

CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO



cenidet



“MOVIWEB: PLATAFORMA PARA SOPORTAR EL
ACCESO A SITIOS WEB DESDE DISPOSITIVOS
MÓVILES”

T E S I S
PARA OBTENER EL GRADO
DE MAESTRO EN CIENCIAS
EN CIENCIAS COMPUTACIONALES

PRESENTA
ING. JUAN CARLOS OLIVARES ROJAS

DIRECTOR DE TESIS
DR. JUAN GABRIEL GONZÁLEZ SERNA

CODIRECTOR DE TESIS
DRA. AZUCENA MONTES RENDÓN

DEDICATORIA

A Dios por sobre todas las cosas (a nuestro padre celestial y a nuestra madre divina; a Jesucristo: camino, verdad y vida).

A mis padres: Pedro Olivares Salceda y Eva Rojas González por su infinito amor y apoyo; simplemente no se que sería de mi vida sin ustedes.

A mis hermanos: Iván y Pedro Alberto por todas las vivencias que hemos tenido junto y por su comprensión.

A toda la FAMILIA (ustedes bien saben quien son, y para no excluir a nadie simplemente diré: ABUELITOS, TIOS, PRIMOS y SOBRINOS) por todas sus atenciones siempre y a quienes les debo todo lo que soy.

A todas aquellas personas que desde siempre han creído en mí, y a las que no... también.

Juan Carlos Olivares

Juan Carlos Olivares Rojas
Octubre de 2006

AGRADECIMIENTOS

Al Centro Nacional de Investigación y Desarrollo Tecnológico (*cenidet*) por brindarme la oportunidad de seguir preparándome en la vida.

A la SEP y al CoSNET (ahora integrado a la DGEST) por haber brindado el apoyo económico con la beca 102004189PJ para estudios de postgrado.

A quien fuera mi tutor y director de tesis el Dr. Juan Gabriel González Serna por brindarme su asesoría en este trabajo de tesis y por sobretodo, por su valiosa y sincera amistad. A mi codirector de tesis la Dra. Azucena Montes Rendón por su coasesoría y por sus valiosos comentarios que enriquecieron este trabajo.

A los revisores de esta tesis: Dr. René Santaolaya Salgado, M.C. Mario Guillén Rodríguez y en especial al M.C. José Antonio Zárate Marceleño por sus contribuciones y aportaciones a este trabajo.

A todos los profesores, personal académico y manual de este instituto. En especial a los profesores-investigadores Dr. Rodolfo Abraham Pazos Rangel, Dr. Víctor Jesús Sosa Sosa, Dr. Máximo López Sánchez, entre otros.

A todos los profesores del Instituto Tecnológico de Morelia de los cuales obtuve su apoyo siempre: M.C. Felipe Morales López, M.C. Rogelio Ferreira Escutia, M.C. Cristóbal Villegas, M.C. Anastasio Antolino, entre otros. En especial a la M.C. Miriam Zulma Sánchez por las facilidades brindadas en los últimos días.

A todos mis compañeros de la generación 2004-2006 *xoloezcuitles* (es como en teoría de conjuntos: el orden no importa) por laboratorios, Ing. Software: Adrián, Sandy e Ismael; Inteligencia Artificial: Jorge, Jonathan, Edgar, y los paisanos: Paco, Chavo, Chío (¡que lindo es Michoacán!); en especial mis compañeros de Sistemas Distribuidos: Hilda, Michelle y Rafael por todas las vivencias que compartimos juntos durante nuestra estancia en el cenidet.

A todas las demás personas que he conocido durante mi breve estancia en el cenidet de otras generaciones como los SD4: Adriana, Pedro, Lirio, Chuy y Daniel; Gabo, Morgan, Toño, Sonia, Santos y Rosy; los jorges (Ceyca y Vanoye) así como Maricela; e inclusive de otras áreas, generaciones y otras maestrías.

A todas mis amistades y a todas las personas que en el transcurso de mi vida he tenido la grata oportunidad de conocer y convivir. En especial aquellas personas que conocí en mi breve estancia en la ciudad de la eterna primavera: Cuernavaca.

En fin son muchas las personas a las cuales tengo que agradecer y para las cuales ya no tengo palabras y para no omitir a alguien mejor ya no lo pongo, sólo me queda decirles a todos:

¡Mil gracias!

RESUMEN

A pesar del gran auge que han tenido los dispositivos móviles en nuestros días, el acceso a Internet a través de esta clase de dispositivos es sumamente limitado. Esto se debe a que los sitios Web no se diseñan tomando en cuenta las características y limitantes de estos dispositivos con el propósito de hacer los recursos Web más accesibles. Así por ejemplo, las restricciones de pantalla, la poca capacidad de memoria y de almacenamiento, los enlaces de comunicaciones no persistentes, los altos costos de conexión y el deficiente ancho de banda, han frenado la visualización de sitios Web a través de dispositivos móviles.

El presente trabajo pretende “poner la Web en los bolsillos de los usuarios”. Para garantizar que los usuarios puedan visualizar correctamente los recursos Web, se necesitan dos cosas: un servicio de gestión de conexiones que permita visualizar contenidos Web sin importar el estado de conexión del dispositivo, un servicio de acaparamiento de sitios Web y un mecanismo que adapte el contenido Web a las características propias del dispositivo móvil específico (transcodificación). En este trabajo se presenta una herramienta que integra estos servicios y permite mejorar la experiencia de navegación de los usuarios con dispositivos móviles.

El objetivo de esta tesis es hacer que los recursos de la Web sean accesibles desde cualquier dispositivo, en cualquier momento, y en cualquier lugar y como los necesite el usuario. Los beneficios que se obtienen son posibles reducciones en el tamaño de los recursos Web al transcodificarlos y acapararlos en el dispositivo móvil, ahorro de energía de las baterías, así como reducción en los tiempos de acceso a los recursos. Esto ayuda a incrementar los niveles de acceso a la Web a través de dispositivos móviles.

ABSTRACT

In spite of the great popularity which the mobile devices have in our days, the access to Internet through this type of devices is extremely limited. This due because Web sites have not been developed taking into account the characteristics and constrains from these devices in view of making the Web resources more accessible. Thus for example, the screen-restrictions, the low capacity of memory and storage, the communications through no persistent connections, the high costs and the deficient bandwidth of connection, among others, these characteristics have restrained the visualization of Web sites through mobile devices.

The present work tries “to put the Web in the users’ pockets”. In order to guarantee that the users can visualize correctly the Web resources, two things are needed: a mechanism that control disconnections and allow to visualize Web content without concerning the connection state of the device (hoarding), and a mechanism that adapts the Web content to the own characteristics of the specific mobile device (transcoding). In this work a tool that integrates these two mechanisms and allows improving the navigation experience of the users in the mobile Web is presented.

The goal of this thesis consists of which the Web resources are accessible independent of the device, when, where and how the users need. In addition, with this work it is possible to reduce the size of the Web resources when transcoding and hoarding them on the mobile device, energy save of the batteries, as well as speed up access times to the resources, which with takes to diminish costs of accessing to the Web or per time air or by volumen of information This aid to increase the access levels to the Web through mobile devices.

TABLA DE CONTENIDO

CAPÍTULO 1 INTRODUCCIÓN	1
1.1 Introducción	1
1.2 Descripción del problema	1
1.3 Objetivos	2
1.4 Justificación	3
1.5 Beneficios	5
1.6 Antecedentes	6
1.7 Trabajos relacionados (estado del arte)	6
1.8 Alcances de la tesis	11
1.9 Organización del documento	11
CAPÍTULO 2 MARCO TEÓRICO	12
2.1 Acaparamiento	12
2.1.1 Replicación	13
2.1.2 Precarga	14
2.1.3 Caching	14
2.1.4 Diferencias entre acaparamiento, replicación, precarga y caching	15
2.2 Transcodificación	15
2.3 Dispositivos móviles	17
2.3.1 Clasificación	18
2.3.1.1 PDAs (Pocket PC)	18
2.3.1.2 Teléfonos inteligentes (Smartphones)	19
2.3.2 Herramientas de programación	20
2.3.2.1 Conclusiones (herramienta seleccionada)	21
CAPÍTULO 3 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN	22
3.1 Introducción	22
3.2 Arquitectura propuesta	22
3.3 GAP	26
3.3.1 Observador	30
3.3.2 GDL (Gestor de Desconexión Local)	30
3.3.3 GAL (Gestor de Acaparamiento Local)	31
3.4 GAT	34
3.4.1 MT (Mecanismo Transcodificador)	34
3.4.2 MA (Mecanismo Acaparador)	40
CAPÍTULO 4 PRUEBAS	44
4.1 Introducción	44
4.2 Caso de prueba 1: Configuración del GAP	44
4.3 Caso de prueba 2: Recursos sin acaparar y sin transcodificar	47
4.4 Caso de prueba 3: Recursos sin acaparar pero transcodificados	49
4.5 Caso de prueba 4: Recursos acaparados sin transcodificar	53
4.6 Caso de prueba 5: Recursos acaparados y transcodificados	55

4.7 Otras pruebas	56
4.7.1 GAT en otros dispositivos móviles heterogéneos	56
4.7.3 Pruebas de rendimiento	57
4.7.3.1 Pruebas de transcodificación y acaparamiento	59
4.7.3.2 Tiempos de acceso del GAP en dispositivos móviles	65
<i>CAPÍTULO 5 CONCLUSIONES</i>	71
5.1 Conclusiones	71
5.2 Aportaciones	72
5.3 Artículos publicados	72
5.4 Premios y otros resultados	74
5.5 Trabajos futuros	74
<i>ANEXOS</i>	77
Anexo A Costos de acceso a Internet a través de redes de telefonía celular	77
Anexo B Especificación de métodos del GAP	79
Anexo C Archivo de configuración del GAT MT	87
Anexo D Archivo de configuración del GAP	91
Anexo E Archivo de configuración del GAT MA	95
Anexo F Instalación del GAP	98
<i>REFERENCIAS</i>	99

LISTADO DE FIGURAS

<i>Figura 1.1 Problemas más comunes en dispositivos móviles.</i>	2
<i>Figura 1.2 Problema de la visualización de sitios Web en dispositivos móviles (problema del corrimiento).</i>	3
<i>Figura 1.3 Artículos que más cargan los Estadounidenses al salir de casa por las mañana.</i>	4
<i>Figura 1.4 Usuarios de telefonía móvil en México.</i>	4
<i>Figura 1.5 Arquitectura de Moviware.</i>	7
<i>Figura 2.1 Esquema general del proceso de acaparamiento.</i>	12
<i>Figura 2.2 Tipos de transcodificación de documentos.</i>	17
<i>Figura 2.3 Clasificación de dispositivos móviles.</i>	19
<i>Figura 3.1 Modelo general de solución propuesta.</i>	23
<i>Figura 3.2 Arquitectura del sistema por bloques.</i>	24
<i>Figura 3.3 Arquitectura del prototipo desarrollado en el lado cliente (dispositivo móvil).</i>	25
<i>Figura 3.4 Arquitectura del prototipo desarrollado en el lado del servidor.</i>	25
<i>Figura 3.5 Diagrama de casos de uso del GAP.</i>	26
<i>Figura 3.6 Diagrama de clases del GAP parte 1.</i>	27
<i>Figura 3.7 Diagrama de clases del GAP parte 2.</i>	28
<i>Figura 3.8 Diagrama de secuencia cuando el recurso no está en la caché y existe patrón de acaparamiento.</i>	28
<i>Figura 3.9 Diagrama de secuencia para el escenario donde el recurso está en la caché sin importar la conexión.</i>	29
<i>Figura 3.10 Diagrama de secuencia para la visualización de error cuando no existe conexión y el recurso no está en la caché.</i>	29
<i>Figura 3.11 Escenario de visualización de error cuando existe conexión, el recurso no está en la caché y no se pudo obtener recurso en línea.</i>	30
<i>Figura 3.12 Diagrama de secuencia para la visualización de errores cuando existe conexión, el recurso no está en la caché y se pudo obtener en línea pero no transcodificar.</i>	30
<i>Figura 3.13 Diagrama de secuencia del escenario cuando existe conexión, el recurso no está en la caché, y existe patrón de acaparamiento.</i>	31
<i>Figura 3.14 Diagrama de secuencia para el escenario cuando existe conexión, el recurso no está en la caché, y el recurso se obtiene en línea.</i>	31
<i>Figura 3.15 Diagrama de secuencia que modela el escenario cuando existe conexión, el recurso no está en la caché, y el recurso se obtiene en línea mediante transcodificación.</i>	32
<i>Figura 3.16 Diagrama de actividades del procesamiento de acaparamiento.</i>	32
<i>Figura 3.17 Diagrama de actividades del proceso de desconexión en el GAP.</i>	33
<i>Figura 3.18 Arquitectura de la caché del GAP en el dispositivo local.</i>	33
<i>Figura 3.19 Diagrama de actividades del proceso de sincronización de la caché en el GAP.</i>	34
<i>Figura 3.20 Modelo vista controlador aplicado al prototipo desarrollado.</i>	35
<i>Figura 3.21 Proceso general de transcodificación en el GAT MT.</i>	36
<i>Figura 3.22 Diagrama de actividades del proceso de transcodificación.</i>	40
<i>Figura 3.23 Proceso de transcodificación del documento.</i>	41
<i>Figura 3.24 Diagrama de clases del GAT MT.</i>	41
<i>Figura 3.25 Diagrama de clases del GAT MA.</i>	42
<i>Figura 3.26 Sincronización de la caché en el GAT MA.</i>	42
<i>Figura 3.27 Estructura genérica de un sitio Web.</i>	43
<i>Figura 4.1 Interfaz gráfica y estructura interna del GAP.</i>	45
<i>Figura 4.2 Carga del archivo de configuración e interfaz de configuración de desconexiones en el GAP.</i>	46
<i>Figura 4.3 Visualización de peticiones en la interfaz del GAP.</i>	47
<i>Figura 4.4 Visualización de recursos Web sin transcodificar ni acaparar y bitácora generada por el GAP.</i>	48
<i>Figura 4.5 Visualización de recursos Web en HTML reformateado.</i>	49
<i>Figura 4.6 Visualización de recursos Web en WML.</i>	50
<i>Figura 4.7 Recurso transcodificado en XHTML-MP.</i>	50
<i>Figura 4.8 Recurso transcodificado en formato PDF.</i>	51
<i>Figura 4.9 Visualización de un recurso Web transcodificado en formato PS, y texto plano.</i>	52
<i>Figura 4.10 Transcodificación de recursos Web utilizando el GAT MT.</i>	52

<i>Figura 4.11 Mensaje que indica que el proceso de transcodificación se está realizando de manera asíncrona.</i>	53
<i>Figura 4.12 Estructura de la caché del GAT MA.</i>	54
<i>Figura 4.13 Descarga de recursos contenidos en un patrón de acaparamiento.</i>	54
<i>Figura 4.14 Proceso de acaparamiento de manera asíncrona.</i>	55
<i>Figura 4.15 Visualización de recursos Web acaparados cuando existe conexión y cuando no.</i>	56
<i>Figura 4.16 En el lado izquierdo, se muestran los mensajes mostrados al realizar la comprobación de restricciones al descargar un sitio Web acaparado.</i>	57
<i>Figura 4.17. Visualización de un recurso Web en HTML reformateado.</i>	58
<i>Figura 4.18. Transcodificación de un recurso Web en diferentes formatos y plataformas.</i>	58
<i>Figura 4.19 Tiempos obtenidos de la transcodificación de recursos Web.</i>	61
<i>Figura 4.20 Porcentaje de ahorro de energía utilizando dispositivos móviles diversos a través del GAP.</i>	61
<i>Figura 4.21 Tamaño y porcentaje de ahorro obtenidos al transcodificar recursos Web.</i>	62
<i>Figura 4.22 Tamaños y porcentaje de ahorro obtenidos al acaparar sitios Web.</i>	63
<i>Figura 4.23 Tiempos de acaparamiento obtenidos con y sin transcodificación.</i>	63
<i>Figura 4.24 Porcentaje de páginas que cumplen con los niveles de accesibilidad WCAG 1.0.</i>	64
<i>Figura 4.25 Porcentajes de eficiencias en el cumplimiento de los niveles de accesibilidad WCAG 1.0.</i>	64
<i>Figura 4.26 Tiempos de respuesta del GAP en base a emuladores.</i>	66
<i>Figura 4.27. Comparativa de los tiempos de respuesta del GAP en diversos dispositivos Pocket PC.</i>	67
<i>Figura 4.28 Comparativa de tiempos promedios de procesamiento en el GAP en plataformas PC y PPC.</i>	68
<i>Figura 4.29 El desempeño de equipos Smartphone es inferior al de Pocket PC.</i>	70
<i>Figura 5.1 Implementación del GAP en J2ME para ejecutarse en más plataformas móviles.</i>	75

LISTADO DE TABLAS

<i>Tabla 1-1 Estado del arte.</i>	8
<i>Tabla 2-1 Comparativa entre los conceptos de replicación, acaparamiento, precarga y caching.</i>	16
<i>Tabla 3-1 Tabla de verdad-contradicción que explica el funcionamiento básico del sistema.</i>	23
<i>Tabla 3-2 Ejemplo de encabezado de solicitud HTTP.</i>	35
<i>Tabla 3-3 Documento HTML original.</i>	36
<i>Tabla 3-4 Documento transformado en XHTML.</i>	37
<i>Tabla 3-5 Documento en XML.</i>	38
<i>Tabla 3-6 Archivo intermedio generado.</i>	38
<i>Tabla 3-7 Extracto de la plantilla generada para PDF.</i>	39
<i>Tabla 3-8 Documento generado en WML.</i>	39
<i>Tabla 3-9 Encabezado de respuesta generado.</i>	40
<i>Tabla 4-1 Subcaso 1.1: carga del archivo de configuración.</i>	45
<i>Tabla 4-2 Subcaso 1.2: configuración del sistema.</i>	46
<i>Tabla 4-3 Subcaso 1.3: otras opciones</i>	46
<i>Tabla 4-4 Subcaso 2.1: visualización de mensajes en la interfaz del GAP.</i>	47
<i>Tabla 4-5 Subcaso 2.2: visualización de la bitácora.</i>	47
<i>Tabla 4-6 Subcaso 2.3: visualización de errores.</i>	48
<i>Tabla 4-7 Subcaso 2.4: visualización de recursos Web en línea sin transcodificar ni acaparar.</i>	48
<i>Tabla 4-8 Subcaso 3.1: Recursos Web transcodificados en diversos formatos.</i>	49
<i>Tabla 4-9 Subcaso 3.2: funcionamiento de la caché del mecanismo transcodificador.</i>	50
<i>Tabla 4-10 Subcaso 3.3: Proceso de transcodificación asíncrona.</i>	51
<i>Tabla 4-11 Subcaso 3.4: visualización de errores en el proceso de transcodificación.</i>	52
<i>Tabla 4-12 Subcaso 4.1: obtención de recursos de un patrón de acaparamiento asíncrono o asíncrono.</i>	53
<i>Tabla 4-13 Subcaso 4.2: envío de un sitio acaparado al cliente.</i>	54
<i>Tabla 4-14 Subcaso 5.1: visualización de recursos acaparados y transcodificados.</i>	55
<i>Tabla 4-15 Subcaso 5.2: obtener y validar un sitio Web acaparado en el GAP.</i>	55
<i>Tabla 5-1 Selección del puerto 10800 del GAP por parte de la IANA.</i>	74
<i>Tabla A-1 Costos de acceso a Internet en México desde un dispositivo móvil haciendo uso de la red de telefonía celular.</i>	77
<i>Tabla A-2 Beneficios tangibles de la utilización del prototipo implementado.</i>	78
<i>Tabla A-3 Clases que conformar el prototipo GAP.</i>	79
<i>Tabla A-4 Métodos de la clase archivoXML.</i>	79
<i>Tabla A-5 Métodos de la clase Configuración.</i>	80
<i>Tabla A-6 Miembros de la clase Desconexión.</i>	81
<i>Tabla A-7 Métodos de la clase GAL.</i>	81
<i>Tabla A-8 Métodos que conforman la clase GAP.</i>	82
<i>Tabla A-9 Métodos de la clase GDL.</i>	82
<i>Tabla A-10 Métodos de la clase Log.</i>	83
<i>Tabla A-11 Métodos de la clase Observador.</i>	83
<i>Tabla A-12 Métodos de la clase paraconf.</i>	84
<i>Tabla A-13 Métodos de la clase Plataforma.</i>	84
<i>Tabla A-14 Métodos de la enumeración Plataformas.</i>	84
<i>Tabla A-15 Métodos de la clase Principal.</i>	84
<i>Tabla A-16 Métodos de la clase Procesos.</i>	85
<i>Tabla A-17 Métodos de la clase WinProcesos.</i>	85
<i>Tabla A-18 Métodos de la clase Sincronizador.</i>	85
<i>Tabla A-19 Métodos de la clase singleton.</i>	85
<i>Tabla A-20 Métodos de la Clase Validador.</i>	85
<i>Tabla A-21 Métodos de la clase ZIP.</i>	86
<i>Tabla A-22 Archivo de configuración del GAT MT.</i>	87
<i>Tabla A-23 Descripción del archivo de configuración del GAT MT.</i>	88
<i>Tabla A-24 Archivo índice de la caché.</i>	89
<i>Tabla A-25 Archivo de configuración del GAP.</i>	91
<i>Tabla A-26 Descripción de la etiqueta GAT.</i>	91
<i>Tabla A-27 Ejemplo del archivo de bitácora del sistema (mensajes.log).</i>	92
<i>Tabla A-28 Descripción de la etiqueta GAP.</i>	92

<i>Tabla A-29 Estructura del archivo contenedor de patrones.</i>	93
<i>Tabla A-30 Archivo patrón del sitio Web http://www.cenidet.edu.mx/</i>	93
<i>Tabla A-31 Descripción de la etiqueta Almacenamiento.</i>	93
<i>Tabla A-32 Descripción de la etiqueta Desconexión.</i>	94
<i>Tabla A-33 Archivo de configuración del GAT MA.</i>	95
<i>Tabla A-34 Descripción del archivo de configuración del GAT MA.</i>	95
<i>Tabla A-35 Ejemplo de del archivo de configuración de patrones.</i>	97
<i>Tabla A-36 Archivo patrón del sitio http://www.cenidet.edu.mx/</i>	97
<i>Tabla A-37 Archivo de control de direcciones del GAT MA.</i>	97

GLOSARIO

- Agente:** Es un componente de software intermedio que capacita a un objeto que resida en un cliente para enviar un mensaje a un método que esté encapsulado en otro objeto que resida en el servidor [Way95].
- Araña:** También recibe el nombre de spider, robot o crawler, es un programa encargado de analizar un documento Web, obtener los enlaces contenidos en el documento, descargarlos y nuevamente analizarlos. El proceso es recursivo por lo que generalmente se tiene una condición de paro que puede ser un número máximo de niveles, de documentos, o enlaces fuera del dominio analizarlo. Algunas de sus funciones son la indexación de documentos en los buscadores Web, la descarga de sitios Web, la búsqueda de información, etc.
- ARM:** *Advanced RISC Machines*. Máquinas RISC avanzadas. Es diseñado y manufacturado por Acorn Computer Group. Es un microprocesador de 32 bits diseñado en arquitectura RISC, el cual permite optimizar la velocidad de procesamiento usando técnicas como el *pipelining* (entubamiento) [GS05].
- Bluetooth:** Es el estándar 802.15 para redes WPAN propuesto por el IEEE. Originalmente fue creado por un conjunto de empresas del sector informático para solucionar el problema de la interconexión de periféricos de manera inalámbrica. Son redes inalámbricas de corta distancia, no alcanzan distancias mayores a los 10 metros. Trabaja en la frecuencia de los 2.4 Ghz. [Fir04].
- CISC:** *Complex Instruction Set Code*. Conjunto de códigos de instrucción complejos. Es una arquitectura de microprocesadores que contienen una gran cantidad de instrucciones, para resolver diversos problemas, tiene la ventaja que en una sola instrucción se resuelve un solo problema, su desventaja radica en el tiempo que se tarda en buscar una instrucción.
- CSD:** Entre las redes basadas en conmutación de circuitos destaca CSD (*Circuit Switch Data*, conmutación de circuitos de datos) y HSCSD (*High Speed Circuit Switch Data*, conmutación de circuitos de datos de alta velocidad) cuya característica principal es que se cobra por tiempo de conexión. CSD fue originalmente creado para redes de telefonía celular TDMA (*Time Division Multiple Access*, acceso múltiple por división de tiempo) aunque actualmente se puede utilizar en otros tipos de redes de telefonía celular. Tiene una velocidad de transmisión de 9.6 Kbps. Funciona de manera similar a una llamada de voz solo que simplemente se envían datos a través de la línea. HSCSD es una mejora de CSD y se ejecuta sobre redes GSM. Las velocidades pueden ser de 14.4, 57.6 o hasta 115 Kbps, por lo que se considera una red 2.5 G.
- Flash ROM:** *Flash Read Only Memory*. Memoria flash de solo lectura. Tipo de Memoria de solo lectura que mediante un proceso eléctrico puede escribir datos los cuales tienen la característica de ser persistentes; es decir, los datos se mantienen a pesar de no estar conectados a la energía eléctrica. Actualmente son el medio más utilizado para almacenar información en dispositivos móviles y empotrados.
- GSM/GPRS:** GSM (*Global System for Mobile communications*, sistema global para las comunicaciones móviles) es un estándar mundial para teléfonos móviles digitales. Corresponde a redes de telefonía celular 2G. Es una red ubicua ya que cubre a 210 países y 2000 millones de personas. Puede operar en las bandas de 850, 900, 1800 o 1900 Mhz. Se pueden enviar datos a velocidades 9 Kbps, con la ventaja de ser totalmente digital (a diferencia de CSD). GPRS (*General Packet Radio Service*, servicio general de paquetes por radio) es una tecnología

digital de telefonía móvil 2.5G. Es una red basada en conmutación de paquetes cuya característica principal es que se tarifa en base a datos transmitidos ya que la información se envía por paquetes y no por tiempo. Alcanza velocidades promedios de 54 Kbps, hasta un máximo teórico de 236.8 Kbps. Algunos autores consideran a GPRS = GSM + IP.

- Middleware:** No existe una definición única para el término middleware, debido principalmente a que su significado depende del ambiente de la aplicación. Middleware es una arquitectura de tres capas: clientes, intermediarios, y servidores. Esta infraestructura facilita la interacción entre módulos de software distribuido. La función principal de una capa middleware es ocultar la complejidad del ambiente de red, ya que el middleware enmascara la heterogeneidad de plataformas de cómputo, sistemas operativos, lenguajes de programación, tecnologías de red, etc. [Pre02].
- MIPS:** Es un microprocesador del tipo RISC de 32 bits desarrollado por la compañía japonesa NEC, cuya característica principal es el entubamiento. Cada instrucción en promedio es realizada en dos ciclos de reloj. Este tipo de microprocesadores son muy usados en ambientes de servidores, estaciones de trabajo y mainframes. Desde su diseño se trató de portar toda la funcionalidad y robustez con que cuentan estos microprocesadores al ambiente móvil
- Palm:** PalmOS es un sistema operativo para dispositivos móviles creado por la compañía Palm [palm06], su característica principal es que es ligero y versátil. Tiene una interfaz muy intuitiva y para muchos expertos la mejor para esta clase de dispositivos. Soporta una gran variedad de microprocesadores móviles. El sistema no es multitarea pero permite conmutar entre aplicaciones guardando su estado para posteriormente recuperar donde se quedó la aplicación, dando la sensación de multitarea y sobre todo agilizando los tiempos de ejecución (se debe recordar que en un dispositivo móvil, sólo se puede ver una aplicación a la vez).
- Pipelining:** “Entubamiento”. Modo de ejecución que tienen algunos microprocesadores de ejecutar dos o más instrucciones de manera paralela, esto se logra canalizando cada nueva instrucción después de un ciclo de reloj de la anterior instrucción.
- Proxy (intermediario):** Servidor que actúa como intermediario entre el cliente y el servidor, haciendo las solicitudes del cliente y pasando los resultados nuevamente al cliente. Servidor especial encargado, entre otras cosas, de centralizar el tráfico entre Internet, de forma que evita que cada una de las máquinas de la red interior tenga que disponer necesariamente de una conexión directa a la red.
- RISC:** *Reduce Instruction Set Code*. Conjunto reducido de códigos de instrucciones. Es una arquitectura de microprocesadores que se caracteriza por tener pocas instrucciones, las cuáles son lo más sencillas posibles; además, se cuentan con gran cantidad de memoria auxiliar (registros) para que su velocidad sea más rápida. En pocos ciclos de reloj se realizan muchas instrucciones.
- Shell:** Con este nombre se designa a la interfaz que presenta al usuario un sistema operativo. Puede ser a través de comandos o en el caso de los dispositivos móviles a través de una interfaz gráfica.
- SH3:** Es desarrollado por la compañía japonesa Hitachi, pertenece a la familia de microprocesadores Super H (Hitachi), los cuales son microprocesadores de 32 bits con instrucciones de 16 bits [HNS+98].
- Sistemas empotrados:** *Embedded Systems*. Otros nombres que reciben son sistemas embebidos o incrustados. De acuerdo con [emb06], son sistemas de propósito

especial en donde tanto el hardware como el software se encuentran encapsulados en una sola pieza y son utilizados en una gran variedad de aplicaciones específicas. Ejemplo de ellos son los cajeros automáticos, las terminales punto de venta, las máquinas despachadoras de alimentos, los electrodomésticos inteligentes, entre otros. Según un estudio reciente, los norteamericanos interactúan en promedio con 90 dispositivos empotrados al día.

Squid: Es un servidor Proxy caché cuya funcionalidad principal es mantener los recursos solicitados con mayor frecuencia por los usuarios de una intranet, con el fin de proporcionar respuestas en tiempos más eficientes.

Symbian: Es un sistema operativo para dispositivos móviles de 32 bits. Es un sistema ligero cuya característica principal es que es muy confiable. Es la evolución del sistema operativo EPOC utilizado en dispositivos empotrados de tiempo real. Está diseñado ex profeso para teléfonos celulares para sacarle provecho a las redes de telefonía celular de 2.5 G y 3G. Es un sistema muy seguro. Se originó debido a que en la antigüedad los teléfonos celulares utilizaban diferentes sistemas operativos aún incluso del mismo fabricante, provocando una gran incompatibilidad entre aplicaciones [sym06].

WAP: *Wireless Application Protocol*. Protocolo de aplicaciones inalámbricas. Es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas. Se trata de la especificación de un entorno de aplicación y de conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos se pueden utilizar para acceder a correo electrónico, grupo de noticias y otros.

WBMP: *Wireless BitMaP*. Mapa de bits inalámbrico. Formato para representar imágenes en WML cuya característica principal es que se encuentran optimizados para dispositivos móviles. Son imágenes en blanco y negro de baja resolución.

WiFi: *Wireless Fidelity*. Fidelidad Inalámbrica. Es el estándar 802.11 propuesto por la IEEE para redes inalámbricas WLAN. Utiliza la banda ISM (*Industrial Scientific Medic*, industrial científica y médica) la cual está a una frecuencia de 2.4 Ghz. (802.11b), o bien puede utilizar la banda de los 5 Ghz (802.11a). Actualmente está muy extendido el estándar 802.11g que es compatible con los estándares anteriores y puede alcanzar velocidades hasta de 54 MB (en algunos casos hasta 108). No tiene alcances mayores a los 100 metros [Mon05].

Windows CE (Windows Mobile): Windows CE es el sistema operativo para dispositivos electrónicos empotrados cuya finalidad es mantener una versión lo más fiel Windows de escritorio en dispositivos de cómputo con características limitadas. Es un sistema operativo de 32 bits multitarea y de tiempo real. Porta la mayoría de las APIs de Win32 (Windows de computadoras de escritorio) pero siendo totalmente diferentes en su arquitectura [wce06]. Dentro de los sistemas empotrados se encuentran ubicados los dispositivos móviles y debido a su alto grado de penetración en el mercado, surgió una variante para dispositivos móviles a la que simplemente se le denomina Windows Mobile.

WLAN: *Wireless Local Area Network*. Es un sistema de comunicación de datos inalámbrico flexible muy utilizado como alternativa a la LAN cableada o como una extensión de ésta.

WML: *Wireless Markup Language*. Lenguaje de marcado inalámbrico. Es un lenguaje de marcado basado en XML. Está estructurado en una unidad denominada deck (baraja) la cual puede contener una o más entidades llamadas cards (tarjetas). Las etiquetas en su gran mayoría son HTML, se han eliminado aquellas que causan mucho problema en los dispositivos móviles. [wml06].

- WPAN:** *Wireless Personal Area Networks*. Red Inalámbrica de Área Personal. Una PAN es una solución de redes, que realza nuestro espacio personal ya sea de trabajo o privado, por una red con una variedad de dispositivos personales.
- XHTML:** *eXtensible HyperText Markup Language*. Lenguaje de marcado de hipertexto extensible. Es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0. Los tipos de documentos de la familia XHTML están basados en XML, y diseñados fundamentalmente para trabajar en conjunto con agentes de usuario basados en XML.
- XHTML-MP:** *XHTML-Mobile Profile*. Es una subespecificación de XHTML con algunas etiquetas extras. Es desarrollado por la compañía Nokia y se ha convertido en un estándar de facto para visualizar recursos Web en dispositivos móviles hoy en día [mp06].
- XML:** *eXtensible Markup Language*. Lenguaje de marcado extensible. Lenguaje desarrollado por el W3C para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML, aunque no es una extensión ni un componente de éste.
- XSL:** *eXtensible Stylesheet Language*. Lenguaje extensible de hojas de estilo. XML sólo define y estructura datos pero por sí solo no define cómo se deben presentar. Con este lenguaje se permite indicar cómo debe mostrarse al usuario cada etiqueta XML definida en nuestro lenguaje.
- XSLT:** *eXtensible Stylesheet Language Transformation*. Lenguaje extensible de hojas de estilo para transformación. Este estándar permite convertir la estructura de un documento XML a otro formato (basado en XML) adecuado para su lectura. Por lo tanto, permite definir una presentación para un documento de XML. Es decir, es un lenguaje para procesar estructuras y convertirlas en textos con legibilidad adecuada. De esta forma, una hoja de estilo de XSLT contiene una serie de reglas que determina cómo va a ocurrir la transformación. Cada regla se compone de un patrón (pattern) y una acción o plantilla (template). De este modo, cada regla afecta a uno o varios elementos del documento de XML.

CAPÍTULO 1

INTRODUCCIÓN

“Si supiera qué es lo que estoy haciendo, no lo llamaría investigación aún, ¿verdad?”
Albert Einstein

1.1 Introducción

En el pasado el paradigma de la computación era: *“una computadora múltiples usuarios”*. A finales de los años 70s y principios de los 80s, el paradigma de la informática cambió a *“una computadora un usuario”*; debido a la aparición de las computadoras personales (PC). Esto se refleja en los slogan de compañías como Microsoft: *“Una PC en cada escritorio y en cada hogar”* [mic75] o como Apple: *“Una persona, una computadora”* [app76]. A finales de los 80s el paradigma informático cambió a *“muchas computadoras muchos usuarios”*, debido a la aparición de las redes de computadoras y sobre todo a la aparición de la Web y popularización del Internet. Así, algunos slogan como el de Sun Microsystems reflejan esta etapa: *“La red es la computadora”* [sun87].

En la actualidad, gracias a los grandes descubrimientos en ciencia y tecnología, es posible que las personas puedan entre otras cosas, comunicarse y compartir información entre sí en cualquier momento, a toda hora y en todo lugar; a esto se le ha denominado cómputo penetrante o ubicuo (*pervasive computing*). Uno de los avances tecnológicos que ha hecho posible todo esto es, sin duda, los dispositivos móviles. Esto ha traído como resultado un cambio radical en el paradigma de la computación por lo que a partir del año 2000 se empieza a hablar del nuevo paradigma: *“una persona múltiples computadoras”*. Como alguna vez dijo Bruno Kreisky (canciller austriaco): *“lo que las redes de ferrocarril, carreteras y canales fueron en otra época, las redes de telecomunicaciones y computación son hoy”* [DD98].

La Web ha resultado una revolución en los medios de comunicación como lo fue la radio y la televisión en su tiempo. A pesar de que todas las estadísticas e indicadores (*ver sección 1.4*) señalan que los dispositivos móviles están en constante crecimiento y dominan ya gran parte del mercado, no se están utilizando para acceder a la Web. Esto se debe a que los dispositivos móviles presentan ciertas limitantes.

1.2 Descripción del problema

Los dispositivos móviles pese a su creciente popularidad, presentan muchas limitantes, entre las cuales destacan las siguientes:

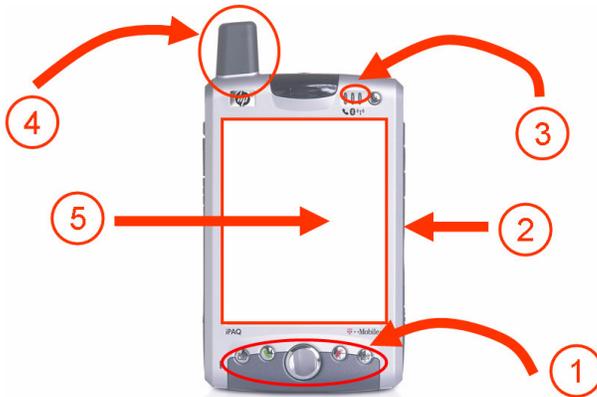


Figura 1.1 Problemas más comunes en dispositivos móviles.

1. Métodos de entrada de información deficientes (teclados pequeños, reconocimiento de escritura ineficaz, etc.).
2. Cuentan con pocos recursos en comparación con una PC de escritorio (poca memoria RAM, poco espacio de almacenamiento, pocos periféricos, microprocesadores más lentos, etc.).

3. Suministro finito de energía (entre más capacidad de procesamiento mayor consumo de energía).
4. Debido a que utilizan interfaces de red inalámbricas y a su alta movilidad, las desconexiones son frecuentes.
5. El despliegue de la información es limitado debido a que se tienen pantallas pequeñas, con poca resolución y gama de colores.

Estas limitantes son las que más han influenciado en la poca utilización de dispositivos móviles para acceder a la Web. Otros problemas que no corresponden a las limitaciones de los dispositivos móviles pero si influyen en el acceso a la Web desde dispositivos móviles son los altos costos de conexión a Internet (ver Anexo A) y el poco ancho de banda (lo cual hace que la descarga de la información sea más lenta).

Todos estos problemas conllevan a una mala experiencia de navegación por parte de los usuarios y a que elijan otras alternativas de acceso a la Web. Los problemas que nos enfocamos a resolver son los últimos dos y que a continuación se describen brevemente.

Por un lado, las aplicaciones Web requieren servicios de transporte de datos que mantengan enlaces persistentes y orientados a conexión. Por esta razón la arquitectura cliente/servidor no es adecuada para escenarios de cómputo móvil, ya que el modelo de interacción de esta arquitectura es un gran consumidor de tiempo e ineficiente en presencia de eventos de desconexión.

En lo que se refiere a las dimensiones de la pantalla en dispositivos móviles, el problema que identificamos es que se necesita mucha interacción por parte del usuario. El usuario tiene que realizar desplazamientos horizontales y verticales para visualizar la página Web de forma completa (ver Figura 1.2). Aunado a esto algunos navegadores en dispositivos móviles no soportan algunos objetos Web (e.g. tablas, frames, formularios, scripts, animaciones, applets, etc.), por lo que a veces no se pueden visualizar o si lo hacen es de manera incorrecta.

1.3 Objetivos

El objetivo general de esta tesis es el diseño e implementación de un prototipo de servicio intermediario para plataforma Windows CE, que gestione el acaparamiento de páginas Web transcodificadas tomando en consideración las características y

limitaciones de los dispositivos móviles. De tal forma que se garantice la correcta visualización de recursos Web sobre esta clase de dispositivos.



Figura 1.2 Problema de la visualización de sitios Web en dispositivos móviles (problema del corrimiento).

Entre los objetivos particulares se encuentran:

1. Evaluación de nuevas tecnologías y herramientas de programación para dispositivos móviles caso concreto: Pocket PC (PPC).
2. Desarrollo de software propietario en plataformas no convencionales, específicamente para microprocesadores ARM, MIPS y SH3.
3. Integración y adaptación del módulo de gestión de acaparamiento de sitios Web [Ver03] perteneciente a la arquitectura Moviware [Gon03], para la gestión de los recursos que se acaparán en el dispositivo móvil PPC.
4. Integración y adaptación del módulo transcodificador de contenidos Web [Uri04] perteneciente a la arquitectura Moviware, para realizar la transformación de las páginas Web que se van a acaparar en el dispositivo móvil PPC.

La hipótesis que se planteó con esta tesis fue probar que con la utilización de servicios de acaparamiento y transcodificación a páginas Web, es posible visualizar sitios Web en dispositivos móviles sin importar el estado de la conexión ni las limitaciones de dichos dispositivos.

1.4 Justificación

Los dispositivos móviles ya forman parte de nuestra vida diaria. Un estudio realizado por la compañía SonyEricsson [gprs06] muestra que el 90% de los estadounidenses

salen de sus casas por las mañanas con un teléfono celular, sólo superados por el uso del llavero y de la cartera o monedero (ver Figura 1.3). Otros dispositivos como PDAs, agendas electrónicas, reproductores de música, consolas de videojuegos portátiles y cámaras digitales están creciendo poco a poco pero su uso todavía es limitado. También se puede observar que estos últimos dispositivos se están integrando en dispositivos móviles por lo que se prevé que este mercado siga creciendo en los próximos años.

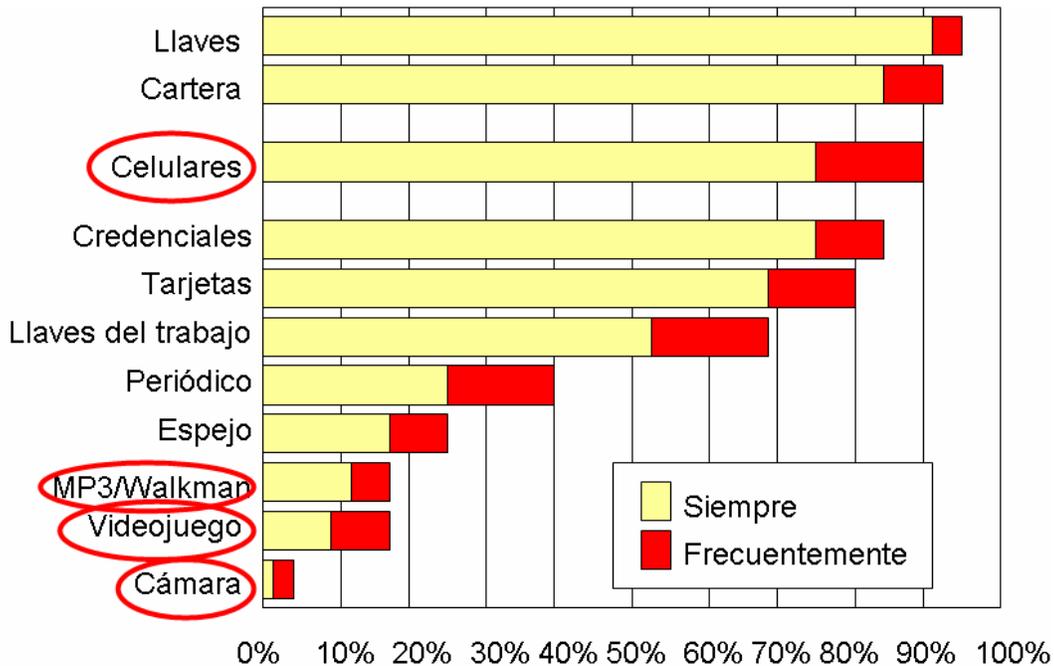


Figura 1.3 Artículos que más cargan los Estadounidenses al salir de casa por las mañana.

En México según el último censo de población el 2005 [ine05] existen un total de 105 millones de habitantes de los cuales casi 50 millones tienen un teléfono celular [cft06] (ver Figura 1.4). Es decir, en México prácticamente el 50% de la población tiene un teléfono celular.

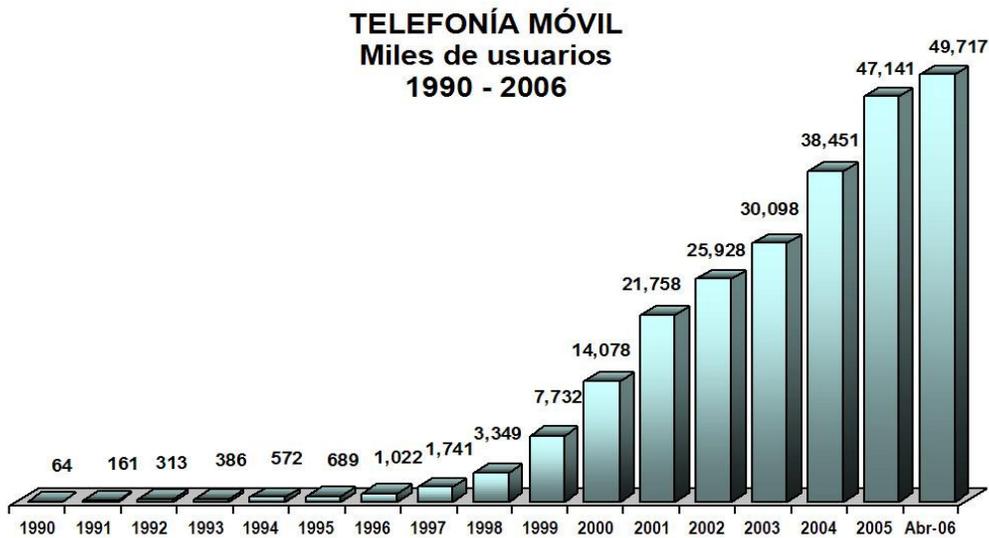


Figura 1.4 Usuarios de telefonía móvil en México.

Además, de acuerdo a los estudios realizados por [inf06], se cree que en cinco años los dispositivos portátiles y los teléfonos celulares reemplazarán a las computadoras como medio de acceso a Internet en América latina. La principal razón es que las PCs son caras, su vida útil muy corta, y buena parte de la población considera que son difíciles de usar. En este sentido, no se cree que los dispositivos móviles sustituyan a las PCs, sino simplemente el acceso a la Web cada día se realizará más a través de dispositivos móviles, ya que de manera general son más baratos y más versátiles; aunque como todo, también existen sus excepciones.

Toda las estadísticas mostradas en esta sección reflejan que el uso de dispositivos móviles va en creciente aumento debido principalmente a su tamaño y a que su poder de procesamiento y versatilidad van creciendo día con día. A pesar de estos avances, el acceso a Internet desde dispositivos móviles aún es sumamente limitado. De acuerdo con [Pau06], se estima que los usuarios de dispositivos móviles acceden sólo 30 minutos al mes a la Web, envían en promedio 80 mensajes SMS y 15 mensajes MMS y en promedio 300 minutos de voz. Estas estadísticas reflejan que existen problemas para acceder a la Web desde dispositivos móviles (ver sección 1.2).

1.5 Beneficios

Entre los beneficios que se pueden obtener de esta tesis se encuentran:

1. Visualización de páginas Web en modo de desconexión en dispositivos móviles, de manera transparente para el usuario (no se requiere de una conexión persistente para acceder a recursos de la Web).
2. Agilizar los tiempos de acceso a páginas Web, al tener sitios Web acaparados de manera local, cuando se presenten desconexiones (de acuerdo a pruebas realizadas el acceso a la caché es hasta 85% más rápido que acceder a recursos externos).
3. Facilidad de administración y por ende de programación, al no tener páginas distintas para distintas plataformas. Dentro de este beneficio se obtienen los siguientes:
 - a. Incrementar la cuota del mercado y el alcance de la audiencia.
 - b. Contenido reutilizable por múltiples formatos o dispositivo.
 - c. Reduce el mantenimiento del sitio.
 - d. Permite la reutilización de contenido.
 - e. Menor carga del servidor.
 - f. Menor ancho de banda requerido.
4. Ahorro de energía en dispositivos que dependen de un suministro finito. Esto como consecuencia de trabajar en modo de desconexión (de acuerdo a pruebas realizadas, se estima un ahorro aproximado del 9% de la batería de un dispositivo móvil cuando se utiliza nuestro prototipo que cuando no).

5. Ahorro en tiempo aire de equipos que se conecten a través de la red de telefonía celular para transmitir información (no se requiere conexión y se ahorra en el tamaño de los recursos y por ende en tiempo de conexión).

En otras palabras, esta tesis pretende maximizar el rendimiento al visualizar sitios Web en dispositivos móviles minimizando los costos que esto involucra.

1.6 Antecedentes

Este proyecto forma parte de la arquitectura del proyecto *Moviware* [Gon03]. La finalidad de *Moviware* consiste en dar soporte a clientes móviles inalámbricos que operan en un ambiente de red inestable, sujetos a experimentar diversos eventos mientras se encuentran trabajando. Es una plataforma prototipo que se compone de los siguientes módulos:

1. Gestor de desconexión y reconexión [Ala02] y [Jua05].
2. Generador de patrones de acceso a sitios Web basado con algoritmos de minería de reglas de asociación [Val02] y [Her05].
3. Gestor de acaparamiento de recursos informáticos [Ver03].
4. Transcodificador de contenidos Web [Uri04].

La arquitectura de *Moviware* se muestra en la

Figura 1.5. En el óvalo con línea continua se muestra el área principal donde se centra este proyecto. Es en este punto, donde se tiene la mayor aportación de esta tesis, ya que se ha desarrollado un servicio intermediario para dispositivos móviles; es decir, existen pocos servicios para dispositivos móviles que se implementen directamente sobre esta plataforma móvil.

En los óvalos con líneas punteadas se muestran los módulos que se adaptaron y modificaron con la finalidad de servir a los dispositivos móviles. Por otra parte, en el óvalo con líneas punteadas se muestra el módulo del cual se obtienen los patrones de acceso a los sitios Web para realizar acaparamiento.

1.7 Trabajos relacionados (estado del arte)

Hasta el momento, no se ha encontrado en la literatura especializada referencias a trabajos relacionados, sólo se han encontrado trabajos que tiene cierta similitud en ciertas áreas pero no en su totalidad. Por una parte, se tienen trabajos que realizan acaparamiento y por otra parte, trabajos que realizan transcodificación pero de manera independiente

Los trabajos relacionados con el acaparamiento en su gran mayoría están enfocados hacia base de datos, sistemas de archivos distribuidos y precarga de datos (algoritmos de obtención de patrones de acaparamiento). En lo que respecta a trabajos de

transcodificación están más enfocados a la personalización de documentos; otros trabajos se encargan de resolver alguna problemática particular del proceso de transcodificación; por ejemplo, qué hacer con las imágenes, frames, tablas, formularios, etc., o cómo se debe transformar el documento para no perder semántica ni contenido.

En la Tabla 1-1, se muestran los trabajos relacionados más importantes en forma de cuadro comparativo. A continuación se describe brevemente cada uno.

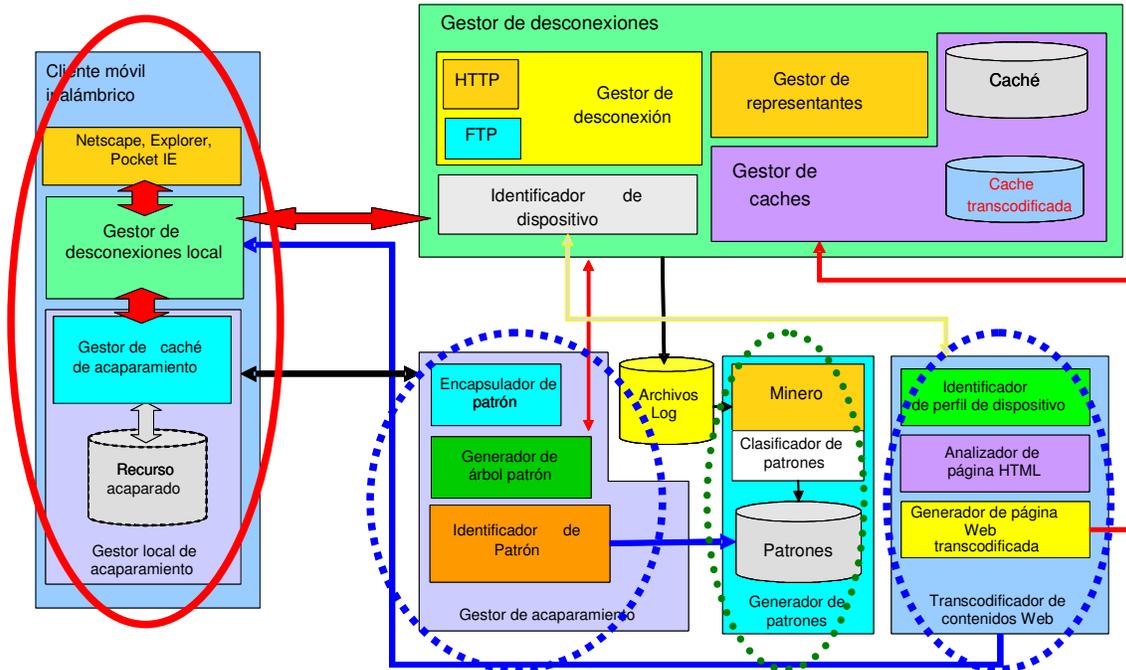


Figura 1.5 Arquitectura de Movieware.

Skweezer [skw06] es un portal Web que se encarga de transformar documentos para tratar de adaptarlos a dispositivos móviles. Al ser un portal tiene la ventaja de ser multiplataforma (sólo se necesita que la plataforma posea un navegador Web). El usuario introduce en un formulario la URL que desea transcodificar y el sistema devolverá al usuario la página solicitada. La desventaja de este esquema es que no permite una navegación libre dado que el usuario necesita ir introduciendo las URLs a transcodificar. El formato de la transcodificación es HTML (reformateado). No cuenta con una caché por lo que el proceso de transcodificación se tiene que repetir cada vez que se hace la solicitud a un recurso ya transformado. No realiza acaparamiento por lo que no se puede trabajar en desconexión tampoco implementa un modelo asíncrono de comunicación. Es un servicio gratuito con publicidad, y se tiene una versión de cuota (venden soluciones para portales).

Avantgo [ava06] es una herramienta comercializada por suscripción; es decir, se cobra de manera periódica por el servicio (aunque se puede tener acceso a algunas características de manera gratuita). Este programa está basado en canales (micrositios o miniportales) que se orientan a temas específicos: clima, deportes, cine, etc. De manera formal no se realiza transcodificación, pero al tener un formato propietario permite su correcta adaptación al dispositivo móvil. El problema radica en que se tienen que diseñar las páginas de acuerdo a este formato, y que se necesita un cliente específico para cada plataforma que permita visualizar el documento.

Tabla 1-1 Estado del arte.

Trabajo	Procesador	Plataformas	Transcodificación	Cache	Acaparamiento
Skweezer	x86, MIPS, SH3, ARM, m68x, PowerPC, DragonBall	Windows, Windows Mobile, Symbian, PalmOS, Linux y otros	HTML	No	No
AvantGo	MIPS, SH3, ARM, DragonBall	Windows Mobile, Symbian y PalmOS	HTML propietario (basado en canales).	Si	No
WebClipping	ARM, DragonBall	Windows Mobile, PalmOS,	HTML propietario (recortes)	Si	No
RepliGo	x86, MIPS, SH3, ARM, m68x, PowerPC, DragonBall	Windows, Windows Mobile, Simbian, y PalmOS	No (realiza conversión de documentos de office)	No	No
World Off-line	MIPS, SH3, ARM	Windows Mobile	No	Si	No
Isilo	x86, MIPS, SH3, ARM, DragonBall	Windows, Windows Mobile y PalmOS	No	Si	No
Hoarding Content in M-Learning Context	ARM	Windows Mobile	Si (personalización de documento)	Si	Si
Google Web Accelerator	x86	Windows	No	Si	No
Proxy Server for Handhelds	x86, MIPS, SH3, ARM, m68x, PowerPC, DragonBall	Windows Mobile, Symbian, PalmOS, Linux y otros	XML, XHTML, WML	No	No
Prototipo	x86, MIPS, SH3, ARM, m68x, PowerPC, DragonBall	Windows Mobile, Symbian, PalmOS, Linux y otros	HTML, WML, XHTML-MP, PDF, TXT, PS y XML.	Si	Si

AvantGo no realiza de manera formal el proceso de acaparamiento pero sí cuenta con un sistema de caché en el dispositivo móvil. El proceso consiste en realizar una replicación con diferentes niveles de profundidad (en base a los enlaces o hipervínculos; es decir, funciona como una simple araña Web). El usuario elige el límite de niveles, por lo que en general se trata de una replicación total del contenido del sitio. Está disponible para dispositivos PPC, Palm OS y algunos celulares con sistema operativo Symbian. El proceso de acaparamiento que se realiza en esta tesis está basado en patrones de acceso a sitios Web, lo cual es completamente diferente a esta herramienta.

Webclipping [pwc06] es un producto que se ejecuta en arquitecturas con sistema operativo Palm OS y en algunos dispositivos PPC. También está basado en un formato propietario llamado *PQA* (Palm Query Application) para tener una especie de transcodificación. No realiza acaparamiento pero cuenta con un sistema de caché en el dispositivo local. El proceso consiste en una replicación selectiva de los recursos indicados. Los PQA son pequeñas aplicaciones desarrolladas para obtener “recortes” (*clippings*) de páginas Web normales, los cuales permiten que el usuario defina sus preferencias de visualización del contenido Web. Estos recortes son en cierta medida una especie de personalización de contenidos. El usuario debe seleccionar explícitamente los recursos que requiere; mientras que en nuestra tesis la selección de los recursos se realiza de manera explícita.

RepliGo [rg06] es un sistema que se encarga de convertir ciertos documentos (generalmente documentos de office y alguna página Web) a un formato propietario que se adapta a las características del dispositivo móvil (se necesita de un cliente específico por plataforma, las plataformas actualmente soportadas son: Windows, Windows Mobile, Symbian y PalmOS), por lo que propiamente no se considera transformación. No es una herramienta en línea por lo que el usuario debe descargar el documento para después convertirlo. No cuenta con un sistema de caché y no realiza acaparamiento. El proceso consiste en la transferencia (replicación) y actualización de un documento en diversos clientes. Es una herramienta propietaria con costo por licencia.

World Off-line [wol06] es un navegador Web para dispositivos Windows Mobile que permite visualizar páginas Web en modo desconexión. No realiza transcodificación de documentos, pero cuenta con una caché para poder trabajar en modo desconexión con el dispositivo. No permite visualizar recursos Web de manera normal (on-line). No realiza acaparamiento, permite realizar una replicación basada en niveles de un sitio Web. Es una herramienta comercial con costo por licencia.

Isilo [isi06] es otro cliente Web para dispositivos móviles muy parecido a World Off-line. Se está portado a más plataformas (Windows y PalmOS). Además de que permite visualizar recursos Web en modo conexión como cualquier otro navegador.

Hoarding Content in M-Learnig Context [Tri06] es un trabajo que tiene una gran similitud con el tema de tesis; sin embargo, hay algunas diferencias. La primera consiste en el área que se cubre. En esta tesis, el enfoque es de dominio generalizado; es decir, para cualquier sitio (con sus respectivas limitantes); mientras que en este trabajo relacionado, se enfoca hacia una aplicación de aprendizaje móvil (*M-Learning*) denominada m-Eldit (versión móvil del programa de *e-Learning* ‘Eldit’¹).

Otro aspecto similar de este trabajo con esta tesis, consiste en que este trabajo realiza acaparamiento basado en el seguimiento (*tracking*) del usuario; es decir, la predicción de los recursos a acaparar se realiza de manera más sencilla, debido a que deduce los posibles recursos a acaparar en base a los cursos faltantes que tiene que completar dicho usuario. En este trabajo relacionado, el acaparamiento se realiza de manera especial para cada usuario; mientras que en esta tesis el acaparamiento se realiza de manera general para todos los usuarios. Para dejar más claro esto, supongamos que el usuario X ha

¹ Diccionario electrónico alemán-italiano, por sus siglas en alemán.

tomado los temas uno, dos y cinco de un curso de 6 lecciones, por lo que lo más probable es que a futuro, el usuario ocupe los temas tres, cuatro y seis. En un sitio Web, determinar qué recursos se deben acaparar a los usuarios, es más complejo, ya que las preferencias de los usuarios son extremadamente variantes.

Este trabajo también realiza un tipo especial de transcodificación, la cual se realiza de forma personalizada gracias a que se cuenta con el perfil del usuario. La personalización realizada en este trabajo, consiste simplemente en el cambio de parámetros a visualizar, como son los colores de la interfaz, el tamaño de las letras etc.; mientras que en esta tesis, la transcodificación se realiza de manera genérica para todos los sitios Web. Cuenta con un sistema de caché en el dispositivo móvil.

Google Web Acelerador [gwa06] es un programa en fase beta cuya finalidad es hacer la carga de páginas Web más rápida. Los métodos para lograrlo son los siguientes:

1. Envía páginas solicitadas a través de servidores de Google; es decir, funciona como servidor Proxy. Se parte del hecho de que al ser Google el motor de búsqueda en Internet más utilizado, existen grandes probabilidades de que el recurso se encuentre en la caché de los servidores de Google.
2. Almacena páginas en la computadora para hacer más accesible el diseño de páginas Web; es decir, cuenta con un sistema de caché en el dispositivo que agiliza los tiempos de acceso a los recursos. En este sentido no realiza acaparamiento y además no existe versión para ningún dispositivo móvil y hasta la fecha no hay planes de hacerlo.
3. Descarga páginas que sólo han sido modificadas desde la última vez que se accedió a ella. Precarga algunas páginas Web. La precarga se realiza en base a etiquetas especiales (`<link rel="prefetch" href="http://url/to/get/">`), encabezados (`x-moz: prefetch`) o por medio de un seguimiento estadístico de los movimientos de los usuarios.
4. Administra la conexión a Internet para evitar retardos o latencias en el acceso a la información de la Web.
5. Comprime datos antes de enviarlos. Se asemeja mucho a esta tesis que comprime recursos Web, la diferencia radica en que en esta tesis se comprime un sitio Web completo en formato ZIP, mientras que este programa lo hace por recurso.

Proxy Server for Handhelds [Dab05] es un trabajo de investigación que permite transformar recursos Web en distintos formatos: WML, XML y XHTML. Es un servidor Proxy implementado con .NET. La diferencia fundamental radica en que dicho Proxy se ejecuta en una computadora de escritorio, el prototipo desarrollado en esta tesis gran parte del proceso se realiza en el dispositivo móvil, delegando a una computadora de escritorio las actividades más pesadas (transcodificación e identificación del patrón). Por otra parte, este trabajo no cuenta con un sistema de caché y sobre todo no realiza acaparamiento, precarga o replicación.

1.8 Alcances de la tesis

Entre las restricciones que tiene este trabajo de tesis se encuentran las siguientes:

1. Debido a la gran heterogeneidad que presentan los dispositivos móviles, el prototipo realiza acaparamiento, sólo en plataformas basadas en Windows Mobile, en especial dispositivos PPC y Smartphone.
2. El servicio de transcodificación aplica para cualquier dispositivo móvil que tenga un navegador Web.
3. Debido a la gran diversidad de arquitecturas para los dispositivos móviles, los microprocesadores para los cuales se ejecuta el prototipo son: SH3, ARM y MIPS (no se garantiza que se ejecute sobre arquitecturas de microprocesadores diferentes).
4. El acaparamiento en dispositivos móviles está limitado a las características propias de cada dispositivo móvil; es decir, cuanto tamaño se ocupa para la caché, que recursos se desean acaparar, entre otras cosas.
5. No se realiza reintegración de páginas Web, sólo se hace la sincronización del contenido de las cachés. Esto debido a que el usuario no puede modificar las páginas Web.
6. El prototipo es compatible con módulos anteriores de la arquitectura Moviware para dar soporte a clientes convencionales.
7. Los formatos de transcodificación soportados son los siguientes: HTML reformateado, WML, XHTML-MP, PDF, PostScript, texto plano y XML. Las capacidades de transformación están limitadas a sus componentes básicos (código de marcado y no se garantiza que haya otros elementos, imágenes, frames, formularios, etc.).

1.9 Organización del documento

En el capítulo 2 Marco teórico, se profundiza en toda la teoría y conceptos sobre las tecnologías involucradas durante el desarrollo de esta tesis. En el capítulo 3 Análisis y diseño de la solución, se explica la metodología de solución que se llevó a cabo para resolver nuestro problema de investigación. En el capítulo 4 Pruebas, se detalla el plan de pruebas que se siguió para validar el prototipo desarrollado así como se da la interpretación de los resultados obtenidos. En el capítulo 5 Conclusiones, se presenta un sumario con las principales contribuciones y aportaciones obtenidas durante el desarrollo de esta tesis, así como se da pie a trabajos futuros sobre esta misma línea de investigación. Finalmente, en la sección de anexos, se muestra un análisis de los costos obtenidos al acceder a Internet a través de redes de telefonía celular, así como otros temas relacionados al diseño del prototipo.

CAPÍTULO 2

MARCO TEÓRICO

La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas y, por regla general pueden ser expresadas en un lenguaje comprensible para todos.

Albert Einstein

2.1 Acaparamiento

El acaparamiento (*hoarding*) se puede definir como el *proceso de replicación y procesamiento en desconexión de datos previamente seleccionados y copiados localmente en el cliente móvil* [Val02].

Como dato anecdótico, muchos animales como los roedores, hacen uso del concepto de acaparamiento, para ello acumulan alimento para sobrevivir en épocas de invierno. De hecho, algunos lingüistas creen que la palabra hámster (un tipo de roedor) deriva del inglés *hoarding* que significa acaparamiento [dmc05].

En informática, el concepto de acaparamiento es un proceso integral, el cual incluye tres etapas: desconexión, reintegración y propiamente el acaparamiento (ver Figura 2.1). A continuación se describe cada una de las partes que componen el proceso completo de acaparamiento.

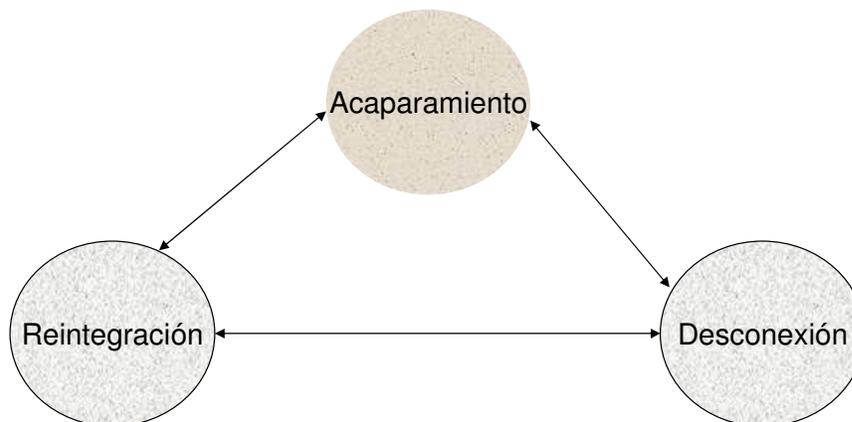


Figura 2.1 Esquema general del proceso de acaparamiento.

El acaparamiento surge del hecho de que las desconexiones, tanto planeadas como accidentales, no están consideradas en las arquitecturas tradicionales distribuidas, en particular con la arquitectura cliente/servidor.

Los recursos requeridos para las operaciones son cargados en la unidad móvil. La reubicación de los datos es simple, debido a que los datos son inaccesibles para otros sitios y para otros nodos.

Para gestionar las desconexiones de los equipos móviles, se deben acaparar los recursos periódicamente para poder trabajar en modo desconexión. La pregunta medular en esta fase consiste en ¿cómo anticipar las necesidades futuras de los usuarios móviles? Una alternativa consiste en que los usuarios definan explícitamente los datos que quieren; otra alternativa es de manera implícita, la cual consiste en examinar el historial de acceso a los datos por parte del usuario, para tratar de identificar los recursos que podría solicitar. Esta última alternativa es la que se utiliza en este trabajo de tesis.

Se deben considerar los siguientes aspectos a la hora de realizar acaparamiento:

1. ¿Cuál debe ser la unidad de acaparamiento: bloques de archivos, archivos, grupos de archivos o directorios? En nuestro trabajo la unidad de acaparamiento corresponde a un sitio Web, el cual a su vez está conformado por diversos recursos Web: páginas, imágenes, audio, documentos, etc.
2. ¿Cuándo se debe acaparar? Cuando se necesita la información (i.e., cuando la información es crítica). En nuestro caso se debe acaparar recursos cuando el cliente móvil esté en modo conexión para poder garantizar un asincronismo en el dispositivo y poder trabajar en modo desconexión. Los recursos acaparados deben ser enviados de manera transparente y en tiempos de ocio de conexión.
3. ¿Qué recursos se deben acaparar? En esta pregunta existe una gran diversidad de respuestas; por ejemplo, se deben acaparar los recursos de mayor prioridad, los recursos definidos por el usuario, o los recursos más accedidos. En este trabajo de tesis, se acaparan todos aquellos recursos que se encontraron al analizar la bitácora de un servidor Web y que representa un árbol patrón o modelo de navegación que probablemente utilizarán los clientes. El usuario puede determinar que tipos de recursos (extensiones) son permitidos dentro del patrón.
4. ¿Cómo acaparar? Esta pregunta es sumamente difícil de contestar, debido a que existen diversas metodologías de solución, las cuales resuelven problemas específicos. De forma general el acaparamiento se realiza a través de los usuarios, ya que éstos proporcionan información ya sea de forma explícita o implícita. Esta pregunta se resuelve con el mecanismo denominado GAP implementado en esta tesis.

2.1.1 Replicación

El proceso de replicación implica el almacenamiento o respaldo de la información [rh05]. Con respecto a la teoría de base de datos, la replicación es el proceso de crear y manejar versiones duplicadas de una base de datos. No solo incluye la copia de información, sino involucra que los cambios realizados en una réplica se reflejen en la base de datos al reintegrar la réplica (proceso de sincronización).

Entre las características que posee la replicación se encuentran las siguientes:

1. Aumentar la confiabilidad al tener respaldos independientes de cada archivo.

2. Permitir el acceso al archivo aunque falle un servidor de archivos. El lema aquí es el “espectáculo debe continuar”.
3. Repartir la carga de trabajo entre varios servidores.

En lo referente a cómputo móvil y/o ubicuo, la replicación de la información debe ser dinámica; esto es, los datos y los servicios deben seguir a los usuarios.

2.1.2 Precarga

Es un proceso concurrente; es decir, en línea, que transfiere de la caché los archivos requeridos durante periodos de bajo tráfico en la red [CF03].

De manera general, la precarga es una técnica que reduce el costo de la falta de caché durante desconexión. Es decir, la carga se hace antes de la desconexión y está lista para su uso en cualquier momento.

La precarga se basa en el hecho de que la mayoría de los archivos (información) son leídos de manera secuencial, a este hecho se le llama lectura anticipada. La precarga hace énfasis en la reutilización de la información, por lo que trata de lograr un ahorro de lecturas. Para saber que recursos se deben precargar, generalmente se utilizan técnicas que tratan de predecir o “*adivinar*” los posibles recursos que el usuario podría utilizar.

La precarga puede traer un exceso en el tráfico de la red, y si no se hace de manera adecuada, puede traer como resultado la transmisión de páginas que nunca pueden ser requeridas por los clientes. La precarga puede ser mejorada basada en la sintaxis de los prefijos URL o modelos estadísticos, capturando patrones de acceso temporales de páginas Web.

No se precargan enlaces con efectos negativos (páginas dinámicas, dar de alta, borrar un recurso, etc.). Una precarga errónea puede incrementar la carga de trabajo de los servidores Web o que la información pueda ser inútil para un cliente móvil.

2.1.3 Caching

El caching es la replicación y almacenamiento de datos en el cliente o cerca del Proxy (intermediario) siguiendo la petición de datos hecha al servidor remoto por el cliente. Las copias son creadas y gestionadas por el cliente o Proxy (basado en los patrones de acceso de los clientes) y no están controladas de ninguna forma por el servidor que provee los datos.

Las cachés de sitios Web pueden utilizar precarga (petición de documentos antes de que sean pedidos por el cliente) para disminuir la latencia de respuesta de futuras peticiones del cliente. Identificar las páginas que serán pedidas por el cliente es la clave de esta metodología. Estudios revelan que la eficiencia del caching oscila entre el 40% y 50% [Wan99].

El caching de páginas Web reduce el consumo de ancho de banda, disminuyendo el tráfico de la red evitando congestiones. Se reduce la latencia de acceso a los datos por dos razones:

1. Los documentos frecuentemente accedidos están cargados cerca del Proxy caché y no en el servidor, por lo que el retardo en la transmisión es minimizado.
2. El caching de páginas Web reduce la carga de trabajo de los servidores Web. Si el servidor remoto no está disponible, los clientes pueden obtener una copia del servidor a través del Proxy.

La caché es un sistema cuya funcionalidad es copiar páginas Web recientemente visitadas por el usuario, las cuales se almacenan en disco duro y son accedidas de manera local, ya que se almacenan en el dispositivo, sin necesidad de volver a acudir a la red para descargar la página.

2.1.4 Diferencias entre acaparamiento, replicación, precarga y caching

En la Tabla 2-1, se resume los conceptos de acaparamiento, replicación, caching y precarga que en ocasiones se les suele confundir con mucha frecuencia.

El atributo desconexión indica si el concepto se ideó como mecanismo para poder trabajar sin necesidad de estar conectado a una entidad centralizadora. En general, cuando aparece el valor Si/No indica que el mecanismo en ese atributo no se diseñó específicamente para esa cuestión, pero que actualmente puede ser empleado para ello. El atributo en línea indica si el proceso se realiza en un tiempo considerablemente corto o no. El atributo tolerante a fallas indica si el mecanismo se creó como medio para evitar fallas en sistemas distribuidos. El atributo predicción, indica si el mecanismo se diseñó para tratar de anticipar los posibles datos que un usuario podría utilizar en el futuro. El atributo optimización de acceso, indica si el mecanismo se creó para mejorar el acceso a los datos a través de un sistema distribuido. El atributo esquema de predicción, indica la forma o método en que se realiza la predicción de la información; en este punto, los algoritmos más comunes son basados en costo, por heurísticas, por posiciones consecutivas o por LRU (Last Recent Used, o si se trata de Web caching: LRV Last Recent Visited). El atributo reducir latencia, señala si el mecanismo fue ideado para agilizar los tiempos de respuesta al acceder a la información. El atributo selección de datos indica el esquema de cómo se obtiene los datos, puede ser explícito cuando el usuario hace la selección manualmente, implícito cuando la selección se hace sin intervención humana o una mezcla de ambos. Por último, el atributo reintegración indica si los datos al ser modificados son posteriormente conciliados con los demás datos en busca de tener los datos siempre actualizados evitando las inconsistencias.

2.2 Transcodificación

Transcodificación (*transcoding*) se define como el proceso de convertir un formato o código a otro, con la finalidad de que este nuevo formato o código se adapte a la plataforma indicada para su correcta visualización [Cha03]. En palabras más sencillas,

la transcodificación es la translación o transformación de contenidos a formatos adaptables para varias plataformas.

Tabla 2-1 Comparativa entre los conceptos de replicación, acaparamiento, precarga y caching.

Concepto	Desconexión	En línea	Tolerante a fallas	Predicción	Optimización acceso	Esquema de predicción	Reducir latencia	Selección de datos	Reintegración
Replicación	Si/No	Si/No	Si	Si/No	Si	Heurística /Costos	Si/No	Híbrido	Si
Acaparamiento	Si	No	No	Si	No	Minería de datos (Heurística)	No	Implícito	Si/No
Precarga	Si/No	Si	No	Si	Si	Posiciones consecutivas	No	Implícito	No
Caching	Si/No	Si	Si/No	No	Si	LRU	Si	Explícito	Si/No

Para dejar más claro el concepto de transcodificación, se pondrá el siguiente ejemplo: el sistema de transmisión de televisión norteamericano NTSC², es usado en muchos países incluido México, pero desgraciadamente es incompatible con el sistema europeo de transmisión: el PAL³. Ambos sistemas manejan diferentes formas de visualizar las imágenes en pantalla, mientras en NTSC se transmiten 525 líneas en una frecuencia de 60 Hertz (hz), el sistema PAL maneja 625 líneas con 50 hz como frecuencia. Debido a estas diferencias, no se pueden visualizar secuencias de video grabadas en un formato a otro, aunque se trate del mismo video. Para poder realizar una conversión de un formato a otro se necesita de un transcodificador. Éste recibe una señal de video en un formato dado y la transforma a otro formato conservando la misma secuencia de imágenes que componen el video.

Cabe hacer mención que el término transcodificación sólo se aplica cuando existe una transformación de códigos, cuyo objetivo fundamental tiene que ver con la adaptación de su contenido para su correcta visualización; por ejemplo, la transformación de un código de un lenguaje de programación de alto nivel (*Pascal*) a otro de alto nivel (*C*), no se considera transcodificación, por que, si bien se realiza una transformación de código de un lenguaje a otro, dicha transformación no tiene como objetivo final una adaptación del mismo para su correcto despliegue.

² National Television Standards Comittee, comité estándar de televisión nacional

³ Phase Alternating Line, línea alternando de fase

El mecanismo de transcodificación de contenidos Web, lleva a cabo una reorganización y agrupación de los elementos contenidos en la página Web de acuerdo a la delimitación del lenguaje de entrada (HTML). Como resultado final de la transcodificación, se obtienen páginas Web cuyo formato de presentación o visualización es óptimo para un dispositivo de despliegue limitado, tratando de respetar fielmente la semántica (estructura) original del documento o página.

Para [NC06] la transcodificación de documentos debe adecuarse fielmente a las restricciones con las que cuentan los dispositivos móviles. La transcodificación debe ser lo más fielmente posible. Por fidelidad se entiende como la calidad de la transformación. Por ejemplo una imagen, video o sonido con una alta calidad consume más espacio y poder de procesamiento de cómputo que una de baja calidad. En la Figura 2.2, se muestra las diversas modalidades en las que se puede aplicar la transcodificación de documentos. En este trabajo de tesis, se hace uso de la transcodificación de documentos de Internet.

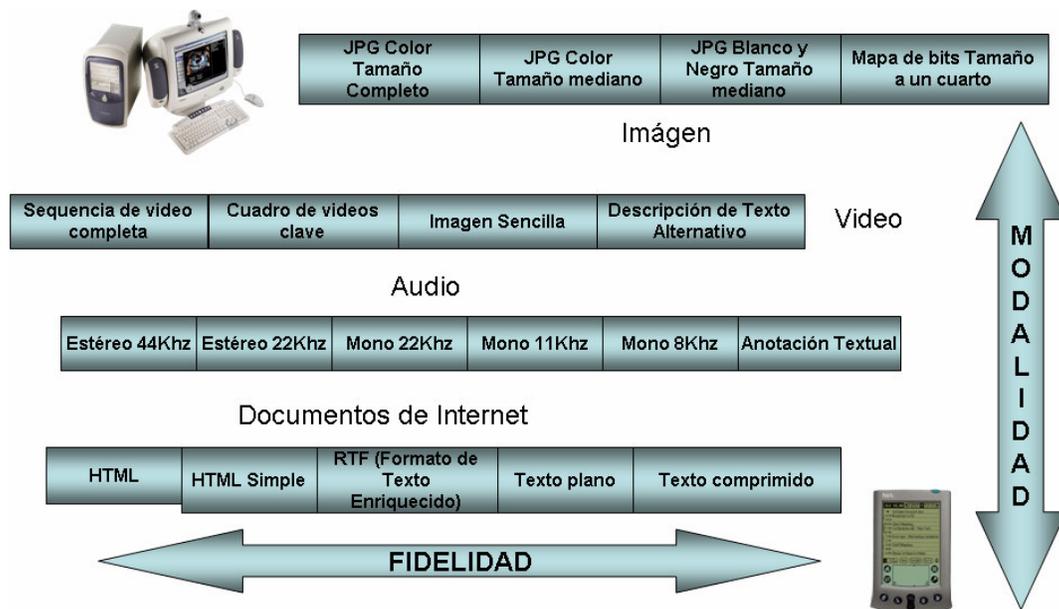


Figura 2.2 Tipos de transcodificación de documentos.

2.3 Dispositivos móviles

Los dispositivos móviles son aparatos electrónicos que sirven para la comunicación, procesamiento e intercambio de datos y pueden ser llevados por sus usuarios para enviar, recibir o compartir datos con otros dispositivos. Las características de estos equipos son las siguientes según [Bia05]:

1. Se pueden trasladar fácilmente por su forma y tamaño adecuado (de aquí viene su movilidad y portabilidad).
2. Poseen interfaces de red inalámbricas (son capaces de conectarse a redes de voz y datos).

3. Tienen autonomía eléctrica, es decir, pueden ser operados con independencia de la red eléctrica por un lapso de tiempo determinado.
4. Actualmente tienen restricciones de hardware y software suficientes para requerir un tratamiento especial (poseen CPUs y memorias limitadas si se comparan con arquitecturas tradicionales pero las cuales son adecuadas para almacenar y ejecutar muchos programas).
5. Poseen formas de almacenamiento persistente (Flash ROM).

2.3.1 Clasificación

Ante la gran demanda y auge de dispositivos móviles ha surgido la necesidad de clasificarlos para su mejor comprensión. Actualmente esto es sumamente difícil por que existen en el mercado dispositivos que pudieran caer en más de dos clasificaciones.

El problema de la clasificación se debe a varios factores. Uno de estos factores consiste en que los dispositivos móviles son muy dinámicos y actualmente realizan más funciones de las que originalmente se tenían contempladas; por ejemplo, los teléfonos actuales permiten realizar actividades de gestión de información personal tal como lo haría un PDA, ya no se limitan simplemente a realizar/recibir llamadas. Una clasificación de dispositivos móviles se muestra en la Figura 2.3.

2.3.1.1 PDAs (Pocket PC)

Se define una Pocket PC (PPC) según Microsoft como *“un dispositivo de mano que permite grabar, enviar y recibir e-mails, contactos, citas, mostrar archivos multimedia, juegos, intercambiar mensajes de texto con MSN Messenger, navegar por la Web y más”* [ppc06].

Los dispositivos PPC están clasificados dentro del segmento de PDAs (Personal Digital Assistant, Asistente Personal Digital) que no es otra cosa que una computadora de bolsillo u organizador electrónico.

El primer antecedente de un dispositivo PDA fue en 1993 el Newton de la compañía Apple. En 1996 surge la Palm Pilot (la cual revolucionó el mercado informático), por las mismas fechas aparecen los primeros dispositivos Windows CE de Microsoft. En el 2000 aparece la versión Windows CE 3.0 y con ella la plataforma Pocket PC.

Si bien Windows CE 3.0 es el sistema operativo, la palabra Pocket PC (PPC) designa dispositivos con ciertas características impuestas por Microsoft a los fabricantes para sus dispositivos; es decir Microsoft pone una serie de restricciones al hardware de dispositivos móviles para poder ejecutar su sistema operativo y dar una certificación al dispositivo móvil.

En el 2002 aparece la plataforma Pocket PC 2002. En el 2003, Microsoft evoluciona su concepto de cómputo móvil y presenta la plataforma Windows Mobile que agrupa tanto a dispositivos PPC como teléfonos inteligentes. El cambio de nombre es para hacer

distinción entre los sistemas empotrados y móviles. Técnicamente ejecutan el mismo sistema operativo pero con diferente shell (interfaz de usuario).



Figura 2.3 Clasificación de dispositivos móviles.

En el 2004 aparece la versión Windows Mobile 2003 SE (Second Edition). La versión más reciente es Windows Mobile 5, la cual apareció a finales del 2005. La próxima versión de la plataforma se tiene contemplada que aparezca en el 2007 y lleva por nombre clave *foton*, se espera que los Pocket PC y Smartphones tengan el mismo shell (actualmente cada clase dispositivo tiene su propia shell lo que dificulta la transportabilidad de las aplicaciones).

Además de las características que posee un PDA por ser dispositivo móvil, también cuenta con otras características como: pantalla sensible al tacto (se utiliza un lápiz como dispositivo apuntador), tiene funcionalidad de PIM (Personal Information Manager, gestor de información personal) lo cual hace que funcione como organizador o agenda electrónica, cuenta con botones de acceso rápido a las aplicaciones más importantes, generalmente no cuenta con teclados alfanumérico pero algunos modelos los tienen, siempre cuentan con una batería de respaldo.

2.3.1.2 Teléfonos inteligentes (Smartphones)

Los teléfonos inteligentes son un híbrido entre los dispositivos tradicionales PDAs y los teléfonos celulares [sp06]. Entre esta mezcla de teléfono y PDA surgen dos vertientes, la primera de ellas es la mezcla entre un teléfono celular con funciones PDA, a los que se ha denominado Smartphone. Por otra parte, los PDAs que tiene como agregado una función de radio que les permite funcionar como teléfono, a dichos dispositivos se les llama PDA + teléfono. Ejemplo de un PDA+Teléfono es el dispositivo Treo (utiliza PalmOS) o el Pocket PC Phone Edition (es una PPC ordinaria que además cuenta con un interfaz de radio para comunicarse a redes de telefonía celular).

Una clasificación ampliamente aceptada [mwm06] dice que los teléfonos no tienen pantalla sensible al tacto y se pueden operar con una sola mano. Esto aplica en dispositivos Windows Mobile for Smartphone. Otra característica que tienen estos

Smartphone es que poseen el mismo sistema operativo que las PPC, pero su interfaz gráfica del shell es distinta.

La primera plataforma de teléfonos inteligentes con Windows Mobile apareció en el 2002 y a tenido las mismas versiones que su contraparte PPC (versiones 2003, 2003 SE y 5).

2.3.2 Herramientas de programación

A continuación se mencionan algunos entornos de desarrollo para la creación de aplicaciones en dispositivos móviles caso concreto en plataforma PPC (pueden usarse para otra clase de dispositivos). De manera generalizada se piensa que el desarrollo de aplicaciones en dispositivos móviles es similar al desarrollo de aplicaciones en sistemas tradicionales de escritorio; es decir, se tiene la creencia que el desarrollo de aplicaciones en dispositivos móviles es idéntico a las aplicaciones en PCs pero en “chiquito”, pero esto no es así. Para el desarrollo de aplicaciones móviles se deben tener en cuenta muchos factores, de los cuales los más sobresalientes según [Lut04] y [Mue06] son:

1. Entender el problema a resolver (“*ensuciarse las manos*”); es decir, conocer como operará la aplicación en un ambiente real.
2. Estudiar las capacidades y limitaciones de los dispositivos móviles con el fin de saber de manera anticipada que se puede hacer en el dispositivo y que cosas son imposibles de implementar. Se deben tomar en cuenta las limitantes de espacio de almacenamiento primario y secundario, la duración de las baterías, el poder de procesamiento, entre otras.
3. Se debe desarrollar una interfaz adecuada que minimice las acciones por parte del usuario y que se adapte al tamaño de las pantallas de despliegue.
4. Se debe mantener un diseño sencillo del sistema para que éste sea fácilmente modificable.
5. Se deben tomar en cuenta todas las implicaciones que conlleva la movilidad de los dispositivos como es el caso de la seguridad.
6. Por último, se deben probar las aplicaciones realizadas, nuevamente probarlas y volverlas a probar hasta que estén libres de errores.

Las herramientas de programación se dividen tomando en cuenta el tipo de código que generan: código nativo y código gestionado. El código nativo es aquel que se ejecuta en una plataforma específica, tiene la ventaja que su ejecución está optimizada para esa plataforma pero no puede ser ejecutada en otras plataformas de manera sencilla (por ejemplo una aplicación realizada en Windows no se ejecuta en Linux), en algunos casos se tiene una transportabilidad de código fuente, ya que este se lleva a otro compilador en la plataforma adecuada y podría ejecutarse. El código gestionado es aquel que se diseña para lo que se conoce como una máquina virtual, la cual puede ser implementada en diversas plataformas por lo que se obtiene una compatibilidad binaria; es decir, la

misma aplicación se ejecuta en diversas plataformas. Tiene la desventaja de que el código gestionado no está optimizado para una plataforma específica.

2.3.2.1 Conclusiones (herramienta seleccionada)

Se realizó una comparativa entre las herramientas de programación anteriormente descritas y se obtuvieron las siguientes conclusiones:

Se determinó que .NET CF con lenguaje en C# es en estos momentos la mejor herramienta de programación para plataforma PPC. C# es un lenguaje moderno y orientado a objetos, con una sintaxis muy similar a la de C++ y Java. Combina la facilidad de uso de Visual Basic con el poder y la flexibilidad de C++. La misma aplicación que se ejecuta bajo Windows podría funcionar en un dispositivo móvil de tipo PDA, por lo que con C# no nos atamos a ninguna plataforma en particular. C# gestiona automáticamente la memoria, y de este modo evita los problemas de programación tan típicos en lenguajes como C o C++.

Nuestra segunda opción corresponde eMbedded Visual C++ (eVC++), la cual se descartó por la complejidad del lenguaje en sí, ya que para generar el código para distintos microprocesadores es necesario tomar algunas consideraciones. Se recomienda su uso cuando se desee obtener mayor velocidad de ejecución de las aplicaciones y/o un uso intensivo de hardware.

La tercera opción corresponde a la tecnología Java 2 Micro Edition (J2ME), la cual se descartó para nuestro proyecto debido a que no existen máquinas virtuales para todas las plataformas y dispositivos PPC, además de que estas máquinas virtuales no son gratuitas. Se recomienda su uso cuando se tiene software legado que se quiere ejecutar en plataforma PPC.

Como cuarta opción se tiene el desarrollo de aplicaciones en plataforma Linux, se descartó dicha opción por el hecho de que no es posible instalar Linux en todo tipo de dispositivo PPC, además de que instalar el sistema operativo y las herramientas de programación es sumamente complejo. Se recomienda su uso cuando se quieran utilizar lenguajes muy específicos en el desarrollo. Cuando se necesite realizar aplicaciones que aprovechen las características multitarea y multiusuario que proporciona Linux.

La quinta opción corresponde a eMbedded Visual Basic (eVB), la cual se descartó por ser un lenguaje interpretado y con pocas características. Se recomienda su uso cuando se desee realizar desarrollos rápidos y sencillos.

Por último se descartó utilizar ASP .NET (y sus variantes) por ser una tecnología que no permite realizar muchas cosas en el dispositivo móvil. Se recomienda su uso cuando el procesamiento se realiza en mayor medida en el lado del servidor, y en el cliente sólo se implementará una interfaz a través de la Web que se ajustará a cada tipo de dispositivo.

CAPÍTULO 3

ANÁLISIS Y DISEÑO DE

LA SOLUCIÓN

“La formulación de un problema, es más importante que su solución”

Albert Einstein

3.1 Introducción

Para solucionar el problema de la visualización de contenidos Web en dispositivos móviles, se han propuesto varias alternativas: diseñar nuevos protocolos, modificar protocolos o implementar servicios intermediarios que resuelvan el problema.

En lo referente a diseñar nuevos protocolos se puede citar al protocolo WAP y al lenguaje de marcado WML, los cuales funcionan de manera análoga a HTTP-HTML en la Web tradicional. El problema radica en que WAP sólo trabaja con equipos móviles y traería consigo la misma fragmentación de la Web (páginas especiales para toda clase de dispositivos). Además, WAP se diseñó originalmente para dispositivos con capacidades de cómputo limitada (pantallas monocromáticas, bajo ancho de banda, etc.) lo que en estos momentos se está solucionando por medio de conexiones de banda ancha inalámbrica (WCDMA, UMTS, 802.11g, WiMax, etc.) y equipos cada vez más potentes. La mejor solución sería crear un nuevo protocolo, pero el problema es que éste debe ser totalmente compatible con los existentes.

En lo referente a la utilización de servicios intermediarios, es la solución más extendida para solucionar el problema de la visualización de recursos Web y muchos otros problemas que presenta la Web, tal es el caso de los corta fuegos que solucionan alguno de los problemas de seguridad de la Web como el control de acceso o los proxys caché, que tratan de reducir la latencia de acceso a la información. El esquema de intermediarios es ampliamente usado ya que no necesita modificar ningún aspecto funcional de las aplicaciones cliente y servidor; de hecho, los procesos cliente y servidor no notan la existencia de dichos servicios intermediarios. Estos servicios se encargan del trabajo pesado y son transparentes hacia los usuarios. Este trabajo de tesis funciona bajo el esquema de servicios intermediarios.

3.2 Arquitectura propuesta

El esquema de solución que se propone, consiste en la adaptación del esquema cliente/servidor orientado a clientes móviles. En medio de los clientes y servidores se

encuentra la capa de intermediarios, tanto del lado del cliente como del lado del servidor.

En la arquitectura propuesta en esta tesis, el Proxy ubicado del lado del cliente recibe el nombre de GAP (Gestor de Acaparamiento para Pocket PCs) y el del lado servidor GAT (Gestor de Acaparamiento y Transcodificación) tal y como se muestra en la Figura 3.1. Nótese que el GAT hace uso de otro Proxy caché llamado Squid, pero podría ser cualquier otro Proxy. EL GAT está conformado por dos componentes principales: el Mecanismo Acaparador (MA) y el Mecanismo Transcodificador (MT).

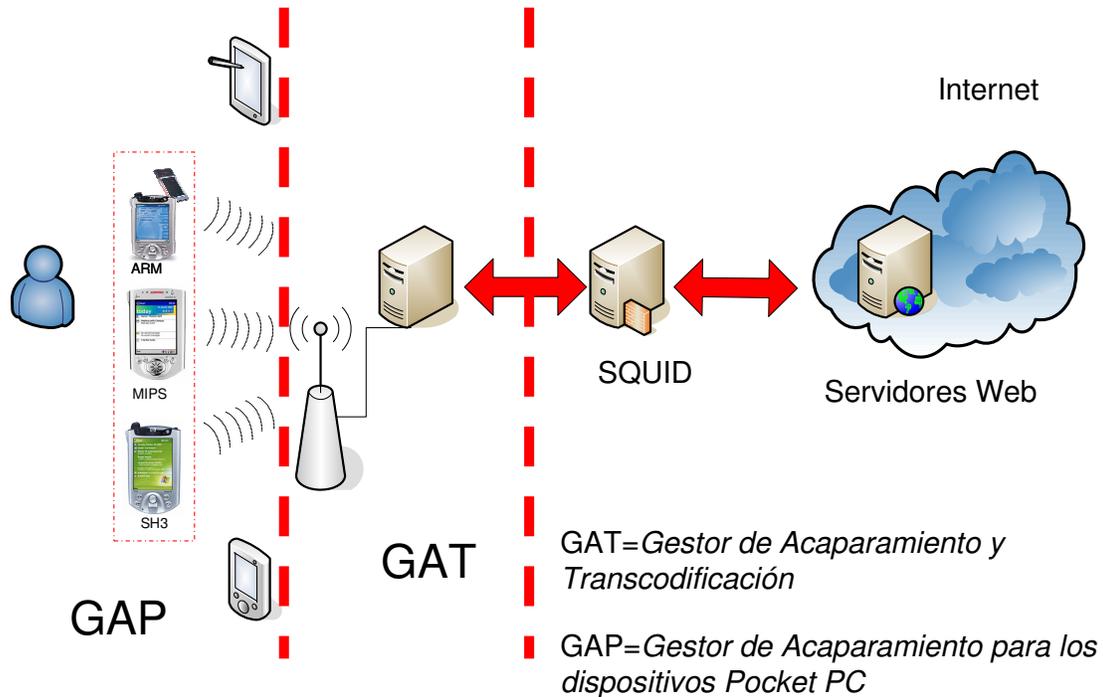


Figura 3.1 Modelo general de solución propuesta.

La arquitectura del sistema en forma de bloques se muestra en la Figura 3.2; mientras que en la Figura 3.3 y Figura 3.4, se puede observar la arquitectura del sistema de manera más detallada tanto del lado cliente como del lado del servidor.

El funcionamiento general del sistema es el siguiente: el usuario introduce una URL desde su navegador (que previamente ha sido configurado para redireccionar su salida hacia el GAP). El GAP recibe la petición y determina el estado de las siguientes variables: recursos acaparados (A), existencia del patrón de acaparamiento (P) y estado de la conexión (C). En base a dichas variables (0 indica la ausencia y 1 la presencia de ese factor), se realiza una acción en particular, tal y como se muestra en la Tabla 3-1.

Tabla 3-1 Tabla de verdad-contradicción que explica el funcionamiento básico del sistema.

A	P	C	Resultado
0	0	0	Muestra el mensaje de recurso no acaparado.
0	0	1	Trata de obtener el recurso en línea. Si puede lo muestra, en caso contrario muestra mensaje informativo de error de conexión
0	1	0	Muestra el mensaje de recurso no acaparado.
0	1	1	Trata de obtener recurso en línea (si puede lo muestra, sino muestra

			mensaje informativo de error de conexión) y de obtener patrón de acaparamiento del GAT MA.
1	0	0	Muestra el recurso desde la caché local.
1	0	1	Si el sistema está configurado para dar prioridad a la caché local, se muestra directamente el recurso desde la caché local (esto nos permite trabajar en modo desconexión). En caso contrario, trata de obtener el recurso en línea, en caso de éxito muestra el recurso desde la Web, en caso de falla muestra el recurso que está en la caché local.
1	1	0	Muestra el recurso desde la caché local.
1	1	1	Muestra el recurso desde la caché local o desde la Web según sea el caso. Verifica que el patrón no se haya modificado, en caso de que el patrón se haya modificado trata de obtener el sitio Web acaparado.

El GAT MA comprueba si tiene ya los recursos Web del patrón acaparado en su caché. En caso de no tenerlos, los descarga de la Web, los transcodifica y los comprime para posteriormente enviárselos al GAP.

Una vez que el MA ha enviado el sitio Web acaparado, el dispositivo móvil debe descomprimir el sitio Web y actualizar su lista de patrones. Este proceso ocurre de manera transparente sin que el usuario lo perciba.

El MT se encarga de recolectar los documentos y en caso de ser recursos HTML, los transforma si es que los parámetros de configuración así lo indican. La transcodificación se realiza en línea, por lo que el proceso se ve va volviendo cada vez más lento si el documento es demasiado grande.

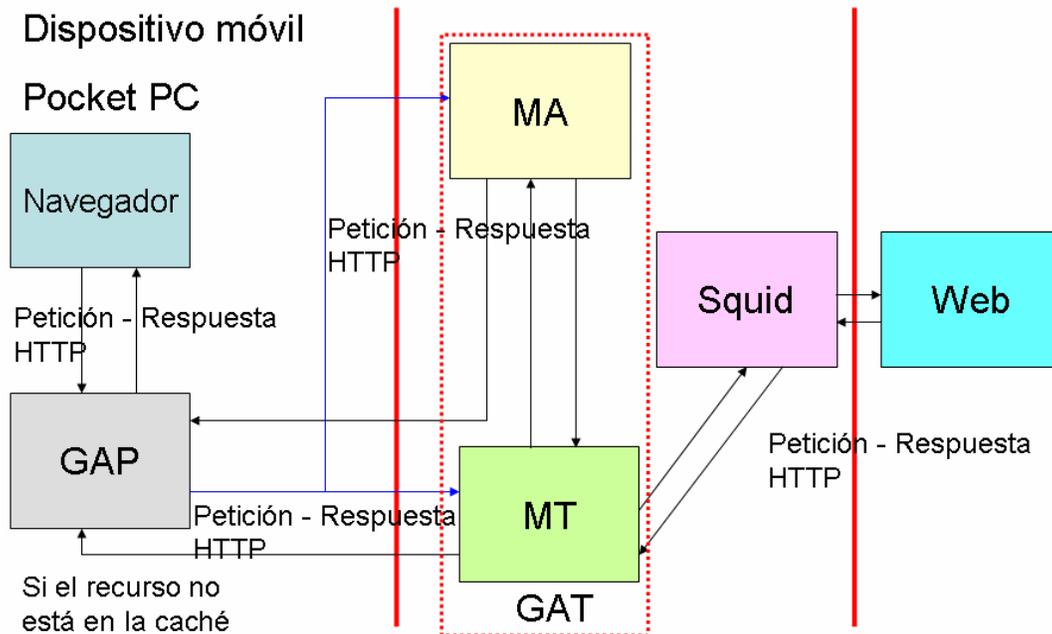


Figura 3.2 Arquitectura del sistema por bloques.

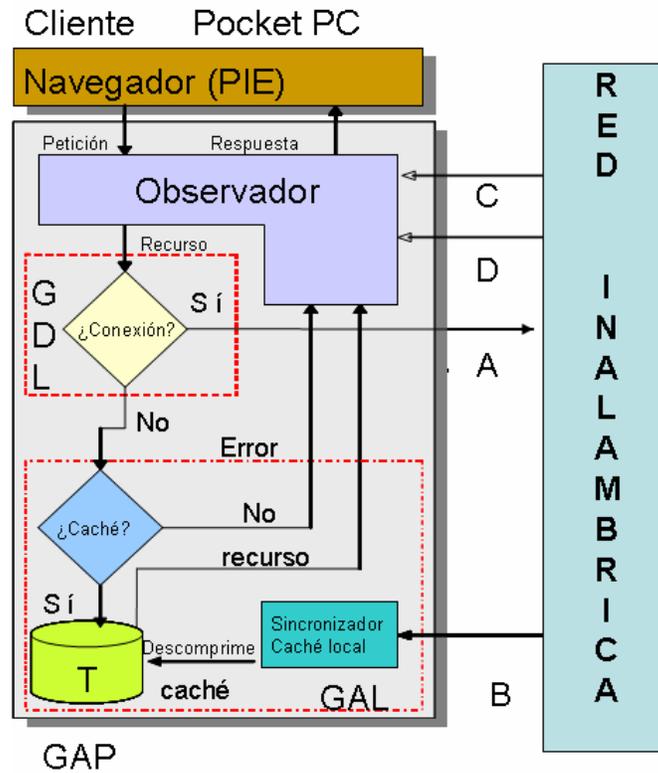


Figura 3.3 Arquitectura del prototipo desarrollado en el lado cliente (dispositivo móvil).

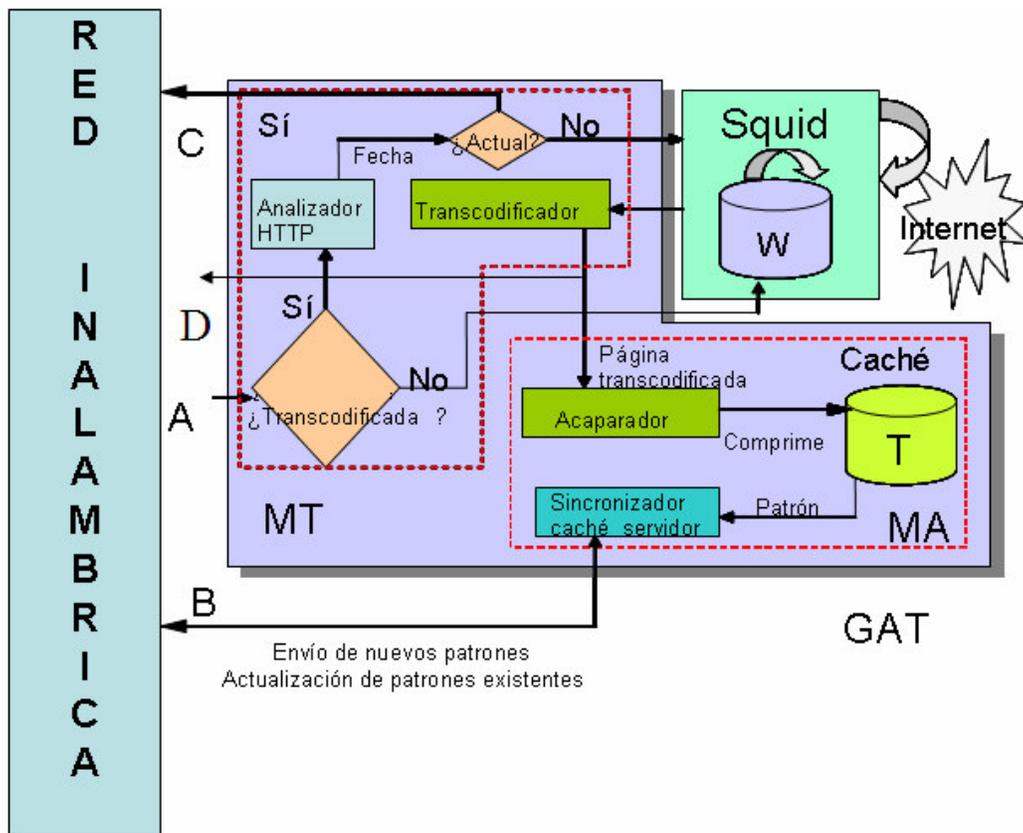


Figura 3.4 Arquitectura del prototipo desarrollado en el lado del servidor.

3.3 GAP

El GAP se encarga de gestionar los eventos de desconexión que se pudieran presentar en el dispositivo móvil, por lo que acapara recursos en la caché del dispositivo local. El GAP está conformado básicamente de tres módulos principales: Observador, Gestor de Acaparamiento Local (GAL) y Gestor de Desconexión Local (GDL).

El Observador se encarga de procesar cada petición y devolver el resultado al navegador. El GAL se encarga de la manipulación y control de la caché en el dispositivo. Es el usuario quien decide que recursos se desean acaparar, así como limitar el espacio de almacenamiento. El GDL se encarga de determinar el estado de la conexión del dispositivo.

En la Figura 3.5, se muestran las actividades que puede realizar un usuario sobre el sistema. Entre dichas actividades se encuentran visualizar recursos Web en línea, acaparados y transcodificados, la configuración del sistema, la visualización del estado de las peticiones y de los errores.

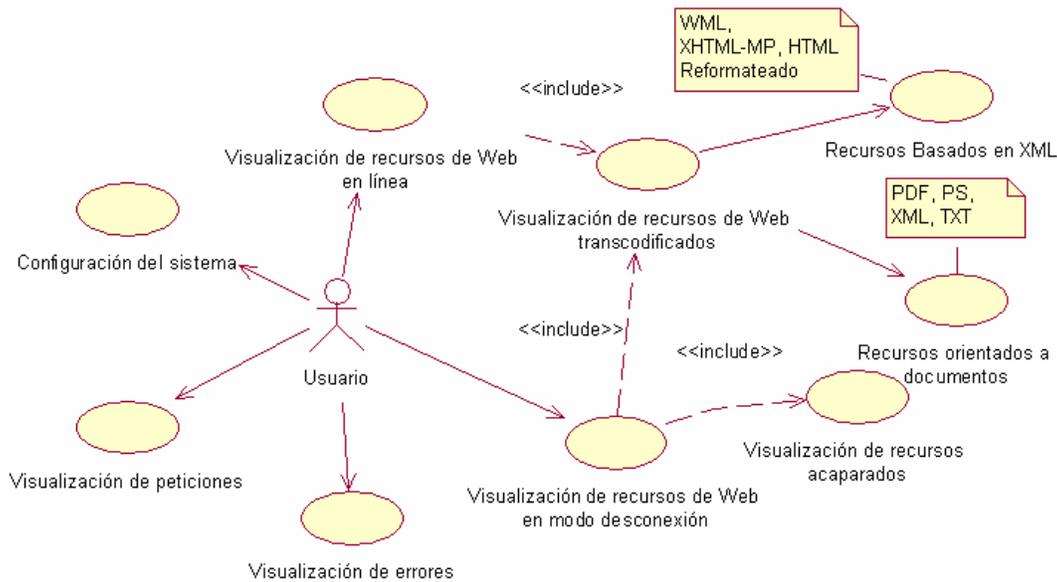


Figura 3.5 Diagrama de casos de uso del GAP.

En la Figura 3.6 y Figura 3.7, se muestra el diagrama de clases correspondientes al módulo GAP. El detalle de los métodos implementados en el GAP se muestra en el Anexo B.

De la Figura 3.8 a la Figura 3.15, se muestran los distintos diagramas de secuencias para cada uno de los diferentes escenarios que se pueden presentar durante la ejecución del sistema.

En la Figura 3.8, se muestra la forma en que se obtiene un sitio Web acaparado. Cuando el usuario realiza una petición se verifica si se tiene el recurso acaparado y en caso de no tenerlo y de existir un patrón de acaparamiento, se procede a descargar dicho sitio Web. De manera paralela se procesa la petición del usuario obteniendo el recurso Web en línea o mostrando un mensaje de error cuando no se pudo obtener el recurso.

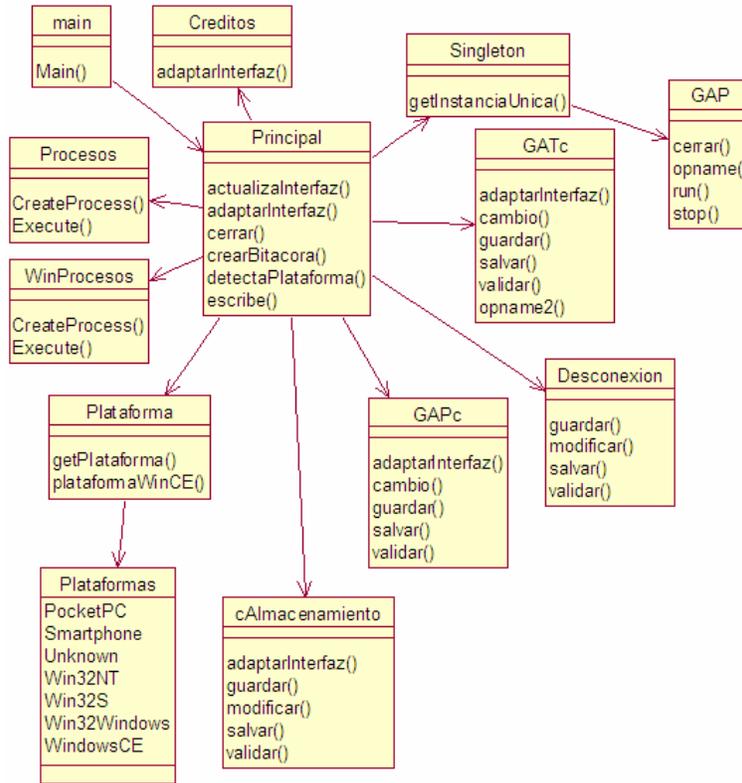


Figura 3.6 Diagrama de clases del GAP parte 1.

En la Figura 3.9, se muestra el escenario cuando el recurso se encuentra en la caché. Se puede observar que el recurso se muestra sin importar el estado de la conexión del dispositivo, y que el acceso será más rápido ya que sólo se interactúa con el Proxy local y no con los otros proxys.

En la Figura 3.10, se muestra lo que ocurre cuando no se tiene un recurso en la caché local del dispositivo y se ha determinado de que no existe conexión a Internet. Nótese que el GAL no hace nada por obtener el recurso en línea y muestra un mensaje informativo.

En la Figura 3.11, se muestra la secuencia que se presenta cuando el recurso no está en la caché local y se presenta un error al tratar de obtener un recurso Web. El GAL muestra al usuario un mensaje informativo de error HTTP 404 Not Found (recurso no encontrado) que encapsula a todos los tipos de errores que se pueden presentar al obtener un recurso desde la Web.

En la Figura 3.12, se muestra la secuencia que se obtiene cuando no se puede transcodificar un recurso. Nótese que el proceso de transcodificación se realiza debido a que no se tiene el recurso acaparado en la caché local.

En la Figura 3.13, se muestra el proceso que se sigue cuando no existe un patrón de acaparamiento de un sitio. Para el usuario este proceso es transparente ya que se realiza en segundo plano.

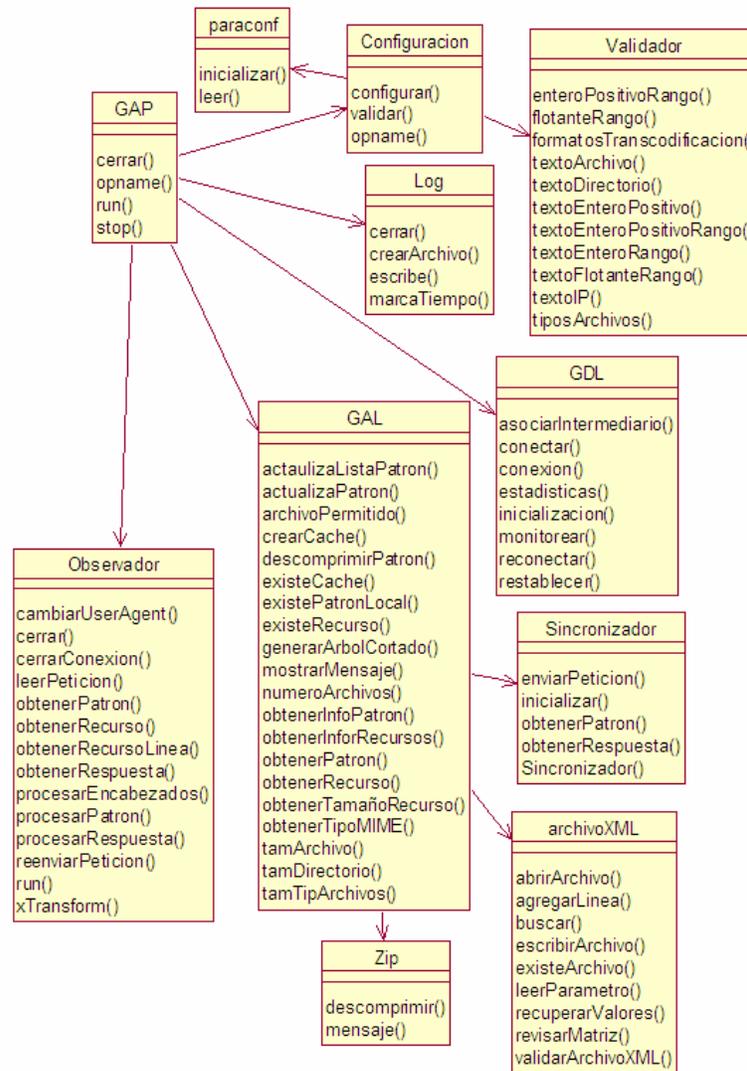


Figura 3.7 Diagrama de clases del GAP parte 2.

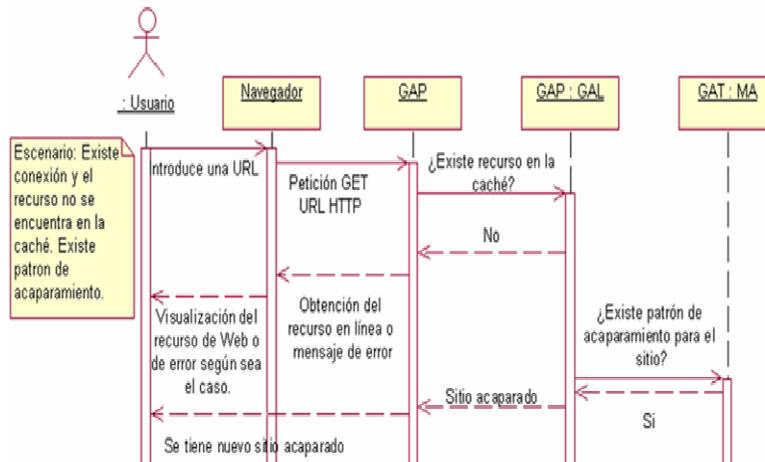


Figura 3.8 Diagrama de secuencia cuando el recurso no está en la caché y existe patrón de acaparamiento.

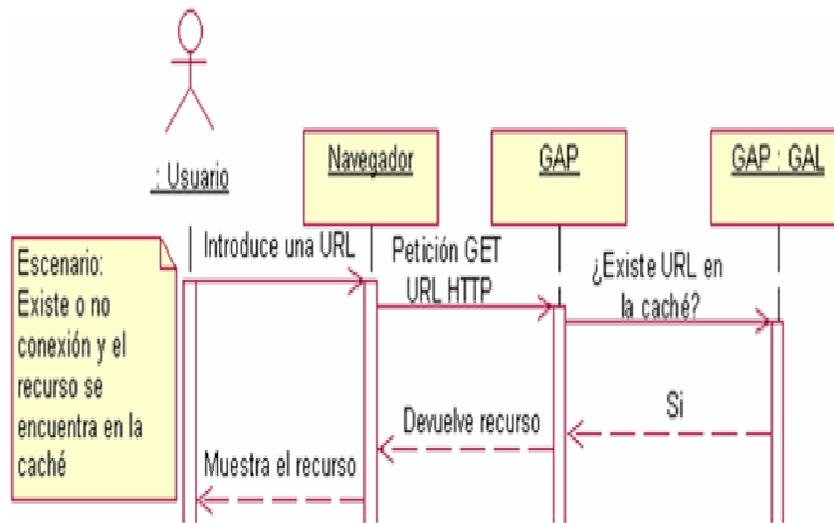


Figura 3.9 Diagrama de secuencia para el escenario donde el recurso está en la caché sin importar la conexión.

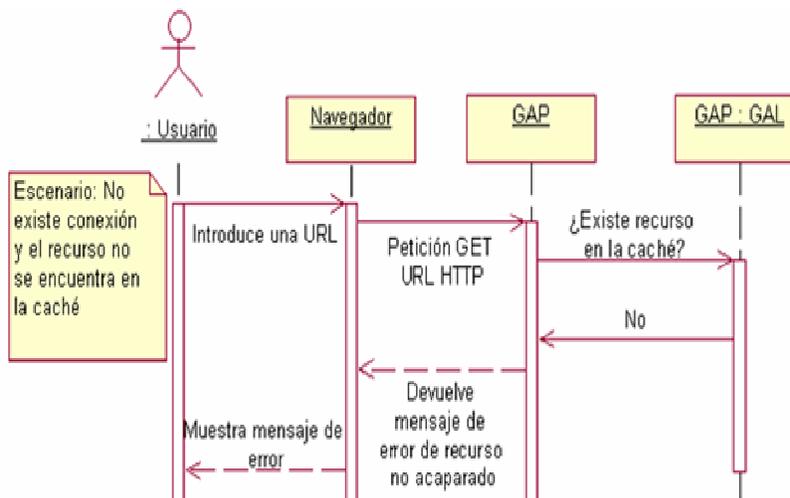


Figura 3.10 Diagrama de secuencia para la visualización de error cuando no existe conexión y el recurso no está en la caché.

En la Figura 3.14, se muestra la secuencia que existe cuando se obtiene un recurso en línea debido a que el recurso no se encuentra en la caché del dispositivo local. En este escenario el prototipo funciona con un Proxy Web normal.

En la Figura 3.15, se muestra el escenario en donde se obtiene un recurso Web que ha sido transcodificado. En este punto, el GAT MT genera un encabezado específico para cada formato de transcodificación.

En las siguientes subsecciones se describen cada uno de los componentes principales del GAP: Observador, GAL y GDL.

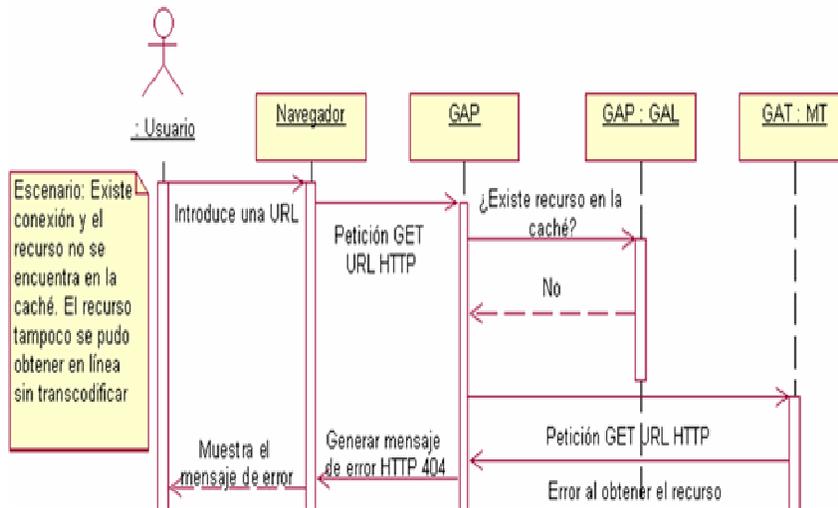


Figura 3.11 Escenario de visualización de error cuando existe conexión, el recurso no está en la caché y no se pudo obtener recurso en línea.

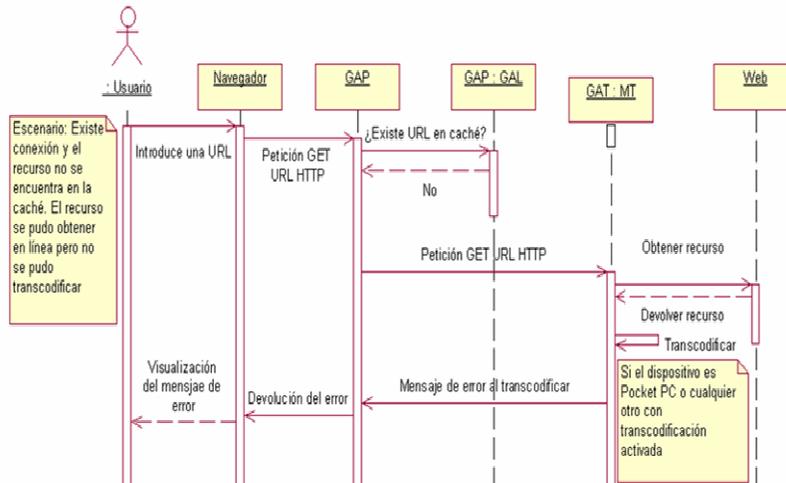


Figura 3.12 Diagrama de secuencia para la visualización de errores cuando existe conexión, el recurso no está en la caché y se pudo obtener en línea pero no transcodificar.

3.3.1 Observador

También llamado *vigía*, su objetivo fundamental es revisar y procesar las peticiones tanto de entrada como de salida que van o vienen dirigidas desde el navegador Web. El objetivo de esta clase consiste en brindar seguimiento a cada una de las peticiones realizadas por el navegador para llevarlas a buen término. La especificación del proceso de obtención de un recurso Web está definida en la Figura 3.16.

3.3.2 GDL (Gestor de Desconexión Local)

Tiene la función de revisar el medio físico de conexión y determinar si el cliente se encuentra conectado o no. Se encarga de controlar las desconexiones. En base a las

estadísticas determina en que estado se encuentra el sistema: en línea o fuera de línea. El proceso de monitoreo de una desconexión se muestra en la Figura 3.17.

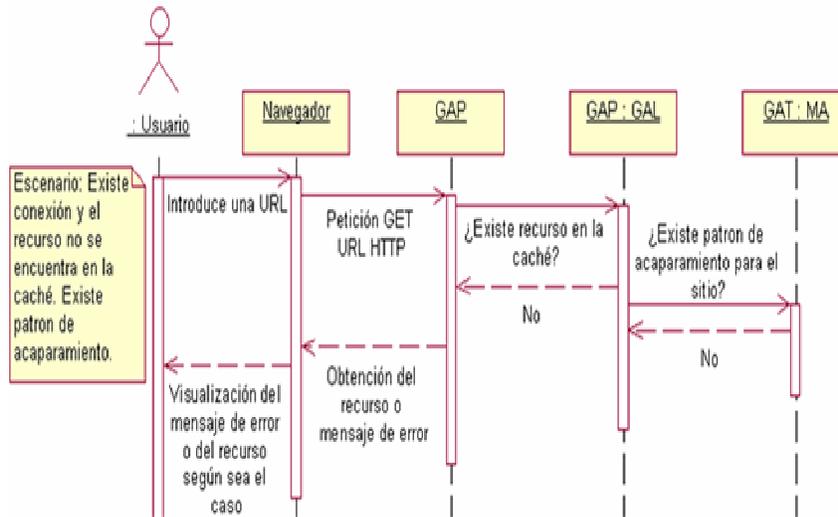


Figura 3.13 Diagrama de secuencia del escenario cuando existe conexión, el recurso no está en la caché, y existe patrón de acaparamiento.

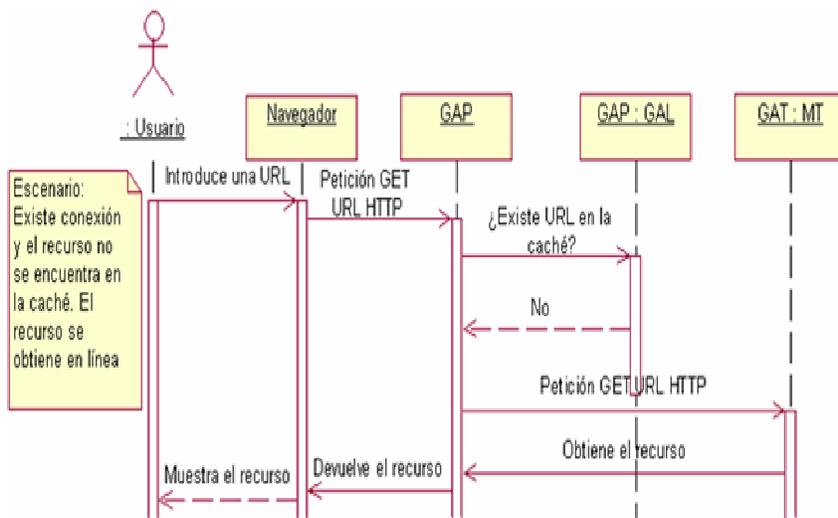


Figura 3.14 Diagrama de secuencia para el escenario cuando existe conexión, el recurso no está en la caché, y el recurso se obtiene en línea.

3.3.3 GAL (Gestor de Acaparamiento Local)

Su función consiste fundamentalmente en: revisar si existe un recurso acaparado. En caso de existir, se manda el recurso solicitado (en este caso el Observador cambia el encabezado de respuesta para hacer creer al navegador que la solicitud viene del exterior) al navegador local. En caso de no existir el recurso acaparado, se manda un mensaje de error. El observador construye una página Web con un mensaje de error que el usuario visualizará en su navegador. La otra actividad fundamental consiste en controlar la sincronización de la caché transcodificada local con la caché del GAT MT. El diagrama de la arquitectura de la caché se muestra en la Figura 3.18.

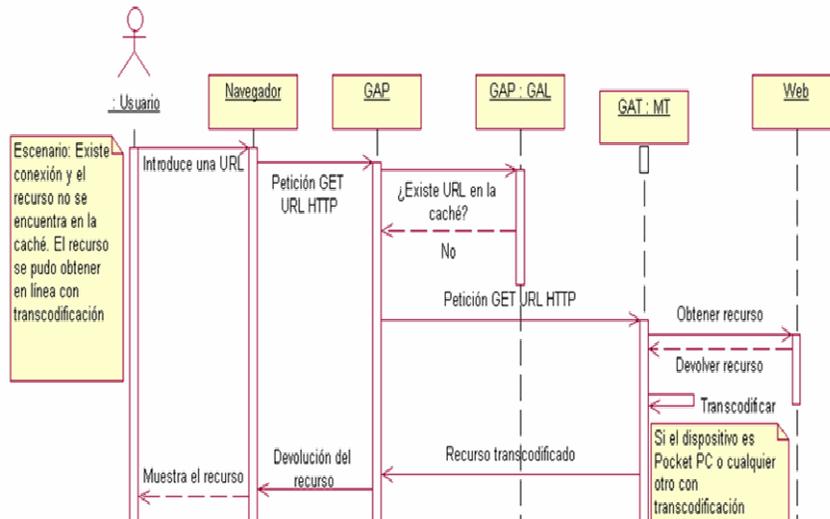


Figura 3.15 Diagrama de secuencia que modela el escenario cuando existe conexión, el recurso no está en la caché, y el recurso se obtiene en línea mediante transcodificación.

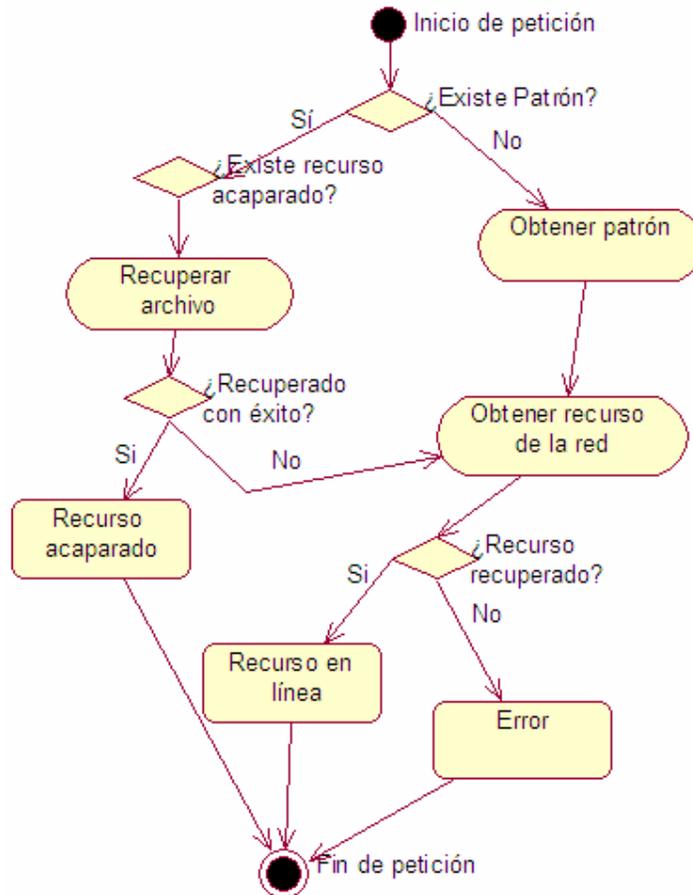


Figura 3.16 Diagrama de actividades del procesamiento de acaparamiento.

En la Figura 3.18, se puede apreciar que el GAP obtiene la ubicación del índice contenedor de patrones (patrones.xml) del archivo de configuración del GAP (config.xml). En dicho archivo contenedor de patrones se encuentran referencias hacia los patrones de sitios Web en particulares. Dentro de dichos archivos de patrones se

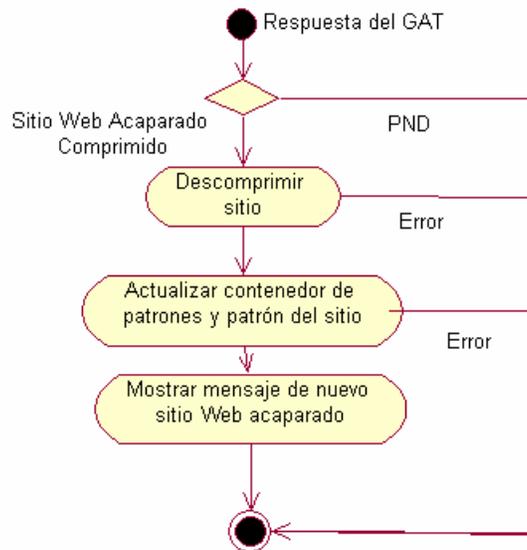


Figura 3.19 Diagrama de actividades del proceso de sincronización de la caché en el GAP.

En el Anexo D se muestra información sobre el archivo de configuración del GAP.

3.4 GAT

El GAT se divide en dos componentes principales. El MA (Mecanismo Acaparador) encargado de gestionar los patrones de acceso a sitios Web para poder replicarlos en el dispositivo móvil y el MT (Mecanismo Transcodificador) encargado de transformar una página de un formato a otro en vista de su correcta visualización en dispositivos móviles.

3.4.1 MT (Mecanismo Transcodificador)

Es el encargado de transcodificar los recursos Web solicitados por los clientes móviles. Este módulo es parte de Moviware [Uri04] y se adaptó para que se integrara con el GAT MA y así de esta manera, se pudieran obtener patrones de sitios Web acaparados. Las modificaciones más importantes consistieron en idear una variante del mecanismo de transcodificación para que transcodificara documentos a varios formatos. El otro cambio consistió en permitir que el proceso se pudiera realizar de manera asíncrona.

También se encarga de analizar el contenido de los encabezados HTTP de las solicitudes realizadas por los clientes. En este caso nos interesa encontrar información como el recurso solicitado, el tipo de cliente que realiza la petición (sistema operativo y tipo de microprocesador) y la fecha de modificación del recurso solicitado, entre otros campos. Este analizador identifica el tipo de dispositivo que realiza la conexión.

El GAT MT se encarga básicamente de interpretar el formato que contiene la petición a un recurso Web (ver Tabla 3-2), revisando la etiqueta *X-Transform* (sino la tiene se asume de manera predeterminada formato HTML) y si se trata de un dispositivo móvil, convierte la página al formato adecuado al dispositivo.

El mecanismo de transcodificación se asemeja de manera muy generalizada al modelo vista controlador, tal y como se observa en la Figura 3.20. En donde el modelo corresponde al recurso Web transformado a un esquema de XML, el controlador es propiamente el GAT MT que maneja la forma en que se va a transformar el documento y finalmente la vista es representada por el documento en todos los formatos disponibles.

Tabla 3-2 Ejemplo de encabezado de solicitud HTTP.

```
GET http://www.cenidet.edu.mx/ HTTP/1.0
Accept: */*
UA-OS: Windows CE (Pocket PC) –Version 3.0
UA-Color: Color16
UA-Pixeles: 240x320
UA-CPU: ARM SA1110
UA-Voice: False
UA-Language: Mozilla/2.0
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/2.0 (Compatible; MSIE 3.02; Windows CE; PPC; 240x320)
Host: www.itmorelia.edu.mx
Proxy-Connection: Keep-Alive
X-Transform: XHTML-MP; Complete; Synchronous
X-Hoard: Asynchronous
```

Dentro de la petición se han agregado dos características adicionales al formato de transcodificación, el modo de transcodificación y el tipo de sincronía. Los grados de transcodificación pueden ser *none*, *partial* y *complete*; que indican que la página no se transcodifica, que se transcodifica pero si ocurre un error se retorna la página sin transcodificar, y que la página se transcodifica y si ocurre un error éste se muestra al usuario.

En lo referente al modelo de sincronía, el valor *Synchronous* nos indica que se sigue un esquema tradicional de petición/respuesta; mientras que el valor *Asynchronous* indica que el usuario realiza una petición, este se desconecta y finalmente se vuelve a conectar obteniendo el resultado de su transacción.

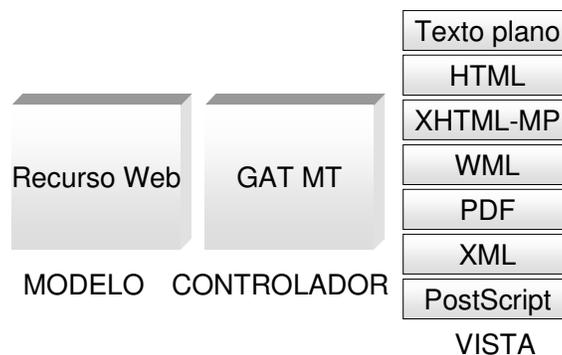


Figura 3.20 Modelo vista controlador aplicado al prototipo desarrollado.

Una vez identificado el formato se procede con la transformación del recurso. El proceso general de transcodificación se ilustra en la Figura 3.21.

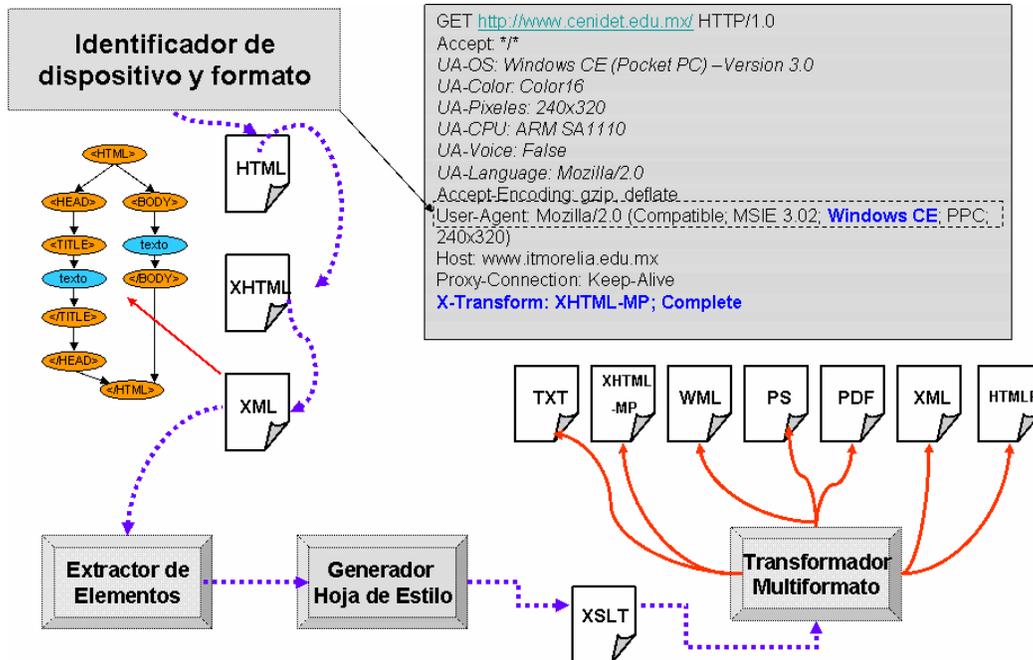


Figura 3.21 Proceso general de transcodificación en el GAT MT.

El primer paso consiste en determinar el tipo de recurso, si el recurso es diferente de HTML el transcodificador funciona como cualquier otro Web proxy. Si el recurso es HTML se procede a su descarga, es necesario tener el documento completo ya que de lo contrario no es posible operar con él. Otros formatos de entrada puede ser páginas dinámicas como JSP, PHP, ASP, pero al final estas páginas deben generar código basado en HTML.

En la Tabla 3-3, se muestra el código fuente de un recurso HTML que será transcodificado.

Tabla 3-3 Documento HTML original.

```

<HTML>

<Head><title>Prueba del transcodificador</title></head>
<Body>
  <center> <h1>Prueba del transcodificador</h1> </center>
  <p>Página de prueba<br> Diseñado por
  <a href="mailto:jcolivares04c@cenidet.edu.mx">Juan Carlos Olivares
  Rojas </a><hr> Centro Nacional de Investigación y Desarrollo
  Tecnológico <i>(cenidet)</i> <a href="http://www.cenidet.edu.mx">
  </a>
<table>
  <tr>
    <td>Area:</td>
    <td>Sistemas Distribuidos</td>
    <td><a href="http://www.sd-cenidet.com.mx"> http://www.sd-
    cenidet.com.mx </a></td>
  
```

```

</tr>
<tr>
  <td> Especialización: </td>
  <td> Cómputo móvil </td>
  <td> <a href.="http://www.cenidet.edu.mx/~wm-serna/">
http://www.cenidet.edu.mx/~wm-serna </a></TD>
</tr>
</table>
</body>
</html>

```

Una vez descargado el recurso, se procede a convertir el documento de HTML a XHTML (ver Tabla 3-4). Esta transformación se realiza con JTidy [jti06] que es un convertidor de XHTML para Java. Los errores y alertas generados se guardan en el archivo erroresXHTML.txt

Tabla 3-4 Documento transformado en XHTML.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head><title>Prueba del transcodificador</title></head>
  <body>
    <center> <h1>Prueba del transcodificador</h1> </center>
    <p>Página de prueba<br /> Diseñado por <a
href.="mailto:jcolivares04c@cenidet.edu.mx">Juan Carlos Olivares Rojas
</a><hr /> Centro Nacional de Investigación y Desarrollo Tecnológico
<i>(cenidet)</i> <a href.=  </img>
</p>
    <table>
      <tr>
        <td>Area:</td>
        <td>Sistemas Distribuidos</td>
        <td><a href.="http://www.sd-cenidet.com.mx"> http://www.sd-
cenidet.com.mx </a></td>
      </tr>
      <tr>
        <td> Especialización: </td>
        <td> Cómputo móvil </td>
        <td> <a href.="http://www.cenidet.edu.mx/~wm-serna/">
http://www.cenidet.edu.mx/~wm-serna </a></td>
      </tr>
    </table>
  </body>
</html>

```

Una vez obtenido el documento XHTML si es que no ha ocurrido un error grave (no se pudo cerrar alguna etiqueta, etiquetas inválidas en el documento, etc.), el modelo se convierte a XML añadiendo los encabezados respectivos para posteriormente subirlo a memoria y procesarlo (ver Tabla 3-5).

Estando el documento en memoria es posible reformatear al documento extrayendo toda la información relevante como es: el título, el texto, las imágenes y los enlaces; con base en esta información primero se construyen un archivo intermedio reflejando el nuevo formato y una plantilla de transformación. Para realizar esta transformación se genera un archivo intermedio eliminando todos los elementos que causan ruido en el documento y que no se pueden mostrar de manera adecuada en el dispositivo, tal es el caso de tablas, formularios, marcos, etc. A este documento intermedio (ver Tabla 3-6) será al que se le aplique el proceso de transformación a otros formatos si es que no se presentó ningún error a la hora de realizar la transformación.

Tabla 3-5 Documento en XML.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<html>
  <head><title>Prueba del transcodificador</title></head>
  <body>
    <center> <h1>Prueba del transcodificador</h1> </center>
    ...
  </body>
</html>
```

Tabla 3-6 Archivo intermedio generado.

```
<html>
  <head><title>Prueba del transcodificador</title></head>
  <body>
    <h1>Prueba del transcodificador</h1>
    <p> Página de prueba <br /> Diseñado por <hr /> Centro Nacional de
      Investigación y Desarrollo Tecnológico <i>(cenidet)</i>
       </img> Area Sistemas Distribuidos
      Especialización: Cómputo móvil
    <p>Enlaces</p>
    <p> <a href="http://www.cenidet.edu.mx/~wm-serna/">
      http://www.cenidet.edu.mx/~wm-serna </a></p>
    <p><a href="http://www.sd-cenidet.com.mx"> http://www.sd-
      cenidet.com.mx </a></p>
    <p>< a href="mailto:jcolivares04c@cenidet.edu.mx">Juan Carlos
      Olivares Rojas </a>
    </p>
  </body>
</html>
```

Esta plantilla depende del formato definido en la etiqueta *X-Transform*. La transformación se basa en dos tecnologías de XSL: XSLT y XSL-FO. La primera para transformar documentos basados en XML a otros documentos basados en XML (WML, XHTML, o cualquier otro lenguaje de marcado como podría ser RDF, OWL, MathML, UIML, VoiceML, etc.). La segunda tecnología se enfoca a documentos no basados en XML como es el caso de PDF, RTF, PS, entre otros (ver Tabla 3-7 y Tabla 3-8).

Tabla 3-7 Extracto de la plantilla generada para PDF.

```
<?xml version='1.0' encoding='iso-8859-1'?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="html">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        ...
      </fo:layout-master.set>
      <fo:block font-family="Helvetica" font-size="18pt" text-align="end" space-
before="2cm">
        GAT MT X-transform Convertidor de recursos Web a PDF
      </fo:block>
      <fo:block font-weight="bold" color='blue'>
        Prueba del transcodificador</fo:block>
      <xsl:value-of select="body" /><xsl:text> </xsl:text>
      <fo:block font-size="12pt" color='red'> * Imágenes *</fo:block>
      <fo:block><fo:external-graphic src='cenidet.jpg' height='1.4cm'
width='2.0cm' scaling='uniform' /></fo:block>

      <fo:block font-size="12pt" color='red'> * Enlaces *
      <fo:basic-link external-destination='http://www.cenidet.edu.mx/~wm-serna'
color='blue'>http://www.cenidet.edu.mx/~wm-serna/
      </fo:basic-link>
      ...
    </fo:root>
  </xsl:template>
</xsl:stylesheet>
```

Tabla 3-8 Documento generado en WML.

```
<wml>
  <card title="Prueba del transcodificador">
    <p><big>Prueba del transcodificador</big></p>
    <p>Página de prueba Diseñado por Centro Nacional de Investigación y Desarrollo
Tecnológico (cenidet) Area Sistemas Distribuidos Especialización: Cómputo
móvil </p>
    <p>* Imágenes *</p>
    <p> </img> </p>
    <p>* Enlaces * </p>
    <p><a href="http://www.cenidet.edu.mx/~wm-serna">
http://www.cenidet.edu.mx/~wm-serna </a></p>
    <p><a href="http://www.sd-cenidet.com.mx"> http://www.sd-cenidet.com.mx
</a></p>
    <p><a href="mailto:jcolivares04c@cenidet.edu.mx">Juan Carlos Olivares Rojas
</a></p>
  </card>
</wml>
```

Una vez realizada la transformación se devuelve el recurso en caso de éxito o se muestra el mensaje informativo de error. Con este proceso de transcodificación, el usuario podrá visualizar de mejor manera las páginas Web en su dispositivo móvil. Es importante en la respuesta construir un encabezado de respuesta con el tipo MIME respectivo para que el usuario (ver Tabla 3-9).

Tabla 3-9 Encabezado de respuesta generado.

HTTP/1.0 200 Ok
Server: cenidet MoviWeb GAP
Content-Type: text/wml
Content-Length: 100

El proceso de transformación se visualiza en las Figura 3.22 y Figura 3.23.

En las Figura 3.24, se muestra el diagrama de clase del GAT MT en donde se marcan todos aquellos métodos y propiedades que se agregaron y/o modificaron del módulo transcodificador, las pequeñas modificaciones no se ven reflejadas.

En el Anexo C, se muestra el archivo de configuración del GAT MT, así como de la descripción de cada campo.

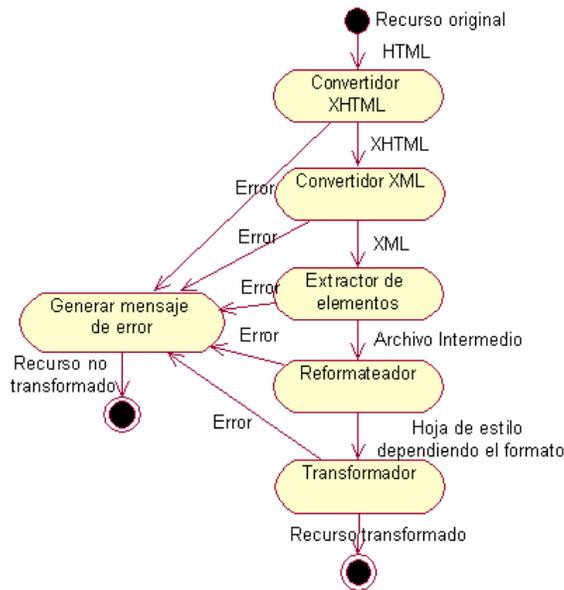


Figura 3.22 Diagrama de actividades del proceso de transcodificación.

3.4.2 MA (Mecanismo Acaparador)

Se encarga de dos funciones: acaparamiento de un sitio Web transcodificado y sincronización de la caché transcodificada. Este módulo ya existía en *Moveware* [Ver03] pero sin transcodificar, la meta fue integrar este módulo al prototipo desarrollado, de tal forma que se detecte el dispositivo desde el cual se está realizando una petición de un recurso Web y se envíe al cliente, ya sea en su forma original o en su forma transcodificada.

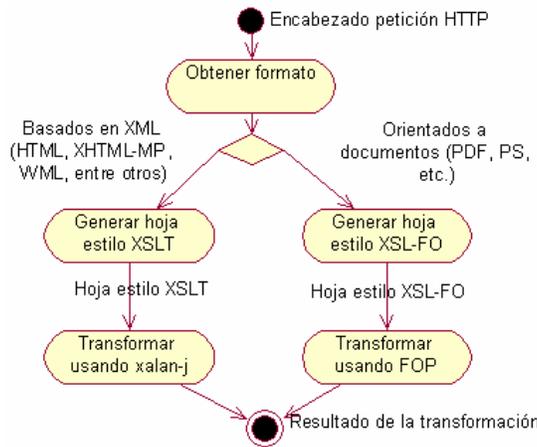


Figura 3.23 Proceso de transcodificación del documento.

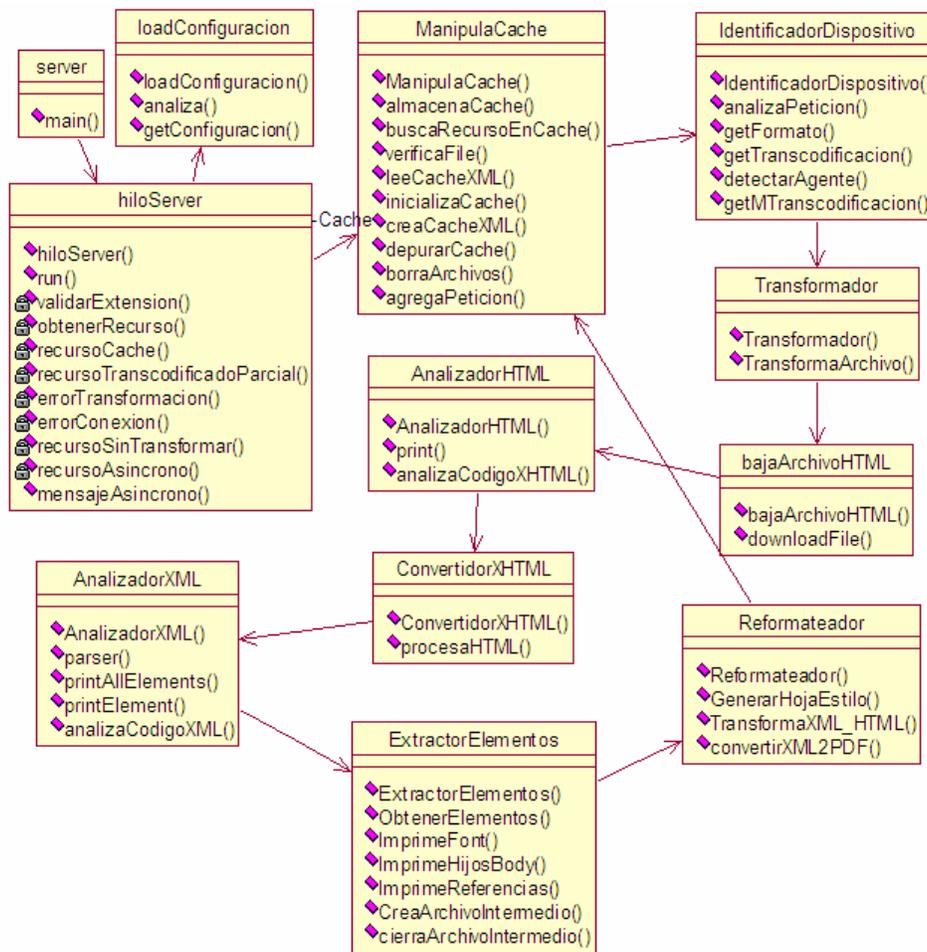


Figura 3.24 Diagrama de clases del GAT MT.

En la Figura 3.25, se muestran los métodos y propiedades que se agregaron y/o modificaron del módulo acaparador en el lado servidor de Moveware. Las pequeñas modificaciones no se muestran (ya que de lo contrario habría que remarcar prácticamente todo el diagrama). El proceso de sincronización de la caché en el lado

servidor (GAT MA) se muestra en la Figura 3.26. En la Tabla A-33 se muestra el archivo de configuración del GAT MA.

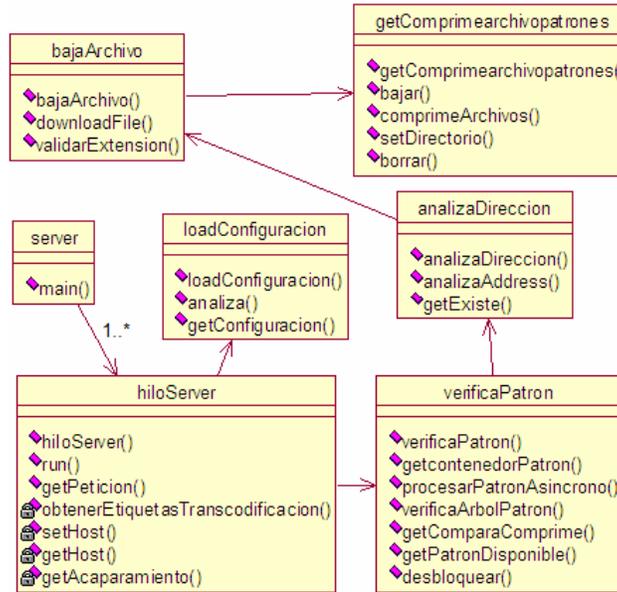


Figura 3.25 Diagrama de clases del GAT MA.

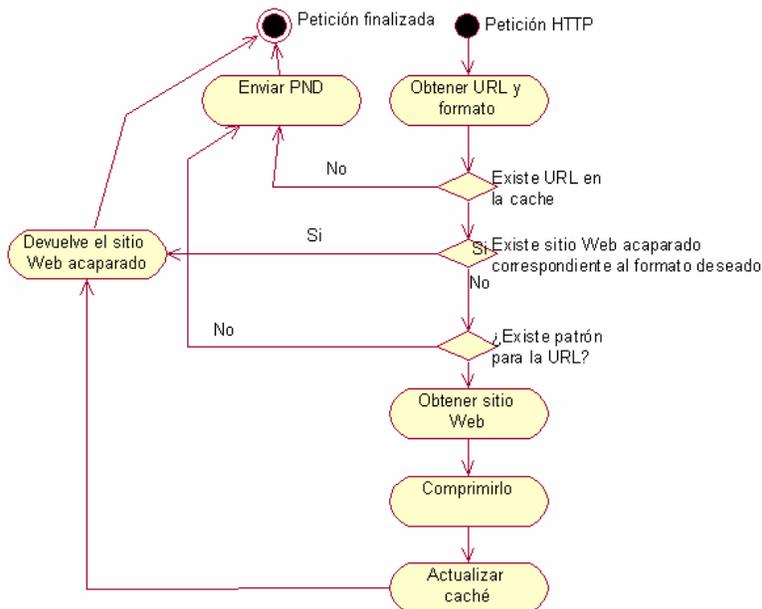
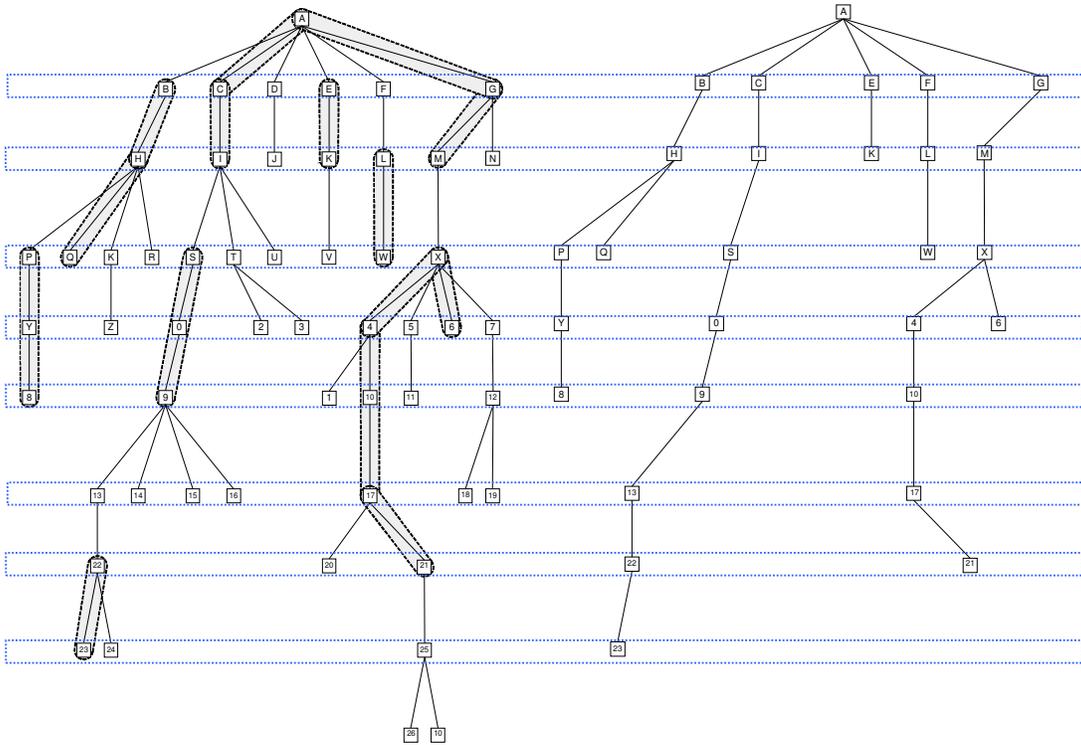


Figura 3.26 Sincronización de la caché en el GAT MA.

En la Figura 3.27 se muestra el diagrama de acaparamiento de un sitio Web. En dicha figura se muestra un árbol que representa un sitio Web así como los patrones obtenidos. De dichos patrones se obtiene un árbol que se recorta para realizar acaparamiento. Este árbol recortado es el que se comprime y envía a los dispositivos móviles.



a) Estructura de árbol de un sitio Web b) Subestructura del sitio Web (árbol patrón)

Figura 3.27 Estructura genérica de un sitio Web.

CAPÍTULO 4

PRUEBAS

“Si buscas resultados distintos, no hagas siempre lo mismo”
Albert Einstein

4.1 Introducción

Para probar el prototipo desarrollado se implementó un plan de pruebas dividido en dos partes. En la primera, se realizaron pruebas de factibilidad sobre el prototipo implementado, mientras que en la segunda se realizaron pruebas auxiliares que complementan este plan de pruebas.

En lo que respecta a las pruebas de factibilidad, se evaluaron cinco casos generales de prueba, ya que en un solo escenario no se pueden probar todas las características que posee el prototipo.

En lo que respecta a las pruebas auxiliares, éstas se dividieron en tres partes con el objetivo de mostrar que el prototipo implementado puede funcionar en otros dispositivos con sistema operativo Windows Mobile, en una gran diversidad de dispositivos móviles heterogéneos y finalmente, las pruebas de rendimiento para conocer que tan efectivo es nuestro prototipo.

Las pruebas se realizaron tomando como base un dispositivo iPAQ rx3115 de la compañía HP con Windows Mobile 2003, aunque también algunas pruebas se realizaron con un dispositivo iPAQ H3630 de la compañía Compaq con Pocket PC 2000, un dispositivo Jornada 540 de la compañía HP con Windows Mobile 2002 y un Axim de la compañía Dell con Windows Mobile 5.

La instalación del GAP se detalla en el Anexo F. Para ejecutar el programa GAP se debe configurar el navegador Web del dispositivo móvil con la dirección IP 127.0.0.1 para indicar que el Proxy se ejecuta localmente y el puerto de conexión corresponde al 10800 (el cual se registró ante la IANA [iana06]). Para las pruebas que involucran transcodificación se necesita que el GAT MT esté en ejecución escuchando por el puerto 2700. Para las pruebas que involucran acaparamiento se requiere que el GAT MA esté en ejecución que generalmente corresponde al puerto 1800.

4.2 Caso de prueba 1: Configuración del GAP

En este caso de prueba se verá la interfaz gráfica así como las opciones con las que cuenta el GAP. El objetivo de este caso de prueba es comprobar el correcto

funcionamiento de todas las características con las que cuenta el prototipo desarrollado para dispositivos PPC.

El sistema está compuesto por una interfaz gráfica la cual permite visualizar todos los detalles de ejecución del sistema como son las notificaciones del sistema así como visualizar el estado que guardan las peticiones. El menú de opciones está optimizado para desplegarse tanto en dispositivos Pocket PC y Smartphone para que funcione con las “teclas suaves” (tecla opción izquierda y tecla opción derecha). El menú de opciones cuenta con cuatro opciones las cuales son: GAP, GAT, Créditos y ayuda. Las dos primeras opciones presentan submenús, tal y como se muestra en el lado izquierdo de la Figura 4.1.

El GAP está estructurado de tal forma que pueda ejecutarse en tarjetas de almacenamiento. En la carpeta ayuda se encuentran todos los archivos de ayuda, en la carpeta *cache* se encuentran todos los recursos acaparados, en la carpeta *config* se encuentran los archivos de configuración y control, en la carpeta *etc* se encuentran archivos auxiliares y finalmente en la carpeta *patrones* se guardan todos los archivos contenedores de patrones. En el lado derecho de la Figura 4.1, se muestra la estructura interna que presenta el GAP.

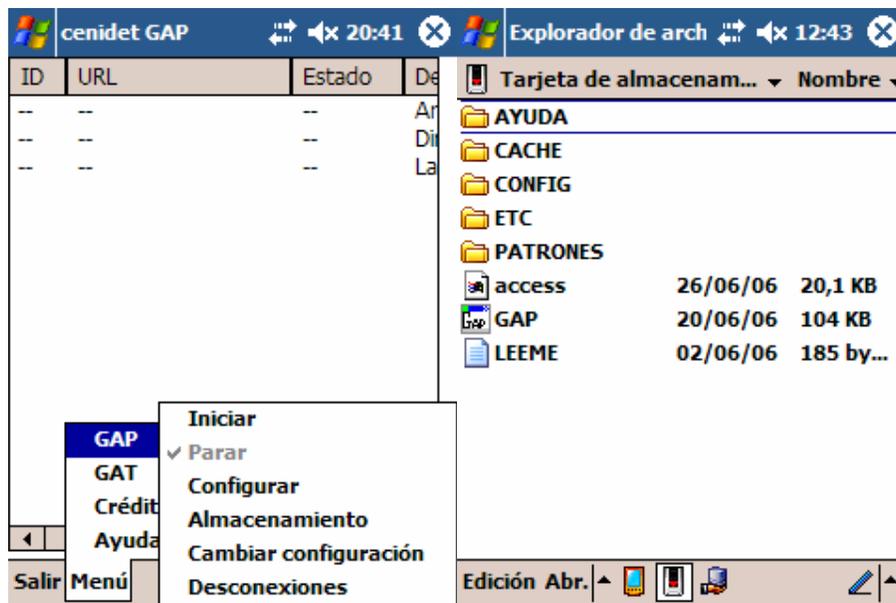


Figura 4.1 Interfaz gráfica y estructura interna del GAP.

A continuación se describe cada una de los subcasos de pruebas.

Tabla 4-1 Subcaso 1.1: carga del archivo de configuración.

<i>Objetivo:</i>	Probar que el sistema lee el archivo de configuración del sistema.
<i>Descripción:</i>	Se ejecuta el sistema y se cargan diversos archivos de configuración, si el archivo no tiene el formato adecuado se toman valores predeterminados. Las opciones de configuración leídas son mostradas en el área de notificación.
<i>Resultado esperado:</i>	Se lee correctamente cada parámetro de configuración.
<i>Resultado obtenido:</i>	Se puede leer de manera adecuada el archivo de configuración

correctamente.

Tabla 4-2 Subcaso 1.2: configuración del sistema.

<i>Objetivo:</i>	Comprobar que el sistema a través de su interfaz es capaz de configurar el sistema.
<i>Descripción:</i>	Se ejecutan cada una de las opciones de configuración del sistema, y se observa si se escriben correctamente en el archivo de configuración y si pueden ser leídos nuevamente.
<i>Resultado esperado:</i>	El sistema puede cambiar las opciones de configuración.
<i>Resultado obtenido:</i>	El sistema puede dar de alta y cambiar la configuración del sistema.

Tabla 4-3 Subcaso 1.3: otras opciones

<i>Objetivo:</i>	Verificar que todo los módulos opcionales funcionen correctamente (probar conexión, mostrar ayuda y créditos).
<i>Descripción:</i>	Se muestran las opciones faltantes, como lo son la ayuda, probar si el GAT MA y MT están conectados, entre otros.
<i>Resultado esperado:</i>	El sistema de ayuda y notificaciones funciona correctamente.
<i>Resultado obtenido:</i>	El usuario puede visualizar de forma correcta la ayuda y los mensajes de notificación del estado de la conexión.

En la Figura 4.2, se muestra en el lado izquierdo el sistema de notificaciones al cargar un archivo de configuración en el GAP. En el lado derecho de esta misma figura, se muestra la interfaz de configuración del módulo de desconexiones del GAP.

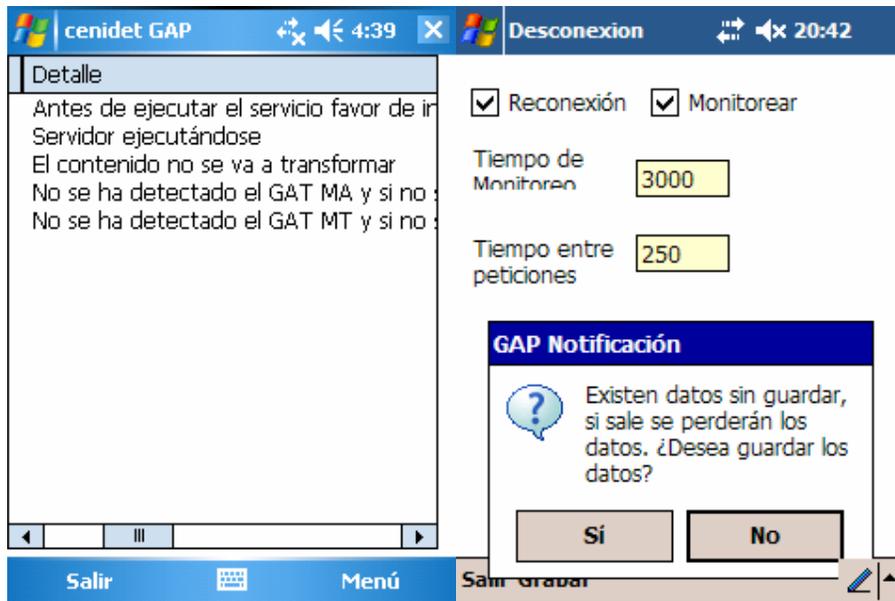


Figura 4.2 Carga del archivo de configuración e interfaz de configuración de desconexiones en el GAP.

4.3 Caso de prueba 2: Recursos sin acaparar y sin transcodificar

En este caso de prueba se pueden obtener recursos sin transformar ni acaparar (recursos en línea). También se muestra otras opciones del GAP que en el caso de prueba anterior no se mostraron ya que es hasta entonces que se hace la evaluación del funcionamiento del GAP.

Tabla 4-4 Subcaso 2.1: visualización de mensajes en la interfaz del GAP.

<i>Objetivo:</i>	Verificar que el sistema muestre el estado real del sistema en la interfaz.
<i>Descripción:</i>	Se obtienen recursos en línea y el sistema muestra el estado de las peticiones; es decir si se pudieron obtener en línea o no.
<i>Resultado esperado:</i>	El sistema debe mostrar cada mensaje en la interfaz del sistema.
<i>Resultado obtenido:</i>	El sistema muestra todos los mensajes de advertencia sobre la interfaz.

En la Figura 4.3, se muestra de manera gráfica las peticiones que son atendidas por nuestro prototipo GAP. En este sentido, el usuario puede dejar el prototipo en segundo plano y este seguirá trabajando atendiendo las peticiones como cualquier otro Proxy.

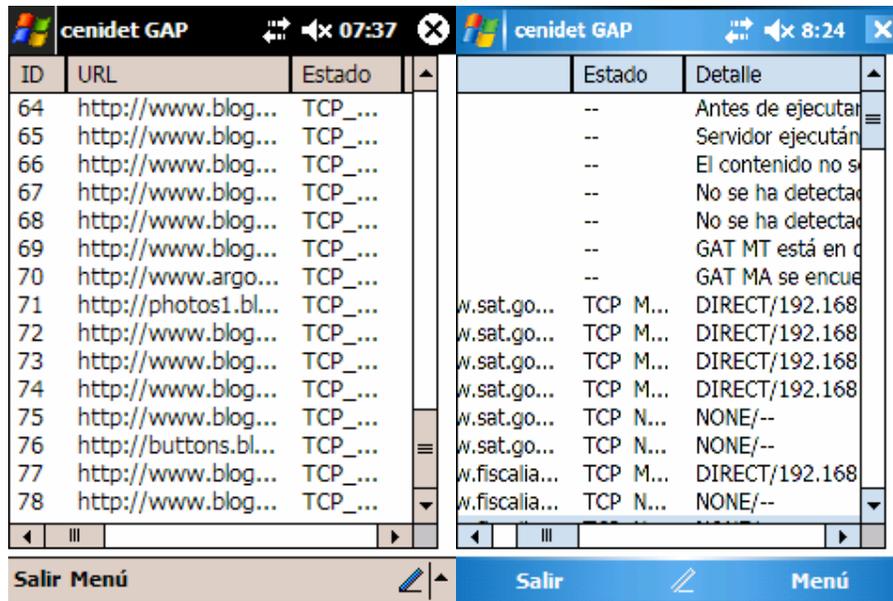


Figura 4.3 Visualización de peticiones en la interfaz del GAP.

Tabla 4-5 Subcaso 2.2: visualización de la bitácora.

<i>Objetivo:</i>	Ver que el sistema genere correctamente el histórico de las peticiones.
<i>Descripción:</i>	Se realizan peticiones hacia recursos Web y se comprueba que el sistema guarde un histórico de las peticiones realizadas. El sistema permite que se guarde una bitácora de las notificaciones del sistema o bien en su caso no se utilicen bitácoras. El formato de la bitácora está en formato nativo de Squid.
<i>Resultado esperado:</i>	La bitácora con las transacciones realizadas al sistema.

Resultado obtenido: El sistema genera una bitácora de manera correcta.

Tabla 4-6 Subcaso 2.3: visualización de errores.

Objetivo:	Ver que el sistema muestre los errores cuando se generen al obtener un recurso en línea.
Descripción:	Se realizaron peticiones a recursos Web cuando no existía conexión en el dispositivo.
Resultado esperado:	El sistema mostrará errores dado que al no existir conexión no puede obtener el recurso en línea además de que el recurso no está acaparado.
Resultado obtenido:	El sistema muestra mensaje de error en el formato apropiado.

Tabla 4-7 Subcaso 2.4: visualización de recursos Web en línea sin transcodificar ni acaparar.

Objetivos:	Ver que el sistema obtenga un recurso de la Web sin transformarlo.
Descripción:	Se realizaron peticiones sobre recursos Web
Resultado esperado:	El sistema obtiene un sitio Web cualquiera como si no se utilizará el sistema.
Resultado obtenido:	El sistema muestra las peticiones obtenidas directamente de la Web.

En la Figura 4.5, en el lado izquierdo se muestra la visualización de un recurso Web sin acaparar y transcodificar; nótese que el recurso no cabe en las dimensiones de la pantalla del dispositivo por lo que se necesitan barras de desplazamiento para poder visualizarlo. En el lado derecho de la misma figura, se muestra la bitácora generada por el sistema, la cual está en el formato nativo de Squid.

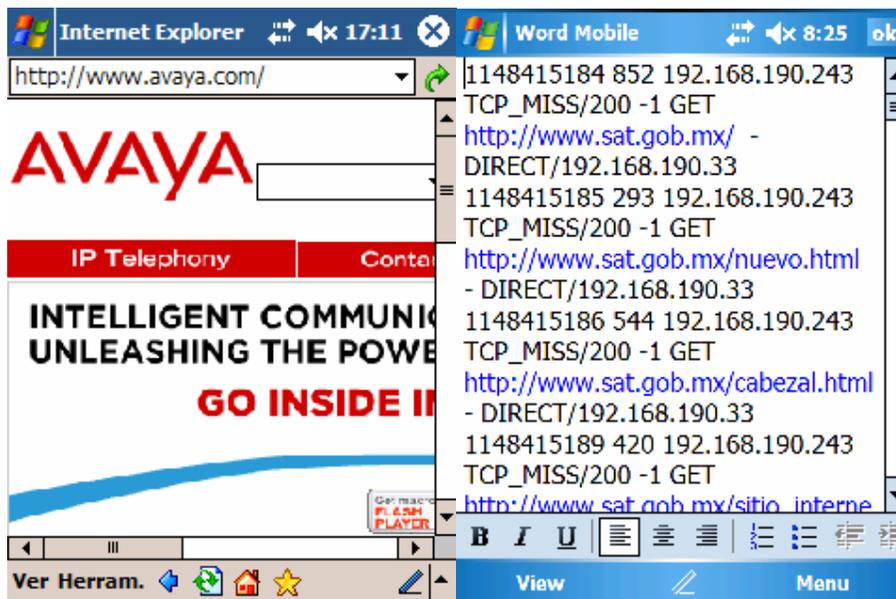


Figura 4.4 Visualización de recursos Web sin transcodificar ni acaparar y bitácora generada por el GAP .

4.4 Caso de prueba 3: Recursos sin acaparar pero transcodificados

Los dos casos de pruebas anteriores se centraron en el funcionamiento del GAP. En este caso de prueba nos centramos en comprobar el funcionamiento del GAT MT.

Tabla 4-8 Subcaso 3.1: Recursos Web transcodificados en diversos formatos.

<i>Objetivo:</i>	Ver que el sistema pueda convertir las páginas Web en distintos formatos.
<i>Descripción:</i>	Se obtuvieron distintos recursos Web y se transcodificaron en distintos formatos para comprobar que se realizaba la transcodificación de documentos Web. Los formatos de transcodificación evaluados fueron: HTMLR, WML, XHTML-MP, PDF, PS, XML y texto plano.
<i>Resultado esperado:</i>	El recurso Web pudo transformarse exitosamente en los formatos deseados.
<i>Resultado obtenido:</i>	Los recursos Web en algunos casos se pudieron transformar en el formato adecuado y en otros no. Los que no se pueden transformar se debe a que no se pudo obtener el recurso de la Web o a que están mal diseñados (se abunda más sobre el tema en las pruebas de accesibilidad).

En la Figura 4.5, se muestra la visualización del sitio <http://www.cenidet.edu.mx/> transcodificado al formato HTMLR. Nótese que el recurso se ajusta a las dimensiones de la pantalla.

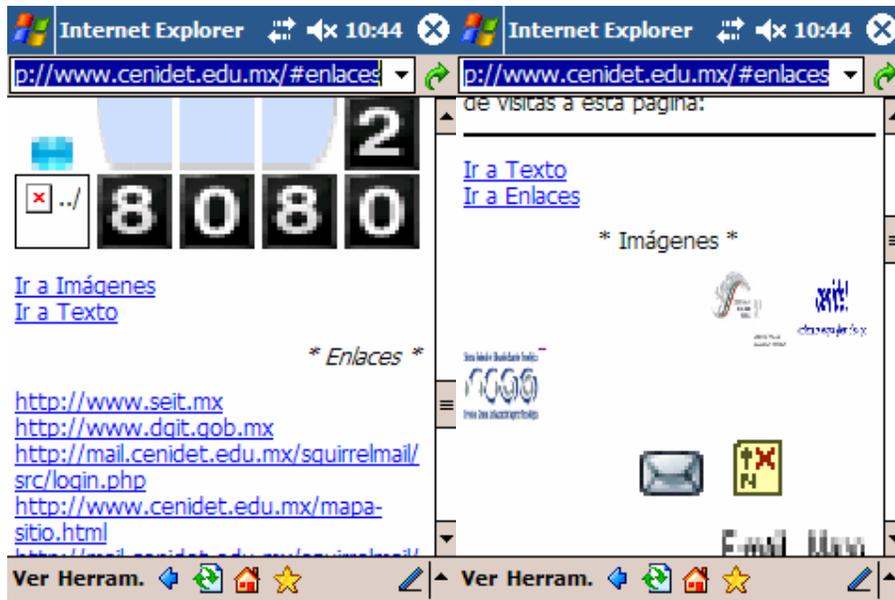


Figura 4.5 Visualización de recursos Web en HTML reformateado.

En la Figura 4.6, se muestra la visualización del buscador Web de turismo experimental a través de ontologías <http://192.168.190.31/turistar/> que ha sido transcodificado al formato WML. Nótese que las imágenes en este formato deben estar en formato WBMP y nuestro trabajo no hace la conversión de formatos.



Figura 4.6 Visualización de recursos Web en WML.

En la Figura 4.7, se muestra la página principal del buscador de recursos Web mediante ontologías realizado en el cenidet. Nótese que si la página tiene una imagen de fondo, esta se conserva.



Figura 4.7 Recurso transcodificado en XHTML-MP.

Tabla 4-9 Subcaso 3.2: funcionamiento de la caché del mecanismo transcodificador.

<i>Objetivo:</i>	Obtener recursos transcodificados de la caché del GAT MT para comprobar su funcionamiento.
<i>Descripción:</i>	Se realizaron varias peticiones de recursos Web para comprobar que el sistema de la caché funciona de manera adecuada.
<i>Resultado esperado:</i>	El GAT MT guarda peticiones de recursos Web transcodificados, para en futuras peticiones ya no sea necesario

volver a realizar la transformación.

Resultado obtenido: El GAT MT es capaz de obtener recursos Web ya transcodificados desde su caché y mostrarlo al cliente

En la Figura 4.8, se muestra la transcodificación de la página principal del cenidet al formato PDF. Nótese que el dispositivo móvil debe tener un visor de PDF para poder visualizar la página transcodificada.

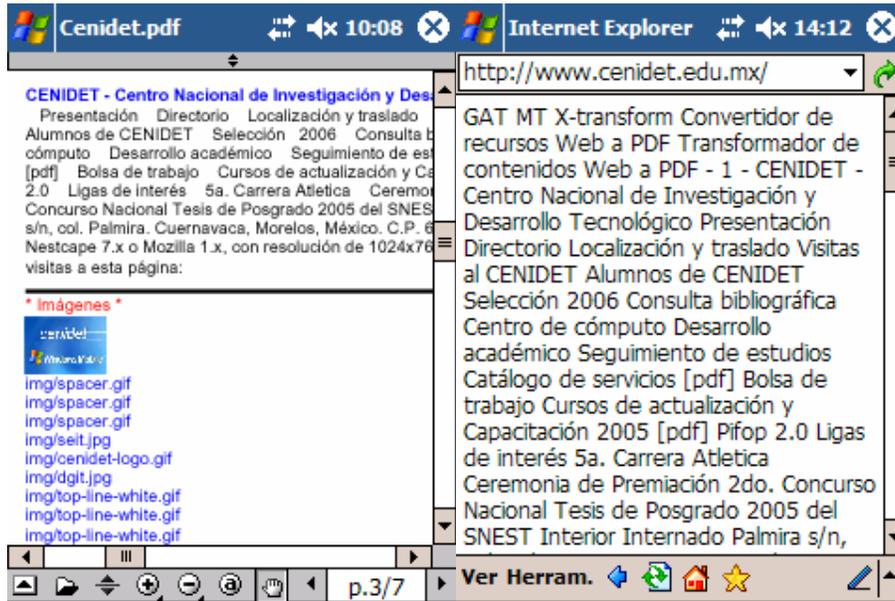


Figura 4.8 Recurso transcodificado en formato PDF.

En la Figura 4.9, se muestra en el lado izquierdo la conversión de una página Web al formato PS. En este caso, no se puede mostrar el documento por qué el dispositivo móvil no cuenta con un visor de PS. En el lado derecho de la misma figura se muestra el mismo recurso pero ahora transcodificado en formato de texto plano.

Tabla 4-10 Subcaso 3.3: Proceso de transcodificación asíncrona.

<i>Objetivo:</i>	Mostrar el funcionamiento del mecanismo asíncrono del GAT MT.
<i>Descripción:</i>	Se realizaron varias peticiones a recursos Web de manera asíncrona. En este esquema se manda un mensaje de que la respuesta se está procesando, si el cliente vuelve a intentar más tarde y aún no se termina de transformar se muestra un mensaje de error. Si la petición ya está transcodificada se manda de manera inmediata.
<i>Resultado esperado:</i>	El GAT MT es capaz de procesar una petición de manera asíncrona
<i>Resultado obtenido:</i>	El GAT MT puede realizar transcodificación de contenidos Web de manera asíncrona.

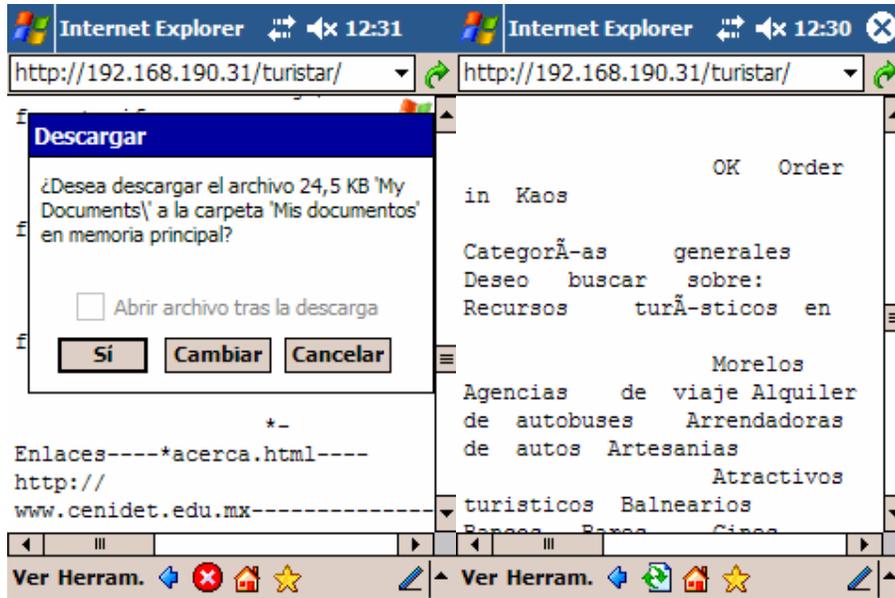


Figura 4.9 Visualización de un recurso Web transcodificado en formato PS, y texto plano.

Tabla 4-11 Subcaso 3.4: visualización de errores en el proceso de transcodificación.

<i>Objetivo:</i>	Mostrar errores que se producen al transcodificar un recurso Web.
<i>Descripción:</i>	Se realizaron varias peticiones de recursos Web variando el grado de transcodificación.
<i>Resultado esperado:</i>	El GAT MT es capaz de mostrar notificaciones al producirse un error en el proceso de transcodificación.
<i>Resultado obtenido:</i>	El GAT MT muestra notificaciones de errores de transcodificación dependientes del formato.

En la Figura 4.10, se muestra los mensajes que se generan en el GAT MT cuando se transcodifica una página Web. Esto nos sirve para ver en que etapa del proceso transcodificación va una petición.

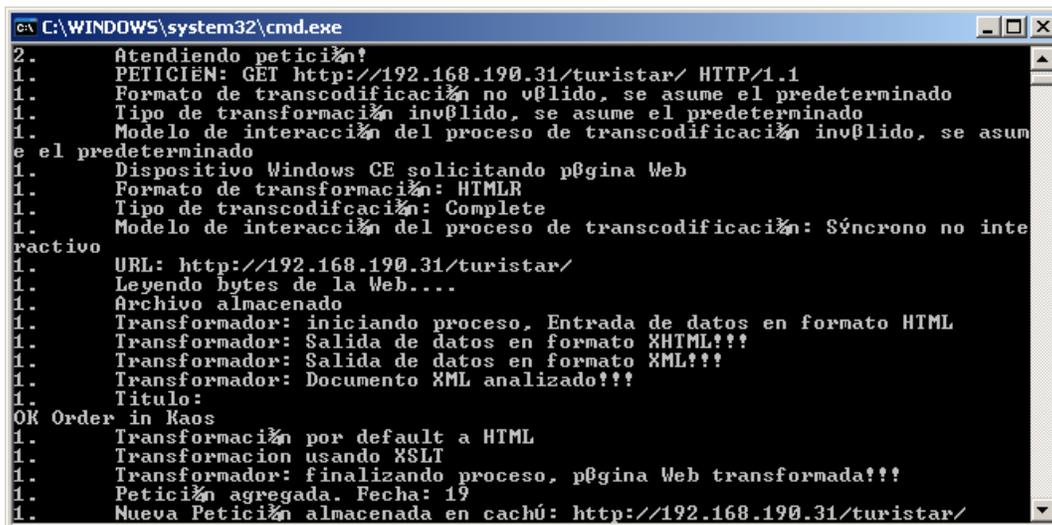


Figura 4.10 Transcodificación de recursos Web utilizando el GAT MT.

En la Figura 4.11, se muestra la pantalla que se despliega cuando se da el proceso de transcodificación se realiza de manera asíncrona. El usuario posteriormente vuelve a conectarse esperando ver si su recurso ya está transcodificado.

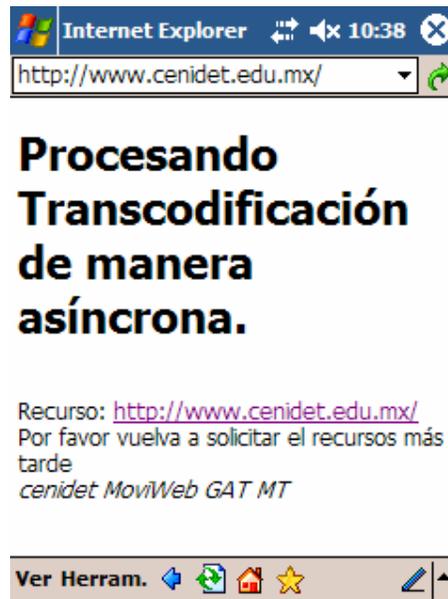


Figura 4.11 Mensaje que indica que el proceso de transcodificación se está realizando de manera asíncrona.

4.5 Caso de prueba 4: Recursos acaparados sin transcodificar

En este caso de prueba se muestra el funcionamiento del GAT MA en lo referente a su funcionamiento interno, es decir obtenemos recursos acaparados pero sin transcodificar.

La generación de patrones de acaparamiento se realiza a través de otro módulo de la arquitectura Moviware denominada Minero Web Cenidet [Her05]. El problema radica en encontrar bitácoras de los servidores Web, por lo que en muchas ocasiones para realizar pruebas se utilizan archivos log sintéticos o patrones de acaparamiento sintéticos.

Tabla 4-12 Subcaso 4.1: obtención de recursos de un patrón de acaparamiento asíncrono o síncrono.

<i>Objetivo:</i>	Verificar que se puedan interpretar los patrones de acaparamiento, obtener los recursos y comprimirlos.
<i>Descripción:</i>	Se realizan varias peticiones para determinar si se pueden obtener los recursos de un patrón de acaparamiento. El proceso puede ser síncrono o asíncrono. Cuando el proceso es asíncrono el GAT MA envía un mensaje PA para indicar que el cliente se desconecte. Si el proceso es síncrono una vez terminada la compresión del patrón, éste se envía al cliente de lo contrario se hará hasta la próxima conexión. La estructura que posee la caché es la misma que contiene el sitio Web pero de manera recortada debido al proceso de acaparamiento.
<i>Resultado esperado:</i>	Se generan archivos ZIP de un patrón de acaparamiento.
<i>Resultado obtenido:</i>	Se obtienen los recursos Web del patrón de acaparamiento que

el cliente visitó y se comprime en un archivo ZIP.

En la Figura 4.12, se muestra la forma en como está estructurada la caché en el mecanismo acaparador. Se puede apreciar que siempre comienza con el formato de transcodificación en mayúsculas seguido por el nombre del sitio Web.

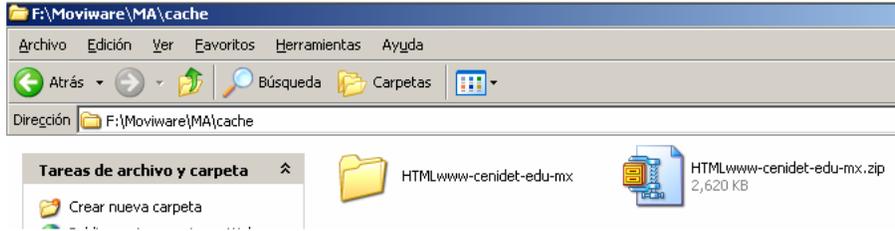


Figura 4.12 Estructura de la caché del GAT MA.

Tabla 4-13 Subcaso 4.2: envío de un sitio acaparado al cliente.

Objetivo:	Comprobar que el sistema puede enviar un sitio Web acaparado que se encuentra en el GAT MA.
Descripción:	Se realizaron peticiones sobre diversos sitios Web. Si ya se cuenta con el sitio Web acaparado se envía al cliente. En caso contrario se manda un mensaje PND que indica que el patrón no está disponible. Este mismo mensaje se envía cuando ocurre un error en el proceso de acaparamiento.
Resultado esperado:	El sistema puede enviar sitios Web acaparados al dispositivo cliente
Resultado obtenido:	El GAT MT envía sitios Web acaparados

En la Figura 4.13, se muestra como el GAT MA descara archivos por medio del GAT MT y si estos son HTML a su vez se transcodifican. Una vez acaparados todos los recursos de un patrón de acaparamiento, se procede a comprimirlos para posteriormente estar listos cuando un dispositivo móvil lo solicite.

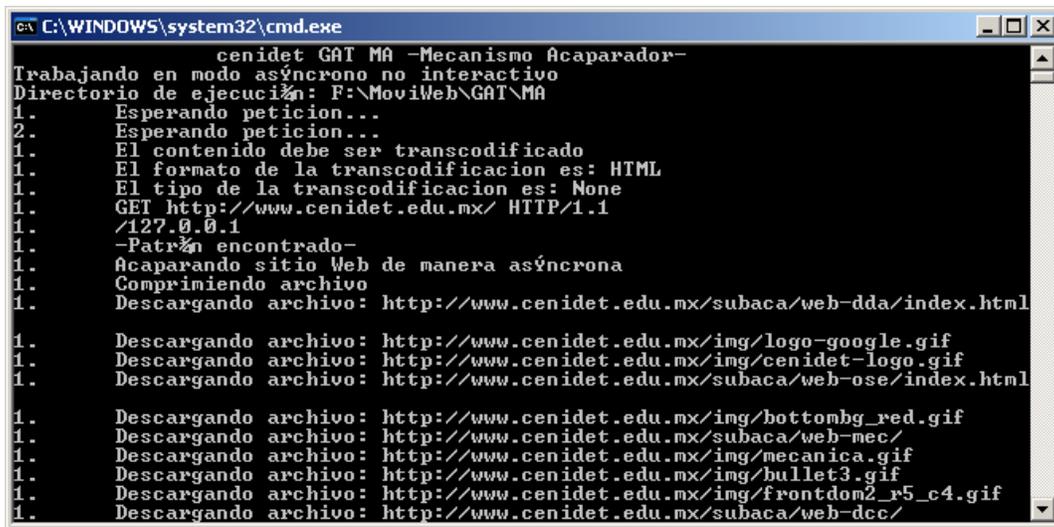


Figura 4.13 Descarga de recursos contenidos en un patrón de acaparamiento.

4.6 Caso de prueba 5: Recursos acaparados y transcodificados

El objetivo de esta prueba es probar el sistema en su totalidad y detallar aquellas características faltantes en los casos de pruebas anteriores. En este caso de prueba se verá que se obtienen recursos transcodificados y acaparados en diversos formatos.

Tabla 4-14 Subcaso 5.1: visualización de recursos acaparados y transcodificados.

<i>Objetivo:</i>	Ver que el GAP muestre un recurso que se encuentre acaparado y lo muestre correctamente.
<i>Descripción:</i>	Se realizaron varias peticiones de recursos Web los cuales se tenían en caché.
<i>Resultado esperado:</i>	Obtener recursos Web desde la caché local sin que ocurra ningún problema.
<i>Resultado obtenido:</i>	Se obtuvieron diversos recursos Web como páginas Web (en distintos formatos), imágenes y otros elementos dentro del patrón del sitio Web acaparado en la caché local.

En la Figura 4.14, se muestra el proceso de acaparamiento de manera asíncrona. El dispositivo móvil realiza una petición para obtener un sitio Web acaparado, si el sitio está acaparado en el GAT MA se le envía el archivo comprimido del sitio en caso contrario, si existe un patrón de acaparamiento se desconecta al cliente, realiza el proceso de acaparamiento y hasta que nuevamente el cliente se vuelva a conectar envía el sitio acaparado.

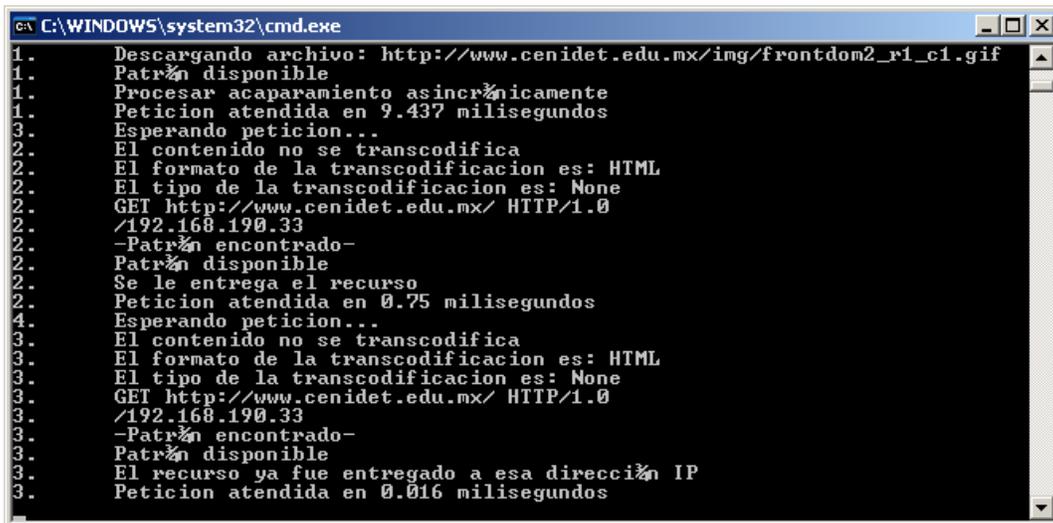


Figura 4.14 Proceso de acaparamiento de manera asíncrona.

Tabla 4-15 Subcaso 5.2: obtener y validar un sitio Web acaparado en el GAP.

<i>Objetivo:</i>	Obtener un sitio Web acaparado en el dispositivo móvil y validar que cumpla con las restricciones.
<i>Descripción:</i>	Se realizaron varias peticiones a diversos recursos Web de los cuales se tenían disponibles patrones de acaparamiento. Si el sitio Web acaparado existe, el GAP descarga dicho sitio, lo descomprime, valida las restricciones impuestas a los recursos, actualiza su lista de patrones y la caché del dispositivo local.
<i>Resultado esperado:</i>	El GAP puede descargar sitios Web acaparados remotamente

para actualizar su caché.
Resultado obtenido: En el 75% de los casos se pudo descargar y descomprimir el sitio Web acaparado. En algunas ocasiones se presentan errores sobre todo si el archivo es demasiado grande, no se comprimió bien (por ejemplo, un archivo no se pudo descargar y generó un entrada ZIP errónea), el nombre del archivo es muy grande, entre otras. Estos problemas se presentan dado que se está utilizando un componente de terceros, ya que no existe un componente en .NET CompactFramework que realice compresión/descompresión de archivos ZIP.

En la Figura 4.15, en el lado izquierdo se muestra un mensaje de notificación cuando se presenta una desconexión. En el centro de la misma figura, se muestra recursos Web que se encuentran acaparados en la caché local del dispositivo; mientras que en el lado derecho de la misma figura, se muestra un mensaje informativo cuando el recurso no está en la caché y no puede obtenerse en línea debido a que el sistema se encuentra en modo desconexión.



Figura 4.15 Visualización de recursos Web acaparados cuando existe conexión y cuando no.

En la Figura 4.16, se muestra la validación de recursos al descomprimir un sitio Web acaparado. En este sentido, se valida que no sobre pasen en número, en tamaño por recurso, en tamaño por sitio y en tamaño total de la caché. En caso de no cumplir, los recursos son eliminados de la caché.

4.7 Otras pruebas

Se realizaron algunas pruebas en vista de obtener y calcular un mejor promedio de los beneficios obtenidos. A continuación se muestra los resultados de dichas pruebas.

4.7.1 GAT en otros dispositivos móviles heterogéneos

En las siguientes imágenes se muestran la versatilidad de nuestro para ejecutarse de manera total en dispositivos basados en Windows: PCs de escritorio, dispositivos empotrados con Windows CE, Smartphones y otros dispositivos móviles cuyo sistema

operativo sea Windows Mobile; así como también puede ejecutarse de parcial en una gran variedad de dispositivos móviles.

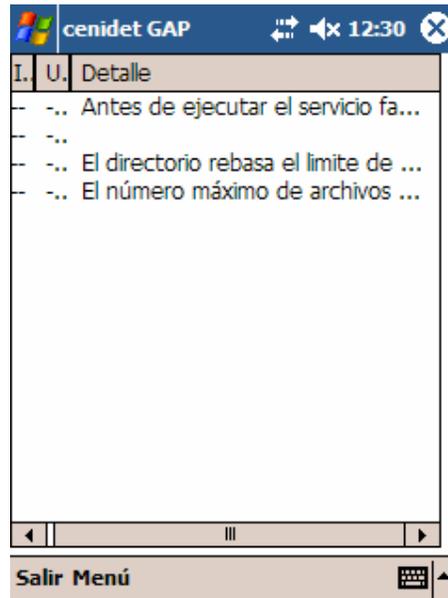


Figura 4.16 En el lado izquierdo, se muestran los mensajes mostrados al realizar la comprobación de restricciones al descargar un sitio Web acaparado.

En la Figura 4.17, se muestra la estructura que se obtiene al transcodificar un documento. Al principio del documento se muestra los elementos texto (párrafos) de la página. Posteriormente se muestran todas las imágenes agrupadas en una sola sección, para finalmente mostrar todas las imágenes en forma resumida.

Con este procedimiento obtenemos un documento más fácil de leer y accesible en los diferentes dispositivos móviles, aunado a que se reduce el tamaño de los recursos, se obtiene por consiguiente un tiempo de respuesta mucho más rápido que con llevan a una reducción significativa de costos.

En la Figura 4.18, se muestran la transformación del mismo recurso Web original a diversos formatos, visualizado a través de diversos dispositivos. El uso de diversos formatos permite que más clientes puedan visualizar el recurso, ya que de forma tradicional no pudieran soportar el formato del documento.

4.7.3 Pruebas de rendimiento

El objetivo de esta prueba fue medir la calidad del prototipo desarrollado para poder apreciar de manera más tangible y cuantitativa los beneficios que se pueden obtener con este trabajo de tesis. Las pruebas se realizaron sobre el tiempo de transcodificación, acaparamiento, caché, tamaño de los recursos entre otras cuestiones que a continuación se describen.



Figura 4.17. Visualización de un recurso Web en HTML reformateado.



Figura 4.18. Transcodificación de un recurso Web en diferentes formatos y plataformas.

Se debe recordar que el GAP ejecutándose en plataformas Windows tradicionales recibe el nombre de WinGAP y en dispositivos Smartphone recibe el nombre de SmartGAP. Esto simplemente para diferenciarlos ya que se trata del mismo programa con pequeñas variaciones en su interfaz dependiendo de la plataforma.

4.7.3.1 Pruebas de transcodificación y acaparamiento

El objetivo de estas pruebas fue conocer el rendimiento que tiene el mecanismo acaparador así como cuantificar sus beneficios. Se realizaron varias pruebas con el prototipo desarrollada para evaluar su desempeño. Las variables que nos interesa conocer son el desempeño de la batería, tiempos de procesamiento, así como el tamaño de los recursos.

Las pruebas consistieron tomando en cuenta un conjunto de 100 sitios Web tomados a partir de una encuesta por correo electrónico a usuarios promedio. Para algunos casos de prueba este conjunto se redujo hasta 29 sitios dado que no se pudieron transformar muchos recursos Web.

Para realizar las pruebas se tomaron como referencia:

- Una laptop HP Pavilion dv1000 con un microprocesador Intel Centrino a 1.7 Ghz con 512 MB RAM.
- Un Smartphone con Windows Mobile 2003, 32 MB de RAM y procesador ARM a 300 Mhz.
- Una PPC Compaq iPAQ H3630 Pocket PC 2000 con 32 RAMRAM y procesador StrongARM a 210 Mhz.
- Una PPC HP Jornada 5400 Pocket PC 2002 32 MB RAM y procesador ARM a 210 Mhz
- Una HP iPAQ rx3115 con 64 RAM y procesador ARM a 300 Mhz.

Contestaron la encuesta 12 personas donde solamente 3 personas han utilizado algún dispositivo móvil para acceder a la Web. Sólo dos personas accedieron por curiosidad mientras que una persona utiliza su dispositivo móvil en moderadas ocasiones.

De estos sitios todos se pudieron visualizar con nuestro desarrollo tanto en PC como en una Pocket PC.

El tamaño promedio de la página principal es de 30,476.81 (aprox. 30 Kb.), los cuales contienen un promedio de 56.99 objetos (imágenes, otros recursos).

Se formularon las siguientes hipótesis de investigación:

H_1 = El proceso de transcodificación se realiza en un tiempo similar al que se tardaría en ver un recurso normal, por lo que el usuario no percibe el hecho de que se está realizando transcodificación de documentos.

H_2 = Con el uso del prototipo y en general del acaparamiento se disminuye considerablemente el consumo de energía del dispositivo, al trabajar de manera

asíncrona y en modo de desconexión; consumiendo menor energía al no tener que acceder los recursos a través de la Web constantemente.

H_3 = Con la transcodificación se obtienen en general recursos Web de menor tamaño ya que al quitar y reformatear algunos elementos de la página se reduce su tamaño.

H_4 = Con el acaparamiento se reduce considerablemente el tamaño de los sitios Web al realizar un recorte del sitio Web en base a un patrón de acaparamiento.

H_5 = El tiempo de acceso a recursos Web se reduce significativamente al tener recursos acaparados en los dispositivos móviles.

Para H_1 se tomaron los tiempos de acceso a los sitios que se pudieron transcodificar obteniendo los siguientes tiempos promedios de transformación (ver Figura 4.19). La estadística más importante es que sólo 29 sitios (páginas principales) se pudieron transcodificar. Este problema se detalla a fondo en las pruebas de accesibilidad.

Los resultados de transformación son desde que el GAT MT recibe la solicitud y devuelve el recurso solicitado transcodificado. El promedio de transformación de página fue de 2,173.6 ms (2.2s) que si se compara con el valor del GAT MT de 1,589.53 existe una diferencia de 36.74% (584.07 milisegundos) más lento al visualizar un recurso transformado.

Con respecto a la transformación pudo observarse que es proporcional al tamaño de la página principal (entre más pesada se requiere mayor tiempo invertido para la descarga del recurso y del procesamiento del documento). También se observó que entre más elaborado es la plantilla de transformación mayor es el tiempo de la transcodificación. En general el uso de XSL-FO (PDF y PostScript) es más lento que el uso de XSLT, esto debido a la complejidad del formato de visualización (el PDF y PS requieren de mayor cantidad de elementos para desplegar la información), dado que el tiempo de extracción de los elementos, descarga, etc. es similar.

Se pudo determinar que el uso de transcodificación no es tan lento (en dispositivos móviles el tiempo de acceso promedio es de más de 3 segundos [ver pruebas de tiempos de acceso, sección 4.7.3.2]), además gracias a que se cuenta con un mecanismo de caché el proceso de transcodificación sólo se realiza una vez. Por este motivo, H_1 es cierto.

Para H_2 se tomó primeramente el consumo de energía en los dispositivos móviles. El tiempo de visualizar los 100 sitios Web requiere de un tiempo aproximado de 52 minutos. Las características que tenían los dispositivos móviles para estandarizar las pruebas consistían en no tener ningún proceso de usuario ejecutándose (sólo procesos de sistema, navegador Web y GAP), el brillo de la pantalla a máxima intensidad, las opciones de ahorro de energía e hibernación deshabilitadas, y la interfaz de red encendida. Una vez obtenido el consumo de energía sin visualizar sitios Web, se procedió a registrar el consumo de energía funcionando el sistema pero sin hacer uso de acaparamiento y posteriormente con acaparamiento, obteniendo los siguientes incrementos en el consumo de energía (ver Figura 4.20).

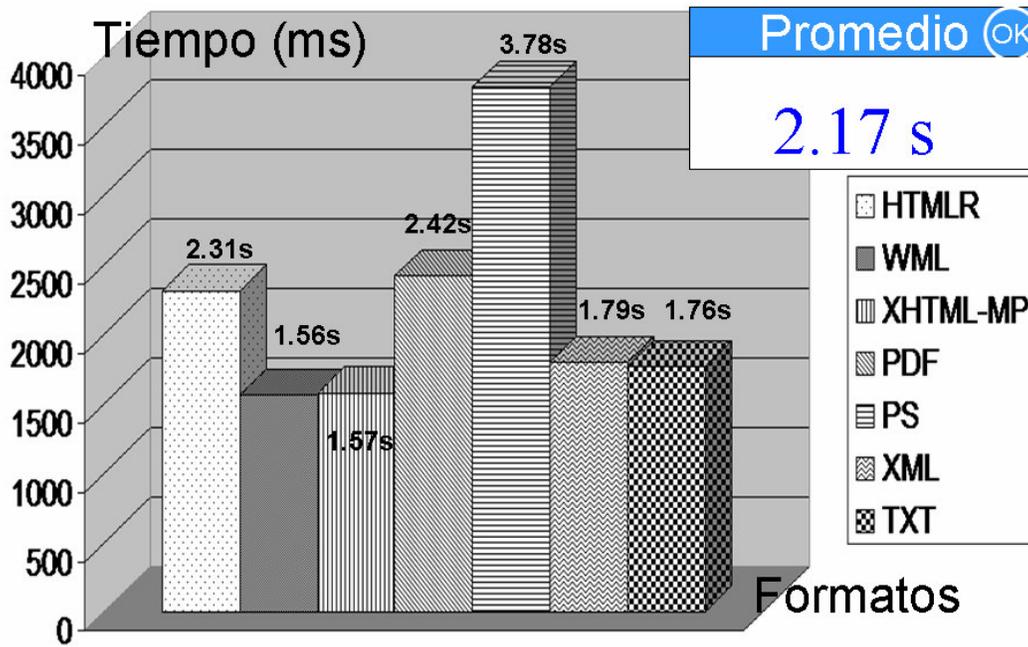


Figura 4.19 Tiempos obtenidos de la transcodificación de recursos Web.

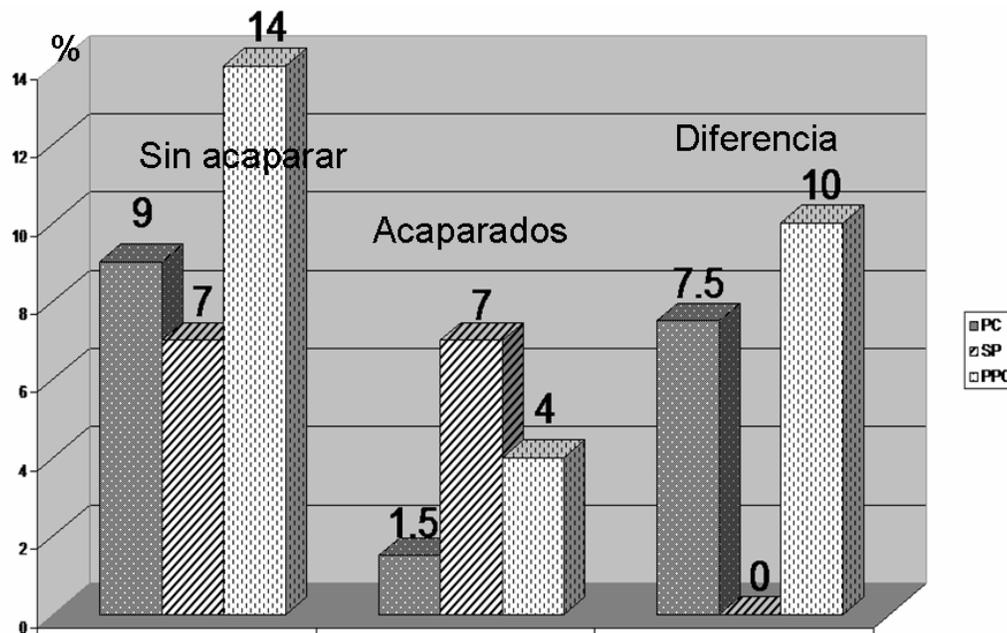


Figura 4.20 Porcentaje de ahorro de energía utilizando dispositivos móviles diversos a través del GAP.

Por lo que al obtener un promedio de las diferencias obtenidas se obtuvo que la diferencia en el ahorro de energía fue de 8.75% menor usando acaparamiento que sin utilizarlo. Por lo que H_2 es cierta.

Con respecto a H_3 , se procedió a transcodificar recursos Web en diferentes formatos para ver el tamaño de los recursos. Los resultados obtenidos se muestran en la Figura 4.21.

Se observa que existe una disminución considerable en el tamaño de los recursos transformados, no sucede lo mismo en algunos casos en especial en el formato PS y XML en donde debido a errores de transformación el tamaño de tres recursos se disparó enormemente influyendo sobre los resultados, por lo que al final se decidió omitir los resultados obtenidos para ese formato y no tener un sesgo mucho mayor. En promedio se obtuvo un promedio de 33.9% de reducción del tamaño del recurso transcodificado. Por lo que H_3 es cierta.

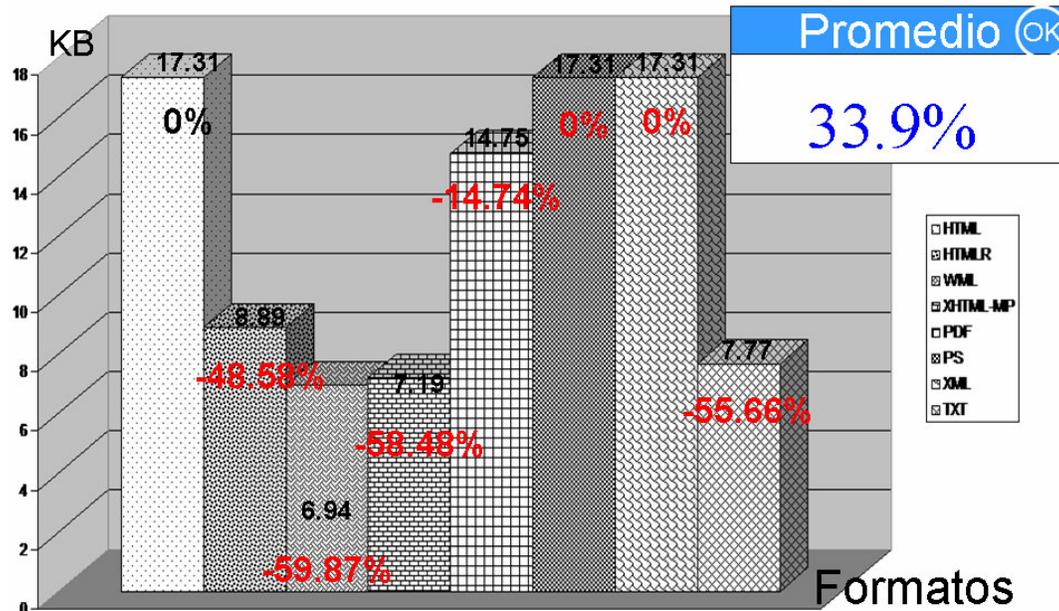


Figura 4.21 Tamaño y porcentaje de ahorro obtenidos al transcodificar recursos Web.

Para H_4 se realizó el acaparamiento en base a patrones obtenidos. Obteniéndose los resultados mostrados en la Figura 4.22.

Gracias al acaparamiento un sitio Web en promedio puede reducirse un 34.85%, y si a eso le aunamos que utilizamos el algoritmo de compresión .ZIP, se pueden obtener mejoras hasta de un 86.62%. En base a estas pruebas se concluye que H_4 es cierta.

Para H_5 se tomaron los tiempos que se registraron en la bitácora del GAP al tener sitios Web acaparados en el dispositivo. El promedio de acceder a un recurso Web en línea es de 2509.76 milisegundos, mientras que obtener los mismos recursos pero en la caché del dispositivo requirió 368.82 ms, obteniendo un acceso 85.30% más rápido. Por lo que se demuestra que H_5 es cierta.

Se observó que se habían realizado pruebas en base al tamaño de los sitios Web acaparados (Figura 4.22), pero no en base al tiempo en que tardan en acaparse, por lo que se procedió nuevamente a modificar el sistema para registrar los tiempos de acaparamiento. Los resultados de dicha prueba se muestran en la Figura 4.23.

Los tiempos se muestran en milisegundos por lo que el tiempo promedio del acaparamiento sin transcodificar (TAHTML) es de 22.46 segundos, mientras que el tiempo de acaparamiento de un sitio Web que ha sido transcodificado en formato HTMLR (THMLR) es de 63.29 segundos lo que lo hace un 281.79% más lento.

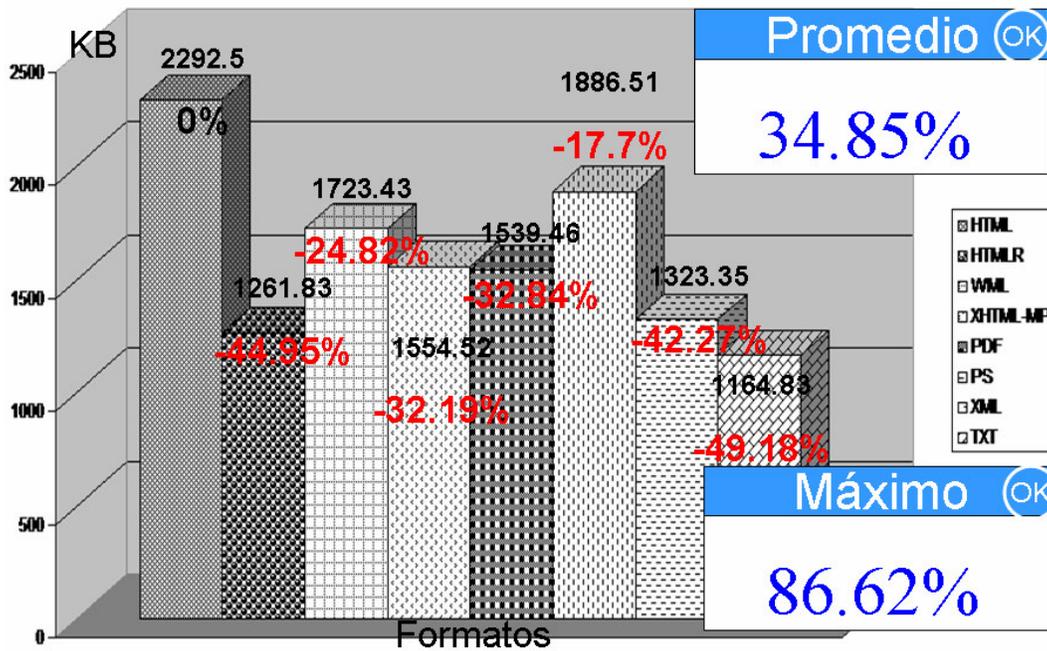


Figura 4.22 Tamaños y porcentaje de ahorro obtenidos al acaparar sitios Web.

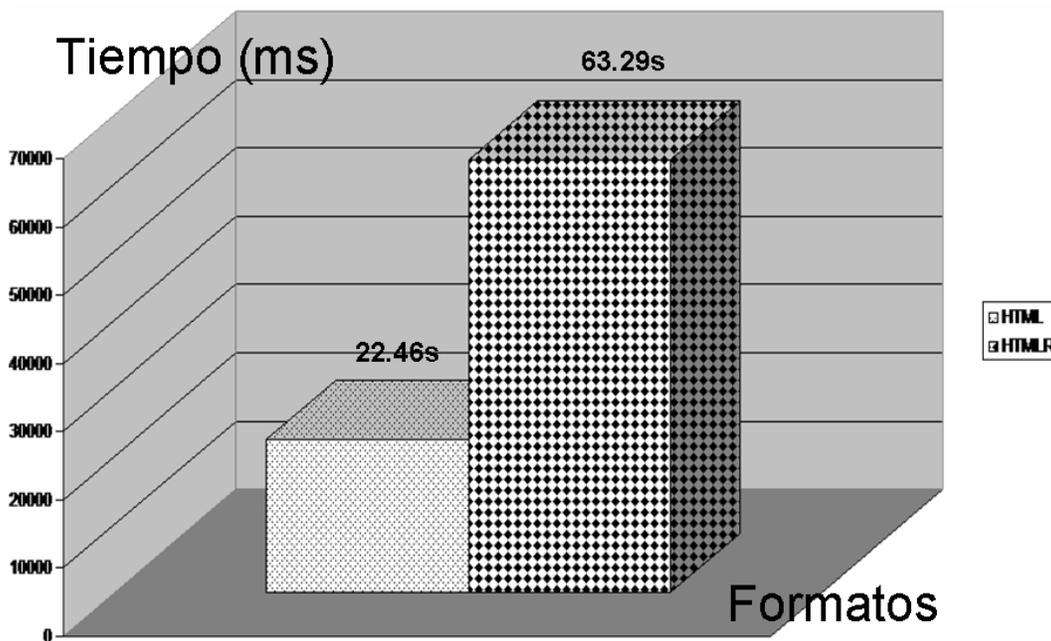


Figura 4.23 Tiempos de acaparamiento obtenidos con y sin transcodificación.

Las pruebas de accesibilidad surgieron de la necesidad de conocer por que el mecanismo transcodificador tenía un porcentaje tan bajo de transcodificaciones exitosas. Se descubrió que el problema principal es que las páginas Web no están correctamente escritas (HTML no es tan formal como XML) por lo cual no son accesibles en dispositivos móviles. En lo que respecta a la accesibilidad, de los 100 sitios visitados se obtuvieron las siguientes estadísticas (ver Figura 4.24 y Figura 4.25).

En dichas estadísticas, podemos ver que son pocos los sitios Web que cumplen con la normatividad WCAG (Web Content Accessibility Guidelines, guías de accesibilidad del

contenido Web) [wcag06] del W3C (WorldWideWeb Consortium) para garantizar el acceso a páginas Web.

La WCAG propone estructurar el contenido Web internamente en el código de marcado para que las páginas puedan ser accesibles con alguna discapacidad. En ese sentido, podemos ver a los dispositivos móviles como “dispositivos con capacidades limitadas”, en donde por ejemplo un dispositivo móvil puede ser ciego al no mostrar imágenes en pantalla, hacerlo en blanco y negro o hacerlo en pocos colores. También podría ser sordomudo al no poder desplegar sonidos, etc.

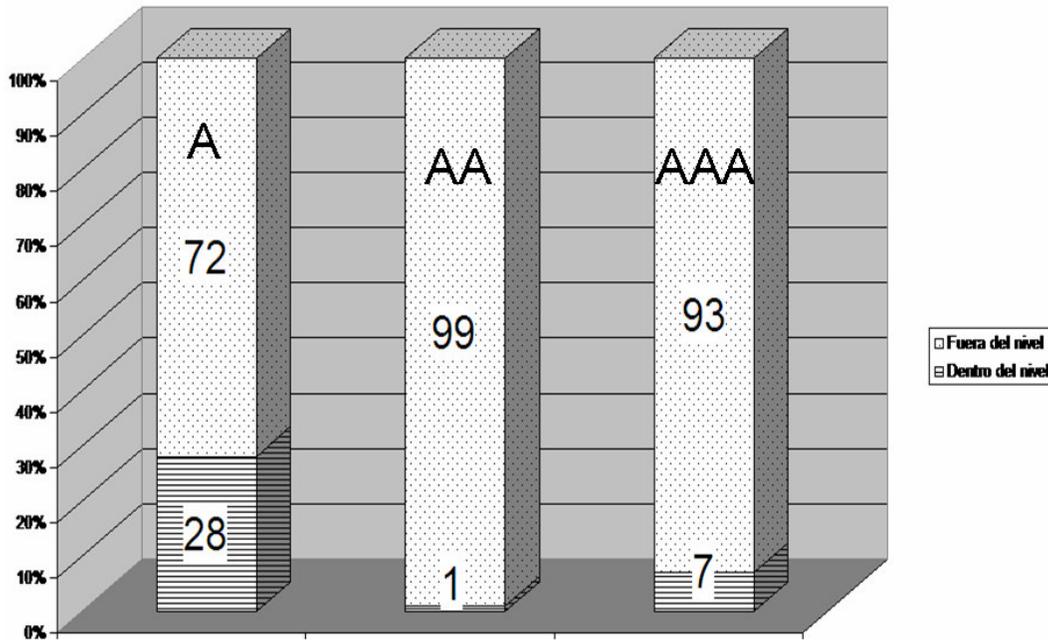


Figura 4.24 Porcentaje de páginas que cumplen con los niveles de accesibilidad WCAG 1.0.

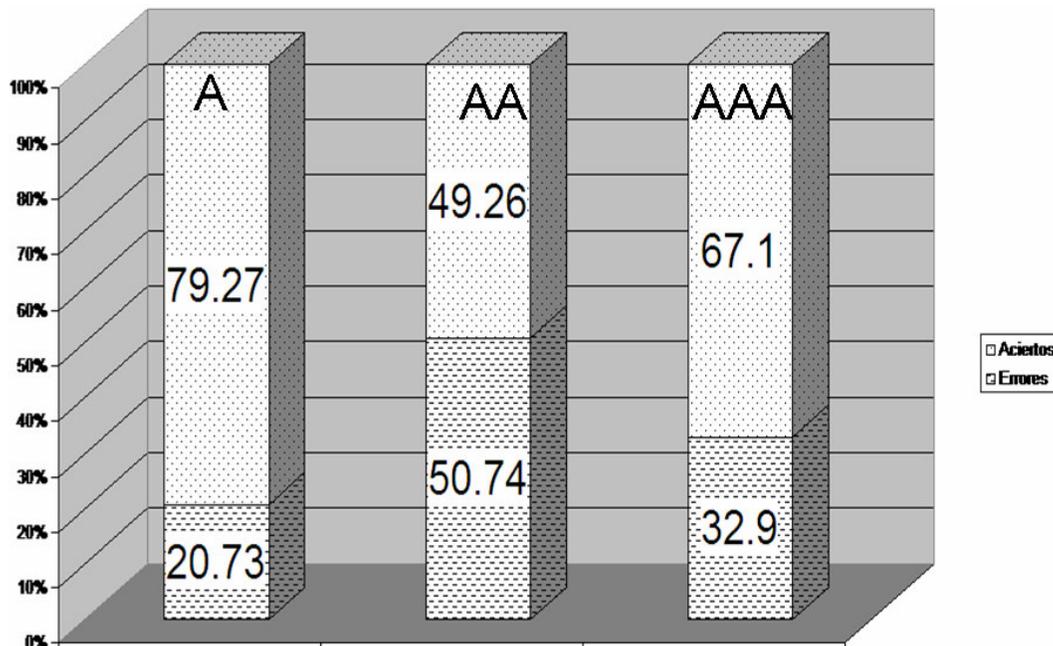


Figura 4.25 Porcentajes de eficiencias en el cumplimiento de los niveles de accesibilidad WCAG 1.0.

LA WCAG propone tres niveles de certificación de accesibilidad, si una página Web cumple con algunas ellas podrá portar un logo distintivo en su página. En el nivel A, los desarrolladores TIENEN que cambiar dichas recomendaciones sobre su sitio Web para hacerlo accesible. En el nivel AA, se DEBEN satisfacer dichos cambios; mientras que en el nivel AAA, PUEDEN satisfacerse los cambios.

Para que se reciba la certificación AA se debió haber pasado las guías del nivel A y del nivel AA. Para recibir la certificación AAA se debió haber pasado las normatividades del nivel A, AA, y AAA.

4.7.3.2 Tiempos de acceso del GAP en dispositivos móviles

Esta prueba surgió del hecho de que queríamos conocer que tan eficiente es el GAP en diversos dispositivos, emuladores y plataformas móviles; así como de determinar cuanto tiempo tarda en mostrarse un recurso Web en los dispositivos móviles.

El caso de prueba consistió de la ejecución del GAP en diversas plataformas Pocket PC (emuladores y dispositivos) registrando el tiempo de acceso al obtener el recurso principal de 100 sitios Web diferentes. Los tiempos se obtuvieron a través del análisis la bitácora del sistema. Antes de realizar las pruebas se plantearon las siguientes hipótesis:

H_6 = El tiempo de acceso a través de emuladores es mucho más lento que a través de un dispositivo normal, debido a que se emula el proceso.

H_7 = El tiempo de acceso a través de emuladores es mucho más rápido debido a que no existe latencia en los tiempos de acceso ya que al ejecutarse de manera local no se ve sometido a factores externos que pudieran afectar la comunicación y el acceso a los recursos.

H_8 = El tiempo de acceso se ve afectado por el tipo y versión de plataforma utilizada; es decir, plataformas más recientes tienen accesos más rápidos.

H_9 = El tiempo de acceso a los recursos se ve afectado por las características del dispositivo; es decir; entre más recursos posea el dispositivo mayor es la posibilidad de que se ejecute más rápido.

H_{10} = La velocidad del tiempo de acceso es superior con dispositivos Pocket PC más recientes.

H_{11} = El uso de la nueva versión de .NET CF 2.0 tiene mejores resultado en la velocidad de acceso que la versión 1.0.

H_{12} = El uso de la misma versión del sistema operativo pero con dispositivos de diferentes capacidades de cómputo produce una mejora en los tiempos de acceso.

H_{13} = El uso independiente del GAP y el cliente Web (navegador) en otro dispositivo mejora los tiempos de acceso al tener menos aplicaciones corriendo en la Pocket PC (sistema dedicado).

Los resultados de las pruebas utilizando emuladores se muestran en la Figura 4.26. En donde:

- ePPC03* Emulador PPC 2003,
- ePPC03PE* Emulador PPC 2003 Phone Edition,
- ePPC03SE* Emulador PPC 2003 Second Edition,
- eWince5* Emulador de Windows CE 5.0,
- eWince4.2* Emulador de Windows CE 4.2 (Windows CE .NET),
- PPC03SE* Dispositivo PPC 2003 Second Edition (HP iPAQ rx3115)

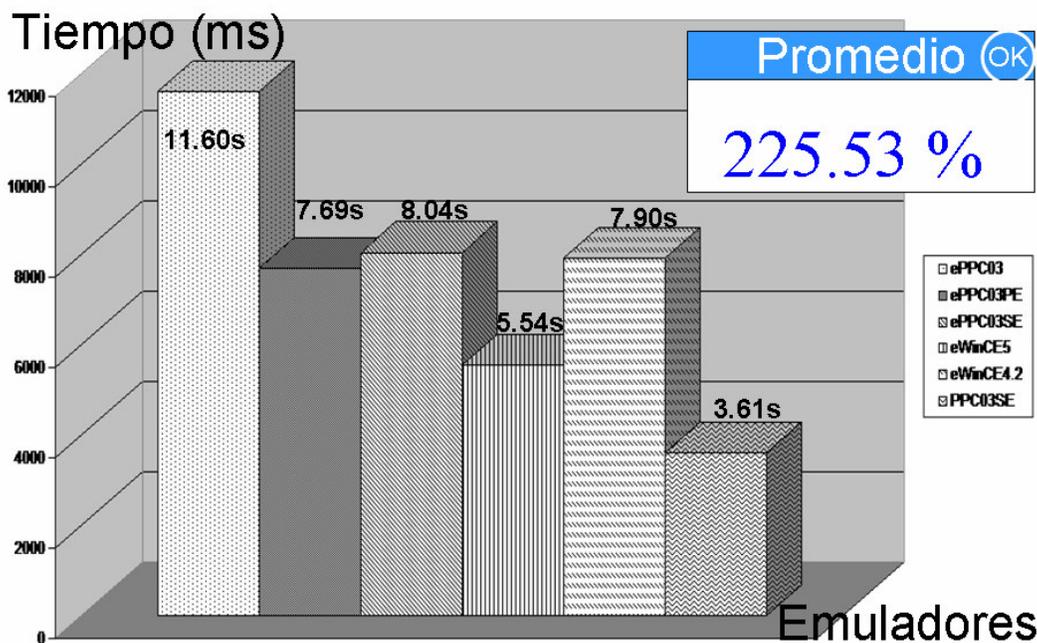


Figura 4.26 Tiempos de respuesta del GAP en base a emuladores.

Los resultados de la prueba utilizando dispositivos PPC se muestran en la Figura 4.27. En donde:

- PPC00* Dispositivo Pocket PC 2000,
- PPC03SE* Dispositivo Pocket PC 2003 SE,
- PC.NETCF2* Dispositivo PC con .NET Compact Framework 2.0,
- PC.NETCF1* Dispositivo PC con .NET CF 1.0,
- PPC03NCF2* Dispositivo Pocket PC 2003 SE con .NET CF 2.0,
- PPC03SE/PC* Dispositivo Pocket PC 2003 SE corriendo el GAP (servidor) y una PC actuando como cliente.
- PPC03SE1* Dispositivo Pocket PC 2003 SE con más memoria (128 en RAM, HP iPAQ rx3715)

El tiempo promedio de acceso del GAP en los emuladores fue de 8,160.158 milisegundos, mientras que el de un dispositivo físico fue de 3618 milisegundos. Por lo

que, la ejecución en emuladores es 225.53 % más lenta que un dispositivo físico. Esto demuestra que H_6 es cierta.

H_7 resultó ser falsa ya que los resultados de las pruebas indican que no se disminuye el tiempo de acceso. Dicho tiempo depende en cierta medida de las capacidades de la PC en donde se corre los emuladores.

H_8 resultó ser cierta ya que la versión más reciente (Windows CE 5.0) es más rápida que las demás versiones de los emuladores. La versión del emulador de Pocket PC 2003 SE es mucho más rápida que su contraparte del emulador de Pocket PC 2003.

H_9 resultó ser cierta. Entre más capacidad de memoria principal, almacenamiento y procesador tenga un dispositivo móvil o emulador, mayor son sus prestaciones. Por ejemplo, puede observarse que la versión del emulador de Pocket PC 2003 Phone Edition es superior a la versión estándar del emulador, esto se debe principalmente que al ser una versión de Pocket PC con un módulo llamado radio (que es lo que lo hace un teléfono) tiene en general mayor prestación y capacidades de almacenamiento y procesamiento que la versión estándar. El uso de memoria persistente en Windows Mobile 5 (Windows CE 5.0) con lleva a mejor rendimiento tal y como se muestra en los resultados, siendo en esta plataforma el mejor tiempo obtenido dentro de los distintos emuladores.

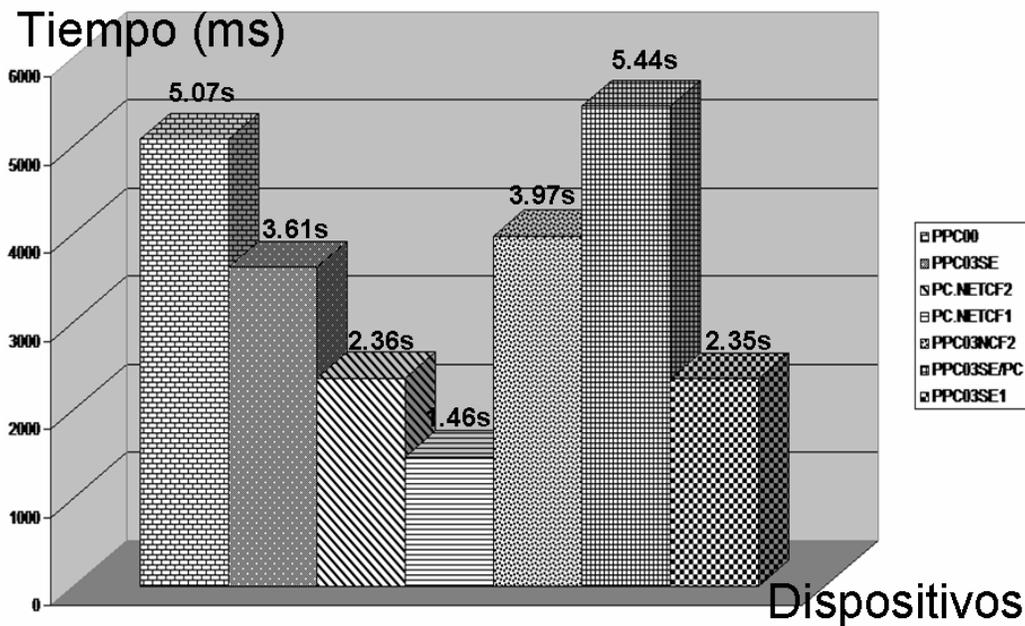


Figura 4.27. Comparativa de los tiempos de respuesta del GAP en diversos dispositivos Pocket PC.

De los resultados obtenidos, se demuestra que en un dispositivo Pocket PC 2000 es 71.29 % más lento que un dispositivo Pocket PC 2003 SE, lo que representa H_{10} se considere como cierta.

Para resolver H_{11} , la prueba se dividió en dos partes, por un lado en dispositivos tradicionales de cómputo PC y por otro con dispositivos PPC.

Haciendo el análisis cuantitativo de los resultados en el dispositivo PC, se aprecia que la versión 2 es más lenta, haciendo un análisis más detallado de los datos se apreció que tienen tiempos casi idénticos sólo varían en algunas peticiones donde la segunda versión tardó mucho más.

En el análisis cuantitativo de dispositivos Pocket PC se aprecia que en los datos obtenidos no existe una gran variación o mejora en los tiempos de acceso, por lo que H_{11} se rechaza. Esto se debe a que necesita optimizar la aplicación para que saque provecho a las nuevas características de la nueva versión del Framework. El ligero incremento en el tiempo de acceso se debe a que .NET CF 2 consume más recursos en el dispositivo dejando menos espacio para las aplicaciones.

Al realizar las pruebas en un dispositivo Pocket PC con 128 en RAM se obtuvo un resultado de 65.11% mejor que usando un dispositivo con 64 Mb., por lo que H_{12} se aceptó como válida.

Para H_{13} se obtuvo que los tiempos de acceso son muy similares con la salvedad de algunas peticiones que se dispararon demasiado. Por lo que se concluye que no existe una mejora sustancial por lo que H_{13} se rechaza. Lo que ayuda el separar el GAP y el navegador es para poder visualizar de mejor manera los procesos que se ejecutan en una Pocket PC, ya que en ésta sólo se puede visualizar una ventana a la vez.

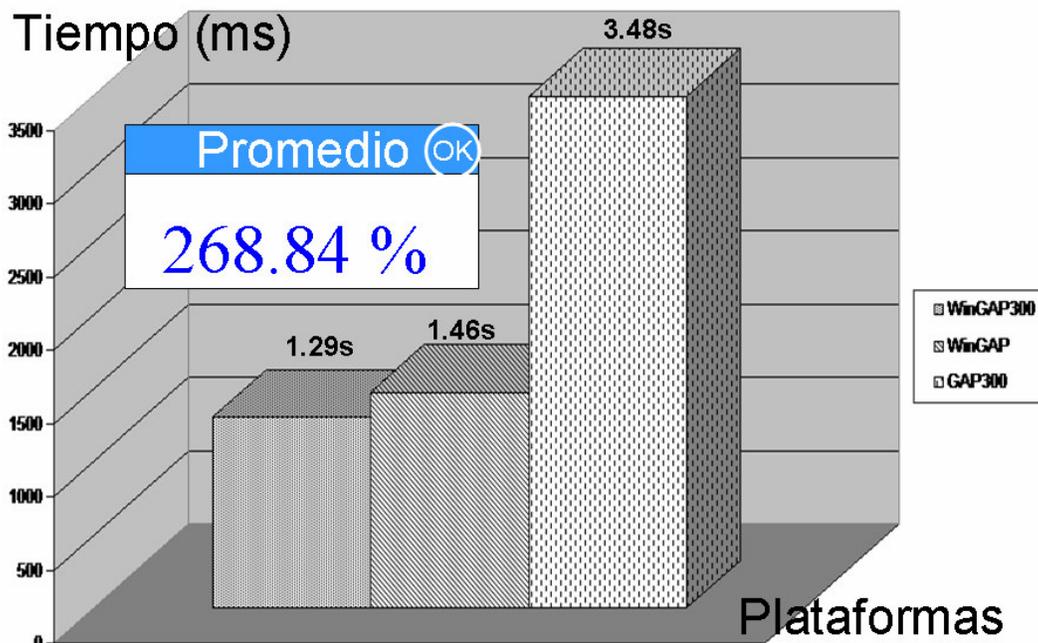


Figura 4.28 Comparativa de tiempos promedios de procesamiento en el GAP en plataformas PC y PPC.

Posteriormente, se realizó otra prueba con el objetivo de ver que tan semejante son los valores del GAP en una PPC y PC con las mismas características de hardware. Se realizaron pruebas comparativas entre una PC de escritorio y una Pocket PC con las mismas características: procesador a 300 Mhz (Centrino en PC, ARM en PPC), 64 MB RAM, 64 ROM en PPC y 10 GB en disco duro en la PPC.

Se tenía instalado en ambos equipos la versión 1.1 de .NET Framework en la PC (ensamblada) con sistema operativo Windows Milenium (ME) y la versión 1.0 de .NET Compact Framework en un PPC HP iPAQ RX3115 con Windows Mobile 2003 SE (Second Edition).

Se midieron los tiempos de acceso del GAP en esta PC64/300 utilizando el mismo corpus de sitios Web empleados en las pruebas anteriores. Se tenía ejecutándose en ambos dispositivos nuestro prototipo denominado GAP (WinGAP). Las pruebas del WinGAP dieron los siguientes resultados (ver Figura 4.28).

De los resultados obtenidos se pudo observar que existe una diferencia abismal entre utilizar un dispositivo PC a un Pocket PC a pesar de tener las mismas características básicas en cuanto a hardware, pero no en arquitectura. Los tiempos obtenidos en la PC con 64 Mb. RAM y procesador a 300 Mhz. Son 268.84 % más rápido (o visto de otra forma el GAP en PPC es 37.19% más lento). Esto se debe a que parte de la memoria RAM en la PPC se utiliza como almacenamiento de programas y datos, dejando en configuración normal menos capacidad de almacenamiento. De hecho, de los 64 MB de memoria RAM disponible en el dispositivo móvil, sólo son accesibles 56 (8 MB son utilizados para el sistema operativo), y de esos 56 restantes de manera predeterminada la mitad (en este caso 28) se utilizan para almacenamiento de datos y programas y la otra mitad se utilizan para memoria RAM tradicional. Afortunadamente, dichos valores son configurables y en este caso, se configuró la memoria lo más que se pudo para RAM exclusivo (53 Mb. aprox.). De allí la necesidad de ejecutar el programa y utilizar almacenamiento externo como tarjetas SD, CompactFlash, etc.

Por otra parte, se puede apreciar que la utilización de una computadora de escritorio con menos capacidad de procesamiento tuvo mejores resultados que con una de mayor capacidad (Procesador Intel Centrino 1.7 Ghz, 512 MB RAM, 40 GB de disco duro). Esto se debe básicamente a las velocidades de conexión; es decir, influye más el tiempo de latencia para la obtención del recurso de la Web que el tiempo de procesamiento de las peticiones. En grandes volúmenes de peticiones, se hubiese notado la diferencia de utilizar un equipo más reciente a uno más obsoleto (sobrecarga del servidor Proxy).

También se realizaron pruebas con dispositivos Smartphone para evaluar su desempeño. Los resultados obtenidos se muestran a continuación. Las hipótesis que se plantearon fueron las siguientes:

H_{14} = Los tiempos de respuesta del GAP en dispositivos Smartphone son similares al de dispositivos PDA y no difiere bastante con respecto a dispositivos tradicionales (PC).

H_{15} = El uso de emuladores de dispositivos Smartphone es más lento que su contraparte de utilizar dispositivos reales, dado que los procesos son emulados.

H_{16} = Los tiempos de respuesta del GAP en plataformas más recientes son mejores debido a que cuentan con mayor capacidad de memoria y procesamiento (microprocesadores más veloces).

Las pruebas se realizaron sobre dispositivos Smartphone Windows Mobile, así como emuladores y otras plataformas. Se obtuvieron los siguientes resultados (ver Figura 4.29). En donde:

<i>eSP03</i>	Emulador de Smartphone 2003,
<i>ePPC03PE</i>	Emulador de Pocket PC 2003 Phone Edition,
<i>eSP03SE</i>	Emulador de Smartphone 2003 Second Edition,
<i>SP03SE</i>	Dispositivo Smartphone 2003 Second Edition,
<i>PPC05/PC</i>	Dispositivo Pocket PC 2005 como servidor y una PC como cliente.
<i>PPC05/PPC03SE</i>	Dispositivo Pocket PC 2005 como servidor y dispositivo Pocket PC 2003 Second Edition como cliente
<i>PC</i>	PC de escritorio

H₁₄ es falsa por las siguientes razones. Como puede apreciarse los tiempos de respuesta obtenidos utilizando plataforma Smartphone (dispositivos y emuladores) es de 9,109.4 ms por 3,882.66 en PPC y 1,467.7 en PC. Lo cual representa una diferencia abismal.

H₁₅ es verdadera. Como puede apreciarse en los tiempos obtenidos, el uso de emuladores lleva consigo una penalización importante en cuanto al tiempo de respuesta. Con esto se demuestra que para ciertas aplicaciones el uso de emuladores no es del todo bueno para ciertas aplicaciones.

H₁₆ es verdadera. Como puede apreciarse, el uso de versiones más recientes trae como resultado tiempos de respuesta más cortos.

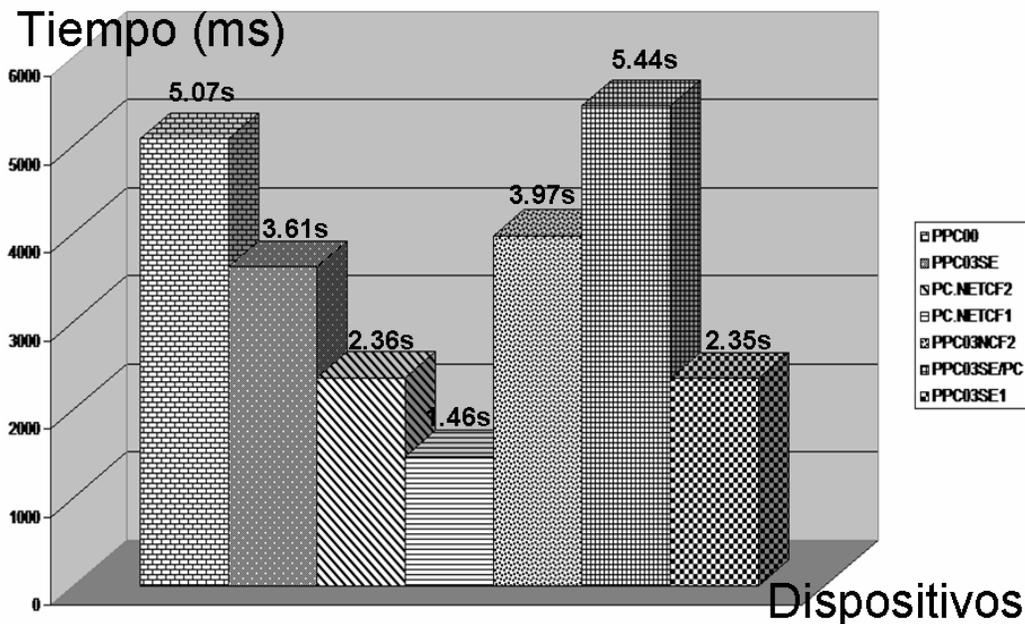


Figura 4.29 El desempeño de equipos Smartphone es inferior al de Pocket PC.

CAPÍTULO 5

CONCLUSIONES

“La imaginación es más importante que el conocimiento”
Albert Einstein

5.1 Conclusiones

Con el presente trabajo se tiene una plataforma de software que permite visualizar sitios Web sin importar las características del dispositivo móvil. Entre estas características se encuentran las limitantes de la pantalla, de la memoria, del procesador, así como de los enlaces de comunicación.

El esquema tradicional de intercambio de información en ambientes distribuidos (modelo cliente-servidor) no funciona de manera adecuada en dispositivos móviles, dado que se implementa un esquema síncrono interactivo altamente consumidor de tiempo. Por el contrario el esquema asíncrono no-interactivo es el que mejor se adapta a las nuevas características de dichos dispositivos. En esta tesis se implementó dicho modelo mediante el acaparamiento de sitios Web, el cual, nos permite garantizar que los usuarios puedan acceder a los recursos de la Web sin importar el estado de la conexión.

Además, este prototipo es capaz de visualizar los recursos de la Web en forma transparente para los usuarios, dado que es posible transformar en línea documentos Web a otros formatos como WML, XHTML-MP, PDF, XML, Postscript, texto plano y HTML reformateado para dispositivos móviles; obteniendo resultados aceptables.

Gracias a estos mecanismos, además de garantizar la correcta visualización de sitios Web a través de dispositivos móviles, es posible ahorrar costos ya que el tamaño de los recursos se reduce gracias al acaparamiento y la transcodificación, se tiene un ligero ahorro de la energía de las baterías de los dispositivos, así como se pueden agilizar tiempos de acceso hasta un 85% gracias a la caché local en el dispositivo móvil.

Los dispositivos móviles son ya una realidad en nuestro entorno, todos indican un crecimiento ascendente. Sin embargo, presentan muchas deficiencias que aún no se han solucionado por lo que hay muchas áreas de oportunidad para realizar investigación en este tema.

En algunos casos se tiene considerado a los dispositivos móviles como simples agendas o juguetes, cuando en realidad tiene una potencia mucho mayor para realizar algunas actividades. Con esta tesis se demuestra que *es viable* el desarrollo de aplicaciones de tipo servidor (servidores Web, servidores de correo electrónico, servidor FTP, etc.) en dispositivos móviles ya que se desarrolló un servicio intermediario denominado GAP.

De los resultados obtenidos se llegó a la conclusión de que existe todavía mucho por hacer para lograr que la Web sea realmente ubicua, es decir accesible por cualquier persona desde cualquier dispositivo. También se observó que la Web es un fenómeno muy complejo de medir dado que intervienen muchos factores que afectan su desempeño.

En la actualidad son cada vez menos frecuente los eventos de desconexiones, pero debido a la naturaleza del medio físico de transmisión inalámbrico, dichos eventos siempre se presentarán y será necesario tomar está y otras consideraciones cuando se realicen aplicaciones en dispositivos móviles. Las desconexiones son una norma en ambientes móviles y lejos de enfocarse en tratar de eliminarlas completamente se debe hacer frente a los eventos de desconexión.

5.2 Aportaciones

La implementación de un servidor Proxy denominado GAP que se ejecuta en dispositivos móviles con sistema operativo Windows CE (Pocket PC y Smartphone). Este servidor es único en su tipo ya que los trabajos relacionados ejecutan toda la lógica del proceso en plataformas tradicionales

La adaptación de un mecanismo de transcodificación para soportar múltiples formatos Web (HTML reformateado, WML, XHTML-MP, PDF, PS, XML y texto plano). Hasta el momento, no se ha encontrado una herramienta que realice dicha transcodificación de la forma y formatos en como la realizamos.

La adaptación de los mecanismos de acaparamiento y transcodificación para que puedan trabajar de manera asíncrona.

La integración y adaptación de diversos mecanismos del proyecto Moveware para tener una plataforma prototipo que sea capaz de visualizar sitios Web sin importar las limitaciones y características de los dispositivos móviles.

Evaluación de tecnologías de frontera como son los dispositivos móviles en lo referente a herramientas de programación, sistemas operativos, hardware y funcionamiento interno de dichos dispositivos.

5.3 Artículos publicados

Del presente trabajo de tesis se publicaron los siguientes artículos:

J. Carlos Olivares R., J. Gabriel González S., Víctor J. Sosa S. y Azucena Montes R., *“MoviWeb: Platform to Solve the Web Content Visualization Problem on Heterogeneous Mobile Devices”*, aceptado en el XV Congreso Internacional de Computación CIC’06, México D.F., México, noviembre 21 al 24 de 2006.

J. Carlos Olivares R., J. Gabriel González S., Víctor J. Sosa S. y Azucena Montes R., *“Evaluation of Pocket PC Devices for its Use as Mobile Servers”*, aceptado en el XIII

Congreso Internacional de Investigación en Ciencias Computacionales CIICC'06, Tampico, Tamaulipas, México, del 15 al 17 de noviembre de 2006.

J. Gabriel González S., Víctor J. Sosa S., Azucena Montes R., J. Carlos Olivares R., “*Using Web Pages Accessible Design for the Correct Web Visualization on Mobile Devices*”, aceptado en el III Congreso de Electrónica, Robótica y Mecánica Automotriz CERMA'06, Cuernavaca, Morelos, México, del 26 al 29 de septiembre de 2006. Memoria disponible de manera impresa ISBN: 0-7695-2569-5, Sección Communications I, pp. 37-42. Aceptado entre los 10 mejores artículos del congreso por lo que se publicará en la revista IEEE Latinoamérica.

J. Gabriel González S., Víctor J. Sosa S., Azucena Montes R., J. Carlos Olivares R., “*Multi-Format Web Content Transcoding for Mobile Devices*”, aceptado en el VII Encuentro Mexicano de Computación ENC'06, San Luís Potosí, San Luís Potosí, México, del 18 al 22 de septiembre de 2006. Memorias disponible de manera impresa ISBN: 0-7695-2666-7 e ISSN: 1550-4069, sección 5, pp. 109-115.

J. Carlos Olivares R., J. Gabriel González S., Azucena Montes R., Víctor J. Sosa S., I. Rafael Ponce M., “*GAP: Una Herramienta para Solucionar el Problema de la Visualización de Contenidos Web en Dispositivos Pocket PC*”. Traducción al inglés de “*GAP: A Tool for Solve The Web Content Visualization Problem in Pocket PC devices*”, revista electrónica IEEE Looking Forward, IEEE Computer Society, vol. 13, verano de 2006.

J. Carlos Olivares R., J. Gabriel González S., Azucena Montes R. y Víctor J. Sosa S., “*Control de desconexiones en la visualización de páginas Web con dispositivos móviles Pocket PC*”, XVI Congreso Interuniversitario de Electrónica, Computación y Eléctrica CIECE'06, Ciudad Obregón, Sonora, México, del 5 al 7 abril de 2006 (memoria disponible en CD).

J. Gabriel González S., Azucena Montes R., Víctor J. Sosa S., J. Carlos Olivares R., “*Arquitectura de una Caché para Almacenar sitios Web en Dispositivos Móviles Pocket PC*”, V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'06), Puebla, Puebla, México, del 1 al 3 de Febrero de 2006 (Memoria disponible de manera impresa, ISBN: 970-94770-0-5, Sección 9: Base de Datos y Minería de Datos pp. 263-270).

J. Gabriel González S., Azucena Montes R., Víctor J. Sosa S., J. Carlos Olivares R., “*Evaluación de Windows CE y Linux Embedded sobre Plataformas iPaq Pocket PC Modelos 3600*”. “6to. Simposium Iberoamericano de Computación e Informática SICI'05”, Monterrey, Nuevo León, México, del 15 al 17 de Noviembre de 2005. (memoria disponibles en CD y de manera impresa con ISBN: 968-5823-21-9).

J. Gabriel González S., Azucena Montes R., Víctor J. Sosa S., J. Carlos Olivares R., “*Gestor de acaparamiento de sitios Web transcodificados para plataforma Pocket PC*”. Publicado en el “3er. Congreso de Innovación e Investigación y Desarrollo Tecnológico CIINDET'05” Celebrado en la ciudad de Cuernavaca, Morelos, México, del 28 al 30 de septiembre de 2005 (memoria disponible en CD).

J. Gabriel González S., Azucena Montes R., J. Carlos Olivares R., “*Comparativa y evaluación de las herramientas de programación para desarrollar aplicaciones en dispositivos Pocket PC*” publicado en el “6to. Congreso Internacional de las Ciencias Computacionales CICC’05”, celebrado en la ciudad de Colima, Colima, México, del 28 al 30 de septiembre de 2005 (memoria disponibles en CD).

5.4 Premios y otros resultados

Durante el desarrollo de este trabajo de tesis se obtuvieron los siguientes logros:

1. Primer lugar en el concurso de creatividad de los institutos tecnológicos en su fase local. Participación en un programa de televisión por cable local referente a la exposición de este proyecto.
2. Aceptación por parte de la IANA (Internet Assigned Number Authority, Autoridad Asignadora de Números en Internet) [iana06] del puerto 10800 del servicio GAP lo que le da validez a este trabajo de tesis, ya que es un reconocimiento a aplicaciones importantes de servicios en Internet. La IANA es el organismo de regular los puertos que están disponibles en Internet (65535 puertos) siendo los primeros 1024 (0-1023) reservados para aplicaciones ampliamente utilizadas en Internet, como es el caso de los protocolos HTTP (80), SMTP (25), SSH (22), etc. se muestra la relación de dichos puertos (última actualización 5 de octubre de 2006).

Tabla 5-1 Selección del puerto 10800 del GAP por parte de la IANA.

```
# 10289-10799 Unassigned
gap 10800/tcp Gestor de Acaparamiento para Pocket PCs
gap 10800/udp Gestor de Acaparamiento para Pocket PCs
# Juan Carlos Olivares Rojas <jcolivares04c@cenidet.edu.mx> March 2006
# 10801-10804 Unassigned
```

5.5 Trabajos futuros

En lo referente al proceso de transcodificación se tienen en mente los siguientes trabajos futuros:

Se puede mejorar el proceso de transcodificación con respecto a las imágenes al comprimirlas, cambiándoles el tamaño y la resolución para que ocupen menos espacio. Es algo difícil de implementar pero incrementaría el desempeño de navegación (aunque tal vez lo haría más lento). Se debe recordar que la transferencia de imágenes representa el 52% de la carga en Internet [ami06].

Otra actividad que se podría implementar es la forma de dejar los archivos de plantillas por separado del código para esta manera hacer más extensible la conversión de documentos (se tendría que modificar también la forma de acceder a la plantilla). La ventaja de realizar esto, es que no se tendría que modificar directamente la plantilla desde código con la compilación necesaria para que los cambios se vean reflejados. Esta actividad es altamente recomendable hacerla.

Una mejora sustancial sería tratar de leer cualquier documento Web y transformarlo a otro. Este problema tiene bastante complejidad ya que se necesita crear herramientas que lean para un formato y transcodifiquen en varios. Esto es posible por que la transformación se realiza a través de XSL, lo cual permite transformar cualquier documento en XML a cualquier otro documento basado en XML.

Es evidente que la mayoría de las páginas Web tienen un diseño estructural inadecuado, motivo por el cual no pueden ser accesibles por cualquier persona e independientes del dispositivo. Por este motivo, es de suma importancia realizar un mecanismo de que en cierta manera reestructure el contenido de la Web de tal forma que sea accesible y pueda visualizarse de manera correcta.

Siguiendo con el diseño correcto y accesible para dispositivos móviles se sugiere como trabajo futuro la creación de un editor Web especialmente diseñado para dispositivos móviles que cumplan con las normas de accesibilidad para dispositivos móviles del W3C denominado mobileOK [mok06] que aun se encuentra en borrador. Esto por que la mejor solución para atacar un problema es desde la raíz.

En lo referente al proceso de acaparamiento se proponen los siguientes trabajos futuros:

Como trabajos futuros se tiene la implementación del módulo GAP en otras plataformas haciendo uso de la tecnología J2ME, esto con la finalidad de abarcar mayor cantidad de plataformas móviles, tal y como se ilustra en la Figura 5.1.

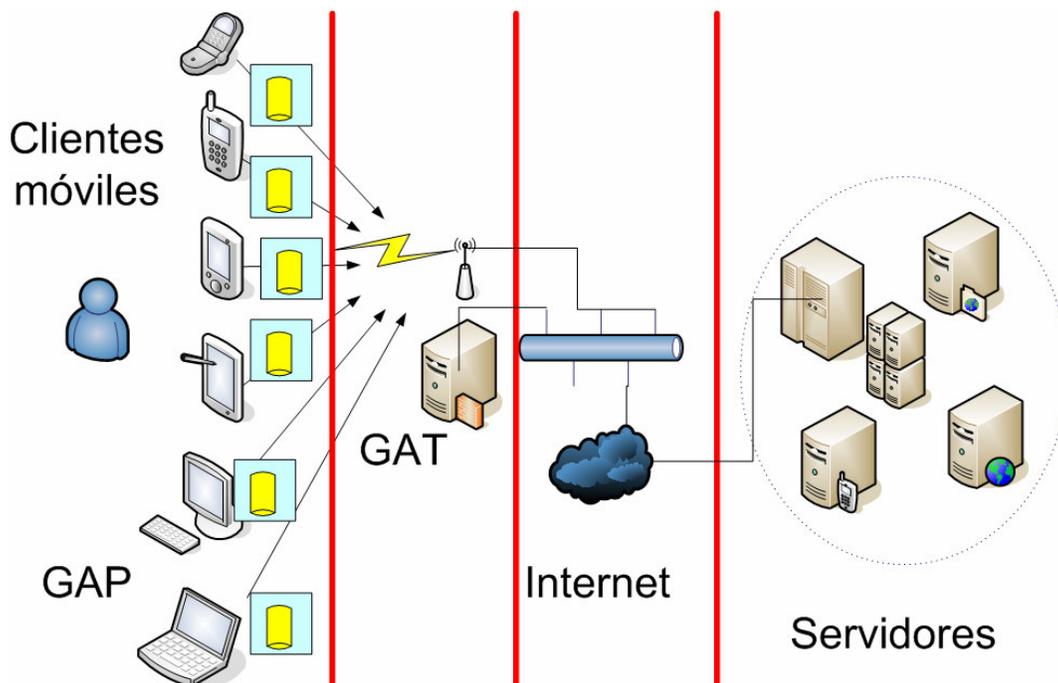


Figura 5.1 Implementación del GAP en J2ME para ejecutarse en más plataformas móviles.

Otro trabajo que se pretende realizar consiste el manejo de un mecanismo totalmente asíncrono para recibir sitios Web usando tecnología SMS/MMS. En este trabajo, el usuario introducirá en un mensaje SMS la URL, el formato de transformación, entre

otras características para posteriormente recibir en un mensaje SMS la petición a su página transcodificada.

Otro trabajo futuro interesante consiste en diseñar un nuevo mecanismo para la identificación de patrones de acceso que sea mucho más eficiente y sobretodo en tiempo real para eliminar las limitaciones que actualmente tiene este proyecto.

Dado que nuestro prototipo GAP cuenta con una bitácora se propone como futura mejora la modificación del prototipo para que interactúe con otros GAP en un esquema de servidores proxys caché cooperativas pero con dispositivos móviles. Esto traería como consecuencia el intercambio de recursos acaparados entre dispositivos móviles que estén en cercanía o en una misma red.

Otro trabajo futuro con respecto a las bitácoras en los dispositivos móviles, consiste en la creación de un minero sobre dispositivos móviles que interprete dichas bitácoras. La idea es factible ya que existen bases de datos en dispositivos móviles, además de que las bitácoras no son tan grandes si se comparan con las de un servidor. Además se pueden utilizar algunos manejadores de base de datos como SQL Server Mobile u Oracle Lite.

Otra posible mejora que existe en lo referente a los esquemas de caché consiste en hacer un esquema híbrido. Por ejemplo, en el GAP, en lugar de poseer una caché con elementos acaparados y transcodificados se propone una caché auxiliar que vaya almacenando los recursos Web que el usuario a visitado (caché normal). En lo que respecta al GAT MT, se propone incluir otro sistema de caché en el cual se almacenen los recursos Web que el usuario ha visitado para que en posteriores visitas sea más fácil obtenerlo.

Otra posible mejora consiste en la implementación de un módulo de configuración que permita al usuario seleccionar los recursos Web que quiera descargar al dispositivo móvil a través de realizar un seguimiento de sus actividades. Dichos recursos se proponen que se envíen a través del proceso de sincronización del dispositivo.

ANEXOS

Anexo A Costos de acceso a Internet a través de redes de telefonía celular

El acceso a Internet a través de un dispositivo móvil usando tecnología celular es considerablemente caro. En México, los costos con el mayor proveedor de telefonía celular (Telcel) son de \$1.5 en prepago o \$1 en plan tarifario con tecnología CSD y con tecnología GPRS cuesta \$0.12 por Kb. o fracción transmitida (se puede obtener un plan de 50 Mb. por \$500), los costos no incluyen IVA. A continuación (Tabla A-1), se muestra una serie de actividades que realiza en promedio un usuario móvil y su costo. Dichas actividades fueron obtenidas de [gprs06], adaptando los costos al mercado mexicano.

Tabla A-1 Costos de acceso a Internet en México desde un dispositivo móvil haciendo uso de la red de telefonía celular.

Tarea	Tamaño (KB)	Tiempo (Segs.)	GPRS	CSD
Login (entrada al sistema)	1.5	27	\$0.24	\$1.5
Leer noticias	2	92	\$0.24	\$3
Buscar una película y ver su sinopsis	3.7	153	\$0.48	\$4.5
Resultados de los partidos del fútbol	5.4	109	\$0.72	\$3
Buscar un numero en un directorio	5.9	100	\$0.72	\$3
Búsqueda de un restaurante y menú	6.3	127	\$0.84	\$4.5
Cargar página Web	6.7	42	\$0.84	\$1.5
Descargar una archivo PDF (68 KB)	72.4	372	\$8.76	\$10.5
Recibir un correo (9 KB)	11.8	74	\$1.44	\$3
Reenviar un correo 9 KB	12.2	74	\$1.56	\$3
Página Web de 70 KB	76.1	455	\$9.24	\$12
Enviar un correo con una nota y un archivo adjunto de 50 KB	81.0	495	\$9.72	\$13.5
Total	285	2120	\$33.12	\$63

Como se puede apreciar, los precios son elevados si se compara con el acceso tradicional pero en algunos casos son convenientes como es el caso de la búsqueda y visualización de algún servicio como en el caso de la cartelera del cine (el usuario no tiene que comprar el periódico [\$5 en promedio], ir al café-Internet [entre \$5 y \$10 por hora] o hablar por teléfono para saber la cartelera [\$1 minuto en teléfono público]). Otro caso sería el de consultar los resultados de partidos de fútbol en donde sale más barato acceder a Internet desde un dispositivo móvil que enviar un mensaje SMS (\$1 + costo del servicio que oscila entre los \$1 y \$5).

También se puede apreciar que en casos donde se requiere mayor contenido o información no es del todo conveniente. Como es el caso de la descarga de archivos o el envío y recepción de correo electrónico. Para este último caso han aparecido esquemas

asíncronos que están tomando mucho auge como el caso de la tecnología Pushmail y los dispositivos móviles Blackberry [bla06].

Haciendo uso del prototipo implementado en esta tesis, es posible obtener los siguientes beneficios (ver Tabla A-2) reales y tangibles para el usuario.

Tabla A-2 Beneficios tangibles de la utilización del prototipo implementado.

Tarea	Tamaño (KB)	Tiempo (Seg.)	GPRS	CSD
Página Web 70 KB	76.1	455	\$9.24	\$12
TRANSCODIFICACIÓN				
Página Web 70 KB	50.22	300.26	\$6.12	\$9
ACAPARAMIENTO				
Sitio Web 70 KB	10.65	63.7	\$1.32	\$3

En el primer caso se muestra los costos que se obtuvieron al visualizar una página de 70 KB haciendo uso de las tecnologías GPRS y CSD sin utilizar nuestro prototipo. En el segundo caso se muestra los costos obtenidos aplicando solamente el mecanismo de transcodificación. Finalmente, en el tercer caso se muestran los valores obtenidos al utilizar el prototipo en su totalidad. Nótese que se están calculando los beneficios tomando en cuenta los casos óptimos tanto de transcodificación (34%) y acaparamiento (86%).

Anexo B Especificación de métodos del GAP

En esta sección se detalla cada una de las clases que conforman el GAP, así como también se describen cada uno de los métodos que componen cada clase.

Tabla A-3 Clases que conformar el prototipo GAP.

MoviWeb.Utilerias	
<i>archivoXML</i>	Clase para leer escribir archivos XML
<i>paraconf</i>	Clase pública que contiene parámetros de configuración para la interfaz gráfica
<i>Plataforma</i>	Clase que se encarga de obtener el tipo de plataforma.
<i>Plataformas</i>	Enumeración para los tipos de dispositivos soportados por .NET CF
<i>Procesos</i>	Clase que se encarga de ejecutar un proceso dentro del sistema operativo.
<i>WinProcesos</i>	Clase que se encarga de gestionar la ejecución de procesos en Windows
<i>Validador</i>	Clase que se encarga de validar datos
<i>Zip</i>	Clase adaptada para descomprimir/comprimir archivos Zip.
MoviWeb	
<i>cAlmacenamiento</i>	Interfaz gráfica de configuración de almacenamiento
<i>Configuracion</i>	Lee un archivo de configuración del sistema en XML.
<i>Créditos</i>	Formulario que muestra la información acerca de créditos de este proyecto
<i>Desconexion</i>	Interfaz gráfica de configuración del módulo de opciones de desconexión
<i>GAL</i>	Gestor de Acaparamiento Local. Encargado de gestionar la caché de sitios Web transcodificados
<i>GAP</i>	Clase principal del Gestor de Acaparamiento para Pocket PC
<i>GAPc</i>	Interfaz gráfica de las opciones de configuración del GAP
<i>GATc</i>	Interfaz gráfica de configuración con las opciones referentes al MoviWeb GAT
<i>GDL</i>	Gestor de Desconexión Local. Encargado de gestionar las desconexiones en los dispositivos móviles.
<i>Log</i>	Bitácora del sistema
<i>main</i>	Clase principal del Sistema
<i>Observador</i>	Clase correspondiente al observador de peticiones
<i>Principal</i>	Formulario principal de ejecución del cenidet MoviWeb GAP
<i>Sincronizador</i>	Clase que se encarga de sincronizar el contenido de la caché en dispositivos móviles.
<i>Singleton</i>	Patrón de diseño para crear una sola instancia de una clase

Tabla A-4 Métodos de la clase archivoXML.

Miembros	Descripción
<i>existeArchivo</i>	Método que determina si existe un archivo o si se puede abrir
<i>leerParametro</i>	Lee un parámetro de un archivo XML
<i>abrirArchivo</i>	Función que se encarga de abrir un archivo XML
<i>escribirArchivo</i>	Escribe un archivo de XML
<i>revisarMatriz</i>	Método que determina las dimensiones de una matriz

	tridimensional
<i>buscar</i>	Buscar en un archivo si existe el parámetro deseado (etiqueta-atributo-valor)
<i>recuperarValores</i>	Recupera los valores de una etiqueta XML dependiente de su posición
<i>validarArchivoXML</i>	Comprueba si un archivo XML es válido sintácticamente (bien formado)
<i>agregarLinea</i>	Método que agrega una línea a un archivo XML antes de que finalice
<i>buscar</i>	Método que se encarga de buscar en un archivo XML si se cumplen dos condiciones
<i>recuperarValores</i>	Método que se encarga de recuperar Valores de un archivo XML basado en dos criterios

Tabla A-5 Métodos de la clase Configuración.

Miembros	Descripción
<i>Configuracion</i>	Constructor de la clase
<i>configurar</i>	Obtiene los parámetros del archivo de configuración
<i>GATIP</i>	Método para acceder a la dirección IP del GAT
<i>GATPuerto</i>	Método para acceder al puerto del GAT
<i>GAPIP</i>	Método para acceder a la IP del GAP
<i>GAPPuerto</i>	Método para acceder al puerto del GAP
<i>directorioCache</i>	Método para acceder al nombre del directorio de la caché
<i>indicePatron</i>	Método para acceder al nombre del archivo contenedor de patrones
<i>estado</i>	Método de acceso a la propiedad conexión del GDL.
<i>transformar</i>	Método que accede a la variable transformador
<i>MAIP</i>	Método para acceder a la propiedad IP del Mecanismo Acaparador
<i>MAPuerto</i>	Método que obtiene la propiedad privada puerto del MA
<i>Espacio</i>	Método que devuelve o asigna el tamaño o espacio de almacenamiento de la caché
<i>Longitud</i>	Método que se encarga de devolver o asignar la longitud máxima de un archivo
<i>Maximo</i>	Método que se encarga de devolver el número máximo de archivos que se pueden almacenar por sitio en la caché
<i>Tipos</i>	Método que se encarga de devolver o asignar el tipo (extensiones) de recursos Web a acaparar
<i>formato</i>	Método que se encarga de poner/obtener el formato de la transcodificación
<i>Reconexion</i>	Método que devuelve el valor de reconexión
<i>Monitorear</i>	Método que devuelve si se debe de monitorear la conexión o no
<i>Tmonitoreo</i>	Método que devuelve el tiempo de monitoreo
<i>Tpeticion</i>	Método que devuelve el tiempo entre peticiones al realizar el monitoreo
<i>Eficiencia</i>	Método que devuelve el porcentaje de eficiencia umbral de las desconexiones
<i>Acaparamiento</i>	Método que devuelve el modelo de interacción del proceso de acaparamiento
<i>Transcodificacion</i>	Método que devuelve el método de interacción del proceso de

	transcodificación
<i>NoCache</i>	Método que indica el valor de si se debe utilizar o no la caché
<i>Tamano</i>	Devuelve el tamaño total disponible para la caché
<i>Bitacora</i>	Devuelve el valor de si se va a utilizar bitácora o no
<i>validar</i>	Método que se encarga de validar los datos

Tabla A-6 Miembros de la clase Desconexión.

Miembros	Descripción
<i>Desconexion</i>	Constructor de la forma de desconexiones
<i>menuItem1_Click</i>	Menú salir. Sale de la aplicación
<i>menuItem2_Click</i>	Menú Grabar. guarda los datos del archivo de configuración
<i>Desconexion_Load</i>	Método que se ejecuta al cargar el formulario, carga los valores del archivo de configuración
<i>guardar</i>	Método que se encarga de guardar las modificaciones a un archivo de configuración
<i>salvar</i>	Método que se encarga de salvar los datos a un medio persistente
<i>modificar</i>	Método que se produce cuando cambia algún valor del formulario
<i>validar</i>	Método que se encarga de validar los datos del formulario

Tabla A-7 Métodos de la clase GAL.

Miembros	Descripción
<i>GAL</i>	Constructor con parámetros de directorio de la caché e índice de patrones
<i>existeCache</i>	Verifica la existencia de la caché en el dispositivo Pocket PC
<i>crearCache</i>	Crea el directorio de la caché
<i>existePatronLocal</i>	Determina si existe un patrón de acaparamiento para el recurso de Web
<i>existeRecurso</i>	Método que determina si existe un recurso en un patrón de acaparamiento
<i>obtenerPatron</i>	Determina si existe un patrón de acaparamiento que se encuentra en el servidor pero aun no ha sido descargado al dispositivo
<i>obtenerInfoPatron</i>	Obtiene información acerca de un elemento del índice patrones
<i>obtenerInfoRecurso</i>	Obtiene información referente a un recurso que se encuentra acaparado
<i>obtenerRecurso</i>	Obtenemos el flujo de bytes del recurso acaparado
<i>mostrarMensaje</i>	Método que devuelve una página de error con el mensaje definido
<i>obtenerTamañoRecurso</i>	Método para obtener el tamaño de un recurso anexando su cabecera de respuesta
<i>descomprimirPatron</i>	Método que se encarga de descomprimir el archivo Patrón obtenido y actualizar su lista patrón de acaparamiento
<i>actualizaListaPatron</i>	Método que se encarga de actualizar el patrón recibido en la lista de patrones. Todo esto con la finalidad de no volver a pedir patrón
<i>tamArchivo</i>	Método que se encarga de determinar el tamaño de un archivo
<i>tamDirectorio</i>	Método que determina el tamaño de un directorio

<i>numeroArchivos</i>	Método que se encarga de determinar el número de archivos que contiene un directorio
<i>archivoPermitido</i>	Método que se encarga de verificar si el archivo se puede guardar en la caché del dispositivo local
<i>actualizaPatron</i>	Método que se encarga de actualizar la lista patrón
<i>tamTipArchivos</i>	Método que se encarga de recorrer cada subdirectorío en vista de validar el tipo y tamaño de cada recurso
<i>generarArbolCortado</i>	Método que se encarga de generar un árbol recortado con los archivos contenidos en el sitio acaparado
<i>obtenerTipoMIME</i>	Método que se encarga de obtener el tipo MIME de un recurso
<i>DirCache</i>	Método que se encarga de obtener el directorio de la caché
<i>obtenerInfoPatron</i>	Método que obtiene la información de un patrón en base a dos valores
<i>NoCache</i>	Método que devuelve si se va a utilizar la caché del dispositivo o no

Tabla A-8 Métodos que conforman la clase GAP.

Miembros	Descripción
<i>GAP</i>	Constructor parametrizado
<i>run</i>	Método principal de atención del proceso
<i>stop</i>	Función que se encarga de parar el servicio
<i>transcodificador</i>	Método que devuelve la propiedad estática de transformación
<i>Formato</i>	Método que se encarga de devolver el formato asignado para la transcodificación
<i>cerrar</i>	Método que se encarga de liberar recursos asociados al GAP
<i>Acaparamiento</i>	Método que se encarga de devolver el método de interacción del proceso de acaparamiento
<i>Transcodificacion</i>	Método que se encarga de devolver el método de interacción del proceso de transcodificación
<i>Bitacora</i>	Método que se encarga de devolver si existe o no la bitácora

Tabla A-9 Métodos de la clase GDL.

Miembros	Descripción
<i>GDL</i>	Constructor para asignar puerto y dirección IP al GDL
<i>inicializacion</i>	Inicializa valores predeterminados
<i>asociarIntermediario</i>	Configura los parámetros del intermediario
<i>estado</i>	Devuelve el estado del sistema. Conexión o desconexión
<i>monitorear</i>	Función que se encarga de monitorear el estado de la conexión
<i>conexion</i>	Revisamos el estado de la conexión
<i>conectar</i>	Método que se encarga de verificar la existencia del GAT
<i>estadisticas</i>	Obtenemos estadísticas para determinar el estado de la conexión
<i>planeación</i>	Método para acceder a desconexiones planeadas
<i>reestablecer</i>	Método que reinicializa los contadores
<i>dirIP</i>	Método para acceder a la dirección del GDL
<i>reconectar</i>	Método que permite recuperarse de una desconexión, en caso de que el usuario se reconecte
<i>MAIP</i>	Método que devuelve coloca la dirección IP del MA

<i>MApuerto</i>	Método que se encarga de devolver o poner el puerto del MA
<i>conectar</i>	Método que se encarga de realizar una conexión y determinar si tuvo éxito o no

Tabla A-10 Métodos de la clase Log.

Miembros	Descripción
<i>Log</i>	Constructor parametrizado
<i>crearArchivo</i>	Método que se encarga de abrir y/o crear el archivo correspondiente a la bitácora
<i>escribe</i>	Método que se encarga de escribir a la bitácora, similar a escribir en línea
<i>cerrar</i>	Método que se encarga de cerrar y liberar recursos de la bitácora
<i>marcaTiempo</i>	Método que devuelve la marca de tiempo en estándar UNIX Se parte de la era UNIX 1 de Enero de 1970
<i>conexion</i>	Revisamos el estado de la conexión
<i>conectar</i>	Método que se encarga de verificar la existencia del GAT
<i>estadisticas</i>	Obtenemos estadísticas para determinar el estado de la conexión
<i>planeación</i>	Método para acceder a desconexiones planeadas
<i>reestablecer</i>	Método que reinicializa los contadores
<i>dirIP</i>	Método para acceder a la dirección del GDL
<i>reconectar</i>	Método que permite recuperarse de una desconexión, en caso de que el usuario se reconecte
<i>MAIP</i>	Método que devuelve coloca la dirección IP del MA
<i>MApuerto</i>	Método que se encarga de devolver o poner el puerto del MA
<i>conectar</i>	Método que se encarga de realizar una conexión y determinar si tuvo éxito o no

Tabla A-11 Métodos de la clase Observador.

Miembros	Descripción
<i>Observador</i>	Asocia un conector al dispositivo
<i>run</i>	Método principal de la clase, corre en forma de hilo
<i>leerPetición</i>	Lee la petición del navegador
<i>procesarEncabezados</i>	Procesa el encabezado de una petición HTTP y la descompone en partes
<i>obtenerRecursoLinea</i>	Función que obtiene el recurso de Web en línea
<i>obtenerRecurso</i>	Método que obtiene la solicitud HTTP desde el GAT MT
<i>obtenerRespuesta</i>	Método que devuelve el recurso solicitado
<i>reenviarPetición</i>	Reenvía la petición del navegador al GAT
<i>cerrarConexion</i>	Cierra la conexión con el navegador
<i>procesarRespuesta</i>	Método que se encarga de procesar la respuesta obtenida
<i>obtenerPatron</i>	Método que se ejecuta en un hilo y sirve para obtener un Patrón de acaparamiento
<i>cambiarUserAgent</i>	Método que cambia la etiqueta UserAgent del encabezado
<i>xTransform</i>	Método que se encarga de agregar un encabezado a la petición HTTP con el siguiente formato: X-Transform: formato donde formato puede ser HTML, WML, XHTML-MP, PDF, etc.
<i>cerrar</i>	Método que se encarga de liberar recursos
<i>procesarPatron</i>	Método que se encarga de descomprimir, actualizar contenedor

de patrones y patrón

Tabla A-12 Métodos de la clase paraconf.

Miembros	Descripción
<i>inicializar</i>	Método que inicializa los valores de configuración
<i>leer</i>	Método que permite leer los parámetros del archivo de configuración y asociarlos a la interfaz

Tabla A-13 Métodos de la clase Plataforma.

Miembros	Descripción
<i>SystemParametersInfo</i>	Método nativo para obtener información acerca del sistema
<i>getPlataforma</i>	Método para obtener la plataforma
<i>plataformaWinCE</i>	Método privado que determina que tipo de dispositivos Windows CE se está ejecutando

Tabla A-14 Métodos de la enumeración Plataformas.

Miembros	Descripción
<i>PocketPC</i>	Plataforma Pocket PC (2000/2002/2003/WM5/PPC Phone Edition)
<i>WindowsCE</i>	Dispositivos Windows CE: Pocket PC, Smartphone, WebPad, Handheld PC, etc
<i>Smartphone</i>	Dispositivos Smartphone: 2002, 2003, Windows Mobile 5 for Smartphone
<i>Win32NT</i>	Windows NT y derivados: NT 4, 2000, XP, 2003
<i>Win32S</i>	Windows 9x: Windows 95, Windows 98, Windows ME
<i>Win32Windows</i>	Otros Windows
<i>Unknown</i>	Plataforma desconocida

Tabla A-15 Métodos de la clase Principal.

Miembros	Descripción
<i>Principal</i>	Ventana principal del GAP
<i>menuItem1_Click</i>	Confirmación de si se desea abandonar la aplicación.
<i>menuItem7_Click</i>	Inicializa el servicio de gestor de acaparamiento
<i>menuItem8_Click</i>	Verifica si se desea parar el servicio o no
<i>actualizaInterfaz</i>	Método que se encarga de actualizar algunos aspectos visuales del programa
<i>menuItem6_Click</i>	Muestra la ayuda del sistema. Quizás sea conveniente documentación en Pocket Word
<i>escribe</i>	Método que muestra gráficamente las acciones del sistema
<i>menuItem5_Click</i>	Muestra la pantalla de créditos
<i>menuItem3_Click</i>	Método que se activa cuando se hace clic sobre el GAP. Tiene la finalidad de habilitar/deshabilitar el menú del servicio
<i>menuItem10_Click</i>	Método que se encarga de probar la conexión con el GAT (MT-MA)
<i>menuItem11_Click</i>	Método que se encarga de enviar al módulo de configuración del GAT
<i>menuItem9_Click</i>	Método que se encarga de mostrar la interfaz gráfica de configuración del GAP
<i>Principal_Load</i>	Método que se invoca cuando se carga el formulario

<i>nArcConf</i>	Método para acceder al nombre del archivo de configuración
<i>menuItem12_Click</i>	Método que se ejecuta al hacer tap sobre Menú > GAP > almacenamiento
<i>detectarPlataforma</i>	Método que obtiene la plataforma actual donde se esta ejecutando el desarrollo
<i>plataforma</i>	Método que se encarga de devolver el tipo de plataforma del sistema
<i>adaptarInterfaz</i>	Método que adapta la interfaz gráfica a las diversas plataformas.
<i>menuItem13_Click</i>	Acción que se ejecuta cuando se accede a Menú > GAP > Cambiar configuración
<i>menuItem14_Click</i>	Método que manda llamar el módulo de configuración de desconexiones
<i>Directorio</i>	Método que devuelve el directorio en donde se esta ejecutando el programa
<i>crearBitacora</i>	Método que crea una bitácora auxiliar
<i>cerrar</i>	Método que se encarga de cerrar la bitácora si es que existe

Tabla A-16 Métodos de la clase Procesos.

Miembros	Descripción
<i>CreateProcess</i>	Envoltura para crear procesos del sistema operativo Windows CE
<i>Execute</i>	Método que se encarga de ejecutar un proceso con sus respectivos parámetros

Tabla A-17 Métodos de la clase WinProcesos.

Miembros	Desconexión
<i>CreateProcess</i>	Envoltura para crear procesos del sistema operativo Windows
<i>Execute</i>	Método que ejecuta un proceso en Windows de escritorio

Tabla A-18 Métodos de la clase Sincronizador.

Miembros	Desconexión
<i>Sincronizador</i>	Constructor parametrizado
<i>inicializar</i>	Método que se encarga de inicializar el sincronizador
<i>enviarPetición</i>	Método que reenvía la petición al servidor de acaparamiento
<i>obtenerPatron</i>	Método que se encarga de determinar si existe un archivo patrón de acaparamiento
<i>obtenerRespuesta</i>	Método para obtener respuesta del recurso acaparado se obtiene "PND" si el patrón no existe o el archivo del sitio acaparado

Tabla A-19 Métodos de la clase singleton.

Miembros	Descripción
<i>instanciaUnica</i>	Instancia única para el GAP
<i>getInstanciaUnica</i>	Método sobrecargado que devuelve una sola instancia de la clase en cuestión

Tabla A-20 Métodos de la Clase Validador.

Miembros	Descripción
<i>textoEnteroPositivo</i>	Método que se encarga de validar un texto como Entero Positivo

<i>textoIP</i>	Método que se encarga de validar una dirección IP dada en forma de texto
<i>textoEnteroRango</i>	Método que se encarga de validar una cadena en un entero comprendido en un rango
<i>enteroPositivoRango</i>	Método que se encarga de validar un entero positivo
<i>formatosTranscodificacion</i>	Método que se encarga de validar los formatos de transcodificación
<i>textoDirectorio</i>	Método que se encarga de validar una cadena de texto como un directorio
<i>textoArchivo</i>	Método que se encarga de validar una ruta como si fuera un archivo válido
<i>tiposArchivos</i>	Método que se encarga de validar una cadena de texto como archivos válidos a acaparar
<i>flotanteRango</i>	Método que se encarga de validar un flotante entre un rango
<i>textoEnteroPositivoRango</i>	Valida una cadena de texto a un número positivo de un rango
<i>textoFlotanteRango</i>	Método que se encarga de validar una cadena de texto a un flotante en un rango establecido

Tabla A-21 Métodos de la clase ZIP.

Miembros	Descripción
descomprimir	Método que se encarga de descomprimir un archivo.
mensaje	Delegado que atiende el proceso de descompresión

Anexo C Archivo de configuración del GAT MT

A continuación (ver Tabla A-22) se muestra el archivo de configuración del módulo cenidet MoviWeb GAT MT. En la Tabla A-23, se detalla la descripción de cada parámetro de configuración.

Tabla A-22 Archivo de configuración del GAT MT.

```
//Archivo de configuración del MoviWeb GAT MT
//Contacto: jcolivares04c@cenidet.edu.mx

//Directorio en donde se ubica la caché
dir_cache          cache

//Dirección IP en donde se ejecuta el Squid u otro Proxy
ip_webserver       192.168.190.33

//Puerto por donde escucha y atiende el Squid u otro Proxy
port_webserver     3128

//IP del GAT MT
ip_servTransformador 192.168.190.33

//Puerto en donde escucha el GAT MT
port_servTransformador 2700

//Días que dura la caché
periodo_cache      5

//Archivo de control de los recursos de la caché
file_cacheXML      cacheXML.xml

//Archivo de salida XML
file_XML            salidaXML.xml

//Archivo de errores generados por el XHTML
file_errXHTML      erroresXHTML.txt

//Archivo de salida en XHTML
file_XHTML          salidaXHTML.html

//Hoja de estilo generada en XSLT
file_XSL            hojaEstilo.xsl

//Archivo HTML intermedio reformateado
file_intermedio     htmlIntermedio.html

//Salida reformateada
file_salidaFormateada salidaFormateada.xml

//Formato de la transcodificación
formato             HTMLR

//Tipo de transcodificación
transcodificacion   Complete

//El sistema detecta automáticamente una gran cantidad de dispositivos
móviles y navegadores.
//Si alguno no se detecta se agrega aquí como posible expansión
```

```
User-Agent WAP | WML | HTML
```

```
//Determina el modelo de interacción del proceso de Transcodificación
1: Asíncrono, 0 síncrono
modelo      0
```

Tabla A-23 Descripción del archivo de configuración del GAT MT.

Atributo	Descripción
<i>dir_cache</i>	Indica el directorio en donde se guardarán los archivos transcodificados. La estructura de este directorio contiene otros directorios que representan cada sitio Web. La estructura de dicha carpeta corresponde al siguiente formato: <i>formato+nombredelsitio</i> . Por ejemplo, el directorio PSwww.itmorelia.edu.mx contiene los archivos transcodificados. Esta caché a diferencia de las caches tradicionales solo contiene archivos transformados. Dentro de la caché, el nombre de los recursos transcodificados tiene el siguiente formato: <i>t- + nombredelrecursosinextension + extensiondeformato-transcodificado</i> . Así por ejemplo, <i>t-index.pdf</i> , indica que el recurso index.html (u otro con los formatos soportados: asp, aspx, jsp, html, php, entre otros) ha sido transformado a formato PDF.
<i>ip_webserver</i>	Dirección IP del servidor Proxy caché Squid u otro servidor Web.
<i>port_webserver</i>	Indica el puerto de atención del servidor Proxy auxiliar.
<i>ip_servTransformador</i>	Es la dirección IP que se le asignará al servidor de transformación.
<i>port_servTransformador</i>	Es el puerto de atención del servicio transformador. Este parámetro de configuración y la dirección IP será necesario saber para configurar el cenidet MoviWeb GAP o en su defecto el navegador de un dispositivo móvil.
<i>periodo_cache</i>	Indica el tiempo en días en que será válido la caché, pasando el tiempo especificado dichos recursos se eliminan.
<i>file_cacheXML</i>	Indica la ruta del archivo en donde se localiza el directorio de la caché. El formato del índice de la caché se muestra en la Tabla A-24. Por cada petición, se registra la dirección URL, la ruta física en donde está almacenado el archivo, la fecha (que corresponde al día en que se hizo el registro) y el formato en que está transformado el recurso.
<i>file_XML</i>	Nombre del archivo XML generado que será utilizado en el proceso de obtención de elementos a través de la tecnología DOM (<i>Document Object Model</i> , modelo de objeto documento). Esta es el área más conflictiva dentro del proceso de transcodificación, ya que si el documento en XML no está bien formado y validado en el proceso anterior (transformación a XHTML) el proceso concluye de forma errónea.
<i>file_errXHTML</i>	Ubicación del archivo en donde se guardan los errores y advertencias del proceso de transformación a XHTML. Este archivo es una versión del documento HTML transformado de acuerdo las restricciones de XML (bien formado y

	válido). Esta es una fase que causa algunos problemas pero en general el proceso de transformación se lleva a cabo.
<i>file_XHTML</i>	Nombre del archivo XHTML generado. La generación del archivo en este formato se hace a través de la API JTIty.
<i>file_XSL</i>	Ruta en donde se ubica el archivo XSL generado. Este archivo contiene la estructura del archivo resultante, que en combinación con los elementos extraídos y las API's necesarias (xalan o FOP) se transcodifica el documento al formato deseado.
<i>file_intermedio</i>	Ruta de la ubicación donde se almacenará el archivo HTML intermedio generado.
<i>file_salidaFormateada</i>	Indica la ubicación de el archivo XML reformateado.
<i>formato</i>	Indica el formato predeterminado en que se va a realizar la transcodificación, este formato se toma en cuenta cuando no se interpreta de manera adecuada el encabezado de transcodificación (X-Transform) o no existe (como en el caso de dispositivos móviles basados en sistemas operativos diferentes de Windows Mobile y que actualmente no soportan el GAP). Los formatos válidos de transcodificación son: HTMLR, WML, XHTML-MP, PDF, PS, TXT y XML.
<i>transcodificacion</i>	Indica el formato de transcodificación predeterminado. El valor recomendado es Partial. Los valores posibles son Complete, Partial y None.
<i>User-Agent</i>	Esta opción de configuración permite indica agentes de usuario (navegadores) opcionales a los que el sistema soporta. El sistema tiene un mecanismo que soporta la mayoría de los dispositivos móviles actuales (teléfonos celulares, PDAs y otros sistemas empotrados). Los valores de este parámetro son opcionales (pueden no ir) y están dado en base a palabras claves como por ejemplo la cadena WAP CHTML aceptará además de los dispositivos móviles que actualmente acepta, aquellos que contienen en su encabezado User-Agent las cadenas WAP y CHTML.
<i>Modelo</i>	Este atributo de configuración señala el mecanismo de transcodificación. El valor recomendado es 0 Síncrono pero puede utilizarse 1 Asíncrono. En el modelo síncrono el cliente tiene que esperar hasta que el recurso sea transformado (en algunos casos pueden ser varios segundos). En el modelo asíncrono el usuario realiza la petición y se desconecta. Tiempo después el usuario se vuelve a conectar obteniendo el status de su petición el cual puede ser procesando o el resultado (página de transcodificación) de la petición.

Tabla A-24 Archivo índice de la caché.

```
<?xml version="1.0" encoding="UTF-8"?>
<Cache>
  <Peticion>
    <Url>http://www.cenidet.edu.mx/subaca/web-dda/index.html</Url>
    <Ruta>C:\MoviWeb\GAT\MT\cache\HTMLwww-cenidet-edu-mx\subaca\web-
dda/index.html</Ruta>
```

```
<Fecha>20</Fecha>
<Formato>HTMLR</Formato>
</Petición>
<Petición>
  <Url>http://www.cenidet.edu.mx/subaca/web-ose/index.html</Url>
  <Ruta>C:\MoviWeb\GAT\MT\cache\HTMLwww-cenidet-edu-mx\subaca\web-
ose/index.txt</Ruta>
  <Fecha>20</Fecha>
  <Formato>TXT</Formato>
</Petición>
<Petición>
  <Url> http://www.cenidet.edu.mx/subaca/web-dda/index.html</Url>
  <Ruta>C:\MoviWeb\GAT\MT\cache\HTMLwww-cenidet-edu-mx\subaca\web-
mec\t-index.xhtml</Ruta>
  <Fecha>20</Fecha>
  <Formato>XHTML-MP</Formato>
</Petición>
</Cache>
```

Anexo D Archivo de configuración del GAP

En este anexo, se muestran los archivos XML que conforman el sistema, como son el archivo contenedor de patrones, un ejemplo de un archivo patrón de un sitio Web y por último, el archivo de configuración del GAP.

A continuación (Tabla A-25) se muestra y describe el archivo de configuración del GAP. Se recomienda configurar el archivo a través de la interfaz, dado que es más sencillo y no es necesario saber el significado de la estructura interna de dicho archivo.

Tabla A-25 Archivo de configuración del GAP.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<MoviWeb>
  <GAT ip="192.168.190.33" puerto="2700" transformador="1"
ipMA="192.168.190.33" puertoMA="1800" formato="TXT"
acaparamiento="1" transcodificacion="0" bitacora="1" />
  <GAP ip="127.0.0.1" puerto="10800" cache="\Tarjeta de
almacenamiento\cache" indice="\Tarjeta de
almacenamiento\patrones\patrones.xml" conexion="1" nocache="1" />
  <ALMACENAMIENTO espacio="6424800" longitud="4096000" maximo="1010"
tipos="*.jpg|*.gif|*.css|*.pdf" tamaño="12400567" />
  <DESCONEXION reconexion="1" monitorear="1" tiempoMonitoreo="3000"
tiempoPetición="300" eficiencia="30"/>
</MoviWeb>
```

La etiqueta GAT hace referencia a los valores de configuración del módulo cenidet MoviWeb GAT. En la Tabla A-26 se describe dicha etiqueta.

Tabla A-26 Descripción de la etiqueta GAT.

Atributo	Descripción
<i>ip</i>	Hace referencia a la dirección IP en donde se ejecuta el módulo de transcodificación GAT MT.
<i>puerto</i>	Hace referencia al puerto en donde escucha el GAT MA.
<i>transformador</i>	Indica el tipo de transcodificación. Los posibles valores son: 0 equivalente a None (no se transcodifica), 1 equivalente a Partial (transcodificación parcialmente), 2 equivalente a Complete (transformación completa).
<i>ipMA</i>	Indica la dirección IP en donde se ejecuta el servicio de acaparamiento correspondiente al GAT MA.
<i>puertoMA</i>	Indica el puerto en donde escucha el cenidet MoviWeb GAT MA.
<i>formato</i>	Indica el formato en el cual se van a transformar los recursos Web si el tipo de transcodificación es Partial o None.
<i>acaparamiento</i>	Indica el modelo de interacción del proceso de acaparamiento, si es 1, indica un modelo de interacción asíncrono; si es 0, indica un modelo de interacción síncrono. Si el modelo de interacción es asíncrono, sólo será asíncrono entre el servidor de acaparamiento GAT MA y el cliente GAP. Las peticiones realizadas entre el GAT MA y GAT MT para la obtención de recursos se realizan siempre de manera asíncrona.
<i>transcodificacion</i>	Indica el modelo de interacción del proceso de transcodificación. Si tiene un valor de 1, indica que el modelo es asíncrono; si es 0, indica

<i>bitacora</i>	<p>un modelo de interacción asíncrono.</p> <p>Indica el tipo de bitácora, si tiene un valor de 0 indica que no existe bitácora, si es 1 indica que se crea la bitácora de acceso a los recursos access.log, si es 2 indica que se utiliza la bitácora access.log y la bitácora de mensajes mensajes.log. En la Tabla A-27, se muestra un ejemplo de bitácora (mensajes.log). La bitácora de acceso (access.log) se detalla en el capítulo de pruebas.</p>
-----------------	---

Tabla A-27 Ejemplo del archivo de bitácora del sistema (mensajes.log).

20/07/2006 06:15:30 p.m.	Bitácora(s) access.log y mensajes.log activada(s)
20/07/2006 06:15:30 p.m.	El contenido se va a transformar Completamente. Formato de la transformación: HTMLR
20/07/2006 06:15:30 p.m.	Modelo de Acaparamiento: Asíncrono no interactivo
20/07/2006 06:15:30 p.m.	Modelo de Transcodificación: Síncrono interactivo
20/07/2006 06:15:30 p.m.	Servidor escuchando en el puerto: 10800
20/07/2006 06:15:30 p.m.	Opciones de configuración de red: Reconexion Activada, Monitoreo activado
20/07/2006 06:15:31 p.m.	Tamaño de la caché: 95.60194 MB Tamaño utilizado: 12.77952 MB. Se va a utilizar el contenido de la caché
20/07/2006 06:15:32 p.m.	No se ha detectado el GAT MA y si no se encuentra activo no se podrán obtener nuevos patrones de acaparamiento
20/07/2006 06:15:33 p.m.	No se ha detectado el GAT MT y si no se encuentra activo no se podrán obtener recursos en línea ni transcodificación de contenidos Web
20/07/2006 06:15:33 p.m.	Desconexión Planeada
20/07/2006 06:22:05 p.m.	Patrón acaparándose asincrónicamente
20/07/2006 06:30:00 p.m.	Patrón acaparándose asincrónicamente
20/07/2006 07:55:46 p.m.	Servicio GAP detenido
20/07/2006 07:55:46 p.m.	Error al crear el hilo #145 de atención para la petición
20/07/2006 07:55:46 p.m.	GAP cerrado

La etiqueta GAP hace referencia a las opciones de configuración del módulo cenidet MoviWeb GAT. La descripción de dicha etiqueta se muestra en la Tabla A-28.

Tabla A-28 Descripción de la etiqueta GAP.

Atributos	Descripción
<i>ip</i>	Hace referencia a la dirección IP donde se ejecuta el sistema. Generalmente tiene el valor de 127.0.0.1 o la dirección IP del dispositivo móvil, pero puede hacer referencia a otro dispositivo móvil o un dispositivo convencional.
<i>puerto</i>	Hace referencia al puerto en donde escucha el servicio GAP, generalmente es el puerto 10800 aceptado por la IANA.
<i>cache</i>	Hace referencia al directorio en donde se ubicarán los recursos Web acaparados. Cabe hacer mención que los directorios en los dispositivos PPC las rutas son absolutas; es decir; es necesario dar la ruta completa para hacer referencia a un objeto en el sistema de archivos. Por esta razón, es aconsejable poner el directorio en la raíz del sistema de archivo o bien en una tarjeta de almacenamiento secundario.

<i>indice</i>	Indica la ruta donde se ubica el archivo contenedor de patrones. Es necesario indicar dicho si se quiere trabajar con recursos acaparados en la caché local. En la Tabla A-29, se muestra la estructura de un archivo contenedor de patrones y en la Tabla A-30, se muestra la estructura de un patrón de un sitio Web.
<i>conexion</i>	Esta opción de configuración indica si su valor es 1 que se trabajará en modo conexión de manera predeterminada; por el contrario, si su valor es 0 indicará que se trabajará en modo desconexión.
<i>nocache</i>	Este parámetro indica si se utilizará la caché (valor 0) o no (valor 1) independiente del estado de la conexión del equipo.

Tabla A-29 Estructura del archivo contenedor de patrones.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<cache>
  <peticion sitio="http://www.cenidet.edu.mx/" patron="cenidet.xml"
  fecha="10/10/2005" formato="PDF"/>
  <peticion sitio="http://www.itcolima.edu.mx/" patron="itmorelia.xml"
  fecha="10/10/2005" formato="WML"/>
  ...
</cache>
```

Tabla A-30 Archivo patrón del sitio Web <http://www.cenidet.edu.mx/>

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<recursos>
  <acaparado nombre="/index.html" ubicacion="index.html" />
  <acaparado nombre="/css/general.css" ubicacion="general.css" />
  ...
</recursos>
```

La etiqueta almacenamiento hace referencia a las opciones de configuración que afectan a la caché local, concretamente con el GAL. La descripción de dicha etiqueta se muestra en la Tabla A-31.

Tabla A-31 Descripción de la etiqueta Almacenamiento.

Atributo	Descripción
<i>espacio</i>	Esta opción hace referencia al tamaño total en bytes disponibles para el almacenamiento de un sitio Web acaparado. Se debe configurar de acuerdo a las restricciones de espacio de almacenamiento del dispositivo por lo que se recomienda ejecutar el prototipo en una memoria de almacenamiento.
<i>longitud</i>	Esta opción indica el tamaño máximo permitido por archivo. Esto se realiza para evitar que unos cuantos archivos saturen la capacidad de los dispositivos.
<i>maximo</i>	Indica el tamaño máximo permitido de recursos Web por sitio acaparado. Si no se cumplen algunas de estas características los recursos serán eliminados.
<i>tipos</i>	Indica las extensiones de archivos que son válidos en la caché del dispositivo móvil. Dicha opción tiene el siguiente formato: *.extension1 *.extension2 ... *.extensionN.

tamaño Hace referencia al tamaño total asignado a la caché del dispositivo local.

La etiqueta desconexión hace referencia a las opciones de configuración con respecto al algoritmo de control y monitoreo de desconexiones, tal y como se muestra en la Tabla A-32.

Tabla A-32 Descripción de la etiqueta Desconexión.

Atributo	Descripción
<i>reconexion</i>	Esta opción si tiene asignado el valor de 1, indica que el sistema pueda conectarse una vez que se haya detectado una conexión y el sistema estuviese en modo desconexión. Un valor de 0, imposibilita la reconexión del dispositivo (útil cuando se desea trabajar en el equipo al 100% desconectado).
<i>monitorear</i>	Indica que el sistema debe realizar una observación de las condiciones de la red para determinar su estado de conexión (si tiene asignado el valor de uno), de lo contrario, no realizará dicha medición (valor de 0).
<i>tiempoMonitoreo</i>	Indica el tiempo que se realizará el algoritmo de control de desconexiones, los valores están dados en milisegundos.
<i>tiempoPetición</i>	Indica el tiempo que habrá entre petición y petición al momento de ejecutar el algoritmo de control de desconexiones.
<i>eficiencia</i>	Indica el umbral que deberá pasar un dispositivo móvil para establecer su estado de conexión. Si es inferior al límite, el dispositivo se encontrará en desconexión; caso contrario estará en conexión.

Anexo E Archivo de configuración del GAT MA

A continuación (Tabla A-33) se muestra el archivo de configuración del módulo cenidet MoviWeb GAT MA, y en la Tabla A-34 se describe cada etiqueta.

Tabla A-33 Archivo de configuración del GAT MA.

```
//Archivo de configuración del GAT MA
//Para más información contacto jcolivares04c@cenidet.edu.mx

//Dirección IP del GAT MA
server_address 192.168.190.33

//Puerto de escucha del GAT MA
listen_port 2700

//Puerto de atención del GAT MA
attention_port 1800

//Formato de transcodificación predeterminado
formato HTMLR

//Tipo de Transcodificación
transcodificacion Complete

//Archivo contenedor de patrones
patrones contenedorPatrones.txt

//Archivo de control de direcciones de recursos entregados
direcciones direccionespcs.txt

//Directorio donde se encuentra la caché
cache cache\

//Directorio donde se ubican los patrones de los sitios (árbol general
y árbol recortados)
árbol patrones\

//Determina si la transmisión de los sitios Web acaparados debe
realizarse de manera asíncrona
modelo 1
```

Tabla A-34 Descripción del archivo de configuración del GAT MA.

Atributo	Descripción
<i>server_address</i>	Hace referencia a la dirección IP en donde se está ejecutando el GAT MT o en su defecto otro Proxy-cache como Squid. En algunas ocasiones se puede utilizar nombres de dominio.
<i>listen_port</i>	Hace referencia al número de puerto utilizado por el servidor (recordar que los puertos van desde el 0 al 65535, donde los primeros 1024 están haciendo referencia a puertos de aplicaciones bien conocidos).
<i>attention_port</i>	Hace referencia al puerto de atención que proporcionará el servicio GAT MA, por lo que para futuras referencias (configuración del GAP se utilizará este puerto y la dirección IP de donde se está ejecutando el servicio).
<i>formato</i>	Hace referencia al formato de transcodificación. Los posibles

	valores que puede tomar este atributo son los diferentes formatos de transcodificación: HTML, HTMLR, WML, XHTML-MP, PDF, PS, XML, y texto plano. Este formato es útil ya que si se desea utilizar el GAT de manera aislada o en dispositivos móviles donde no se ejecuta el GAP, este será el formato de manera predeterminada.
<i>transcodificación</i>	Indica el tipo de transcodificación de manera predeterminada. Los posibles valores de este atributo de configuración son los siguientes: <i>complete</i> , <i>partial</i> y <i>none</i> ; que indican que el recurso se va a transformar completamente (si falla el proceso se envía mensaje de error), si el recurso se va a transformar parcialmente (el recurso se transforma pero si ocurre un error se envía el recurso original), el recurso no se va a transcodificar respectivamente.
<i>patrones</i>	Indica el nombre del archivo de configuración de patrones. Este archivo está en modo de texto plano (*.txt) como se muestra en la Tabla A-35. Dicho archivo sólo nos indica la dirección URL del sitio y la ubicación de su archivo patrón. Dentro de cada patrón el formato que se sigue es el mostrado en la Tabla A-36. Entre corchetes (‘[‘ y ‘]’) se muestra una dirección URL y los archivos que hacen referencia a esa dirección, en este caso los archivos que tiene mayor probabilidad de ser utilizados por el usuario. En este sentido, se necesita tener además del archivo patrón, el archivo general del sitio, el cual se obtiene a través del módulo cenidet Spider. Teniendo estos dos archivos se procede a hacer el recorte del sitio Web; es decir, obtener solo los elementos que se van a replicar evitando dependencias dobles e incluyendo archivos inmersos en las páginas Web como imágenes y otros recursos que a pesar de no estar dentro del patrón son necesarios para el buen funcionamiento del mismo.
<i>direcciones</i>	Indica la ruta en donde se ubicará el archivo de control de recursos acaparados. Dicho archivo es utilizado para evitar enviar el recurso más de una vez al mismo sitio. El formato de dicho archivo se muestra en la Tabla A-37.
<i>cache</i>	Indica el directorio en donde se guardarán los archivos de sitios acaparados. Temporalmente se descargan los archivos contenidos en el patrón de acaparamiento del sitio Web, se comprime dicho archivo siguiendo el formato formato + nombredelsitio (con guiones) + .zip; por lo que el archivo WMLwww-cenidet-edu-mx.zip hace referencia al sitio Web (http://www.cenidet.edu.mx/) acaparado en formato WML. Una vez realizado la descompresión, los archivos descargados son borrados para optimizar el espacio de almacenamiento del servidor.
<i>arbol</i>	Indica el directorio en donde se guardan los patrones de acaparamiento. Se debe recordar que los patrones de acaparamiento se realizan a través de un minero Web (o en su defecto de manera artificial) por lo que este procedimiento no está automatizado y es necesario que el usuario realice este procedimiento.
<i>modelo</i>	Indica el tipo de obtención de los patrones. Puede ser asíncrona representado por el valor 1, o bien, síncrona representado por el valor 0. Cuando el mecanismo es síncrono, la obtención de los recursos acaparados se realiza de manera tradicional; es decir, el

usuario se encuentra todo el tiempo conectado mientras se procesa la petición, esto trae como resultado diversos problemas como una conexión defectuosa (si el proceso de acaparamiento rebasa el límite establecido de timeout), que el equipo este ocioso en espera del archivo, entre otros. Una vez finalizada la petición (sea exitosa o no) se cierra la conexión. En cambio cuando el modelo es asíncrono, el usuario realiza la petición de obtención del proceso de acaparamiento, si el recurso ya se encuentra acaparado se le envía automáticamente, en caso contrario, se desconecta y el servidor trabajando en segundo plano procesa la petición. Cuando el usuario se vuelve a reconectar, se muestra el estado de su petición; en este caso, si el recurso aun no se ha acaparado se envía el mensaje respectivo, en caso contrario se envía el recurso. Este es el modelo predeterminado de acaparamiento, el cual se utilizará cuando no exista o no se pueda interpretar el encabezado de acaparamiento (X-Hoard: Synchronous | Asíncrono). El valor recomendado es asíncrono.

Tabla A-35 Ejemplo de del archivo de configuración de patrones.

[<http://www.cenidet.edu.mx>] , cenidet.txt
 [<http://antares.itmorelia.edu.mx/~fmorales>] , antares-itmorelia-edu-mx-fmorales.txt
 [<http://expertforum.info>] , expertforum-info.txt
 [<http://www.expertforum.info>] , expertforum-info.txt
 [<http://www.acer.com>] , global-acer-com.txt

Tabla A-36 Archivo patrón del sitio <http://www.cenidet.edu.mx/>

[<http://www.cenidet.edu.mx/index.php>]
<http://www.cenidet.edu.mx/alumnos/index.php>
<http://www.cenidet.edu.mx/subaca/web-elec/index.php>
<http://www.cenidet.edu.mx/subaca/web-elec/laboratorio.html>
<http://www.cenidet.edu.mx/seleccion/index.php>
[<http://www.cenidet.edu.mx/alumnos/index.php>]
<http://www.cenidet.edu.mx/subaca/web-elec/index.php>
<http://www.cenidet.edu.mx/seleccion/index.php>
 ...

Tabla A-37 Archivo de control de direcciones del GAT MA.

/192.168.190.33	C:\MoviWeb\GAT\MA/cache\HTMLwww-cenidet-edu-mx.zip
/192.168.190.33	C:\MoviWeb\GAT\MA/cache\HTMLexpertforum-info.zip

Anexo F Instalación del GAP

Para instalar el GAP en dispositivos móviles se puede realizar de diversas maneras. A continuación se describen los pasos más importantes del proceso de instalación.

El primer requisito será contar con el .NET CompactFramework en su versión 1.0 o 2.0. Si el dispositivo contiene Windows Mobile 2003 o superior, el componente ya se encuentra instalado en la memoria ROM del dispositivo.

Sino se cuenta con .NET CF, es necesario descargarlo desde la página Web de Microsoft. La descarga es completamente gratuita, sólo hay que seguir los pasos del instalador, conectar el dispositivo a la PC a través de su base o cable de sincronización o bien de manera inalámbrica con Bluetooth o WiFi.

Una vez instalado el .NET CF, simplemente se copian la carpeta del GAP en el dispositivo móvil ya sea a través del proceso de sincronización o bien a través de una memoria de almacenamiento. Ya con esto se puede utilizar el sistema.

La mejor forma de instalación del prototipo es a través de archivos autoinstalables que se ejecutan en el dispositivo móvil. El prototipo cuenta con uno de estos archivos que se generan a través del IDE de programación utilizado, en nuestro caso Visual Studio .NET 2003. La característica de estos archivos es que son dependientes del microprocesador del dispositivo, por lo que se deberá elegir aquel archivo que sea para procesador ARM, MIPS y SH3. Estos archivos autoinstalables reciben el nombre de cabinet files (*.cab).

La ventaja de utilizar estos archivos autoinstalables es que permiten crear íconos y menús de atajo de la aplicación, lo cual nos facilita el acceso a la misma. Además como los archivos ya no se transfieren a través del proceso de sincronización, el tiempo de instalación es mucho más rápido. También existen versiones de .NET CF autoinstalables en el dispositivo.

La otra forma consiste en crear un instalador en la PC por medio de herramientas como InstallShield, entre otros, en sus versiones más recientes permiten la creación de proyectos para dispositivos móviles. La ventaja de este método es que permite de una manera sencilla instalar una aplicación desde una PC de escritorio, sin saber mucho sobre dispositivos móviles.

REFERENCIAS

- [Ala02] Fernando Alarcón G., “*Mecanismo para Gestión de Conexión en Sistemas Cliente/Servidor Móviles*”, tesis de maestría, *cenidet*, agosto de 2002.
- [ami06] Asociación Mexicana de Internet (AMIPCI), <http://www.amipci.org.mx/>, última consulta: octubre de 2006.
- [ava06] Avantgo, <http://www.avantgo.com/>, última consulta: octubre de 2006.
- [app76] Slogan de Apple computer, <http://www.apple.com/>.
- [Bia05] Pablo A. Bianco, “*Desarrollo de Aplicaciones Basadas en XML Web Services para Dispositivos Móviles con Microsoft .NET Compact Framework*”, tesina de Ingeniería Informática, Facultad de Ingeniería y Tecnología Informática, Universidad de Belgrano, Argentina., abril 2005.
- [bla06] Blackberry’s push technology, última consulta: octubre de 2006, http://www.blackberry.com/products/software/integrations/push_email.shtml.
- [CF03] Mitch Cherniack, Michael J. Franklin, “*Expressing User Profiles for Data Recharging*”, Brandeis University, University of California at Santa Cruz. 2003.
- [cft06] Comisión Federal de telecomunicaciones, <http://www.cofetel.gob.mx/>, última consulta: octubre de 2006.
- [Cha03] Thong Chanchaem, “*A Survey on Internet Content Transcoding for Universal Access*”, Department of Computer Science, Kent State University, mayo de 2003.
- [Cla02] David Clark, “*Mobile processors begin to grow up*”, revista Computer, IEEE Computer Society Press, pp. 22-25, marzo de 2002.
- [Dab05] Ahmad Dabas, “*Proxy Server for Handhelds*”, Universidad Tecnológica Checa, Facultad de Ingeniería Eléctrica, tesis de maestría, Praga, República Checa, mayo de 2005.
- [DD98] Harvey M. Deitel, Paul J. Deitel, “*Como programar en Java*”, Primera edición, Pearson educación, México, 1998, ISBN: 970-17-0044-9, p. 816.
- [dmc05] “*Data management for Mobile Computing, Case Studies*”, http://www.cs.ucv.ac.cy/courses/EPL651/Schedule_Readings/Chapter_3/body_chapter_3.html, Departamento de Ciencias Computacionales, Universidad de Chipre, última consulta: marzo de 2005.
- [dti06] DiarioTI: Diario Tecnologías de la información, <http://www.diarioti.com/>, última consulta: octubre de 2006.
- [emb06] “*Embedded System*”, http://en.wikipedia.org/wiki/Embedded_system, última consulta: octubre de 2006.
- [Fir04] Maximiliano R. Firtman, “*Programación para celulares con Java*”, MP Ediciones SA, ISBN: 9875262277, Argentina, pp. 321.
- [Fir05] Maximiliano R. Firtman, “*Desarrollos móviles con .NET*”, MP Ediciones SA, Argentina, 2005, ISBN: 9875262846, pp. 368.
- [Gon03] J. Gabriel González S., “*Plataforma middleware reflexiva para aplicaciones de cómputo móvil en Internet (Movirware)*”, Centro Nacional de Investigación y Desarrollo Tecnológico (*cenidet*), de septiembre de 2001 a agosto de 2003, financiamiento COSNET: 570.01-P.
- [gprs06] A practical guide to GPRS, SonyEricsson Whitepaper http://www.ericsson.com/mobilityworld/developerszonedown/downloads/docs/gprs/Practical_GPRS1.pdf, última consulta: octubre de 2006.
- [GS05] John Goodacre, Andrew N. Slow, “*Parallelism and the ARM Instruction Set Architecture*”, Computer Magazine, julio de 2005, IEEE Computer Society Press, pp. 42-50.
- [gwa06] Google Web Acelerador, última consulta: agosto de 2006, <http://webaccelerator.google.com/webmasterhelp.html>
- [Her05] Gabriel Hernández M. “*Generador de patrones de navegación de usuarios aplicando Web log mining*”, tesis de maestría, *cenidet*, noviembre de 2005.
- [HNS+98] Toshihiro Hattori, Yusuke Nitta, Mitsuhiko Seki, Susumu Narita, Kunio Uchiyama, Tsuyoshi Takahashi, Ryuichi Satomura, “*Design Methodology of a 200MHz superscalar microprocessor: SH-4*”, IEEE Proceedings of the 35th Design Automation Conference (DAC’98) 0-89791-964-5/98.

- [iana06] IANA, asignación de puertos de servicios de Internet, última consulta: octubre de 2006, <http://www.iana.org/assignments/port-numbers>.
- [ine05] Instituto Nacional de Estadística Geografía e Informática (INEGI). <http://www.inegi.gob.mx/>, última consulta: octubre de 2006.
- [inf06] Infochannel, <http://www.infochannel.com.mx/>, última consulta: octubre de 2006.
- [inf06a] Infochannel, consultado: octubre de 2006 http://www.infochannel.com.mx/busqueda6.asp?id_nota=13718&seccion=buscador
- [isi06] iSilo, <http://www.isilo.com/>, última consulta: octubre de 2006.
- [jti06] JTidy HTML Parser and Pretty-Printer in Java, <http://jtidy.sourceforge.net/>, última consulta: octubre de 2006.
- [Jua05] Fredy Juárez P., “*Servidor Proxy Caché con Soporte a Operaciones en Modo Desconexión en Redes Inalámbricas*”, tesis de maestría, *cenidet*, febrero de 2005.
- [Lut04] Kyle Lutes, “*Software Development for Mobile Computers*”, Revista Pervasive computing, IEEE Computer Society Press e IEEE ComSoc Press, 2004, pp. 10 – 14.
- [mic75] Slogan de Microsoft corporation, 1975, <http://www.microsoft.com>
- [mok06] “*W3C mobileOK Scheme 1.0*”, W3C Working Draft 12 July 2006, consultado en: <http://www.w3.org/TR/mobileOK/>, última consulta: octubre de 2006.
- [Mon05] Matías Montico, “*Wireless: La revolución inalámbrica, guía teórica y práctica*”, MP Ediciones, Argentina, 2005, ISBN: 9875263230, pp. 272.
- [mp06] “*XHTML Mobile Profile*”, WAP-277-XHTMLMP-20011029-a, www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf, última consulta: octubre de 2006.
- [Mue06] Chris Muench, “*Development Tools for Mobile and Embedded Applications*”, <http://www.pocketpcdn.com/tools/index.html>, última consulta: octubre de 2006.
- [mwbp05] “*Scope of Mobile Web Best Practices*”, W3C Working Group Note 20 December 2005, <http://www.w3.org/TR/mobile-bp-scope/>, última consulta: octubre de 2006.
- [mwm06] Microsoft Windows Mobile, <http://www.microsoft.com/windowsmobile/>, última visita: octubre de 2006.
- [NC06] Nicholas Nicoloudis, Andy Chang, “*Synchronization and Mobile Storage*”, CSE3211 Handheld Applications and Operating Systems, Monash University, última consulta: octubre de 2006.
- [palm06] PalmOS, Palm Inc., <http://www.palm.com/>, última consulta: octubre de 2006.
- [Pau06] Linda D. Paulson, “*TV Comes to the Mobile Phone*”, revista Computer, IEEE Computer Society Press, abril de 2006.
- [ppc06] Pocket PC, <http://www.pocketpc.com/>, última consulta: octubre de 2006.
- [Pre02] Pressman Roger S., “*Ingeniería del Software. Un enfoque práctico*” Quinta edición, España, 2002, ISBN: 84-481-3214-9, pp. 601.
- [pwc06] Palm Web Clipping, <http://www.palmos.com/dev/tech/webclipping/>, última consulta: octubre de 2006.
- [rg06] RepliGo, <http://www.cerience.com/>, última consulta: octubre de 2006.
- [rh05] “*Replication Vs Hoarding*”, última consulta: febrero de 2005, <http://www.cs.berkeley.edu/~adj/cs2941.s98/dbmate/sld008.htm>.
- [skw06] Skweezer, <http://www.skweezer.net/>, última consulta: octubre de 2006.
- [sp06] Smartphone, Wikipedia,. <http://en.wikipedia.org/wiki/Smartphone>, última visita: octubre de 2006.
- [sun87] Slogan de Sun Microsystems, <http://www.sun.com/>
- [sym06] SymbianOS version 9.3, <http://www.symbian.com/files/rx/file7999.pdf>, última consulta: octubre de 2006.
- [Tri06] Anna Trifonova, “*Mobile Learning: Gíreles and Mobile Technologies in Education Towards Hoarding Content in M-Learning Context*”, tesis doctoral, Universidad de Trento, Italia, marzo de 2006.
- [upnp06] UPnP (Universal Plug and Play) Forum, <http://www.upnp.org/>, última consulta: octubre de 2006.
- [Uri04] Claudia S. Uriarte C., “*Transformador de Contenidos Web para Asistentes Personales Digitales*”, tesis de maestría, *cenidet*, julio de 2004.
- [Val02] David R. Valenzuela M., “*Mecanismo para Predicción de Acaparamiento de Datos en Sistemas Cliente/Servidor Móviles*”, tesis de maestría, *cenidet*, agosto de 2002.
- [Ver03] Gustavo Verduzco R., “*Gestor de Acaparamiento de Patrones de Sitios Web en Clientes Móviles*”, tesis de maestría, *cenidet*, agosto de 2003.

- [Wan99] Jia Wang, “*A Survey of Web Caching Schemes for the Internet*”, ACM Computer Communication, Review, vol. 25, no. 9, 1999, pp. 36-46.
- [Way95] Meter Wayner, “*Agents unleashed: a public domain look at agent technology*”, Estados Unidos, 1995, ISBN: 0-12-738765-x, pp. 358.
- [wce06] Windows CE Operating System, última consulta: octubre de 2006, <http://msdn.microsoft.com/embedded/window sce/default.aspx>.
- [wcag06] Web Content Accesibility Guidelines 1.0, <http://www.w3.org/TR/WAI-WEBCONTENT/>, última consulta: octubre de 2006.
- [wml06] Especificación WML, <http://www.wapforum.org/>, última consulta: agosto de 2006.
- [wol06] World Off-line, última consulta: octubre de 2006, http://www.pocketgear.com/software_detail.asp?id=5797.