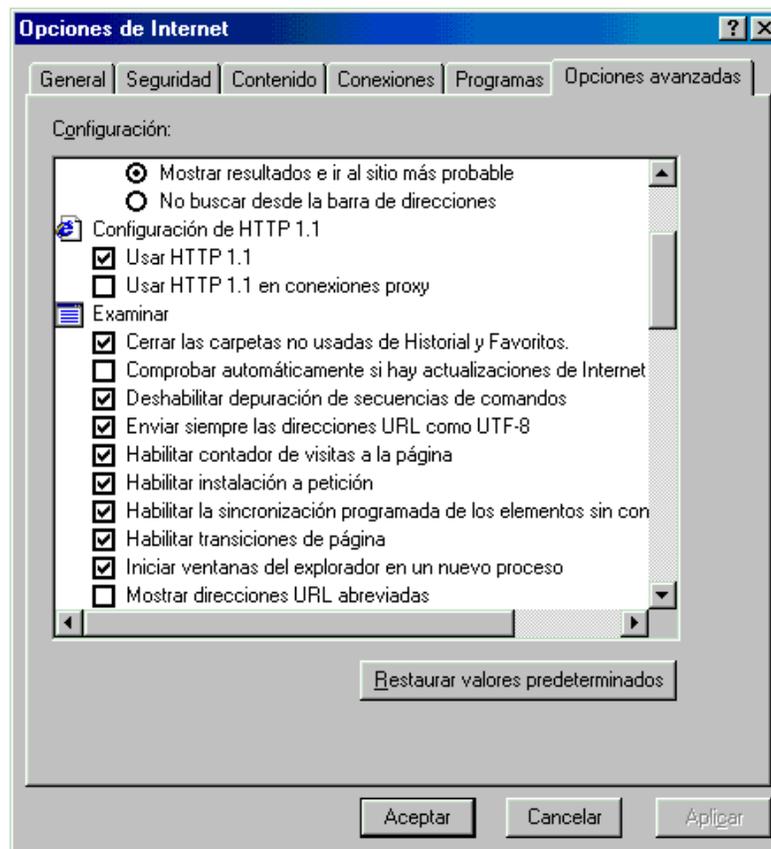
## Anexo D. Habilitar compresión http en .NET [44]

**Importante** Sólo los clientes compatibles con HTTP 1.1 solicitan archivos comprimidos. Los clientes que no son compatibles con HTTP 1.1 solicitan y reciben de IIS los archivos sin comprimir, por lo que no se benefician de los mejores tiempos de descarga que ofrecen los clientes compatibles con HTTP 1.1. Las versiones 4 y posteriores de Internet Explorer, el Explorador de Windows y Mi PC son clientes compatibles con HTTP 1.1 de manera predeterminada.

Para comprobar que Internet Explorer está configurado para utilizar el protocolo HTTP 1.1:

1. Abra la hoja de propiedades **Opciones de Internet**.
  - Si está utilizando Internet Explorer 4, se encuentra en el menú **Ver**.
  - Si está utilizando Internet Explorer 5, se encuentra en el menú **Herramientas**.
2. Seleccione la ficha **Avanzadas**.

Figura 40. Verificación del uso de HTTP 1.1 en Internet Explorer



3. Bajo **Configuración de HTTP 1.1**, compruebe que está seleccionada la opción **Usar HTTP 1.1** (consulte la figura 1 siguiente).

La compresión HTTP se habilita servidor por servidor. Por tanto, si en un único servidor Web con IIS 5.0 tiene varios sitios, habilitará la compresión HTTP para todos los sitios.

1. Desde el **Administrador de servicios Internet**, haga clic con el botón secundario del *mouse* (ratón) en el icono **Equipo** y seleccione **Propiedades**.
2. En la ficha **Servicios de Internet Information Server**, bajo **Propiedades principales**, seleccione **Servicio WWW**.
3. Haga clic en **Modificar**; debe aparecer la hoja **Propiedades principales del servicio WWW**.
4. Seleccione la ficha **Servicio** (consulte la figura 2).
5. Si desea comprimir archivos estáticos para transmitirlos a los clientes que admiten compresión, seleccione **Comprimir archivos estáticos**. Al seleccionar esta opción se comprimirán y almacenarán en caché los archivos que tengan las extensiones ".htm", ".html" y ".txt".
6. Escriba la ruta de acceso a un directorio local en el cuadro de texto **Carpeta temporal** o utilice el botón **Examinar** para encontrar un directorio donde se almacenarán los archivos comprimidos.

**Importante** El directorio que elija en el paso 6 debe encontrarse en una unidad que sea local del servidor Web y debe estar en una partición NTFS (Sistema de archivos de nueva tecnología). El directorio, que no debe estar compartido, no puede ser un directorio comprimido ni debe estar en un volumen comprimido. Para obtener más información acerca de NTFS, busque la palabra clave "NTFS" en la Ayuda en pantalla de Windows 2000.

7. Si desea comprimir el resultado dinámico de las aplicaciones para transmitirlo a los clientes que admiten compresión, seleccione **Comprimir archivos de aplicación**. Al seleccionar esta opción se comprime y se almacena en caché el resultado dinámico de las aplicaciones cuyas extensiones de archivo son ".dll", ".asp" y ".exe".

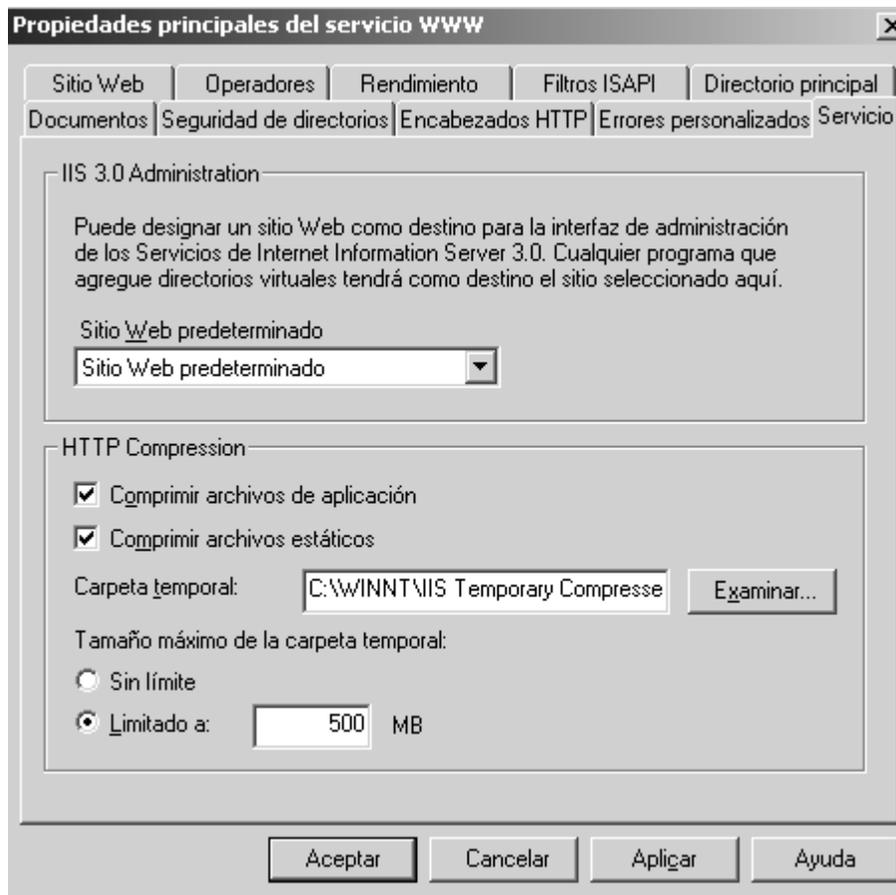
**Importante** No se recomienda comprimir el resultado de los archivos de aplicación a menos que muchos clientes vayan a tener acceso al servidor a través de una conexión lenta, como un módem. Asegúrese también de que el servidor Web tiene suficiente capacidad de procesador como para volver a comprimir los archivos dinámicos cada vez que los solicite un cliente antes de habilitar **Comprimir archivos de aplicación**.

8. Para limitar el tamaño máximo de la carpeta temporal, seleccione **Limitado a** y escriba un número entero en el campo MB.

De manera predeterminada, el tamaño máximo de la carpeta temporal es ilimitado. Esto es válido para los servidores Web que tienen suficiente capacidad de almacenamiento en disco duro para la versión sin comprimir y para la versión comprimida de los archivos estáticos almacenados en la carpeta temporal.

Sin embargo, si la escasez de espacio en disco es un problema, seleccione **Limitado a**. Cuando se alcance el tamaño máximo de la carpeta temporal especificado en el campo MB, IIS eliminará 256 archivos para hacer sitio y que se puedan almacenar en caché en la carpeta temporal los nuevos archivos comprimidos. La configuración de una carpeta temporal demasiado pequeña puede afectar al rendimiento, ya que IIS tendrá que volver a comprimir y volver a almacenar en caché los archivos estáticos, lo que significa un mayor uso de la CPU y un mayor tiempo de acceso a la unidad de disco duro.

**Figura 41. La hoja de propiedades principales del servicio WWW**



## 1. Formas de personalizar la compresión HTTP:

Como se ha indicado anteriormente, la opción para comprimir los archivos estáticos de la hoja **Propiedades principales del servicio WWW** sólo comprime los archivos que tienen las extensiones ".htm", ".html" y ".txt". Además, la opción para comprimir archivos de aplicación de esta hoja de propiedades sólo comprime el resultado de los archivos cuyas extensiones son ".dll", ".asp" y ".exe". Estas extensiones de nombre de archivo se guardan como parámetros en las claves HcFileExtensions y HcScriptFileExtensions de la metabase, respectivamente. No es posible personalizar estos parámetros predeterminados desde la hoja de propiedades; sin embargo, se pueden personalizar modificando la metabase directamente, como se muestra en los escenarios siguientes. Para obtener más información acerca de la metabase, busque la palabra clave "metabase" en la documentación en pantalla de IIS 5.0.

En los escenarios siguientes se describe cómo agregar tipos de archivo o quitar tipos de archivo de la lista indicada anteriormente de tipos de archivo que se pueden comprimir. Es preciso realizar los pasos siguientes además de los enumerados anteriormente en la sección "Para habilitar la compresión HTTP".

### Escenario 1:

La opción para comprimir archivos estáticos se ha habilitado en la hoja **Propiedades principales del servicio WWW** (consulte la figura 2). Además de comprimir archivos con las extensiones ".htm", ".html" y ".txt", también puede decidir comprimir archivos que tengan las extensiones ".doc" y ".xls".

Los comandos enumerados en el paso 3 siguiente reemplazan las extensiones de archivo definidas previamente después de reiniciarse los Servicios de IIS. Los nuevos parámetros (".doc" y ".xls") no se anexan a las extensiones de archivo existentes; por tanto, incluya también los tipos de archivo habilitados antes mediante la hoja **Propiedades principales del servicio WWW**.

1. Abra una ventana de símbolo del sistema.
2. Dentro de la ventana de símbolo del sistema, vaya hasta \inetpub\AdminScripts del servidor en el que vaya a habilitar la compresión HTTP.
3. Escriba las líneas siguientes en la ventana de símbolo del sistema:

```
cscript.exe adsutil.vbs set W3Svc/Filters/Compression/GZIP/HcFileExtensions  
"htm" "html" "txt" "doc" "xls"
```

```
cscript.exe adsutil.vbs set W3Svc/Filters/Compression/DEFLATE/HcFileExtensions  
"htm" "html" "txt" "doc" "xls"
```

```
IISreset.exe /restart
```

Utilice el Explorador de Windows para ver cómo las versiones comprimidas de los archivos aparecen en la carpeta que seleccionó como carpeta temporal en la hoja **Propiedades principales del servicio WWW** (consulte la figura 2). Recuerde que estos

archivos sólo se almacenan en caché en la carpeta temporal la primera vez que los solicita un cliente.

### Escenario 2:

La opción para comprimir archivos de aplicación se ha habilitado en la hoja **Propiedades principales del servicio WWW** (consulte la figura 2). De manera predeterminada, esto comprime el resultado dinámico de los archivos cuyas extensiones son ".dll", ".asp" y ".exe". Decide no comprimir el resultado de los archivos que tienen las extensiones ".exe" o ".dll" y sólo comprime el resultado de las secuencias de comandos de los archivos ASP.

Del escenario 1 hemos aprendido que los comandos enumerados en el paso 3 siguiente reemplazan las extensiones de archivo definidas previamente después de reiniciarse los Servicios de IIS. Los nuevos parámetros no se anexan a las extensiones de archivo existentes; por tanto, omita los tipos de archivo no deseados que se habilitaron antes mediante la hoja **Propiedades principales del servicio WWW**.

1. Abra una ventana de símbolo del sistema.
2. Dentro de la ventana de símbolo del sistema, vaya hasta \inetpub\AdminScripts del servidor en el que vaya a habilitar la compresión HTTP.
3. Escriba las líneas siguientes en la ventana de símbolo del sistema:

```
cscript.exe adsutil.vbs set  
W3Svc/Filters/Compression/GZIP/HcScriptFileExtensions "asp"
```

```
cscript.exe adsutil.vbs set  
W3Svc/Filters/Compression/DEFLATE/HcScriptFileExtensions "asp"
```

```
IISreset.exe /restart
```

Si se ha habilitado la compresión tanto de los archivos de aplicación como de los archivos estáticos, el procedimiento del escenario 2 no afectará a la configuración de los archivos estáticos establecida en el escenario 1 ni a la configuración predeterminada de la hoja **Propiedades del servicio WWW**.

### Escenario 3:

Decide seleccionar la opción **Limitado a** para el tamaño máximo de la carpeta temporal y escribe 500 en el campo MB. De manera predeterminada se eliminan 256 archivos cuando se alcanza el tamaño máximo de la carpeta temporal (ahora está establecido como 500 MB). Cree que el rendimiento mejoraría si se eliminaran menos archivos al llegar a este límite y decide cambiar este valor a 10.

1. Abra una ventana de símbolo del sistema.
2. Dentro de la ventana de símbolo del sistema, vaya hasta \inetpub\AdminScripts del servidor en el que vaya a habilitar la compresión HTTP.
3. Escriba las líneas siguientes en la ventana de símbolo del sistema:

```
cscript.exe                adsutil.vbs                set
3Svc/Filters/Compression/Parameters/HcFilesDeletedPerDiskFree 10
```

```
IISreset.exe /restart
```

El número que sigue inmediatamente a HcFilesDeletedPerDiskFree es el número de archivos que se eliminan. En el ejemplo anterior elegimos 10, pero es válido cualquier número entero.

#### Escenario 4:

De manera predeterminada, cuando se selecciona la opción para comprimir archivos estáticos se comprimen y almacenan en caché todos los archivos que tienen las extensiones ".htm", ".html" y ".txt", independientemente del tamaño de archivo original. En el caso de archivos estáticos muy pequeños, por ejemplo 10.000 bytes (10 KB), la versión comprimida del archivo puede ser realmente mayor que el archivo sin comprimir. Sabiendo esto, decide no comprimir los archivos de menos de 20.000 bytes (20 KB).

1. Abra una ventana de símbolo del sistema.
2. Dentro de la ventana de símbolo del sistema, vaya hasta \inetpub\AdminScripts del servidor en el que vaya a habilitar la compresión HTTP.
3. Escriba las líneas siguientes en la ventana de símbolo del sistema:

```
3. cscript.exe                adsutil.vbs                set
   W3Svc/Filters/Compression/Parameters/HcMinFileSizeForComp 20
```
4. IISreset.exe /restart

El número que sigue inmediatamente a HcMinFileSizeForComp es el tamaño mínimo de archivo estático (en KB) que se comprimirá y almacenará en caché.

## 2. Conclusión:

Ya sabe lo que es la compresión HTTP, cómo funciona y cómo decidir si debe utilizarla. Cuando la compresión HTTP está habilitada, los usuarios del sitio Web pueden tener mejores tiempos de descarga si utilizan clientes compatibles con HTTP 1.1. Recuerde probar siempre el rendimiento del servidor antes de decidir si habilita o no la compresión HTTP. Es conveniente seguir supervisando el rendimiento del servidor de manera continuada, especialmente el contador **% de tiempo de procesador**. La supervisión del contador **% de tiempo de procesador** es especialmente importante si comprime el resultado dinámico de los archivos de aplicación, ya que el contenido dinámico se comprime cada vez que lo solicita un cliente.

También ha aprendido cómo habilitar y personalizar aún más la compresión HTTP en el servidor Web. Los escenarios mostrados anteriormente deben ser una buena referencia cuando esté en condiciones de habilitar esta eficaz característica nueva en su servidor Web de IIS 5.

### 3. Documentos RFC

RFC es un acrónimo de *Request for Comments*, un documento en el que se publica un estándar, un protocolo u otra información relativa al funcionamiento de Internet. El RFC se publica realmente, bajo el control de IAB, después de su discusión y sirve como estándar. Es posible obtener los RFC en lugares como InterNIC.

RFC 2068 Hypertext Transfer Protocol - HTTP 1.1

<<http://www.cis.ohio-state.edu/htbin/rfc/rfc2068.html>>

Este documento es una descripción exhaustiva del protocolo HTTP 1.1.

RFC 1951<<http://www.cis.ohio-state.edu/htbin/rfc/rfc1951.html>>

En este documento se describe la especificación de formato de datos comprimidos Deflate integrado en IIS 5.0 y en las versiones 4 y superiores de Internet Explorer.

RFC 1952<<http://www.cis.ohio-state.edu/htbin/rfc/rfc1952.html>>

En este documento se describe la especificación del formato de archivo GZIP que se utiliza para comprimir y descomprimir archivos. Esta especificación está integrada en IIS 5.0, y en Internet Explorer 4 y posteriores.

### 4. Supervisar el rendimiento del procesador

Ayuda en pantalla de IIS 5.0

Busque las palabras clave *"procesador"* y *"% de tiempo de procesador"*

Ayuda en pantalla de Windows 2000 Server

Busque la palabra clave *"Supervisar la actividad del procesador"*

Guía de recursos de Servicios de Microsoft Internet Information Server 5.0, páginas 195 a 197. Este libro forma parte del Kit de recursos de Microsoft Windows 2000 Server - ISBN: 1572318058. Buena introducción relativa a la supervisión de los procesadores del servidor.

Solutions for Poor Server Performance. Un artículo de TechNet escrito por Gary Duthie donde se explican los contadores de rendimiento.

<http://www.microsoft.com/TechNet/iis/sol.asp>

### 5. Compresión de archivos y carpetas

Guía de operaciones de Microsoft Windows 2000 Server, páginas 141 - 147. Este libro forma parte del Kit de recursos de Microsoft Windows 2000 Server - ISBN: 1572318058

Ayuda en pantalla de Windows 2000 Server

Busque las palabras clave "Comprimir un archivo o una carpeta en una unidad NTFS", "Para cambiar las propiedades de archivos o carpetas", "Mostrar las carpetas y los archivos comprimidos en un color diferente", "compactar", "Para comprimir una unidad NTFS" y "Consideraciones cuando se planean particiones de discos".

Si se siguen las recomendaciones de Microsoft en el enlace [Recomendaciones de Microsoft para la compresión](#), se podrá habilitar la compresión, pero aún no sería

funcional para ASP.NET La razón es que las extensiones .aspx de ASP.NET no es un grupo de extensiones válidos comprimibles.

Para hacer esto posible, se debe editar la Metabase. Se puede editar \Windows\System32\inetsrv\MetaBase.xml con el editor de texto que se guste. Se busca "IIsCompressionScheme". Deben haber dos elementos XML, uno para deflate y otro para gzip. Ambos elementos tienen propiedades llamadas HcFileExtensions y HcScriptFileExtensions. Estas contienen una lista delimitada por espacios de extensiones de archivo para contenido comprimible. Como mínimo se necesita adicionar aspx a la lista de extensiones. No te que si las propiedades no tienen nada a la izquierda, entonces todo el contenido, Note that if the properties are left blank, then all content, a pesar de la extensión del archivo, serán comprimidas.

Para ver en detalle la compresión en IIS remitirse a:

<http://www.microsoft.com/windows2000/en/server/iis/default.asp?url=/windows2000/en/server/iis/htm/asp/apro85ir.htm>.

## Anexo E. Publicaciones

### Ambientación:

El presente trabajo de tesis ha permitido generar dos artículos a publicar. Uno formalizado en La Revista Colombiana de Tecnologías de Avanzada y otro en etapa de análisis por parte del publicador, en el sitio [icinco.org](http://www.icinco.org).

La Revista Colombiana de Tecnologías de Avanzada es una publicación periódica del Instituto de Investigación y Desarrollo de Tecnologías Aplicadas. Este instituto pertenece a la Universidad de Pamplona en la ciudad de Pamplona, Norte de Santander, Colombia. El código ISSN es 1692-7257. En el sitio de la Universidad de Pamplona, <http://www.unipamplona.edu.co>, bajo el menú Investigación de la izquierda, en el submenú Institutos se encuentra el enlace hacia el Instituto de Investigación y Desarrollo de Tecnologías Aplicadas.

En este anexo se presenta la primera parte del artículo que contiene un resumen en español e inglés y la introducción. El artículo original presenta los capítulos ESTADO DEL ARTE, DESARROLLO DE LA PROPUESTA, CONCLUSIONES Y BIBLIOGRAFÍA.

### El resumen del artículo:

#### **ANÁLISIS DE RENDIMIENTO EN EL USO DE SERVICIOS WEB MÓVILES**

Eduardo Carrillo Zambrano; Jesús Evelio Ortega Arévalo; Guillermo Sánchez Jaimes  
Universidad Autónoma de Bucaramanga, Maestría en Ciencias Computacionales  
[jeortegaa@hotmail.com](mailto:jeortegaa@hotmail.com)

**Resumen:** El objetivo del presente estudio es generar unas pruebas tendientes a medir el uso de recursos en los extremos, principalmente en el servidor, cuando se pretende optimizar el ahorro en el consumo del ancho de banda tan escaso en los dispositivos móviles. Para tal fin se emplean dos algoritmos de compresión estándar en las plataformas Microsoft Windows y Linux, el gzip y el XMLPPM. Genera recomendaciones y ambienta con respecto a las actuales opciones para implementar Servicios Web. Contiene una Introducción, que define la importancia del estudio; sigue con el Estado del arte, que presenta una visión de lo que ya existe en evaluación del rendimiento y compara objetivos de los diferentes estudios; sigue con el Desarrollo de la propuesta que aclara a qué se quiere llegar en el estudio, las metodologías de trabajo y los esquemas de banco de pruebas; contiene luego un capítulo de conclusiones que formaliza una serie de definiciones y recomendaciones buscando sacar el mayor provecho de los resultados de las pruebas y con el ánimo de generar sugerencias para optimizar el aprovechamiento del ancho de banda en dispositivos móviles; el documento termina con la Bibliografía, una lista de referencias documentales, apoyo teórico al estudio.

**Abstract:** The aim of the present study is to generate a few tests tending to measure the use of resources in the ends, principally in the server, trying to optimize the use of such a scanty bandwidth in the mobile devices. For such purpose there use two algorithms of standard compression in the platforms Microsoft Windows and in Linux, the gzip and the XMLPPM. It generates recommendations and give concepts about current options to implement Web Services. The Introduction contains information about the importance of

the study; it continues with the State of the art, which presents a vision of what already exists in evaluation of the yield and conceptualization in Mobile Web Services; It continues with the Development of the work that clarifies the goal in the study, the methodologies of work and the schemes of test bench; it contains then a chapter of conclusions that formalizes a series of definitions and recommendations seeking to extract the major profit of the results of the tests and with the intention of generating recommendations to optimize the use of the bandwidth in mobile devices; the document finishes with the Bibliography, a list of documentary references, theoretical support to the study.

## INTRODUCCIÓN

Los Servicios Web son una tendencia que busca automatizar la comunicación entre los dispositivos computacionales que usan la pila de protocolos TCP/IP para emitir servicios o recibirlos.

Existen ya opciones en muchas plataformas para emitir este tipo de servicios y vemos en Microsoft Windows .NET Framework y en sistemas tipo UNIX el proyecto Apache Axis.

Ahora, existe paralelamente una creciente expectativa y uso hacia dispositivos que permiten implementar los conceptos de movilidad. Este concepto se ve representado principalmente en los Personal Digital Assistant, o PDAs de los cuales los más representativos son los Palm de la empresa Palm y los Pocket PC que provienen de varios fabricantes pero que se caracterizan por tener como sistema operativo el denominado Windows CE.

Habida cuenta del gran auge de la tecnología de Servicios Web y de la importancia que reviste el poder automatizar la comunicación en Internet o, viendo otras de sus características más importantes, la posibilidad de solicitar o recibir sólo lo que realmente se requiere en un momento dado, y del crecimiento en la importancia, uso, disponibilidad de recursos y desarrollos en el mundo de los PDAs, se hace muy necesario realizar un análisis del rendimiento de los Servicios Web, tendiente a optimizar el aprovechamiento del ancho de banda de los canales a los cuales se conectan los PDAs. Este análisis de rendimiento busca generar algunas recomendaciones con respecto a la forma en que se puede aprovechar mejor el ancho de banda, dar una idea clara de las tecnologías disponibles en el lado del servidor y en el lado de los clientes, y concluir el costo/beneficio de usar las técnicas recomendadas en el aprovechamiento óptimo del consumo del ancho de banda en los canales móviles.