INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
LABORATORIO DE TECNOLOGÍA DE SOFTWARE

Esquema Adaptativo para la Gestión de Movilidad en Sistemas Cliente/Servidor a través de Internet

TESIS QUE PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

JUAN GABRIEL GONZÁLEZ SERNA

Directores de Tesis

Dr. Felipe Rolando Menchaca García
Dr. Rodolfo Abraham Pazos Rangel

México D.F. **Agosto 2006**

Dedicatorias

A Dios
Gracias señor por todas las bendiciones que me has dado, mi familia y mi trabajo.

A mi Madre
Por ser el tesoro más maravilloso de mi vida
Por su amor, sus bendiciones, su apoyo y su confianza
Por ser mí orgullo y mí más grande ejemplo.

A mi Padre *(Q.E.P.D.)*
Te extraño mucho viejo!

A mi Esposa
Por todo tu Amor y tu paciencia
Por enseñarme a enfrentar mis temores
Te Adoro

A mis Hijos
Yara, Johab, Gabriel y Gael
Por el amor que me dan día a día
y por ser el motor de mi vida
Los Amo.

A mis Hermanos y su Familia
Oscar, Horacio y Claudia
Gracias por mis sobrinos, cuñadas y cuñados
Por ser quienes son
Los Amo

Agradecimientos

Un agradecimiento muy especial a todas aquellas personas que me apoyaron a lo largo de este proyecto. Su amistad, su enseñanza, su dedicación y su ejemplo los llevaré en mí por siempre.

A toda mi familia por creer en mí. A pesar de la distancia siempre estuvieron a mi lado, a todos muchísimas gracias!

Al Centro de Investigación en Computación (CIC-IPN) por darme la oportunidad de realizar este proyecto de vida, y por todo lo que me apoyaron.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (cenidet) en especial a los directivos por confiar en mí y por todo el apoyo que me brindaron.

A Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado durante la realización de mis estudios de doctorado.

A la Asociación Nacional de Universidades e Institutos de Educación Superior (ANUIES) por ser parte importante con su apoyo económico para la culminación de mis estudios de doctorado.

A mis directores de tesis, Dr. Rodolfo Abraham Pazos Rangel y Dr. Felipe Rolando Machaca García, por compartirme sus conocimientos y guiarme a lo largo de este trayecto, pero sobre todo, por el valioso tiempo dedicado para que esto fuera posible.

A mis revisores, gracias por la dedicación a este trabajo, por sus observaciones y sus acertados consejos para mejorarlo.

A mis compañeros del cenidet, por brindarme sus consejos para culminar mi doctorado.

A todos, mil gracias!

TABLA DE CONTENIDO

CAPITULO 4

Validación de la metodología de solución

CAPITULO 5

Trabajos relacionados

CAPITULO 6

Conclusiones y trabajos futuros

ANEXOS

Anexo A. Caso práctico de la generación de patrones

Índice de figuras

Índice de Tablas

Glosario de términos

Bluetooth. Sistema de comunicación inalámbrica que permite la interconexión de diferentes dispositivos electrónicos (PCs, teléfonos fijos o móviles, agendas electrónicas, auriculares, etc.) a corto alcance. Es un estándar creado por importantes empresas del sector de la informática y de las telecomunicaciones. Bluetooth, que en inglés significa literalmente "diente azul", apodo de un jefe vikingo del siglo IX D.C.

Cache. Sistema cuya funcionalidad es la copia de páginas de Web recientemente visitadas por el usuario, las que el ordenador alberga en su disco duro y permite visualizarlas de nuevo con un tiempo de respuesta rápido, ya que la PC muestra esa copia, sin necesidad de volver a acudir a la red para descargar la página.

Frames. El lenguaje HTML ofrece la posibilidad de dividir una página de Web en varias zonas, cada una de las cuales puede tener un contenido independiente de las demás. Cada una de esas zonas constituye un frame.

HTML. HyperText Markup Language. Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia.

Intranet. Red propia de una organización, diseñada y desarrollada siguiendo los protocolos propios de Internet, en particular el protocolo TCP/IP. Puede tratarse de una red aislada, es decir no conectada a Internet.

Middleware. Es el software que sirve de intermediario entre aplicaciones, por ejemplo entre un programa de interfaz con el usuario y un sistema operativo.

PCMCIA. (Personal Computer Memory Card International Association.) Asociación Internacional de Tarjetas de Memoria para Ordenadores Personales. Tarjeta estandarizada de expansión para ordenadores personales. Tecnología que permite conectar fácilmente una gran variedad de dispositivos a un ordenador, normalmente una portátil o un PDA. Para conectar este dispositivo es necesario que el ordenador disponga del mismo tipo de ranura.

PDA. (Personal Digital Assistant.) Es un pequeño ordenador que cabe en el bolsillo, con funciones para organizar la información personal, y algunos equipados con una conexión a Internet.

PIE. (Pocket Internet Explorer.) Navegador de Web integrado en los dispositivos portátiles Pocket PC.

Pocket PC. Computadora de bolsillo que trabaja bajo la plataforma del sistema operativo Windows CE o Pocket PC en sus diferentes versiones.

Proxy. 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.

Punto de Acceso. Dispositivo que regula el tráfico en una red inalámbrica, cuya función es similar a un concentrador en una red cableada.

Scripts. Son pequeñas piezas de código en algún lenguaje de programación, como Java, incrustadas en una página de Web, para conseguir obtener contenido dinámico.

Servidor. Sistema que proporciona recursos (por ejemplo, servidores de archivos, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la red.

Squid. Es un servidor proxy Cache 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.

UML. (Unifed Modeling Language.) El lenguaje para modelado unificado (UML) es un lenguaje para la especificación, visualización, construcción y documentación de de software en un proceso de diseño de sistemas. El lenguaje ha ganado un significativo soporte de la industria de varias organizaciones vía el consorcio de socios de UML, y ha sido presentado al Object Management Group (OMG) y aprobado por éste como un estándar (noviembre 17 de 1997).

URL. (Universal Resource Locutor.) Localizador Universal de Recursos. Sistema unificado de identificación de recursos en la red. Permite identificar objetos WWW, Gopher, FTP, etc. Es una cadena que suministra la dirección Internet de un sitio de Web o de un recurso World Wide Web, junto con el protocolo por el que se tiene acceso a ese sitio o a ese recurso. El tipo más común de dirección URL es http://, que proporciona la dirección Internet de una página de Web.

WLAN. (Wireless Local Area Network.) Término utilizado para referirse a las redes de área local en un ambiente inalámbrico.

XHTML. XHTML 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.

XML. 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.

Hits Número de veces que una página, imagen o archivo de un solo sitio es visto i descargado por una visitante.

.-

Resumen

En esta tesis se aborda el problema de la gestión de eventos de desconexión y acaparamiento de sitios Web en entornos de cómputo móvil heterogéneos. Para abordar este problema en primera instancia fue necesario evaluar el desempeño de la arquitectura de software cliente/servidor con el propósito de identificar sus limitaciones en entornos propensos a frecuentes desconexiones. Para lograr esto, centramos nuestra evaluación en escenarios de redes inalámbricas locales (también conocidas con WLANs), es decir, áreas donde generalmente no se proporciona servicio de red debido a ciertas limitaciones de instalación de cable estructurado, como por ejemplo: jardines, patios, cafeterías, por mencionar algunas, donde se proporciona servicio inalámbrico a los usuarios. Como segunda instancia se desarrollaron estrategias para el manejo de esquemas de acaparamiento y transcodificación de sitios de la Web en dispositivos de cómputo convencionales y no convencionales. Esta segunda vertiente es la principal aportación de este trabajo de tesis ya que hasta este momento no se han encontrado referencias de trabajos que apliquen este esquema de acaparamiento y transcodificación de sitios de Web en dispositivos de cómputo móvil.

Capítulo 1

Introducción

En este capítulo se presenta un panorama general de este trabajo de tesis en donde describimos los puntos más importantes tales como el problema a resolver en esta tesis, los objetivos generales y los particulares ,las contribuciones y la descripción de algunos temas necesario para los capítulos subsiguientes.

1.1 Introducción

En esta tesis se aborda el problema del acceso a Internet mediante dispositivos móviles heterogéneos. Para proponer una solución a este problema en primera instancia fue necesario evaluar el desempeño de la arquitectura cliente-servidor con el propósito de identificar sus limitaciones en escenarios de cómputo móvil. Como segunda instancia se desarrollaron estrategias para el manejo de esquemas de acaparamiento y transformación de sitios de Web en dispositivos de cómputo convencionales y no convencionales. Esta segunda vertiente es la principal aportación de este trabajo de tesis ya que hasta este momento no se han encontrado referencias de trabajos que apliquen este esquema de acaparamiento y transformación de sitios de Web.

Esta tesis se centró en las redes inalámbricas para cómputo móvil las cuales proporcionan cobertura para el acceso a Internet en áreas en donde generalmente no se proporciona servicio, debido a ciertas limitaciones de instalación de cable estructurado, como por ejemplo: jardines, patios, cafeterías, bibliotecas, hospitales y aulas. Las tecnologías inalámbricas más importantes actualmente son las siguientes: WLAN (802.11), Bluetooth y MaNet (802.15), WiMax (802.16) y finalmente las redes de telefonía celular como GSM, GPRS, UMTS y EvDo.

En lo que respecta a las redes celulares podemos identificar dos servicios de transporte de datos utilizados para acceder a Internet, el primera se denomina CSD (Circuit Switch Data) se basa en esquemas de conmutación de circuitos, su esquema de tarifación es por tiempo de conexión dado que se realiza un enlace físico entre emisor y receptor. El segundo servicio de transporte de datos se basa en esquemas de conmutación de paquetes denominada GPRS (General Packet Radio Service) cuyo esquema de tarifación es por volumen de datos transmitidos ya que la información se envía por paquetes.

En resumen en esta tesis se propone un nuevo esquema de gestión de conexión a través de enlaces inalámbricos para acceder a Internet el cual garantiza la continuidad del trabajo del usuario móvil a pesar de la frecuente perdida de la conexión y el ahorro en costo de conexión cuando se utiliza servicios de transporte de datos tarifados ya sea por tiempo de conexión o por volumen de datos transmitidos. La aportación de este trabajo de tesis es el esquema de gestión de movilidad para acceder a Internet el cual permite a las aplicaciones cliente-servidor actuales adaptarse de manera transparente a la dinámica de los escenarios de cómputo móvil actuales, sin necesidad de modificar ninguno de sus esquemas de interacción originales los cuales son inadecuados para escenarios en donde la gestión del esquema de conexión y el modelo de interacción son críticos.

1.2. Problemática general del cómputo móvil

Los adelantos tecnológicos tales como las redes inalámbricas y los dispositivos de cómputo portátiles, han provocado cambios drásticos en el esquema de interacción cliente/servidor tradicional. Esto implica el surgimiento de nuevos problemas, entre los que destacan adaptabilidad a entornos dinámicos, frecuentes perdidas de conexión o de la señal inalámbrica, precarga de recursos informáticos, autentificación de clientes e identificación de patrones de uso. Esta problemática surge por las siguientes razones:

- ***Recursos hardware limitados*** [33]. Las aplicaciones de cómputo móvil tienden cada vez más a ejecutarse en dispositivos con recursos limitados, por ejemplo PDAs, y teléfonos celulares. Estos dispositivos, en algunos casos, cuentan con poca memoria RAM, por ejemplo los teléfonos celulares, existe una gran variedad de microprocesadores para PDAs, ARM, MIPS, MP3, etc. que en su mayoría son incompatibles, otro aspecto importante es el uso eficiente de la batería lo cual es critico en celulares, una gran diversidad de pantallas con diferentes resoluciones, por ejemplo 320x200, 70x50 pixeles y finalmente diseño de software específico para cada plataforma hardware.
- ***Costo de la conexión***. Los dispositivos móviles se conectan a la red por cortos períodos de tiempo mediante de enlaces inalámbricos, principalmente para conexión de voz, de datos o para solicitar un servicio. Sus escenarios de ejecución son sumamente dinámicos, es decir, el ancho de banda es variante, los servicios que están disponible en un momento pueden desaparecer sin previo aviso. En el caso de los teléfonos celulares los costos de conexión varían dependiendo del servicio de transporte de datos, por ejemplo: una conexión tarifada por tiempo, en México tiene un costo promedio de $1.5 pesos el minuto y una conexión tarifada por volumen de datos, es decir Kb por segundo transmitidos, tiene un costo promedio de $0.12 pesos por kilobyte transmitido.
- ***Modelo de interacción cliente/servidor tradicional***. El modelo de interacción de la arquitectura cliente-servidor depende del servicio de transporte utilizado que por lo general es TCP, el modelo de interacción de este servicio de transporte es síncrono, esto implica una interacción constante entre cliente y servidor ya que se basa ya que su interacción es mediante el intercambio de mensajes del tipo solicitud-respuesta el cual es consumidor de tiempo e inadecuado para escenarios de computo móvil.
- ***Sitios Web diseñados para plataformas convencionales***. Actualmente los diseñadores de sitios Web consideran para sus diseños únicamente plataformas convencionales, es decir, dispositivos de cómputo con resoluciones mínimas de 800x600 pixeles, en este sentido los diseñadores no toman en cuenta que actualmente el acceso a Internet se hace mediante dispositivos heterogéneos, con pantallas reducidas, por ejemplo celulares y PDAs.

1.3 Planteamiento del problema

Los dispositivos móviles pese a su gran popularidad presentan muchas limitaciones específicamente en lo que se refiere a la navegación en la Web, en la figura 1 se describen las limitantes que consideramos más relevantes.

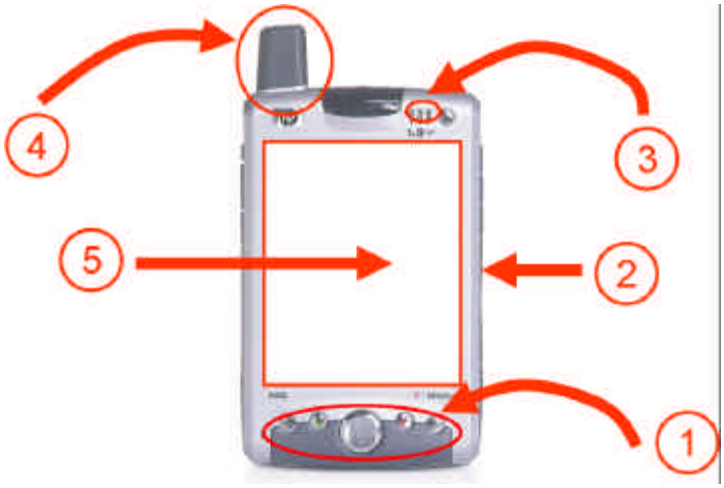


Figura 1.1. Problemas más comunes en dispositivos móviles.

1. Métodos de entrada de información deficientes (teclados pequeños si existen, reconocimiento de escritura ineficaz, etc.).
2. Cuentan con pocos recursos en comparación con una PC de escritorio (limitada memoria RAM, poco espacio de almacenamiento, pocos periféricos, microprocesadores lentos, etc.).
3. Suministro finito de energía (entre más capacidad de procesamiento y uso de periféricos menor tiempo de carga de la batería).
4. Los eventos de perdida de la señal inalámbrica son frecuentes en estas plataformas debido a la movilidad de los usuarios de estos dispositivos.
5. El despliegue de la información es limitado debido a que estos dispositivos tienen pantallas pequeñas, en comparación con una plataforma convencional.

Estas limitantes han provocado que los dispositivos móviles no sean muy populares para acceder a la Web, además, si a esto le sumamos que el costo de acceder a la Web mediante conexiones CSD o GPRS a través de un teléfono celular no es atractivo para la mayoría de los usuarios concluimos que esta tecnología prácticamente fracasaría en estas aplicaciones.

En lo que se refiere a la arquitectura cliente-servidor tradicional, hay que tomar en cuenta que su modelo de interacción requiere enlaces persistente y orientado a conexión. Es decir, una vez

establecido en enlace entre cliente y servidor debe mantenerse activo hasta terminar la solicitud del cliente. Es evidente que el esquema de interacción de la arquitectura cliente/servidor no es adecuado para dispositivos móviles, para este tipo de escenarios se requiere un modelo asíncrono no interactivo en donde los dispositivos no están todo el tiempo conectados y la interacción entre proceso cliente y servidor es casi mínima.

Si acotamos nuestro escenario de cómputo móvil a servicios de acceso a la Web, el primer problema que enfrentaríamos estaría relacionado con el tamaño de las pantallas de los dispositivos móviles, esta limitante provocan que el usuario realice constantes desplazamientos horizontales y verticales (scrolling), con el objetivo de visualizar la página Web. En la figura 1.2 se muestra tres dispositivos con diferente resolución, en el caso de las plataformas celulares y PDAs es evidente el problema del área de despliegue ya que las páginas Web se diseñan considerando una resolución mínima de 800x600 pixeles.

Figura 1.2. Problema de la visualización de sitios Web en dispositivos móviles

Finalmente otro de los problemas que abordamos en esta tesis se relaciona con el costo de acceso a Internet a través de dispositivos móviles. El acceso a Internet a través de un dispositivo móvil usando tecnología celular es considerablemente caro. En México, los costos con el principal proveedor de telefonía celular (Telcel) son:

- Utilizando un enlace con tecnología de conmutación de circuitos, por ejemplo CSD, el costo por minuto es de $1.5 pesos en prepago y $1 peso en plan tarifado.
- Utilizando un enlace con tecnología de conmutación de paquetes, por ejemplo GPRS, tiene un costo de $0.12 pesos por kilobyte o fracción transmitida (se puede obtener un plan de 50 Mb. por $500). En la tabla 1.1 se describen los costos y tiempos de acceso para diferentes tipos de servicios, los cuales consideramos los más representativos.

Tabla 1.1 Costos de acceso a Internet mediante 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 (68k)	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
Ver una 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 a Internet mediante un enlace dedicado, pero en algunos casos son convenientes como es en la búsqueda y visualización de algún servicio como cartelera de cine o el resultado de algún partido de football.

También se puede apreciar que en casos donde se requiere mayor contenido de datos o información no es del todo conveniente, como por ejemplo descarga de archivos o el envío y recepción de correo electrónico.

En resumen, la problemática que se abordo en esta tesis se dividió en tres temas específicos: i) la gestión de conexión a través de enlaces inalámbricos, en este sentido consideramos el problema de las frecuentes desconexiones y del costo de conexión mediante un esquema consumidor de tiempo, el cual no es adecuado para ambientes en donde el tiempo de conexión es crítico para obtener un costo-beneficio, ii) el volumen de información a transmitir y la constante interacción de entre cliente y servidor lo cual se refleja en costo y consumo de energía, y iii) el problema de los criterios de diseño tradicionales de los sitios Web en donde no se consideran los dispositivos con limitantes área de despliegue ni en las limitantes hardware y software.

1.4 Preguntas de investigación

La problemática planteada en la sección 1.2 nos permitió plantear las siguientes preguntas, de las cuales se deriva el objetivo de esta tesis doctoral:

- **Gestión de conexiones**. ¿Se puede aplicar de manera transparente un modelo de interacción asíncrono no interactivo en las aplicaciones cliente/servidor actuales?

- **Acaparamiento de páginas Web**. ¿Se puede precargar un subconjunto de páginas de un sitio Web en un dispositivo móvil en función de patrones de uso del sitio?
- **Transformación de páginas Web**. ¿Es posible reformatear una página Web de acuerdo a las características del dispositivo que la solicita?

1.5 Objetivo de la tesis

El objetivo de esta tesis es evaluar la factibilidad de implementar una arquitectura basada en gestores de acceso a Internet que permita a las aplicaciones cliente/servidor adaptarse de manera transparente a la dinámica de los escenarios de cómputo móvil, estos gestores deben proporcionar servicios de gestión de conexiones asíncronas no interactivas y servicios de precarga y reformateo de páginas Web para dispositivos móviles heterogéneos para garantizar que los usuarios tengan acceso a la Web en cualquier lugar, en todo momento y desde cualquier dispositivo con capacidad de conexión a Internet.

1.5.1 Objetivos específicos

Los objetivos específicos de esta tesis implican el diseño e implementación de tres servicios que consideramos engloban la problemática que se planteo en el apartado 1.2, los objetivos específicos se describen a continuación:

- Desarrollo de un esquema de gestión de conexiones en entornos de cómputo móvil, este esquema debe proporcionar servicios de conexión de tipo asíncronos no interactivos que permitan a las aplicaciones cliente-servidor actuales adaptarse de manera transparente a los escenarios de cómputo móvil sin necesidad de modificar sus esquemas de interacción originales.
- Desarrollo de un esquema de precarga de páginas de un sitio Web en dispositivos móviles, mediante la identificación de patrones de uso del sitio aplicando minería de uso Web,
- Podado y compactación de un sitio Web para reducir su tamaño e impactar el costo de transmisión cuando se utiliza un servicio de transporte de datos basado en tecnologías de conmutación de paquetes (GPRS).
- Desarrollo de un esquema para la transformación de páginas Web a múltiples formatos para dispositivos de cómputo móvil heterogéneos, que permita que diferentes dispositivos visualicen el contenido Web de acuerdo a sus requerimientos.

1.6 Principales contribuciones

Es clara la problemática inherente en los ambientes de cómputo móvil, los cuales imponen nuevos requerimientos de diseño en las arquitecturas software y en los esquemas de interacción cliente/servidor, que difieren de los esquemas tradicionales. Las principales contribuciones de esta tesis para esta área de conocimiento son las siguientes:

- Replanteamiento del modelo de interacción cliente/servidor tradicional al cual denominamos *modelo asíncrono no interactivo*.
- Diseño de un modelo de interacción para las arquitecturas cliente/servidor tradicionales que se denominó *modelo de interacción asíncrono no interactivo*. Este modelo permite que aplicaciones cliente/servidor tradicionales se adapten a cualquier entorno de cómputo móvil sin necesidad de modificar ningún aspecto arquitectónico ni de interacción original.
- Implementación de servicios para la generación de patrones de navegación aplicando Minería de uso de la Web.
- Diseño e implementación de estrategias de acaparamiento de páginas Web en dispositivos de cómputo móvil mediante la identificación de patrones de navegación de sitios de la Web.
- Transformación de contenidos Web a múltiples formatos para dispositivos de cómputo móvil heterogéneos.
- En lo referente a los tiempos de conexión para acceder a recursos Web acaparados en el dispositivo móvil se tiene lo siguiente:
    - Se mejoró un 85% el tiempo de acceso gracias a los recursos Web precargados de manera local en la cache del dispositivo móvil
    - Se redujo la cantidad de solicitudes entre cliente y servidor en un 80%, ya que se replica un subconjunto del sitio Web en el dispositivo móvil por lo que la probabilidad de solicitar una página del sitio no precargada en el dispositivo es del 20%.
- En lo referente al tamaño de los recursos replicados en el dispositivo móvil:
    - El proceso de eliminación o recorte de páginas de un sitio Web reduce en un 35% el tamaño del sitio,
    - la transformación de las páginas que se replican en el dispositivo móvil reduce hasta un 34% el tamaño del recurso y
    - la compresión de un sitio Web previamente recortado y transformado reduce hasta en un 86% el tamaño total del sitio que será replicado en el dispositivo móvil.

1.7 Estructura de la Tesis

Este documento de tesis está organizado de la siguiente manera:

El capítulo dos presenta un panorama general de la problemática inherente a los escenarios de cómputo móvil. De manera específica describimos la problemática de estos entornos en tres áreas: modelos de interacción en entornos de cómputo móvil, precarga de contenidos Web en dispositivos móviles y transformación de contendido Web para dispositivos heterogéneos.

En el capítulo tres presentamos la metodología de solución propuesta, específicamente describimos las estrategias que se propusieron como solución a los problemática planteada para este trabajo en las tres áreas que consideramos: manejo de desconexiones en entornos de cómputo móvil, precarga de contenidos Web en dispositivos móviles y transformación de contendido Web para dispositivos con áreas de despliegue limitadas.

En el capítulo cuatro se describen los resultados que permiten validar la metodología de solución propuesta. Se describen los casos de prueba, el diseño del experimento y los resultados.

En el capítulo cinco se describen los trabajos relacionados con esta tesis y las características de cada uno, se da una visión general de los trabajos de investigación y desarrollo además de describir las diferencias con esta tesis.

El capítulo seis presenta las conclusiones a las que se llego durante el desarrollo de esta investigación. El capítulo concluye dando sugerencias de trabajos futuros.

Capítulo 2

Problemática Relacionada con el Acceso a la Web a través de Redes Inalámbricas

E*n este capítulo se presenta un panorama general de la problemática inherente a los escenarios de cómputo móvil. De manera específica describimos la problemática de estos entornos en tres áreas: manejo de desconexiones en entornos de cómputo móvil, precarga de contenidos Web en dispositivos móviles y transformación de contendido Web para dispositivos con áreas de despliegue limitadas.*

2.1 Descripción general del problema

Los recientes adelantos en la tecnología de gestión de redes inalámbricas y el aumento exponencial de dispositivos de cómputo móvil, tales como computadoras portátiles, teléfonos inteligentes, asistentes personales digitales (PDAs), entre otros, están habilitando una nueva clase de aplicaciones que presentan nuevos desafíos de diseño. Por ejemplo, un usuario equipado con un dispositivo conectado a la red mediante un enlace inalámbrico, puede experimentar pérdidas temporales y sin previo aviso de conexión con la red, o puede descubrir e interactuar con otros nodos conformando espontáneamente redes personales. Para afrontar estos nuevos escenarios se requiere que las aplicaciones reaccionen a los cambios frecuentes de su entorno de ejecución, tales como una nueva localización, variaciones en el ancho de banda, desconexiones frecuentes, desvanecimiento de la señal, entre otros [15][18][23]. Aunado a esto, algunos de estos dispositivos, como los PDAs, que cuentan con recursos limitados tales como carga de batería, pantallas pequeñas, poca memoria, restricciones de manejo de objetos Web en su navegador, por mencionar algunos.

Como resultado de esta problemática y de las pruebas que se realizaron en este trabajo de tesis, consideramos que el esquema de interacción cliente/servidor está cambiando drásticamente. Esto lo podemos afirmar ya que identificamos varias limitantes en la arquitectura cliente/servidor tradicional cuando ésta se ejecuta en entornos con condiciones de trabajo variantes como son las redes inalámbricas, algunas de estas limitantes son las siguientes:

a) el modelo de interacción cliente/servidor es inadecuado para un entorno con condiciones de trabajo variantes,,
b) el manejo de eventos de desconexión no se considera en las arquitecturas cliente/servidor tradicionales,
c) las arquitecturas cliente/servidor no proporcionan servicios de predicción de acceso a recursos informáticos en particular para sitios Web,
d) los mecanismos de replicación o precarga están en su mayoría orientados a aplicaciones de bases de datos, implementando en la mayoría de los casos algoritmos estadísticos
e) la mayoría de los sitios Web están diseñados para plataformas convencionales en donde se considera un área de despliegue (pantalla) con un mínimo de resolución (800x600), lo cual no se aplica en dispositivos PDAs (240x320).

Mientras estos problemas no se solucionen, los equipos portátiles conectados mediante enlaces inalámbricos enfrentarán situaciones que las arquitecturas cliente/servidor convencionales no

manejan de manera adecuada. Con el propósito de estructurar el análisis de la problemática que abordamos en esta tesis dividimos el problema en tres áreas:

- Manejo de desconexiones en entornos de cómputo móvil,
- precarga de contenidos Web en dispositivos móviles y
- transformación de páginas Web para dispositivos de cómputo móvil con pantallas pequeñas.

2.2 Manejo de desconexiones en entornos de cómputo móvil

El modelo de interacción de las arquitectura de software cliente/servidor tradicional no considera un evento de desconexión como un evento común, al contrario, los atribuye a fallas de hardware o del medio físico de transporte [13][14]. Aunado a esto, las solicitudes de información que realizamos frecuentemente en Internet tales como consultas mediante motores de búsqueda, transferencia de datos, consultas a bases de datos, operaciones de comercio electrónico, o simplemente la gestión de los objetos contenidos en una página Web desde un navegador, requieren de una *conexión persistente*[1] para llevar acabo la solicitud de principio a fin.

Las solicitudes mencionadas en el párrafo anterior se realizan generalmente a través de aplicaciones diseñadas con el modelo de interacción de la arquitectura *cliente/servidor* tradicional, a este esquema se le denomina *modelo de interacción síncrono interactivo* (ver sección 2.2.1).

Como resultado de la investigación que se realizó en esta tesis, se identificaron cinco características de este modelo: 1) requiere de una conexión persistente; si la conexión se pierde, la transacción también, 2) cualquier falla de comunicación es atribuida al hardware de red; 4) su esquema de interacción es consumidor de tiempo por su naturaleza *síncrona*; y 5) utiliza un protocolo solicitud-respuesta.

Para comprender más la problemática inherente a los entornos de cómputo móvil es necesario analizar el modelo cliente/servidor móvil. Este modelo presenta las siguientes características:

1. Los eventos de desconexión son frecuentes.
2. Los períodos de conexión están limitados en tiempo debido a factores tales como la carga útil de la batería, movilidad, potencia de la señal, obstáculos físicos, etc.

[1] Nos referimos a una conexión persistente cuando las condiciones del enlace de comunicación garantizan que las desconexiones son eventos atribuidos a fallas de hardware y que son eventos que rara vez se presentan. A nivel de capa de transporte del modelo OSI, una conexión persistente es aquella que se mantiene abierta por el servidor.

3. El alcance de la señal de la señal inalámbrica está sujeto a las características físicas de las instalaciones (configuración y material de construcción del edificio).
4. El ancho de banda puede fluctúan de manera considerable.
5. Debido a la naturaleza del medio físico de transmisión, el aire, no se garantiza es casi imposible controlar los linderos de alcance la señal lo que provoca que el control de acceso a la red por usuarios no autorizados sea ineficiente.

Por lo regular se pueden numerar una gran cantidad de factores que ocasionan eventos de desconexión, algunos de estos factores pueden ser atribuidos a eventos tales como la caída de servidores o que un nodo a lo largo de la red deje de funcionar. Para este trabajo de tesis, el evento de desconexión que nos interesa, es el que se da entre el punto de acceso y el cliente móvil.

2.2.1 Esquema de interacción cliente/servidor tradicional

El esquema de interacción que se presenta cuando un proceso se comunica con otro proceso para solicitar un servicio, se denomina *cliente/servidor*, este esquema de interacción se basa en el intercambio de mensajes del tipo *solicitud/respuesta*, el cual es un proceso de comunicación de tipo *síncrono*, es decir, para que el proceso *A* (*cliente*) envíe un mensaje (*solicitud*) al proceso *B* (*servidor*), es necesario que el proceso *B* mantenga un puerto de escucha en espera de solicitudes (*conectado*), cuando el proceso *B* recibe el mensaje de solicitud envía un mensaje (*respuesta*) de regreso al proceso *A*. Este proceso se muestra en la Figura 2.1. Es evidente que la arquitectura cliente/servidor requiere un enlace de comunicación persistente ya que mantiene una interacción constante con el servidor, por esta razón a este modelo de interacción se denomina *síncrono interactivo*.

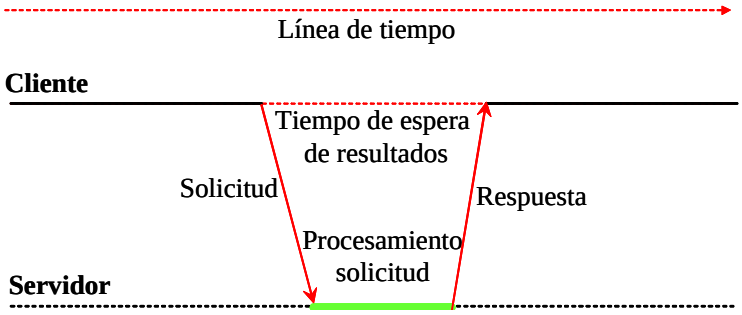


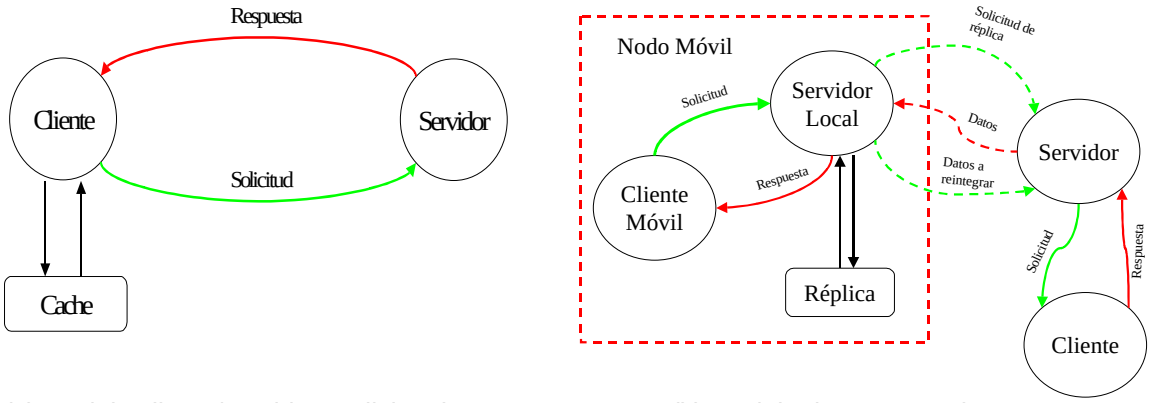
Figura 2.1 Modelo de Interacción basado en sincronismo

En el modelo síncono interactivo el proceso cliente es el que generalmente inicia la comunicación y no mantiene el estado de la misma, a este tipo de peticiones se les denomina

idempotentes[2]. Si el servidor no está disponible o falla, el cliente debe repetir nuevamente la solicitud, sin la posibilidad de recuperar los resultados de la primera solicitud [3][7][12]. Por ejemplo, el caso más común es el de los navegadores Web, en donde el cliente, en este caso el navegador, solicita al servidor Web un archivo *html* que describe el contenido de una página Web. Una vez que el cliente recibe el archivo *html,* analiza el contenido y solicita los objetos referenciados en éste (por ejemplo, las etiquetas *<img src="img.gif">*). Si durante el proceso de solicitud se presenta una desconexión, el navegador no tiene la capacidad de reiniciar (resume) la solicitud en el punto en que se quedo. Por otro lado, el servidor no continua con la(s) solicitud(es) del cliente, ya que los objetos son solicitados por demanda del cliente y no por iniciativa del servidor, por lo tanto, es evidente que se requiere una conexión persistente libre de desconexiones, entre cliente y servidor para llevar a cabo la solicitud de una página Web de principio a fin.

2.3. Precarga de contenidos de la Web en dispositivos móviles.

En la figura 2.2a se muestra el modelo cliente/servidor tradicional en este modelo cada cliente tiene recursos limitados en comparación con el servidor (espacio en disco y poder de cómputo). El modelo que se muestra en la figura 2.2b es una modificación del esquema cliente/servidor, el cual se orienta básicamente a aplicaciones de bases de datos móviles [3][4].



(a) Modelo cliente/servidor tradicional (b) Modelo de acaparamiento

Figura 2.2. Comparación de los modelos cliente/servidor.

Para describir este modelo de acaparamiento analizaremos el acceso a una base de datos de vendedores. El SMBD atiende peticiones de los clientes, a los cuales denominaremos clientes

[2] Una operación idempotente es aquella en la que la repetición de la operación es exactamente igual a la operación realizada una sola vez. Es decir es una operación que puede ejecutarse cuantas veces se quiera sin alterar el resultado de la llamada.

acaparadores (de ahora en adelante CA). A un CA se le permite *acaparar datos localmente* en su equipo por medio de *réplicas locales*, para este ejemplo analizaremos dos escenarios: el primero implica que el CA se encuentra conectado, es decir, existe una conexión persistente entre el cliente y el servidor. En este escenario las transacciones son procesadas directamente por el SMBD. En el segundo escenario, el CA trabajará en modo desconexión, es decir, no existe una conexión persistente entre el cliente y el servidor, en este caso las transacciones son procesadas por un servidor local en el CA (ver figura 2.6b). Si las transacciones del CA se desarrollaron en el segundo escenario es necesario que cuando el CA se conecta nuevamente, el servidor local reconcilie su réplica con la copia del SMBD mediante la reintegración de cualquier actualización local. Los términos acaparar y réplica local los utilizaremos indiferentemente a lo largo de este documento.

Podemos observar que este modelo implica cierto involucramiento por parte del administrador de la BD, ya que las réplicas son gestionadas a través de un software gestor, el cual de manera explícita describe la estructura de la réplica y el nodo en donde se va a replicar físicamente. Esto lo podemos aseverar ya que realizamos algunas pruebas con manejadores de BD como Interbase y Oracle 9i, en las cuales pudimos observar el nivel de intervención requerido por el administrador de la BD, ya que en ninguna de estas dos herramientas pudimos automatizar la replicación. Aunado a esto, el cliente acaparador requiere un servidor local o, dicho de otra forma, un servidor embebido[3]. En resumen un esquema de replicación involucra la intervención del administrador de la base de datos ya que el SMBD no puede gestionar de manera automática la replicación de cualquiera de sus tablas o fragmentos.

Una posible solución para manejar de manera más eficiente las estrategias de replicación, es utilizando un mecanismo de acaparamiento (hoarding) [10][16]. Este esquema permite *identificar y replicar recursos informáticos* en el equipo de clientes propensos a desconexiones. *El problema evidente de estos esquemas es la identificación e implementación de algoritmos basados en probabilidad adecuados al tipo de recurso informático al que se esté accediendo*. Es decir, es necesario evaluar e implementar algoritmos que obtengan información probabilística de acceso a recursos informáticos tales como sitios Web, sistemas de archivos o base de datos [10][25] [26][29].

En base a lo anterior, planteamos las siguientes preguntas de investigación: ¿Cómo podemos identificar patrones o probabilidades y no tendencias de acceso? ¿Cuáles son los

[3] El término "embebido" (también se lo conoce como "incrustado" o "embutido") significa que esos servicios son parte integral del sistema en que se encuentran. Lo interesante de un sistema "embebido" es que puede estar de tal forma incrustado, que puede quedar tan oculto a nuestros ojos, que la presencia de tales servicios no resulte obvia.

mecanismos probabilísticos o de generación de patrones de acceso más adecuados? ¿Los algoritmos de generación de patrones se pueden aplicar en cualquier contexto? ¿Es posible obtener patrones de acceso de un sitio de Web?

Es claro que la problemática que hemos descrito en los puntos 2.2 y 2.3 requiere el replanteamiento de los modelos de interacción y de las arquitecturas de software convencionales, este replanteamiento debe acoplarse a los requerimientos de los nuevos escenarios distribuidos. Por lo tanto*, consideramos necesario replantear y explorar nuevos mecanismos que integren nuevos servicios a las arquitecturas cliente/servidor tradicionales que permita a las aplicaciones basadas en este modelo adaptarse a su ambiente y estar más capacitadas para enfrentar la dinámica de los entornos de red actuales.*

2.4. Transformación de páginas Web para dispositivos de cómputo móvil con pantallas pequeñas

La dificultad que enfrentan los dispositivos portátiles para navegar en Internet, conlleva a replantear la manera de proporcionarles los recursos que se encuentran disponibles dentro de la Web, debido principalmente a los aspectos que se mencionan a continuación:

- En la Web existen millones de recursos que no han sido originalmente diseñados para este tipo de dispositivos, y que actualmente, están siendo solicitados mediante estos dispositivos, sin obtener un buen desempeño en lo que a navegación se refiere.
- Actualmente, los servidores Web pueden identificar el tipo de dispositivo que está realizando la petición, sin embargo no están preparados para personalizar la respuesta en tiempo de ejecución de acuerdo a las características del dispositivo, por ejemplo, resolución de la pantalla y gama de colores, por mencionar algunos.
- Por otro lado, los servidores Web que hoy en día están personalizando la entrega de su contenido, adoptan la estrategia de diseñar previamente los recursos con características especiales para dispositivos con pantallas pequeñas. Esto hace que se vaya creando una duplicidad de recursos en la red, por lo que se tienen que mantener diferentes versiones de una misma página HTML, lo que puede resultar costoso, en cuanto al consumo de recursos de los propios servidores. Esto se puede apreciar en la figura 2.3.

Lo anterior refleja la necesidad que se presenta alrededor de este contexto, lo que conlleva a establecer estrategias de transformación del contenido original de la Web, teniendo que considerar algunas variantes, como por ejemplo, el lugar apropiado para establecer un mecanismo de

transformación, ya que se trata de alterar lo menos posible la arquitectura sobre la cual opera un usuario en la red. Por otro lado, se encuentra la problemática inherente a la estructura de los documentos de HTML, debido a la naturaleza y flexibilidad de este lenguaje. En relación a esto existen investigaciones que han demostrado la deficiencia que se tiene sobre este aspecto [20, 21].

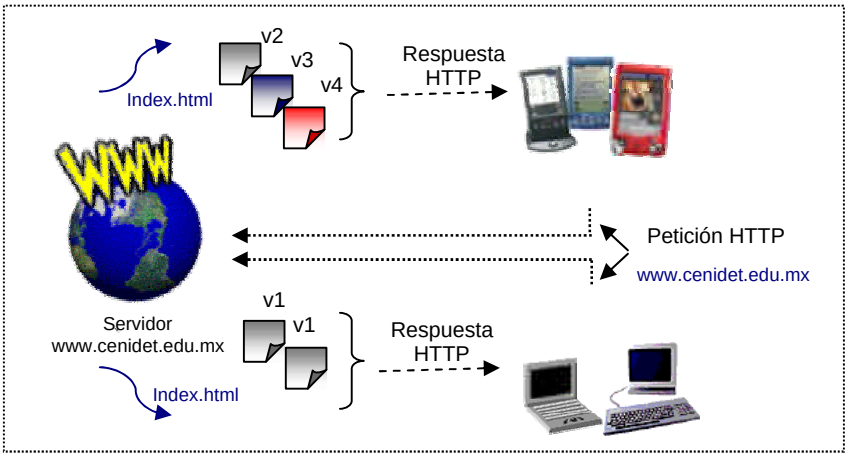


Figura 2.3. Problemática de múltiples versiones del contenido Web

Capítulo 3

Metodología de solución arquitectura Moviware

En este capítulo se describe el modelo conceptual que se propone para la solución del problema de acceso a la Web a través de redes inalámbricas , específicamente describimos los componentes de este modelo que proponen nuevos esquemas para dar tratamiento a la problemática planteada en los tres aspectos que se consideraron en esta tesis: 1) manejo de desconexiones en entornos de cómputo móvil, 2) precarga de contenidos de la Web en dispositivos propensos a desconexiones y 3) transformación de contendido de la Web para dispositivos de cómputo móvil heterogéneos.

3.1 Solución general propuesta

Para resolver la problemática planteada en esta tesis fue necesario replantear un nuevo esquema de interacción cliente/servidor para acceder a Internet mediante dispositivos inalámbricos (ver sección 2.2.1), así como también, el diseñó e implementación de un nuevo esquema de interacción para arquitectura cliente/servidor. La arquitectura que se desarrollo en esta tesis se muestra en la Figura 3.1 la cual denominamos Moviware.

En las siguientes secciones se describe el esquema para la gestión de movilidad que proponemos para solucionar el problema de acceso a la Web a través de redes inalámbricas planteado en esta tesis. Específicamente se describen los componentes de la arquitectura Moviware en donde se proponen nuevos esquemas para dar tratamiento a la problemática planteada en los tres aspectos que se consideraron en esta tesis:

A) gestión de conexiones en entornos de cómputo móvil (Punto 3.2),
B) gestión de servicios de precarga de contenidos de la Web en dispositivos propensos a desconexiones (Punto 3.3), y
C) transformación de contendido de la Web para dispositivos heterogéneos de cómputo móvil (Punto 3.4).

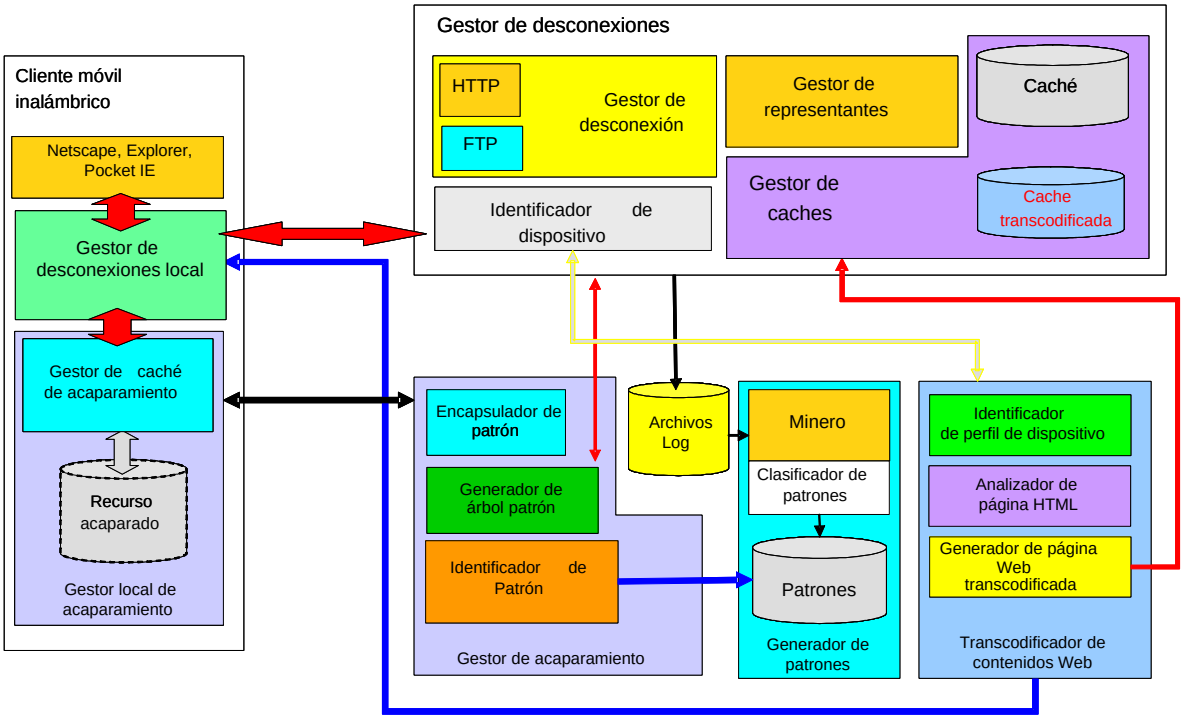


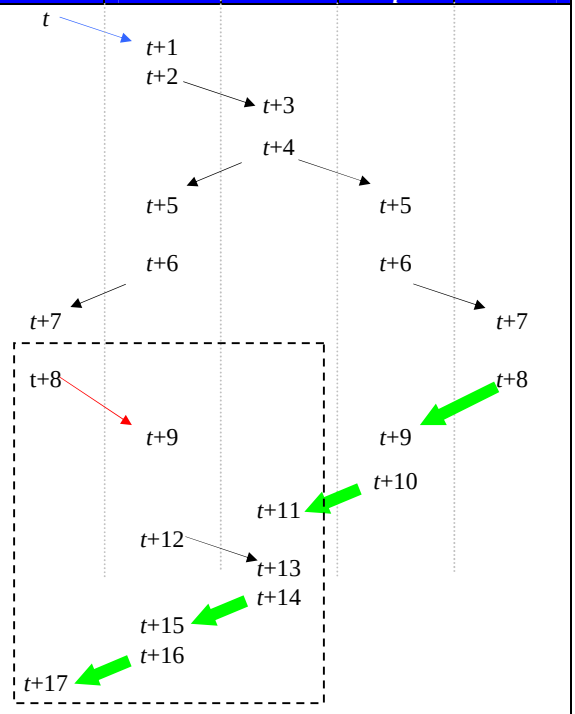
Figura 3.1 Arquitectura de solución general propuesto MoviWare

3.2 Metodología para la gestión de desconexiones en entornos de cómputo móvil

La solución propuesta para dar tratamiento a los eventos de desconexión en redes inalámbricas se compone de en una serie de servicios intermediarios para acceder a Internet, es decir, utiliza una alineación de intermediarios. Las aplicaciones cliente envían sus solicitudes a los intermediarios, cuando reciben la solicitud envían como respuesta un mensaje de confirmación, posteriormente el cliente se desconecta y delega al intermediario la gestión de su solicitud, en este punto, el esquema de interacción cliente/servidor tradicional se modifica ya que se elimina la necesidad de una conexión persistente entre cliente y servidor por lo que el modelo de interacción pasa a ser asíncrono ya que no se requiere de una interacción de tipo síncona (ver sección 3.2.1). En la tabla 3.1 se describe detalladamente el intercambio de mensajes que se da entre los procesos involucrados en el modelo asíncrono no interactivo, en esta tabla se muestran en un recuadro punteado los eventos que determinan el período de tiempo en el que se presenta la *asincronía* de intercambio de mensajes propuesto en esta tesis.

Tabla 3.1. Tabla de interacción de mensajes

Tiempo	Evento	Diagrama
t	*C* envía una petición a *PIC*	
t+1	*PIC* recibe la petición de C	
t+2	*PIC* envía una verificación de conexión a *PS*	
t+3	*PS* recibe la verificación de conexión de *PIC*	
t+4	*PS* envía un ACK de conexión a *PIC*	
	PS envía una petición a *PSquid*	
t+5	*PIC* recibe el ACK de conexión de *PS*	
	PSquid recibe la petición de *PS*	
t+6	*PIC* envía ACK a *C*	
	PSquid envía una petición a *SW*	
t+7	*C* recibe ACK de *PIC*	
	SW recibe la petición de *PSquid*	
t+8	*C* cierra su conexión con *PIC*	
	SW envía respuesta a *PSquid*	
t+9	*PIC* cierra su conexión con *PS*	
	PSquid recibe la respuesta de *SW*	
t+10	*PSquid* envía respuesta a *PS*	
t+11	*PS* recibe la respuesta de *PSquid*	
t+12	*PIC* envía una estado de solicitud a *PS*	
t+13	*PIC* recibe estado de solicitud de PIC	
t+14	*PS* envía respuesta a *PIC*	
t+15	*PIC* recibe la respuesta de *PS*	
t+16	*PIC* envía respuesta a *C*	
t+17	*C* recibe respuesta de *PIC*	
t+*n*		
t+(*n*+1)		



C = Cliente ***PIC*** = Proxy Intermediarío Cliente ***PS*** = Proxy Servidor ***Psquid***= Proxy Squid ***SW*** = Servidor Web

3.2.1 Modelo de interacción asíncrono no interactivo rediseñado

El modelo de interacción asíncrono que se describió en la sección 2.2.1, es funcional, el problema con este modelo es que se diseño considerando que los eventos de desconexión se presentaban esporádicamente, generalmente atribuidos a fallas de hardware. Esta consideración ha provocado que el modelo de interacción cliente/servidor tradicional sea obsoleto para los nuevos escenarios de redes inalámbricas en donde las desconexiones no se consideran fallas si no eventos comunes debido a la naturaleza del medio de transmisión utilizado, caso especifico de las redes IEEE 802.11. El modelo de interacción que se propone en esta tesis se esquematiza en la figura 3.2.

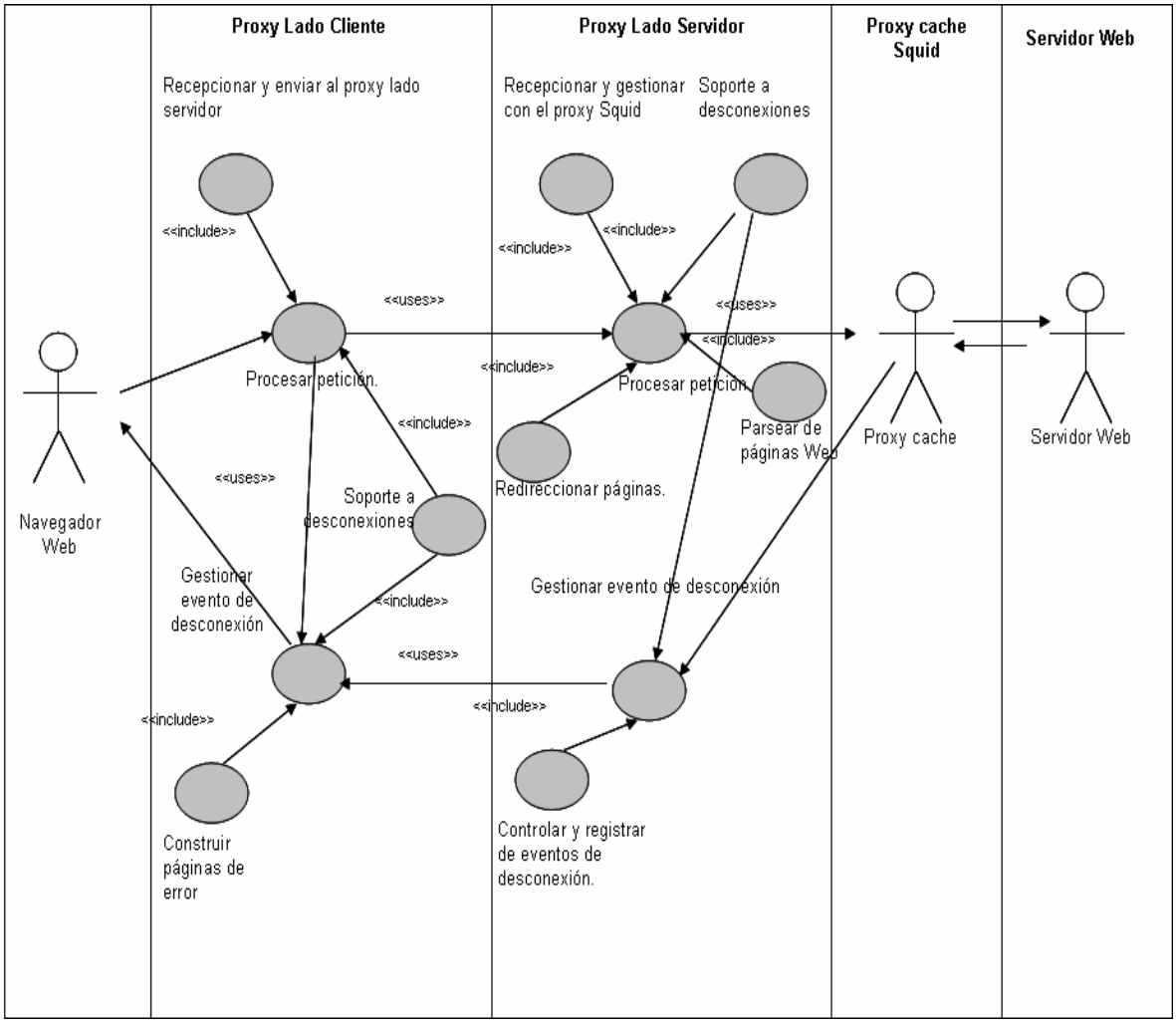


Figura 3.2. Modelo asíncrono no interactivo

Este modelo se conforma por lo siguientes actores, un cliente (C), un proxy intermediario cliente (PIC), un proxy servidor (PS), un proxy externo (PE) y un servidor Web (SW), los componentes PIC y PS permiten al cliente desconectarse durante la transmisión de los datos y dan soporte a desconexiones planeadas y no planeadas, la secuencia del funcionamiento del modelo de interacción asíncrono no interactivo es el siguiente:

1. El C envía una petición URL al PIC.
2. El PIC verifica el estado de la conexión, si se produce una desconexión envía una respuesta de error a C, de lo contrario, reenvía la petición al PS.
3. El PS verifica el estado de la conexión con el PE. Si no está activa, envía al PIC una página de error, de lo contrario, envía una página de confirmación y reenvía la solicitud al PE.
4. El PIC recibe la confirmación y envía una página de estado al C,
5. El C cierra su conexión con el PIC y el PIC cierra su conexión con el PS.
6. El PE envía y recibe la respuesta al SW.
7. El PS recupera la información del PE.
8. Cuando el PIC se reconecta, verifica el estado de la conexión, si se produce una desconexión, regresa una página de error al C, de lo contrario solicita la información del estado de la solicitud del PS.
9. Si el PS ha completado la petición de la página Web, ésta es enviada al PIC, de lo contrario, informa al PIC que aún no se ha completado la solicitud.
10. El PIC envía la información a C.

3.3 Metodología para la gestión de precarga de contenidos de Web en dispositivos propensos a desconexiones

Los algoritmos para el descubrimiento de *patrones de uso de la Web* [27][28][30] permiten identificar grupos de páginas Web que son solicitadas juntas en un gran número de sesiones de usuario. Se pueden mencionar los siguientes ejemplos:

- La página /A.html y la página /B.html fueron accedidas juntas en al menos el 50% de las sesiones de usuario durante el mes de marzo del 2005.
- La página /A.html y la página /B.html fueron accedidas juntas en al menos el 40% de las sesiones de usuario.

Cada uno de estos grupos de elementos o items define un valor conocido como soporte que representa el porcentaje de accesos o frecuencia. Cualquier conjunto de ítems frecuente

puede separarse en *n* reglas también llamada *proposición condicional*[1], donde un indicador de dirección es agregado a la regla. Un conjunto de *ítems* frecuentes formado por los elementos A y B da paso a la formación de dos posibles reglas de asociación representadas por A → B y B →A.

- Una sesión dada en el mes de marzo del 2005 que solicitó la página /A.html tiene un 83% de probabilidad de solicitar la página /B.html.
- Una sesión dada en el mes de marzo del 2005 que solicitó la página /B.html tiene un 89% de probabilidad de solicitar la página /A.html.

El segundo conjunto de ítems frecuentes presentado arriba ahora puede ser visto como sigue.

- Una sesión dada en el mes de marzo del 2005 que solicitó la página /A.html tiene un 66% de probabilidad de solicitar la página selección/B.html.
- Una sesión dada en el mes de marzo del 2005 que solicitó la página /B.html tiene un 92% de probabilidad de solicitar la página /A.html.

Como puede observarse, a partir de un conjunto de ítems frecuentes pueden formarse 2 reglas de asociación y aunque las dos reglas de asociación involucran a los mismos elementos, el porcentaje de probabilidad cambia de una a otra y esto se debe al orden de aparición de los elementos, esta probabilidad de acceso se conoce como nivel de confianza de la regla. En los temas subsecuentes se explica el cálculo del soporte y confianza.

El esquema acaparamiento propuesto en esta tesis para realizar la precarga de contenido Web en dispositivos móviles se muestra en la figura 3.3. En este modelo se compone de 6 capas cada una de ellas realiza una función especifica para la identificación de los patrones que pueden ser acaparados localmente en los dispositivos móviles.

[1] Proposición que puede expresarse como una proposición Si-entonces. Por ejemplo, "Si *un polígono es un hexágono*, entonces *tiene exactamente seis lados*". La primera parte de la condicional se denomina **antecedente**. La segunda parte se denomina **consecuente**.

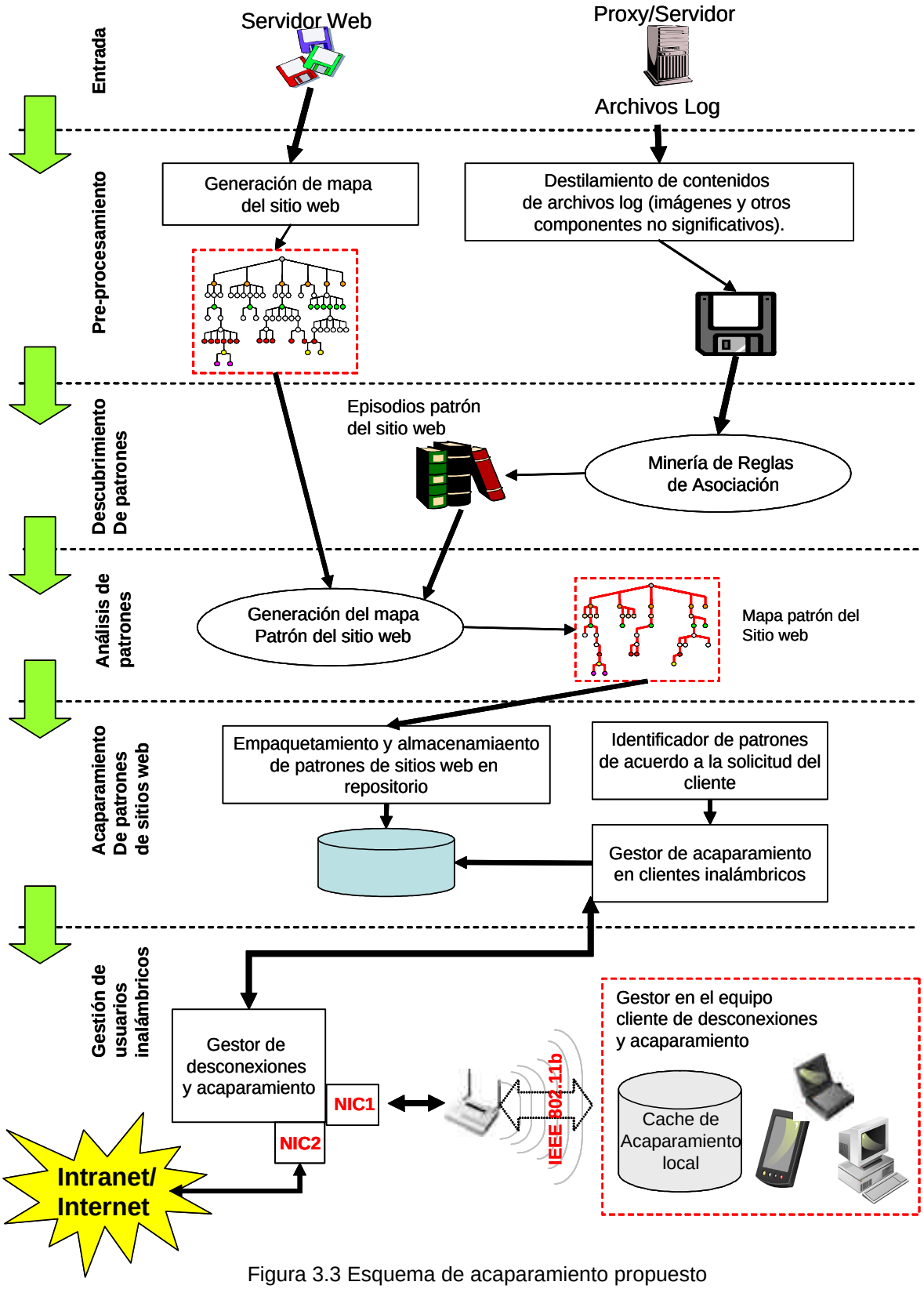


Figura 3.3 Esquema de acaparamiento propuesto

3.3.1 Validación de regla de asociación

En [31] y [32] se presentan estudios sobre las medidas de evaluación sobre reglas de asociación. Cada una de las medidas definidas, toma como base las frecuencias de acceso de cada uno de los elementos que forman la regla de asociación, así como también la frecuencia de acceso de los elementos combinados. Estos valores están dados por las siguientes expresiones.

- $fr(X \bigcup Y)$. Frecuencia del elemento derecho e izquierdo.
- $fr(X)$. Frecuencia del elemento izquierdo.
- $fr(Y)$. Frecuencia del elemento derecho.

Algunos de las medidas que se evalúan en [31] y [32], son las siguientes:

- **Confianza**. Determina la precisión de la regla ya que refleja el grado con el que los ejemplos pertenecientes a la zona del espacio delimitado por el antecedente verifican la información indicada en el consecuente de la regla. La confianza de una regla de asociación está dada por la siguiente expresión.

$$Confianza(X \rightarrow Y) = \frac{fr(X \bigcup Y)}{fr(X)} \quad (1)$$

- **Lift** *(Elevación)*. Es una medida que indica como las predicciones mejoran usando un modelo que podría ser obtenido aleatoriamente. Por ejemplo, suponiendo que un 2% de clientes contactados a través de una lista de correo comprarían un producto. Usando un modelo de minería de datos es posible seleccionar a los clientes con mayor probabilidad de compra. Suponiendo que contactando con un segmento concreto, el 10% compra un producto. Obtendríamos un *lift* de 10/2. Para una regla de asociación la medida *Lift* se expresaría de la siguiente manera.

$$Lift(X \rightarrow Y)\frac{Confianza(X \rightarrow Y)}{fr(X)} \quad (2)$$

- **Leverage** *(Influencia)*. Es una medida que se obtiene a partir de la diferencia entre el soporte del conjunto de *ítems* frecuentes que conforman la regla y un soporte esperado considerando a los elementos izquierdo y derecho independientes. Un alto *leverage* indica un alto soporte en la regla. La medida leverage estará dada por la siguiente expresión.

$$Leverage(X \rightarrow Y) = fr(X \cup Y) - (fr(x) * fr(y)) \quad (3)$$

Ya optimizado el modelo de almacenamiento, es posible describir el algoritmo presentado en [34] llamado A-priori que está diseñado para extraer reglas de asociación a partir de datos almacenados en el modelo.

A continuación se describe las operaciones involucradas en el algoritmo. Se tiene una base de datos binaria en la que cada observación supone una evaluación binaria sobre un conjunto de atributos; un ejemplo de base de datos binaria podrían ser las compras de un cliente, donde cada columna es un producto y las filas son las compras llevadas a cabo por cada cliente, la representación se puede observar en la tabla 3.2.

Tabla 3.2 Base de datos binaria

	A	B	C	D	….
T1	1	0	0	1	….
T2	0	1	1	1	….
T3	1	0	1	0	….
T4	0	0	1	0	….

Atributos de *R*

Observaciones binarias de *R*

De lo anterior se tienen las siguientes definiciones:

- R. Es el conjunto (matriz) de atributos de la base de datos, más las observaciones que cada transacción realiza; en el ejemplo de la tabla 3.2, R estaría compuesta por [A, B, C, D] más las observaciones binarias.
- X. Es cualquier subconjunto de atributos de R.
- $X Patron$. X se podría convertir en patrón si existe una sola fila en la base de datos binaria donde todos los elementos que conforman a X valen 1.
- *Frecuencia del Patrón o Soporte*. Representado por la expresión $fr(X)$ y que está dado por el cociente :

$$fr(X) = \frac{|M(X,r)|}{|r|} \quad (4)$$

Donde $|M(X,r)|$ es el número de veces que aparece X en la base de datos y $|r|$ es el tamaño de la base de datos.

Teniendo las definiciones anteriores y dado un valor mínimo de frecuencia $(\min\_ fr)$ se puede decir que el patrón X es frecuente si:

$$fr(X) >= \min\_ fr$$

$F(r, \min\_ fr)$ se refiere al conjunto de patrones que son frecuentes en la base de datos.

$$F(r, \min\_ fr) = \{X \subseteq R / fr(X) \geq \min\_ fr\} \quad (5)$$

Dada una base de datos con *R* atributos y *X* e *Y* subconjuntos de atributos con *X n Y* = Ø, una regla de asociación se define por la expresión:

$$X \rightarrow Y$$

La expresión $Conf(X \rightarrow Y)$ denota a la confianza de una regla de asociación, calculada por:

$$Confianza(X \rightarrow Y) = \frac{fr(X \cup Y)}{fr(X)} \quad (6)$$

Un valor mínimo de confianza se denota por la siguiente variable: $(conf\_\min)$. Dada una confianza mínima y una frecuencia mínima, una regla de asociación $(X \rightarrow Y)$ se cumple en si:

$$(fr(X \cup Y) \geq fr\_\min) \wedge (conf(X \rightarrow Y) \geq conf\_\min)$$

Teniendo $F_1(r, \min\_ fr)$ que es el conjunto de conjuntos frecuentes de R de longitud 1 y dado un conjunto de patrones de longitud L, los únicos candidatos a conjuntos frecuentes de longitud $L+1$ serán los que tienen subconjuntos dentro de la longitud L. Partiendo de esta definición, el cálculo de las reglas de asociación se puede realizar iterativamente desde los subconjuntos más pequeños hasta los subconjuntos más grandes. En el algoritmo original esta capacidad se pasa por alto ya que se tiene contemplado que las aplicaciones de minería de datos se ejecuten sobre sistemas con grandes recursos de cómputo, es por ello que en trabajos como [35] se propone el cálculo del conjunto potencia del conjunto R, y mediante un proceso iterativo se

eliminan aquellos elementos que no cumplan con el valor de soporte mínimo ($sop\_\min$). Teniendo en cuenta que el sistema se implanta sobre plataformas con restricciones en prestaciones, una de las propuestas derivadas de esta tesis es calcular primero los conjuntos pequeños e ir aumentando el tamaño de aquellos que cumplan con el soporte mínimo, lo cual es una modificación al algoritmo original propuesto en [35]. Suponiendo que se tiene una tabla con las transacciones que se muestran en la tabla 3.3.

Tabla 3.3. Tabla R

	A	B	C	D	E	F
T1	1	1	0	0	1	0
T2	1	0	1	0	1	1
T3	0	1	1	1	0	1
T4	1	1	1	0	0	0
T5	0	1	0	0	1	1
T6	1	0	1	1	0	0
T7	1	1	0	0	1	0
T8	1	1	0	0	0	0
T9	1	0	1	1	0	1
T10	1	1	1	0	1	1
T11	1	1	0	0	1	0
T12	1	1	0	1	1	0
T13	1	1	1	1	1	1
T14	0	1	1	0	0	0
T15	1	1	0	1	0	0
T16	1	1	0	0	1	0
T17	0	1	0	0	1	0
T18	0	1	1	0	1	0
T19	1	0	0	0	1	0
T20	1	1	0	0	1	0
	15	**16**	**9**	**6**	**13**	**6**

Siendo la tabla 3.3 la tabla R se procede a calcular las frecuencias para los ítems de longitud 1, lo que arrojaría el siguiente resultado.

Fr([A])=15/20=75% *Fr([B])=16/20=80%* *Fr([C])=9/20=45%*
Fr([D])=6/20=30% *Fr([E])=13/20=65%* *Fr([F])=6/20=30%*

Estableciendo un valor de frecuencia mínimo del 25%, entonces todos los conjuntos superan esta frecuencia, por lo tanto.

F1= {[A], [B], [C], [D], [E], [F] }

A partir de estos datos se pueden calcular los conjuntos candidatos a ser frecuentes es decir todas las combinaciones de 2 elementos.

C (F1) = {[A, B], [A, C], [A, D], [A, E], [A, F],

[B, C], [B, D], [B, E], [B, F],
[C, D], [C, E], [C, F],
[D, E], [D, F],
[E, F]}

Se procede a calcular las frecuencias de cada uno de los conjuntos generados.

Fr([A, B])=11/20=55%	*Fr([A, C])=6/20=30%*	*Fr([A, D])=5/20=25%*
Fr([A, E])=10/20=50%	*Fr([A, F])=4/20=20%*	*Fr([B, C])=6/20=30%*
Fr([B, D])=4/20=20%	*Fr([B, E])=11/20=55%*	*Fr([B, F])=4/20=20%*
Fr([C, D])=4/20=20%	*Fr([C, E])=4/20=20%*	*Fr([C, F])=5/20=25%*
Fr([D, E])=2/20=10%	*Fr([D, F])=3/20=15%*	*Fr([E, F])=4/20=20%*

Los candidatos que superan el 25% son los siguientes.

F2 = {[A, B], [A, C], [A, E], [A, D] [B, C], [B, E], [C, F]}

Tomando como medida de validez la formula 6.

Se puede probar la confianza de las siguientes reglas de asociación:

conf (A → B)= 11/15	*conf(A → E)=10/15*	*conf(B → E) =11/16*
conf (A → D)=5/15	*conf (B → A)= 11/16*	*conf(E → A)=10/13*
conf(E → B) =11/13	*conf (D → A)=5/6*	*conf (A → C)=6/15*
conf(B → C)= 6/16	*conf(C → F) =5/9*	*conf (C → A)=6/9*
conf(C → B)=6/9	*conf(F → C) =5/13*	

Si se toma como confianza mínima un porcentaje de 66%, entonces aquellas reglas que se consideran significativas, serían las siguientes.

Ras = {A → B, B → A, C → A, A → E, E → A, C → B, B → E, E → B, D → A}

Partiendo del conjunto F2, se puede calcular el conjunto de candidatos para F3.

C (F2) = {[A, B, C], [A, B, D], [A, B, E], [A, C, D], [A, C, E], [A, D, E], [B, C, D],
[B, C, E], [B, D, E], [C, D, E]}

Y calcular su frecuencia.

Fr ([A, B, C])=3/20=15%	*Fr([A, D, E])=2/20=10%*
Fr ([A, B, D])=3/20=15%	*Fr ([B, C, D])=2/20=10%*
Fr ([A, B, E])=8/20=40%	*Fr ([B, C, E])=3/20=15%*

Fr ([A, C, D])=3/20=15% *Fr ([B, D, E])=3/20=15%*
Fr ([A, C, E])=3/20=15% *Fr ([C, D, E])=1/20=5%*

De los candidatos generados, aquellos que superan el 25% son.

F3 = {[A, B, E]}

Y a partir de este conjunto se puede probar la confianza de las siguientes reglas.

Conf (A ? B → E)=8/11=72%
Conf (A ? E → B)=8/10=80%
Conf (B ? E → A)=8/11=72%

Y tomando como confianza un porcentaje de 66%, las reglas significativas serian las siguientes:

{A ? B → E, A ? E → B, B ? E → A}.

Ras = {A → B, A → E, B → A, E → A, B → E, E → B, C → B, C → A D → A}
U {A ? B → E, A ? E → B, B ? E → A}.

El resultado que se obtendría con el algoritmo propuesto en esta tesis el sistema seria el mostrado en la tabla 3.4.

Tabla 3.4. Resultado del algoritmo

ID Regla	Antecedente	Consecuente	Confianza
1	A ? B	E	72%
2	A ? E	B	80%
3	B ? E	A	72%
4	A	B	73%
5	B	A	68%
6	C	A	66%
7	A	E	66%
8	E	A	77%
9	C	B	68%
10	B	E	68%
11	E	B	84%
12	D	A	83%

3.3.2 Mecanismo de acaparamiento de sitios Web

De acuerdo al problema planteado en el punto 2.2 del capítulo 2, los eventos de desconexión en WLANs es por naturaleza un evento común, sin embargo, las desconexiones pueden ser causadas por condiciones tan variadas como: la batería del dispositivo, desvanecimiento de la señal del punto de acceso, o simplemente porque el usuario abandona el rango de alcance del punto de acceso. Lo anterior nos lleva a plantear la siguiente pregunta ¿de qué forma se le puede proporcionar continuidad a las solicitudes de páginas Web de un usuario que está propenso a desconexión? La respuesta a esta pregunta se describió en el punto 3.3, el mecanismo o estrategia propuesta para este problema está en función de los patrones generados por los algoritmos descritos en la sección 3.3.2.

Es conveniente comentar que este mecanismo es una de las principales aportaciones de esta tesis ya que hasta este momento no hemos encontrado ninguna referencia de herramientas de software comercial ni de trabajos de investigación que apliquen acaparamiento parcial de un sitio Web en dispositivos móviles de acuerdo a patrones de navegación, utilizando para esto algoritmos de minería de uso Web. Para la validación de la estrategia de acaparamiento de contenidos Web se implementó un prototipo que permite probar la viabilidad de esta estrategia.

El esquema de acaparamiento que proponemos en esta tesis se muestra en la figura 3.9, el esquema implica la verificación de

3.3.3 Interpretación de los patrones de acceso a sitios de Web

Una vez determinado un patrón de acceso utilizando minería de reglas de asociación, se deben interpretar estos patrones para realizar el acaparamiento de los recursos identificados en el cliente móvil. Para ejemplificar esto vamos a tomar los casos de prueba descritos en la sección 5.5, que se generaron del histórico de acceso del servidor de Web de la dirección URL del departamento de Ciencias Computacionales del Cenidet, http://www.sd-cenidet.com.mx/index.html, a este histórico de acceso se le aplicaron algoritmos de minería de uso de Web[2], de lo cual se obtuvieron los patrones que se muestran en la tabla 3.5. Estos patrones expresan el comportamiento de acceso observado durante un periodo aproximado de un mes. La forma en la que se interpreta el patrón es la siguiente: de acuerdo a la línea marcada con el número 1, los

[2] Tesis de maestría “Mecanismo para Predicción de Acaparamiento de Datos en Sistemas Cliente/Servidor Móviles” por David René Valenzuela Molina desarrollada en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).

usuarios que en su navegador solicitaron la página http://www.sd-cenidet.com.mx/index.html, también solicitaron las páginas que aparece en la línea 1.1, 1.2, 1.3, 1.4 y 1.5.

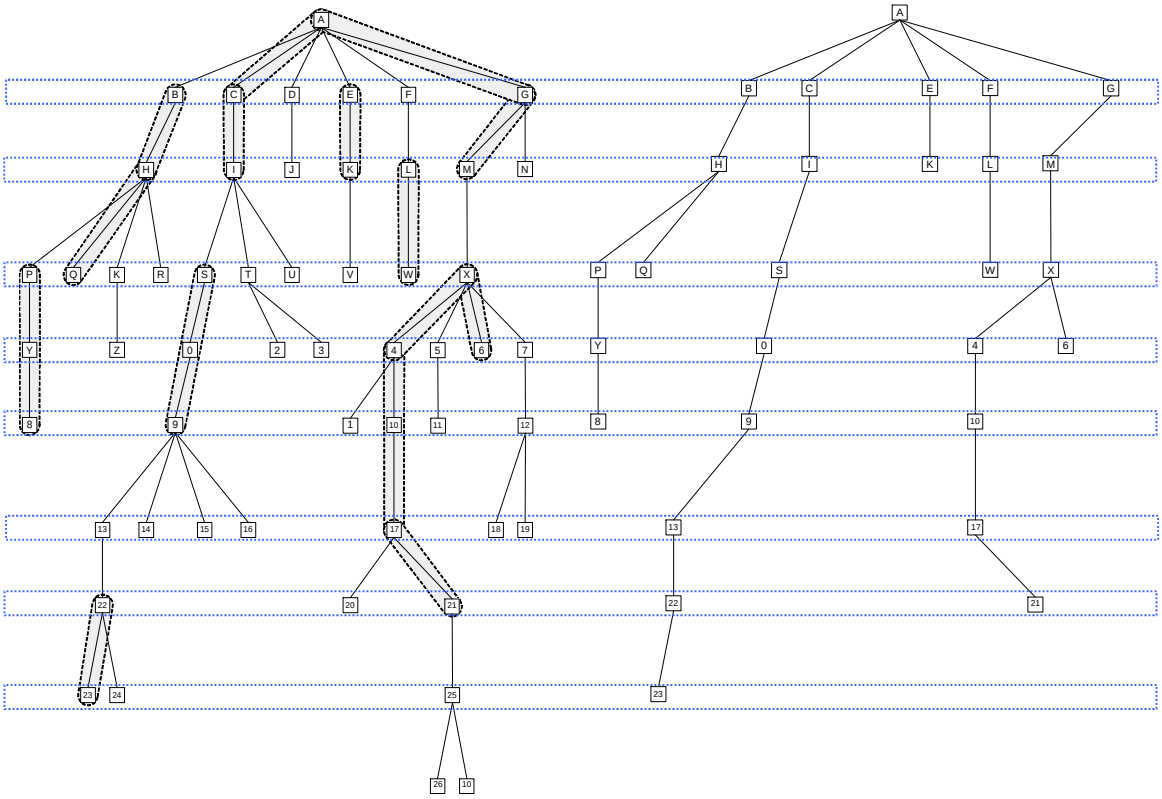
Tabla 3.5 Patrones de acceso generados.

1. [http://www.sd-cenidet.com.mx/index.html]
1.1. http://www.sd-cenidet.com.mx/Web-dcc/Indice.html
1.2. http://www.sd-cenidet.com.mx/Web-dcc/InfoAspirantes.html
1.3. http://www.sd-cenidet.com.mx/Web-sd/index.html
1.4. http://www.sd-cenidet.com.mx/Web-sd/Principal.html
1.5. http://www.sd-cenidet.com.mx/Web-sd/menu.html
2. [http://www.sd-cenidet.com.mx/Web-dcc/Indice.html]
2.1. http://www.sd-cenidet.com.mx/index.html
3. [http://www.sd-cenidet.com.mx/Web-dcc/InfoAspirantes.html]
3.1. http://www.sd-cenidet.com.mx/index.html
4. [http://www.sd-cenidet.com.mx/Web-sd/index.html]
4.1. http://www.sd-cenidet.com.mx/index.html
4.2. http://www.sd-cenidet.com.mx/Web-sd/Principal.html
4.3. http://www.sd-cenidet.com.mx/Web-sd/menu.html
5. [http://www.sd-cenidet.com.mx/Web-sd/Principal.html]
5.1. http://www.sd-cenidet.com.mx/index.html
5.2. http://www.sd-cenidet.com.mx/Web-sd/index.html
5.3. http://www.sd-cenidet.com.mx/Web-sd/menu.html
6. [http://www.sd-cenidet.com.mx/Web-sd/menu.html]
6.1. http://www.sd-cenidet.com.mx/index.html
6.2. http://www.sd-cenidet.com.mx/Web-sd/index.html
6.3. http://www.sd-cenidet.com.mx/Web-sd/Principal.html
7. [http://www.sd-cenidet.com.mx/SRCA/inicioCaptura.html]
7.3. http://www.sd-cenidet.com.mx/SRCA/aspirantesQuePagaron.html
7.4. http://www.sd-cenidet.com.mx/SRCA/formulario1.html
8. [http://www.sd-cenidet.com.mx/SRCA/aspirantesQuePagaron.html]
8.1. http://www.sd-cenidet.com.mx/SRCA/inicioCaptura.html
9. [http://www.sd-cenidet.com.mx/SRCA/formulario1.html]
9.1. http://www.sd-cenidet.com.mx/SRCA/inicioCaptura.html

Los patrones que aparecen en esta tabla nos indican la relación que existe entre la dirección de la página que se muestra en la línea 1 y las páginas que se muestran en los puntos 1.1 al 1.5. Con base en esto podemos decir que el 100% de los usuarios que visitan la página indicada en la línea 1 tienen una probabilidad definida por el soporte establecido en el algoritmo de minería de uso de la Web. Al visitar las páginas mostradas en las líneas 1.1, 1.2, 1.3, 1.4 y 1.5. Dicho de otra forma, hay una probabilidad de un x% de que el usuario que solicita la dirección mostrada en la línea 1 visite las páginas indicadas en las líneas 1.1 a la 1.5. Este mismo criterio se aplica a los patrones que se muestran en las líneas 2 a la 9.

De lo anterior podemos inferir que los patrones extraídos del histórico de accesos (ver Tabla 6.1), son un subconjunto del total de elementos que componen un sitio de Web; en otras palabras, un sitio de Web se puede esquematizar como un árbol en donde cada rama representa un archivo contenedor o un recurso final, esta estructura se define como estructura de un sitio de Web descrito en la sección 3.5. A partir de esta estructura podemos realizar una comparación entre

el árbol general de un sitio (ver Figura 3.4a) y el árbol patrón que se obtuvo de este mismo sitio (ver Figura 3.4b), el cual es un subconjunto del sitio general (árbol patrón).



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

Figura 3.4 Estructura genérica de un sitio Web

En este punto es necesario determinar ¿qué sucede con los recursos que como documentos pdf, doc, txt, archivo de imagen jpeg, jpg, gif, entre muchos otros no aparecen dentro del patrón?, pero que son necesarios para que un usuario móvil en el momento que sufra una desconexión pueda ver una página Web completa y con un formato bien definido. La respuesta no es tan complicada ya que el árbol patrón mostrado en la Figura 3.2b contiene únicamente archivos .html, los cuales son contenedores de objetos ya sean imágenes, gráficas, documentos pdf entre muchos otros, mismos que pueden ser obtenidos mediante un análisis de su contenido.

3.3.4 Mecanismo de acaparamiento

Uno de los elementos fundamentales de este trabajo de investigación, es la forma en la que se le proporciona transparencia al usuario móvil en cuanto al uso de recursos de Web

acaparados; es decir, que el usuario móvil no detecte el momento en el que está desconectado y cuando vuelve a reconectarse.

A fin de ahondar más en este asunto comenzaremos explicando la forma en la que se le proporciona dicha transparencia. Para tal efecto veamos cómo el cliente móvil manipula un recurso de Web y de qué manera lo acapara. En principio de cuentas un cliente móvil a través de su navegador de Web puede realizar diversas solicitudes.

Internamente esta petición viaja del cliente al servidor utilizando el protocolo HTPP/1.X utilizando alguno de sus métodos. En la figura 3.5 se muestra el método de generación de patrones del sitio de Web propuesto en esta tesis.

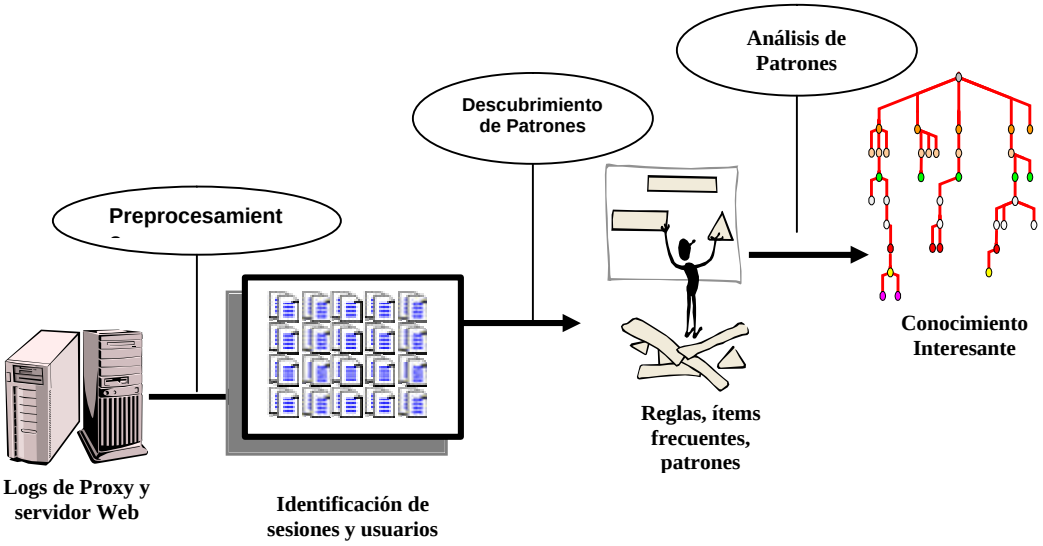


Figura 3.5 Fases de la minería de uso Web.

3.4 Metodología para la transformación de contendido de la Web para dispositivos de cómputo móvil heterogéneos.

En la figura 3.6 se muestra el proceso general de operación seguido por el servidor Transformador, desde el momento en que recibe alguna solicitud de HTTP emitida por un dispositivo PDA como un Pocket PC.

Como se observa en la estructura general de operación, se encuentra integrado el mecanismo de transformación (resaltado con líneas punteadas) el cual se describirá posteriormente en este capítulo, y de igual forma, se ilustra la funcionalidad del sistema Cache soportado por el intermediario.

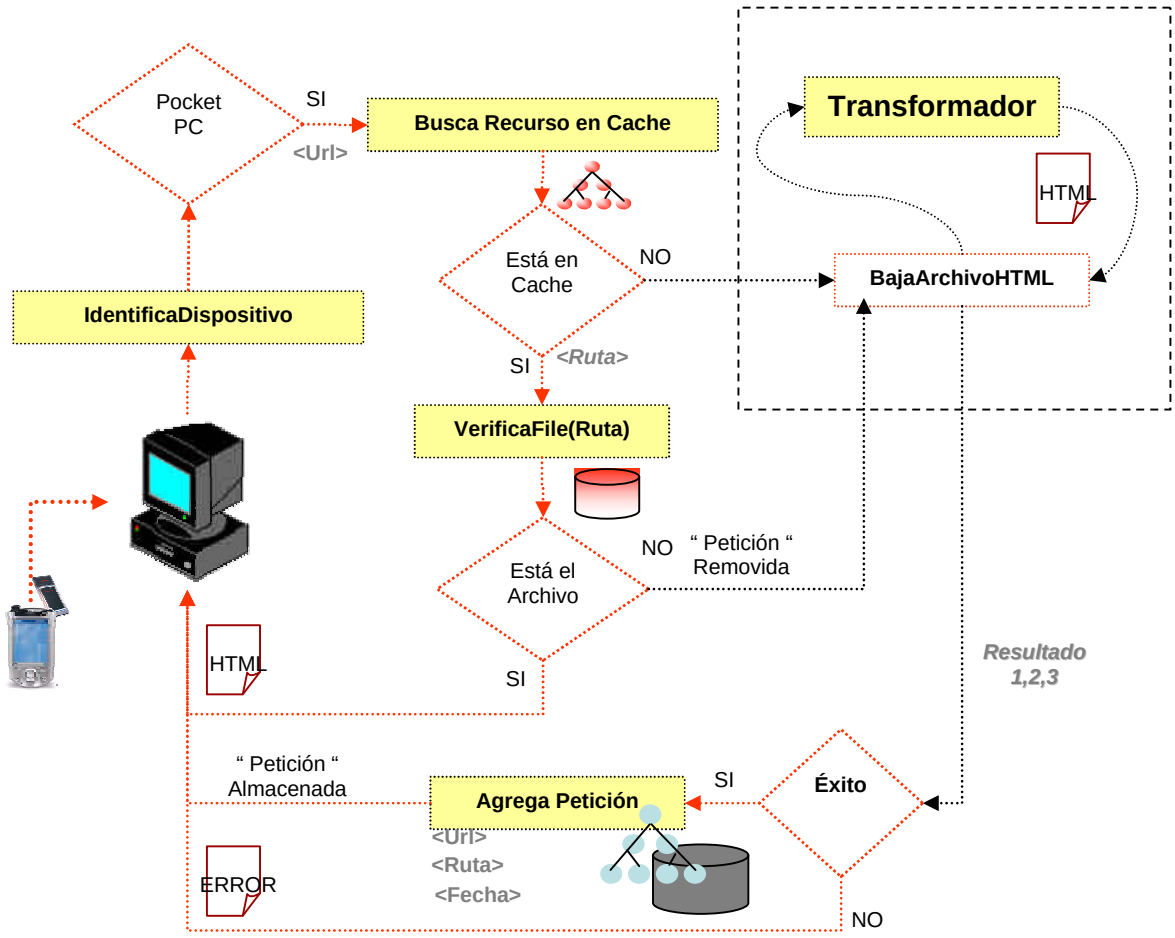


Figura 3.6. Operación del servidor transformador

De esta manera se lleva a cabo el procesamiento de todas aquellas peticiones realizadas desde los dispositivos en cuestión. Por lo tanto, todas siguen el mismo flujo de operación desde el momento en que la solicitud llega al servidor Transformador, hasta que este mismo genera un resultado.

3.4.1. Gestión de la solicitud de HTTP

El servidor Transformador como un servidor intermediario está escuchando por un puerto, el cual se ha establecido como el puerto 2700. De esta forma, recibe solicitudes que realizan los clientes desde sus navegadores, y son del siguiente tipo:

GET http://www.cenidet.edu.mx/ HTTP/1.1

En este caso, un dispositivo Pocket PC solicita el recurso señalado por el URL http://www.cenidet.edu.mx/, pero ¿cómo se sabe que se trata de un dispositivo de este tipo? El protocolo HTTP, permite enviar información adicional después de la línea que especifica el método, el URL y la versión del protocolo, mediante la cual se está tratando de recuperar el recurso. Dicha información se organiza en encabezados, siendo el resto de la petición de HTTP el que se muestra enseguida:

*Accept: */**
UA-OS: Windows CE (POCKET PC) - Version 3.0
UA-color: color16
UA-pixels: 240x320
UA-CPU: ARM SA1110
UA-Voice: FALSE
UA-Language: JavaScript
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/2.0 (compatible; MSIE 3.02; Windows CE; PPC; 240x320)
Host: www.cenidet.edu.mx
Proxy-Connection: Keep-Alive

Por tal razón, es posible determinar el tipo de dispositivo que está atendiendo el intermediario. El encabezado que contiene los datos de interés para llevar a cabo la identificación del cliente es User-Agent: valor. Como se muestra anteriormente resaltado en negritas, se tiene información acerca de la plataforma sobre la cual opera el dispositivo cliente; para este caso, se trata de un Pocket PC con Windows CE, equipado con un navegador Pocket Internet Explorer, el cual es equivalente a Microsoft Internet Explorer en su versión 3.02.

Una vez que se ha identificado el tipo de dispositivo cliente, se procede a analizar la petición para darle un seguimiento especial, diferente al que se le otorgaría a un cliente convencional. Así que, se toma la primera línea (GET http://www.cenidet.edu.mx/ HTTP/1.1) para obtener específicamente el valor correspondiente al contenido de Web solicitado. Una vez que se tiene este dato, se procede a hacer uso del servicio Cache integrado en el servidor Transformador. Por lo que enseguida se explica cómo funciona dicho sistema.

3.4.2. Esquema de funcionamiento del sistema Cache

El sistema Cache almacena páginas de Web transformadas que han sido solicitadas y servidas a dispositivos Pocket PC. Debido a que en este punto ya se tiene localizado el recurso de Web solicitado, se realiza una búsqueda de éste dentro del registro que se tiene de todos aquellos objetos almacenados en disco. Dicho registro contiene información referente acerca de cada una de las peticiones que han sido procesadas con éxito, por lo que, el recurso solicitado en cada una de éstas, fue almacenado en la Cache.

La estructura en la cual se auxilia el sistema Cache para llevar el control de sus elementos, corresponde a un documento de XML, al cual se le ha denominado cacheXML. Como se aprecia, cada item de cacheXML.xml corresponde a una petición de HTTP emitida por un dispositivo Pocket PC, para la cual se guarda el identificador del recurso (<Url>), la ruta en donde se localizará físicamente el documento (<Ruta>) y la fecha en que se ha solicitado éste (<Fecha>). La figura 3.7. muestra la estructura del documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<Cache>
  <Peticion>
    <Url>http://www.cenidet.edu.mx/</Url>
    <Ruta>/selene/proyectos/pruebas/bajados/www.cenidet.edu.mx/t-index.html</Ruta>
    <Fecha>9</Fecha>
  </Peticion>
</Cache>
```

Figura 3.7. Estructura del documento cacheXML.xml

Entonces, cada vez que se almacena un objeto en la Cache, se registra su entrada en esta estructura, por lo tanto, se van agregando nodos <Petición></Petición> con su respectiva información, equivalente a la mostrada en la figura anterior.

De esta manera, se lleva a cabo una búsqueda del recurso incluido en la petición entre esos elementos, comparando el valor del nodo <Url></Url>; si coincide con alguno de ellos, se procede a verificar si el documento existe físicamente en disco, y si es así, se recupera la página de HTML almacenada para enviarla como respuesta al cliente. De lo contrario, si el URL se encuentra registrado, pero no se localiza el recurso en disco o no se encontró ningún registro, se envía una respuesta negativa y el flujo de operación del sistema continúa, para lo cual, se reenvía la petición de HTTP al proxy Squid el cual lanzará la solicitud a la Web. Posteriormente, el servidor Transformador espera por una respuesta. En el momento en que logra recuperar el contenido de Web, enseguida solicita la ejecución del proceso de transformación.

3.4.3. Mecanismo de transformación de contenidos de Web

El diseño general de la estrategia para la transformación de páginas de Web, involucra una serie de fases, mediante las cuales se aplica una especie de preprocesamiento y reorganización al documento original. Para entender mejor esto, se pudo observar la figura 3.8, en la cual se muestra un esquema representativo del proceso que se sigue durante la etapa de transformación para obtener un nuevo documento de HTML.

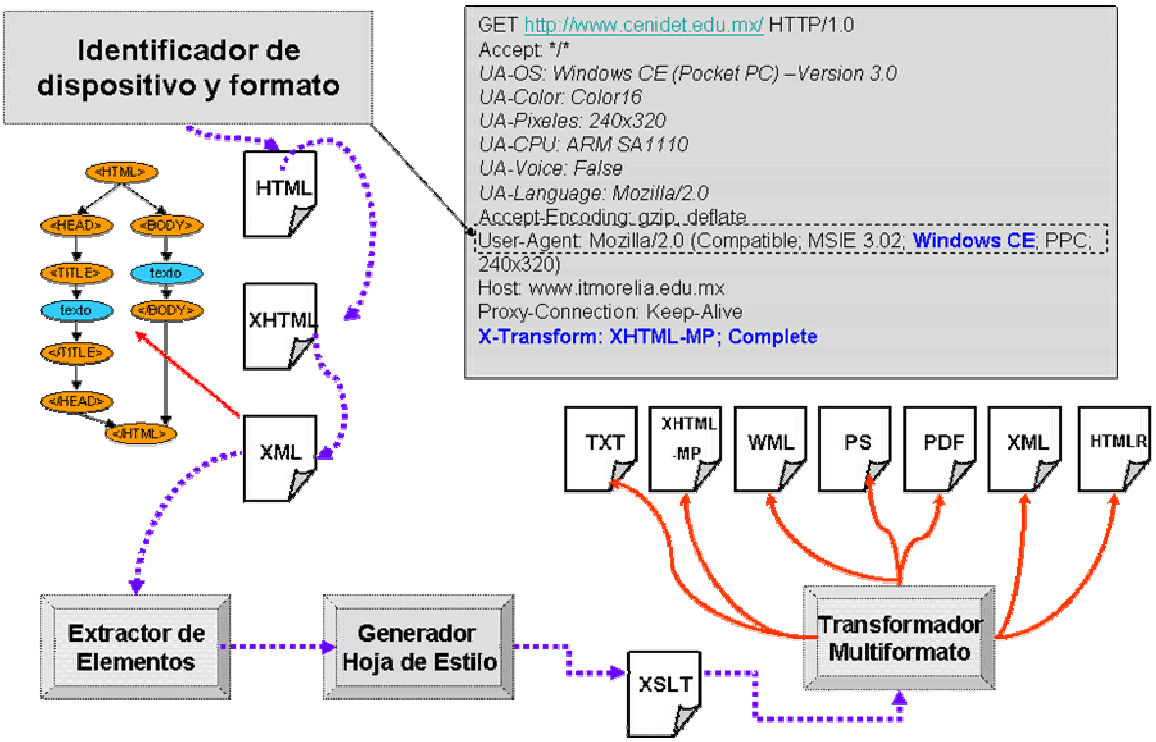


Figura 3.8. Esquema general del mecanismo de transformación

El mecanismo que se sigue para la transformación de los documentos, incluye como primer paso una fase que se denomina, Conversión de XHTML. En este punto se analiza la estructura de la página de Web y se aplican las reglas de sintaxis del lenguaje XML, con el único fin de obtener un documento de HTML bien formado. Por lo cual, se utiliza un depurador de código en HTML denominado JTidy3, el cual toma como entrada el contenido original, y como resultado, se obtiene dicho contenido con una estructura más clara y definida, que caracteriza a un documento de XHTML, lo cual significa que el contenido de Web sigue estando escrito en el lenguaje HTML, pero siguiendo las reglas de XML.

El código que se presenta en la figura 3.9 muestra algunos de los resultados de este proceso, ya que las etiquetas tales como
, <img>, <p>, etc. experimentan cierta transformación durante dicha fase. Esto se debe a que al escribir código en HTML no es necesario una etiqueta de cierre para éstas; en cambio el lenguaje XHTML no permite etiquetas como las anteriores. De este modo toda etiqueta
 que se localice en el archivo HTML se cambia por
, así como <p> se convierte a <p />, etc.

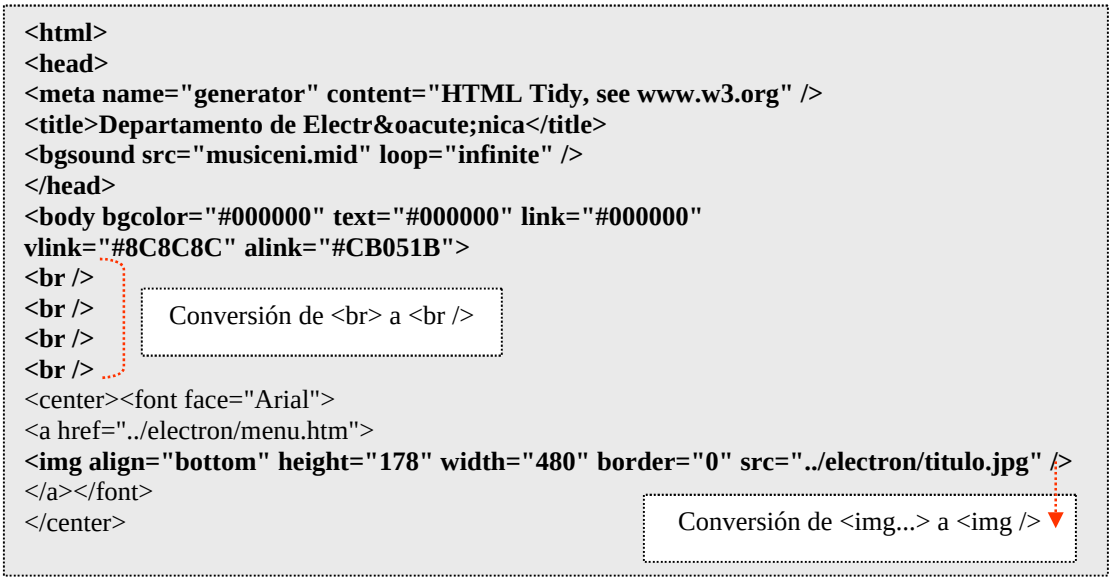


Figura 3.9. Transformación de la fase Convertidor de XHTML

Una vez que se ha logrado el paso anterior, el proceso continúa con la fase número dos, Analizador de HTML. En donde se utiliza un analizador de código en HTML, y se implementa una estrategia para obtener un documento de XML, con la finalidad de estandarizar el contenido para hacer uso de una tecnología que permita rediseñar el documento que se está tratando. De esta

[3] JTidy. Herramienta que permite comprobar la sintaxis de un documento HTML.

forma, ya que se tiene el código de HTML en memoria, se establece una rutina para leer su contenido, explorando cada uno de sus elementos, y al mismo tiempo se va creando un documento de XML, con los elementos que lo conforman. Por tal motivo, ahora el documento debe incluir la cabecera de un archivo de XML como lo muestra la figura 3.10.

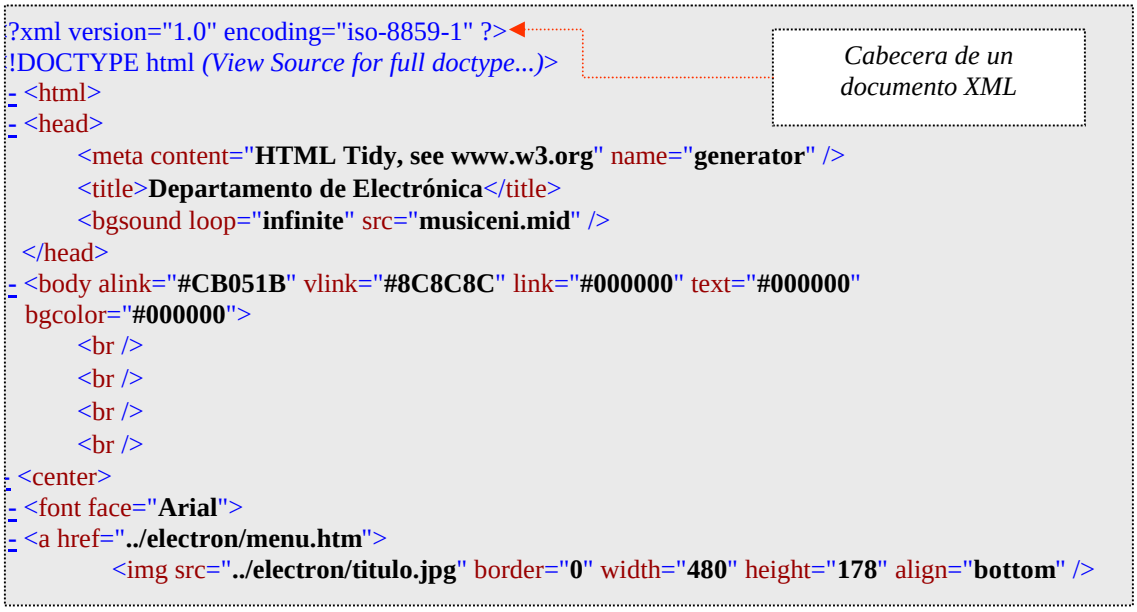


Figura 3.10. Transformación de la fase Analizador de HTML

En este momento, ya se está en condiciones de manipular su contenido con una mayor flexibilidad, así que entra a la fase Analizador de XML. Es aquí donde el documento se pasa a una estructura en forma de árbol, permitiendo entrar a la fase final del proceso, donde realmente se lleva a cabo la transformación de la página de HTML. ¿Cuál es la función de la fase Reformateador dentro del mecanismo de transformación? ¿Cómo se logra obtener de nuevo el documento de HTML? ¿Con qué características se crea la nueva versión del contenido original? Se procede a ver cuál es el flujo de operación de esta fase para encontrar las respuestas a las preguntas anteriores. La figura 3.11 muestra un esquema donde se representa de forma general la tarea que se realiza durante esta fase.

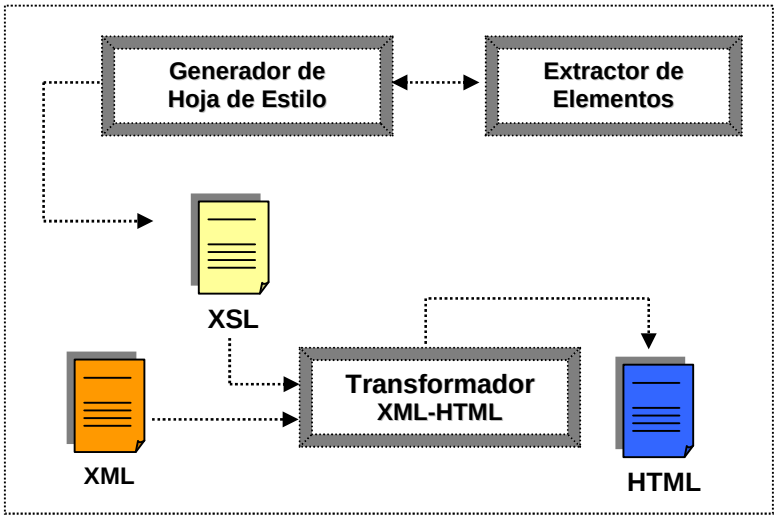


Figura 3.11. Esquema de la fase Reformateador

En primer lugar, el objetivo que se persigue en esta etapa final, es darle un formato a la información que se tiene en una estructura de un documento de XML. Así que, la solución planteada incluye un Generador de Hoja de Estilo. ¿Qué significa esto? Como ya se ha mencionado, una vez que los datos se tienen en formato XML, se aplicará un formato de presentación a los mismos, ya que por sí solos no poseen el estilo de una página de Web, sino que son puramente texto. Por esta razón, se tiene que crear la cara con la que un navegador visualiza dicha información.

Es por eso, que se realiza un análisis del contenido que se tiene como resultado de las fases anteriores, mediante una actividad que se llama Extractor de Elementos, donde se extraen valores que ayudan a determinar la composición del nuevo documento de HTML. Como se observa en la figura 3.11, existe una retroalimentación en la parte de la generación de la hoja de estilo y la extracción de los elementos analizados.

Durante la creación del formato que se va a dar a la información contenida en el documento de XML, se logra obtener un esquema que permite reorganizar los elementos originales, para lo cual se han agrupado por tipo de elemento. De tal forma que aparecerá primero el texto de la página de Web, seguido de un grupo de imágenes en miniatura, y al final se muestran los enlaces que se hayan obtenido durante el análisis. En la figura 3.12 se observa la hoja de estilo generada durante esta etapa.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <xsl:stylesheet version="1.0"  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
- <xsl:template match="html">
- <html>
- <head>
        <title>Página Web para PDAs</title>
    </head>
- <body bgcolor="#FFFFFF">
- <table align="center" border="2">
- <tr bgcolor="white" align="justify">
        <th>Transformador de Contenido Web para PDAs</th>
    </tr>
    </table>
+ <center>
        </body>
    </html>
    </xsl:template>
    </xsl:stylesheet>
```

Figura 3.12. Fase Generador de Hoja de Estilo

Por un lado sólo se tiene el documento de XML, y por otro la hoja de estilo en un documento de XSL por separado. Por tal motivo existe el Transformador XML-HTML ilustrado en el esquema. Se encarga de procesar ambos documentos para obtener nuevamente una página de HTML, pero ahora con características que permiten visualizarla de una manera más sencilla en un navegador de un dispositivo con una pantalla de dimensiones reducidas.

3.4.4. Gestión de la respuesta de HTTP al cliente

Una vez que ya se tiene el resultado del mecanismo de transformación, ¿de qué manera se da respuesta a la petición del cliente que solicitó dicho recurso? Primeramente, se tiene que aprovechar el sistema Cache proporcionado. Para esto es conveniente guardar la página de Web transformada, así que se gestiona el almacenamiento en disco, y posteriormente se procede a construir una respuesta de HTTP, por lo tanto, se construye una cabecera con el formato que se muestra a continuación; por ejemplo:

HTTP 1.1 200 OK
MIME-version: 1.0
Content-type: text/plain
Content-length : 38

<html><body>
617-555-6789
</body></html>

En la respuesta se observa una primera línea, la cual indica que se está trabajando con el protocolo HTTP en su versión 1.1, seguido de un código de estado (200 OK) que indica que la petición se llevó a cabo con éxito, para otros casos existe una diversidad de códigos establecidos dentro de la especificación del protocolo HTTP. Enseguida, se tienen otros encabezados con información referente al tipo y al tamaño en bytes del documento. Finalmente se envía el cuerpo del mensaje, es decir, el contenido de la página de Web a través del canal de datos que se estableció al momento de conseguirse la conexión entre el cliente y el servidor Transformador.

Utilizando el lenguaje de modelado UML4 se presenta el diagrama de secuencias, en la figura 3.13., el cual muestra la operación general del prototipo construido en esta tesis.

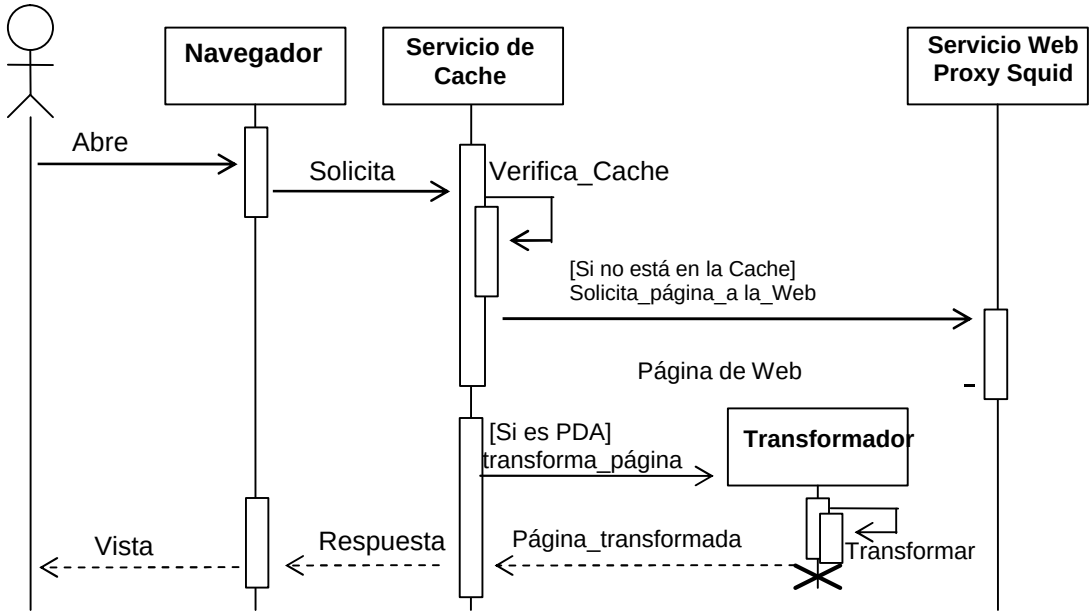


Figura 3.13. Diagrama de secuencias

[4] UML (Lenguaje de Modelado Unificado). Lenguaje gráfico para visualizar cada una de las partes que comprende el desarrollo de software.

Capítulo 4

Validación de la Metodología de Solución

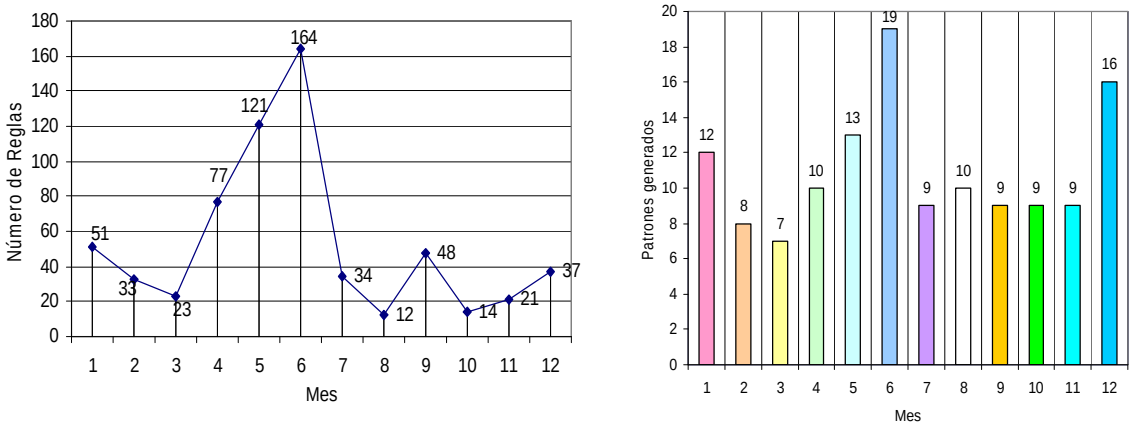
En el capítulo cuatro se describen los resultados que permiten validar la metodología de solución propuesta. Se describen los casos de prueba, el diseño del experimento y los resultados en las tres áreas que consideramos para esta tesis: manejo de desconexiones en entornos de cómputo móvil, generación de patrones de acceso a sitios Web y transformación de contendido Web para dispositivos con áreas de despliegue limitadas.

El plan de pruebas se dividió en 3 secciones, en la primer sección se muestran las pruebas realizadas sobre las funciones generales del generador de patrones de acceso a sitios Web, en la segunda sección se muestran las pruebas realizadas para el manejo de desconexiones en entornos de cómputo móvil y en la tercera sección se muestran las pruebas realizadas al transformador de contendido Web para dispositivos con áreas de despliegue limitadas.

4.1 Generación de patrones de acceso a sitios Web

Se utilizaron archivos log del año 2005 para las pruebas de generación de patrones de acceso a sitios Web, para estas pruebas se recopilo un año de bitácora de acceso al sitio Web www.cenidet.edu.mx, en esta tabla se detalla el tamaño de los archivo, el formato, el número de líneas, el dominio o la red propietaria, la fecha de inicio y la fecha final.

Las pruebas que se realizaron para extraer patrones de acceso de los archivos log del 2005 se agruparon en tres cuatrimestres. Las pruebas se desarrollaron aplicando un soporte mínimo de 5% y una confianza mínima de 10%. El método utilizado para las sesiones de usuario se obtuvieron por sesión, este método determina la sesión de usuario de manera heurística, es decir, las sesiones de usuario se crearon de acuerdo a criterios como: navegador, relación entre vista de página, episodio, tiempo del episodio y liga de referencia. En la figura 4.1a se muestra el total de reglas de asociación obtenidas mensualmente. Los patrones generados para el análisis de estos doce meses del 2005 se muestran en el anexo A.



a) Total de reglas obtenidas por mes b) Total de patrones obtenidos por mes

Figura 4.1 Estadísticas del proceso de minería de uso Web del 2005

Es evidente que los meses de abril, mayo y junio fueron los meses en que se obtuvieron un mayor número de reglas de asociación, lo cual no implica que se obtenga un mayor número de patrones de acceso, en la figura 4.2b se muestra el total de patrones de navegación obtenidos por

mes. En esta figura se puede observar que el mes con mayor número de patrones de navegación fue el mes de junio, seguido por diciembre, los patrones obtenidos por mes se analizan por cuatrimestre. Para este análisis se muestran dos gráficas estadísticas, en la primera gráfica se clasifican las reglas de asociación generadas en clases, es decir, se agrupan de acuerdo a la confianza obtenida en el proceso de minería de uso Web. Es decir, se clasificaron por grupos de acuerdo a su probabilidad. Estas gráficas se agrupan por cuatrimestre y se analizan con detalle en la sección 4.1.2.

4.1.1. Análisis de resultados del proceso de minería de uso Web

Para realizar este análisis se agruparon los resultados en tres grupos de cuatro meses cada uno. Es interesante observar los resultados comparativos por cuatrimestre, para este análisis agrupamos las gráficas de los patrones generados por cuatrimestre, en cada gráfica se muestra la probabilidad estimada de cada uno de los patrones obtenidos por mes, es decir para cada patrón se obtuvo la probabilidad de ser solicitado en posteriores visitas. En la figura 4.2 se muestra los patrones obtenidos en los meses de enero, febrero, marzo y abril de 2005, estos patrones muestran una similitud en función de las solicitudes presentadas en estos cuatro meses, es evidente que los patrones con mayor probabilidad son el 1 y 6.

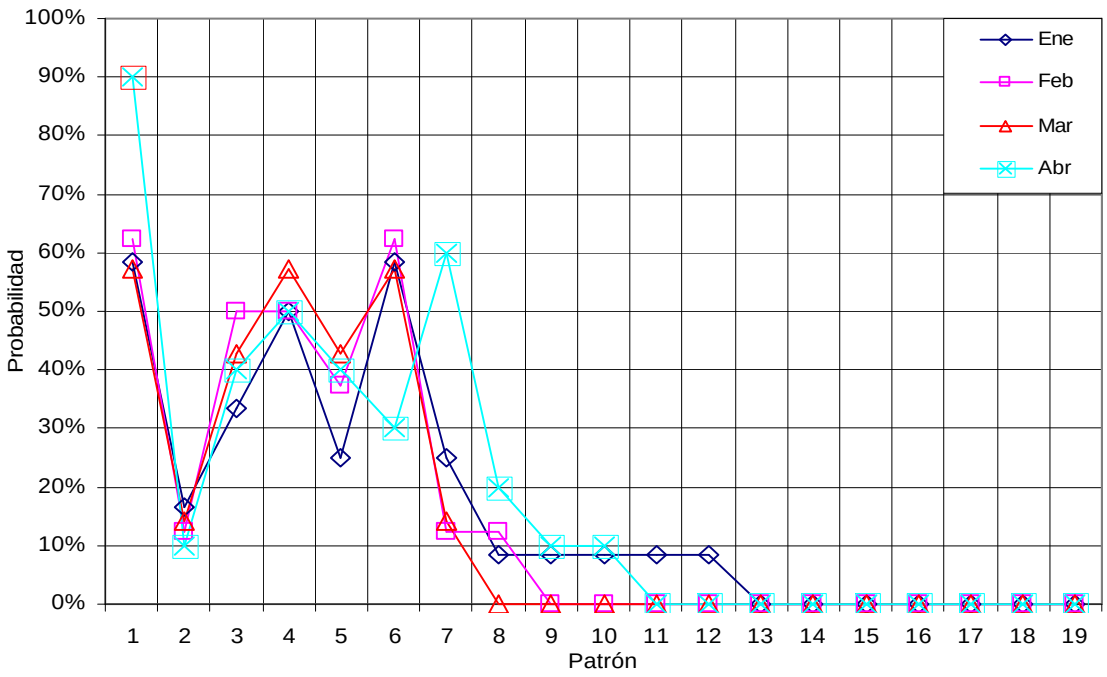


Figura 4.2 Patrones obtenidos en el primer cuatrimestre

Analizando los datos de la gráfica podemos concluir que la probabilidad promedio fue de entre el 60 y 70 por ciento a excepción del mes de abril en donde la probabilidad se promedio entre

el 70 y 90 por ciento, es decir un 42% de las reglas generadas se ubicaron dentro de estas dos clases lo cual nos indica que las sesiones de usuario en este mes fueron más constantes en relación a la información solicitada. En relación a esto podemos inferir que los patrones de navegación con mayo probabilidad en este cuatrimestre fueron los patrones 1, 4 y 6 ya que presentan una probabilidad mayor o igual al 50%, es decir, en función de la gráfica de la figura 4.2 hay entre un 70 y 90 por ciento de probabilidad de que un usuario solicite estas páginas en una sesión.

En la figura 4.3 se muestra los patrones obtenidos en los meses de mayo, junio, julio y agosto de 2005, se puede observar que los patrones muestran diferencias evidentes en relación a las solicitudes presentadas en estos cuatro meses, en este punto es complicado identificar claramente los patrones de navegación que se generaron en este período..

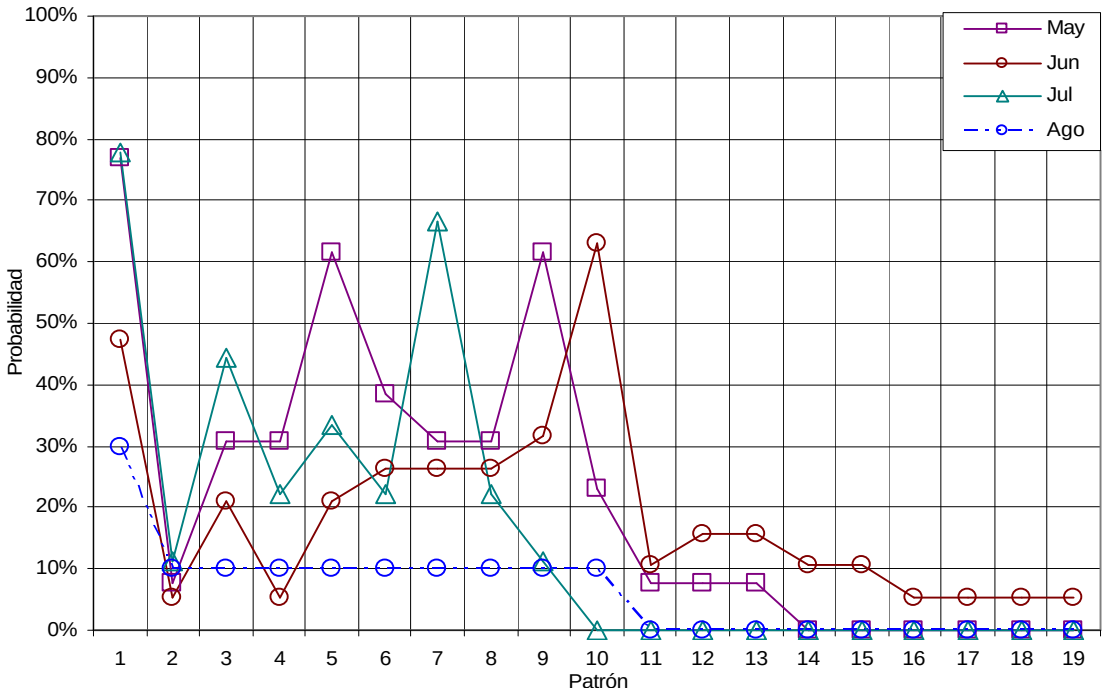


Figura 4.3 Patrones obtenidos en el segundo cuatrimestre

Analizando los datos de la gráfica de la figura 4.3 podemos concluir que la probabilidad promedio fue de entre el 60 y 70 por ciento, en los meses de abril y mayo, por otro lado la probabilidad promedio de los meses de junio y julio fue de entre el 50 y 60 por ciento, es decir un 20% de las reglas generadas se ubicaron dentro de esta clase lo cual nos indica que las sesiones de usuario en este mes fueron poco constantes en relación a la información solicitada.

En función de este análisis podemos inferir que los patrones de navegación con mayor probabilidad en este cuatrimestre fueron los patrones 1, 5, 7, 9 y 10 ya que presentan una

probabilidad mayor o igual al 50%, es decir, hay entre un 70 y 90 por ciento de probabilidad de que un usuario solicite estas páginas en una sesión. En la figura 4.4 se muestra los patrones obtenidos en los meses de septiembre, octubre, noviembre y diciembre de 2005, en esta gráfica no se identifican patrones evidentes en las solicitudes presentadas en estos cuatro meses, es evidente que los patrones con mayor probabilidad son el 1 y 7.

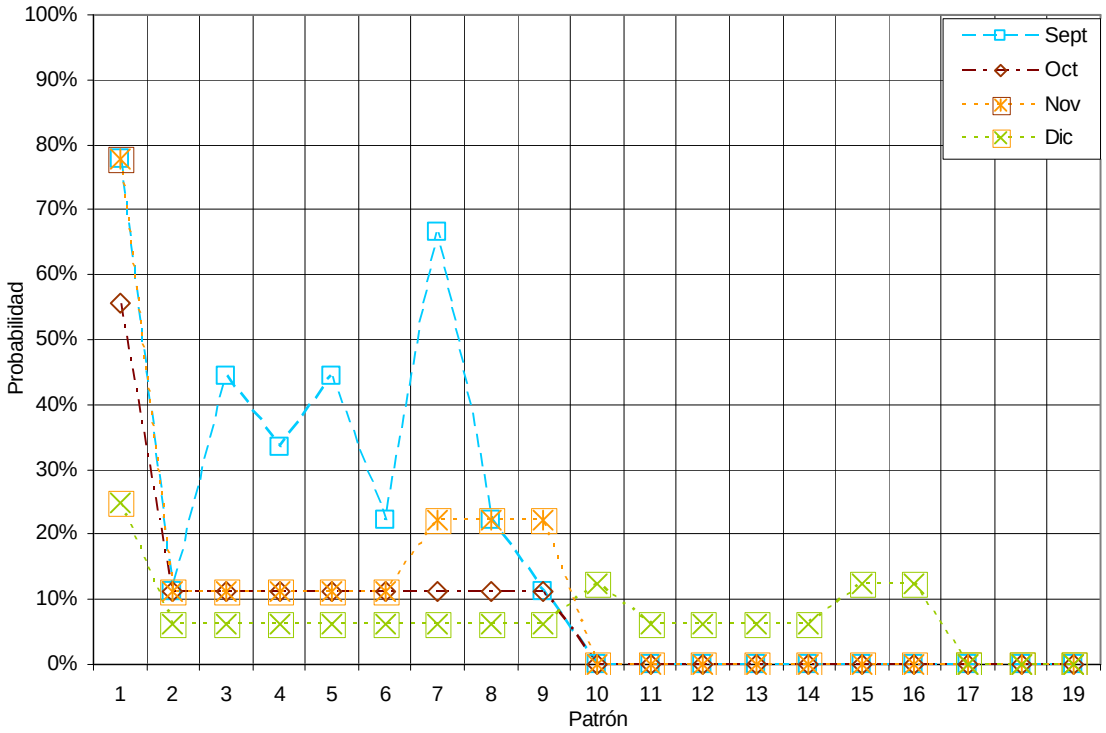


Figura 4.4 Patrones obtenidos en el tercer cuatrimestre

Analizando los datos de la gráfica de la figura 4.4 podemos concluir que la probabilidad promedio fue de entre el 30 y 50 por ciento, es decir un 42% de las reglas generadas se ubicaron dentro de estas clases lo cual nos indica que las sesiones de usuario en este mes fueron poco constantes en relación a la información solicitada.

Con relación a esto podemos inferir que los patrones de navegación con mayor probabilidad en este cuatrimestre fueron los patrones 1 y 7 ya que presentan una probabilidad mayor o igual al 50%, es decir, en función de la gráfica de la figura 4.4 hay entre un 70 y 90 por ciento de probabilidad de que un usuario solicite estas páginas en una sesión.

4.2 Transformación de contendido Web para dispositivos móviles heterogéneos

Estas pruebas tienen como objetivo verificar la metodología de solución propuesta en esta tesis. Por lo tanto, se analizó y se estableció un plan de pruebas adecuado, por medio del cual fuera posible explorar los diferentes escenarios sobre los que se puede operar.

En esta evaluación en particular, se analizaron diferentes criterios para comprobar la validez del prototipo desarrollado, es por esto que en el resto del capítulo se explicará con más detalle la ejecución del plan de pruebas.

4.2.1 Escenarios de prueba

Con el objetivo de llevar a cabo el plan de pruebas, se establecieron dos escenarios sobre los cuales se desarrollaron cada uno de los casos previstos.

4.2.1.1 Escenario del plan de pruebas A

En este escenario participan por un lado los usuarios con dispositivos equipados para acceder a Internet y del otro, la Web donde se encuentra un conjunto de servidores, los cuales proporcionan el contenido solicitado por los clientes: dispositivos PDA o computadoras personales que trabajan bajo plataformas convencionales. Por lo tanto, estos clientes emitirán peticiones a Internet de manera directa como se muestra en la figura 4.5.

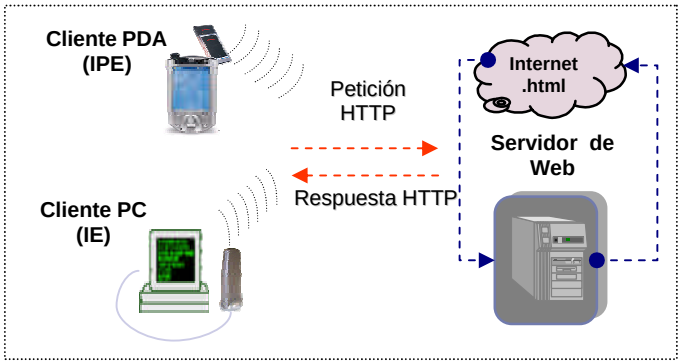


Figura 4.5. Escenario de prueba (Cliente-Internet)

4.2.1.2 Escenario del plan de pruebas B

Como segundo escenario se tiene un arreglo *Cliente – Intermediario – Internet.* Éste involucra a todos aquellos clientes que realizarán solicitudes a la Web a través de un servidor proxy o intermediario al que están enlazados. En este caso en particular, el intermediario corresponde al servidor que proporciona los servicios de transformación para dispositivos Pocket

PC. Pero al mismo tiempo resuelve las peticiones de clientes convencionales como se puede observar en la figura 4.6.

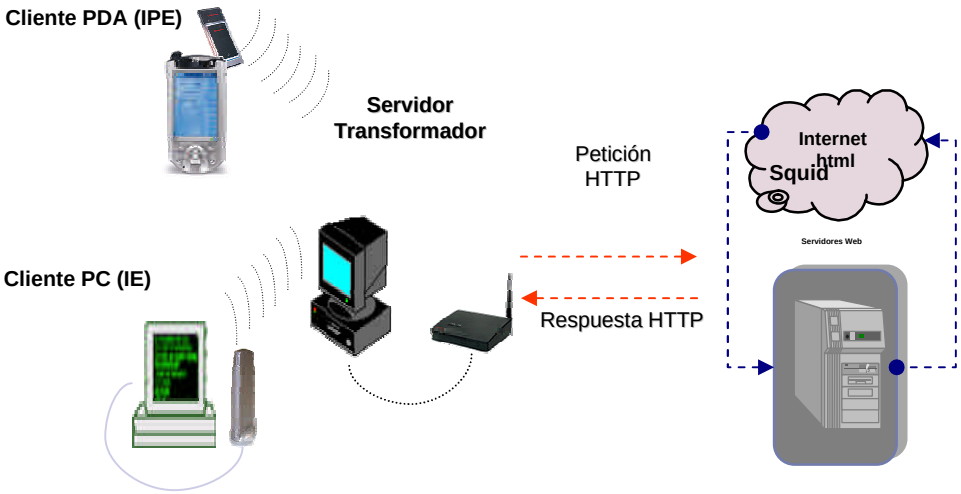


Figura 4.6 Escenario de prueba (Cliente-Intermediario-Internet)

4.2.1.3 Descripción del plan de pruebas

Las pruebas a realizar se definieron para diferentes casos de uso, en los cuales intervienen diferentes dispositivos, estos casos se describen en los siguientes puntos:

- **Caso 1**: Un cliente solicita una página Web directamente a Internet desde una PC convencional.
- **Caso 2**: Un cliente solicita una página Web directamente a Internet desde un Pocket PC.
- **Caso 3**: Un cliente solicita una página Web a través del servidor Transformador desde una PC convencional.
- **Caso 4**: Un cliente solicita una página Web a través del servidor Transformador desde un Pocket PC, y la página no se encuentra en la cache.
- **Caso 5**: Un cliente solicita una página Web a través del servidor Transformador desde un Pocket PC, y la página se encuentra en la cache.

4.2.2 Evaluación experimental

4.2.2.1 Primer escenario de pruebas

Para probar los casos planteados sobre este escenario, es necesario solicitar un recurso Web desde un navegador sin establecer ningún servidor proxy en su configuración, ya sea que se trate de un navegador integrado en un cliente convencional o un navegador como el Internet Pocket Explorer incluido en los dispositivos Pocket PC.

Por lo tanto, los casos de prueba que se sitúan en este escenario, corresponden al caso 1 y caso 2 del plan de pruebas: un cliente (PC convencional o Pocket PC) solicita una página Web de forma directa a Internet. En este contexto, ambos clientes interactuaron con los servidores de Web a través de la red, los cuales poseen el recurso solicitado por éstos. Así que, como resultado de las peticiones HTTP, se obtuvo el contenido Web solicitado en su versión original, indistintamente de la plataforma que realizó la solicitud, debido a que los servidores Web no incluyen mecanismo alguno que les permita enviar una respuesta personalizada. Esto se puede observar en la figura 4.7.

Figura 4.7 Vista original en distintas plataformas

4.2.2.2 Segundo escenario de pruebas

Para ejecutar las pruebas sobre el segundo escenario, se debe realizar el siguiente procedimiento:

1. Establecer los parámetros necesarios en el archivo de configuración *configServer.cfg,* para inicializar el servicio con valores correspondientes al sistema de archivos de la máquina en donde se va a ejecutar el servidor. Los parámetros se refieren a rutas de archivos que el sistema requiere para gestionar adecuadamente cada una de las peticiones HTTP recibidas, además de información necesaria para realizar el enlace con el servidor proxy tales como su IP y el puerto en donde éste escucha peticiones. La figura 4.8 muestra la lista de los parámetros que se han mencionado en el párrafo anterior, como parte de la configuración del sistema.

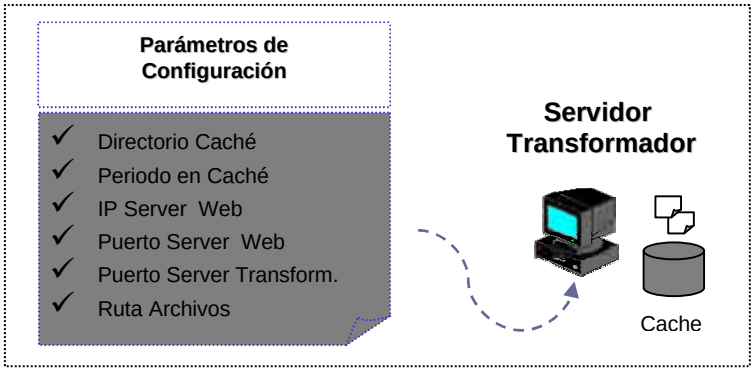


Figura 4.8 Parámetros de configuración para inicializar el servidor Transformador

2. Una vez que se tienen los valores del punto anterior, se procede a inicializar el servidor Transformador para atender las peticiones de los clientes, como se observa en la figura 4.9.

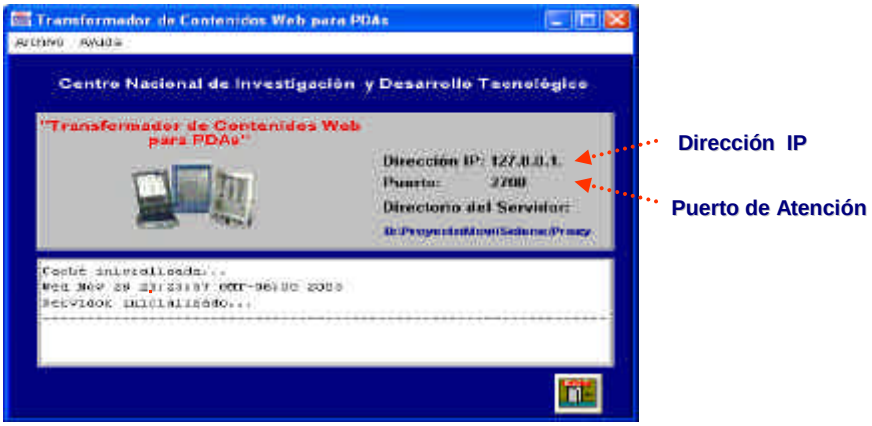


Figura 4.9. Interfaz del servidor Transformador inicializado.

3. Configurar el navegador en conexiones de red, para utilizar un servidor proxy, para lo cual, se establecen la dirección IP y el puerto donde se está ejecutando el servicio que proporciona el servidor Transformador.

4.2.2.3 Tercer escenario de pruebas

El objetivo del tercer escenario de prueba es comprobar que el servidor Transformador atiende las peticiones de clientes clasificados dentro de la categoría de las PC convencionales. El servidor identifica a estos clientes como un tipo de dispositivo que no requiere de los servicios de transformación, por lo cual, sólo reenvía la petición realizada por la PC a Squid, y posteriormente éste se encarga de gestionar la solicitud hacia la Web. Finalmente, el servidor Transformador espera el contenido c Web recuperado por el proxy cache para enviar el recurso original al dispositivo cliente que lo solicitó.

Los resultados de las pruebas reflejan los recursos obtenidos por clientes que se encuentran operando sobre diferentes plataformas pero dentro de la categoría de las computadoras personales. En ambos casos se utilizó el servidor Transformador como servidor proxy, mediante la configuración correspondiente en las opciones de red del navegador, estableciendo la dirección *IP* de la máquina donde está ejecutándose el servidor y el número de *puerto* donde atiende las peticiones.

Siguiendo el plan de pruebas, para analizar los casos 4 y 5, es necesario evaluar el comportamiento del sistema cache integrado en el servidor. Por lo tanto, en la figura 4.10 se muestran los resultados que se obtuvieron al inicializar el servidor Transformador. En esta fase se observa la salida del análisis interno que se lleva a cabo para determinar qué elementos forman parte de la cache; así que se listan los recursos almacenados y el período, correspondiente al número de días en que dicho recurso no ha sido solicitado.

```
Depurando...
http://www.aragoneria.com/natural/geologia/dinosaur.htm-- periodo: 1
http://www.aragoneria.com/natural/flora/arvegeta.htm-- periodo: 0
http://www.aragoneria.com/teruel/maesini.htm-- periodo: 1
http://www.aragoneria.com/trural.htm-- periodo: 1
http://www.ciudadfutura.com/entreollasymates/Light/lightlst.htm-- periodo: 1
http://www.aragoneria.com/teruel/teruelpr.htm-- periodo: 1
http://www.aragoneria.com/natural/hongos/index.htm-- periodo: 0
http://www.computer.org/students/looking/summer97/ieee802.htm-- periodo: 0
http://www.aragoneria.com/natural/geologia/index.htm-- periodo: 0
Sun Nov 30 16:52:35 GMT-06:00 2003
Servidor inicializado...
esperando....
---------------------------------------------------------------

Atendiendo petición!
```

Figura 4.10. Inicialización del sistema cache sin depuración

De acuerdo a la figura 4.10, el proceso de análisis de recursos involucra la evaluación de la estructura física del sistema cache respecto al medio de almacenamiento; para esto, se analizan los datos en formato XML, los cuales guardan información referente a cada uno de los elementos de la cache como se refleja en la figura 4.11.

Figura 4.11. Estructura física del documento cacheXML.xml

En este caso en particular, se observa que los datos que se almacenan respecto a cada una de las peticiones que forman parte de la cache, sirven para identificar físicamente al recurso almacenado. Por lo anterior, el conjunto de información generada durante la manipulación de las peticiones HTTP. Así que, a cada petición almacenada en formato XML, le corresponde un elemento en el sistema de directorios del servidor.

Por ejemplo, después de la recuperación y transformación del documento HTML solicitado en la petición *http://www.cenidet.edu.mx,* se ha explorado la cache para observar el comportamiento del mismo. Es por eso que, después de que la solicitud se resolvió con éxito, el recurso transformado se ha denominado para este caso *t-index.html,* el cual queda almacenado en disco para peticiones posteriores. Ahora, para comprobar el sistema de depuración integrado en la cache, se estableció el parámetro *período en la cache* con un valor de *10,* lo cual significa darle un tiempo de vida de 10 días a los recursos que se iban almacenando. Después de inicializar el servidor Transformador para evaluar este caso, se obtuvo como resultado lo que se presenta en la figura 4.12, que se refiere al segundo caso de inicialización del servidor donde se lleva a cabo el

proceso de depuración de recursos, es decir, aquéllos que han expirado su período de vida en la cache. Una vez que se ha evaluado esta parte, se continúa con el análisis de los dos últimos casos planteados en el plan de pruebas.

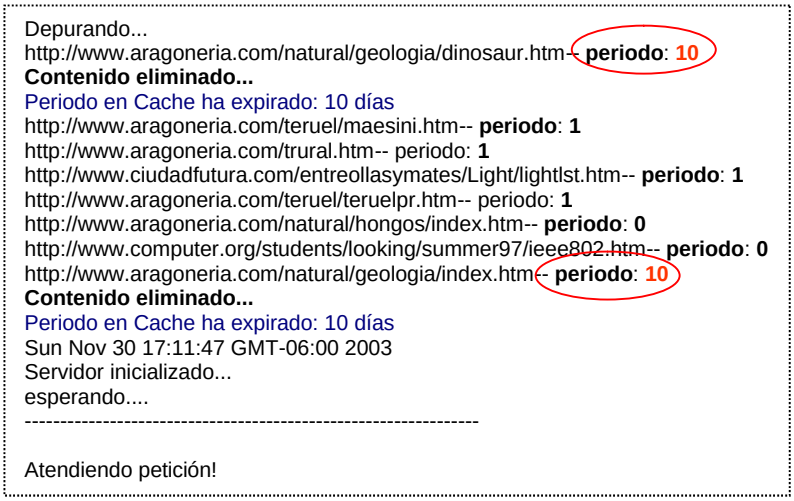


Figura 4.12. Inicialización del sistema Cache con depuración

4.2.2.4 Cuarto escenario de pruebas

El objetivo del cuarto escenario es verificar que se lleve a cabo la transformación del contenido Web solicitado, ya que se trata de una petición originada por un dispositivo Pocket PC, en donde dicha petición no se encuentra disponible en el sistema cache que proporciona el servidor Transformador. Para este caso en especial, el contenido Web se recupera de Internet ya que no se encuentra en la cache, por lo que una vez que se obtiene a través del proxy Squid, pasa al servidor Transformador donde entra en un proceso de transformación y posteriormente se almacena en disco para solicitudes posteriores. En primer lugar, la figura 4.13 muestra el reporte de estado de la cache, generado al inicializar el servidor Transformador antes de realizar las solicitudes por parte del cliente. En dicho listado se observa que el contenido Web requerido por el usuario, no se encuentra aún almacenado en la cache.

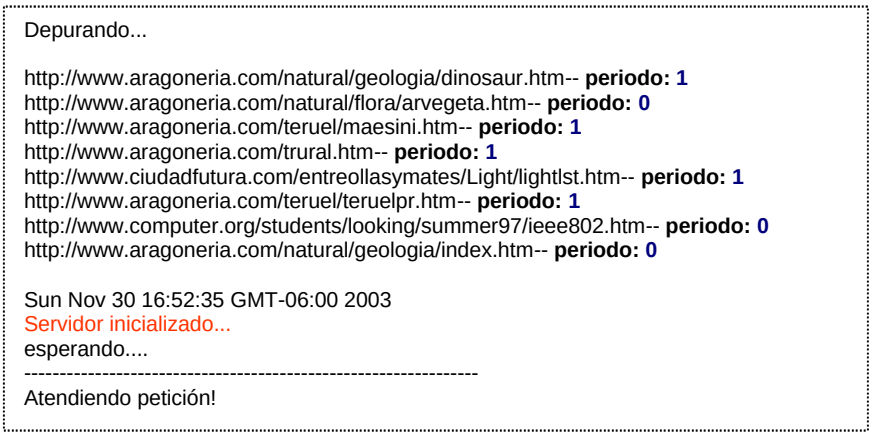


Figura 4.13 Reporte de estado de la Cache para el caso de prueba 4

Las peticiones listadas anteriormente fueron procesadas de manera particular una vez que se identificó el tipo de dispositivo que realizaba la solicitud, lo cual se verifica en la información que se muestra en la figura 4.14, donde se despliega la salida que el servidor envía a su interfaz al momento de estar manipulando las peticiones HTTP: el recurso solicitado marcado con el número 1, la dirección IP del cliente señalada con el número 2 y por último, con el número 3 la plataforma sobre la cual opera el dispositivo.

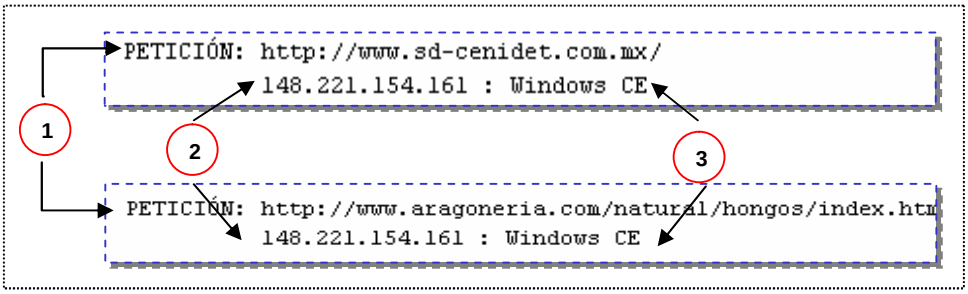


Figura 4.14 Identificación del dispositivo Pocket PC

Como resultado del procesamiento de la solicitud número uno http://www.sd-cenidet.com.mx/, el servidor Transformador envió como respuesta al dispositivo Pocket PC la página resultante del proceso de transformación. El contenido Web que se observa en la figura 4.15, corresponde a la versión transformada, como una variante del contenido Web original. En dicha versión se encuentran los elementos organizados o agrupados en los siguientes bloques: el título de la página, el texto, las Imágenes y los enlaces.

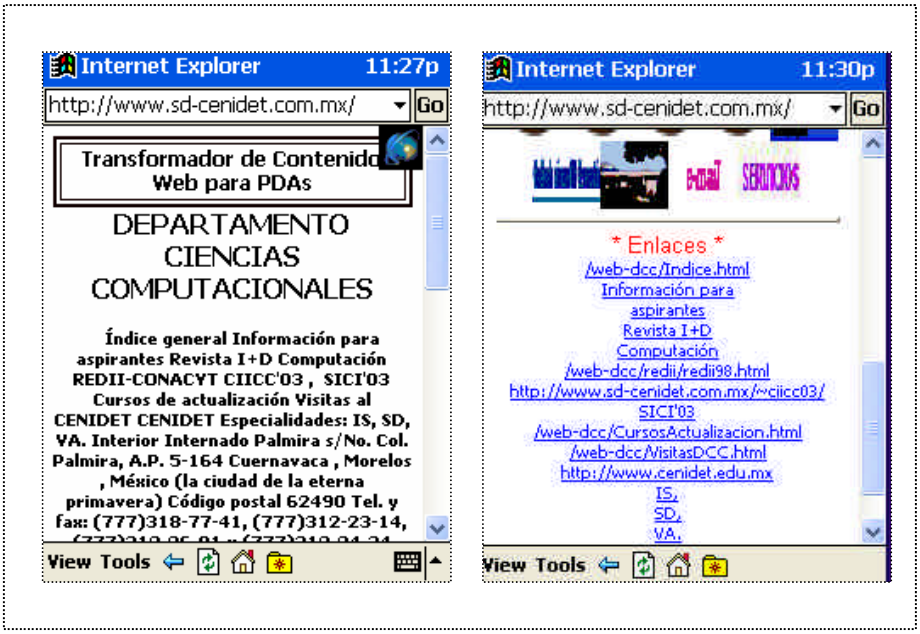


Figura 4.15. Resultado de la petición número uno, no almacenada en la cache

4.2.2.5 Quinto escenario de pruebas

El objetivo del quinto escenario es verificar la recuperación de una página Web que ha sido transformada recientemente, habiendo sido solicitada previamente por otro dispositivo Pocket PC y actualmente se encuentra almacenada en la cache.

Una vez que un dispositivo de este tipo pide un recurso de la Web, el cual fue solicitado recientemente por otro dispositivo Pocket PC, dicho recurso debe encontrarse en disco en la versión transformada, siempre y cuando su tiempo de vida en la cache no haya expirado; por lo tanto, el tiempo de recuperación del contenido de Web se ve reducido notablemente.

Según el objetivo que se persigue en este caso de prueba, es necesario observar lo que muestra la figura 4.16, un reporte de estado de la cache que se obtuvo al inicializar el servidor Transformador, con el fin de capturar esta información para verificar que efectivamente, los recursos Web solicitados por los dispositivos clientes se encuentran almacenados en la cache. Por lo tanto, las peticiones resaltadas entre líneas punteadas corresponden a las dos peticiones que se tomaron como caso de prueba.

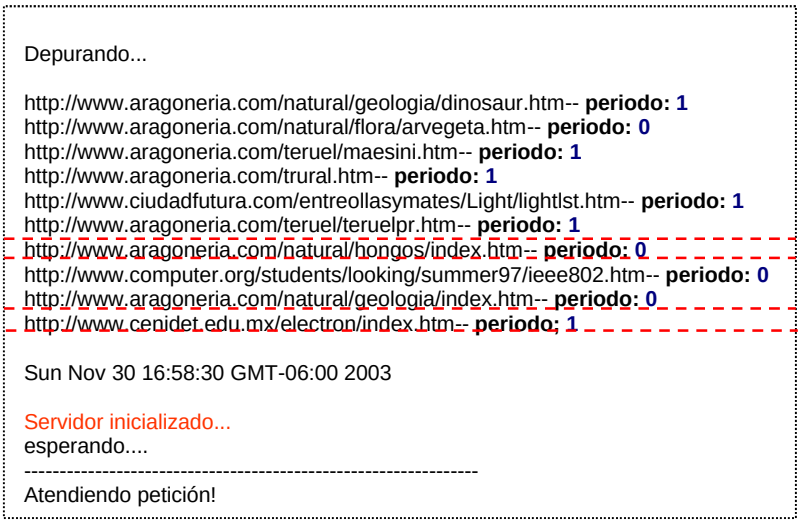


Figura 4.16. Reporte de estado de la cache para el caso de prueba 5

Para comparar los tiempos de respuesta entre una solicitud de la cual se recuperó el contenido Web de Internet, y otra que lo obtuvo de la cache, se tomó una de las peticiones evaluadas en el caso de prueba número 4, la cual corresponde al siguiente *url*: http://www.aragoneria.com/natural/hongos/index.htm

En la figura 4.17 se despliega la salida que el servidor envía a su interfaz como resultado del escenario de éxito sobre el cual se realizó el procesamiento de la petición. Como se puede apreciar en la figura, el recurso solicitado se obtuvo de la cache, es por eso que fue necesario actualizar la fecha de acceso al recurso almacenado. Para esto, el número 14 corresponde al número de día en el mes en que se realizó la recuperación del recurso.

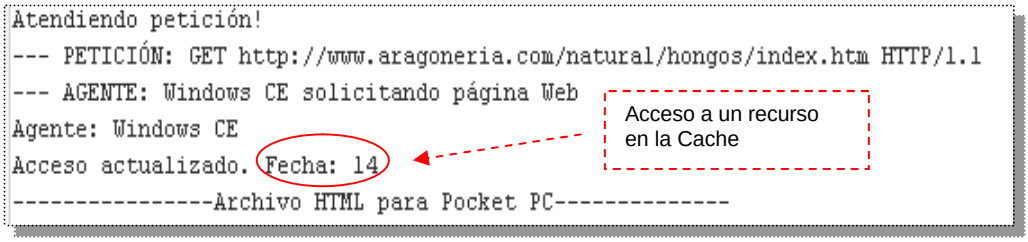


Figura 4.17 Reporte de actualización de cache en petición uno, almacenada en la Cache (caso de prueba 5)

La segunda petición, http://www.cenidet.edu.mx/electron/index.html**,** no se evaluó en el caso de prueba número cuatro, sin embargo, los resultados obtenidos muestran la reducción de los tiempos de respuesta que se consiguen con la utilización del sistema cache integrado en el servidor Transformador.

Por lo anterior, el tiempo de respuesta que se consiguió para la segunda solicitud fue de 0.151 segundos; lo cual se muestra en la figura 4.18 y se encuentra señalado con el número 2.

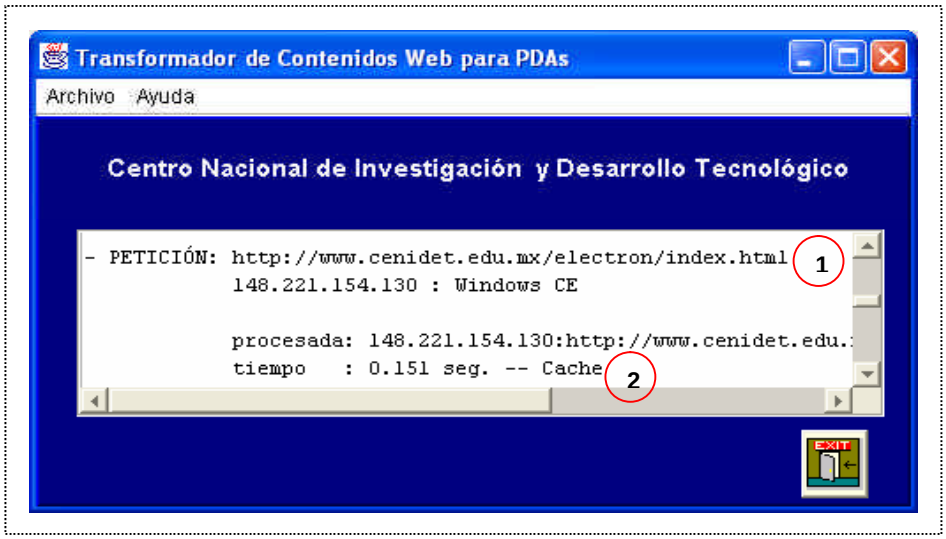


Figura 4.18. Salida del procesamiento de peticiones de HTTP

4.3 Evaluación del proceso de transformación a múltiples formatos

En este caso de prueba se muestra que es posible acceder a cualquier recurso en la Web de manera normal con el GAP, como se muestra en la figura 4.19.

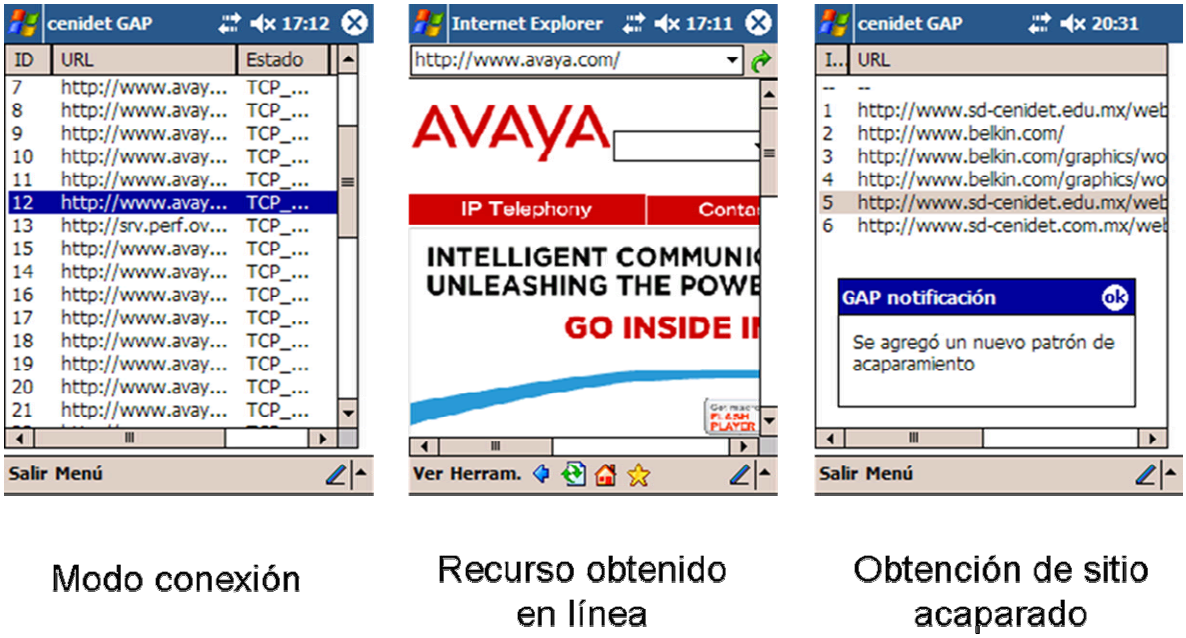


Figura 4.19. Acceso a recursos Web desde el GAP.

En la figura 4.20 se muestra que es posible acceder a recursos acaparados cuando no existe conexión.

Figura 4.20. Acceso a recursos acaparados.

En la figura 4.21 y 4.22 se muestra que es posible visualizar el mismo recurso Web transcodificados en los siguientes formatos: HTML reformateado (versión anterior), WML, XHTML-MP y PDF.

Figura 4.21. Recursos acaparados.

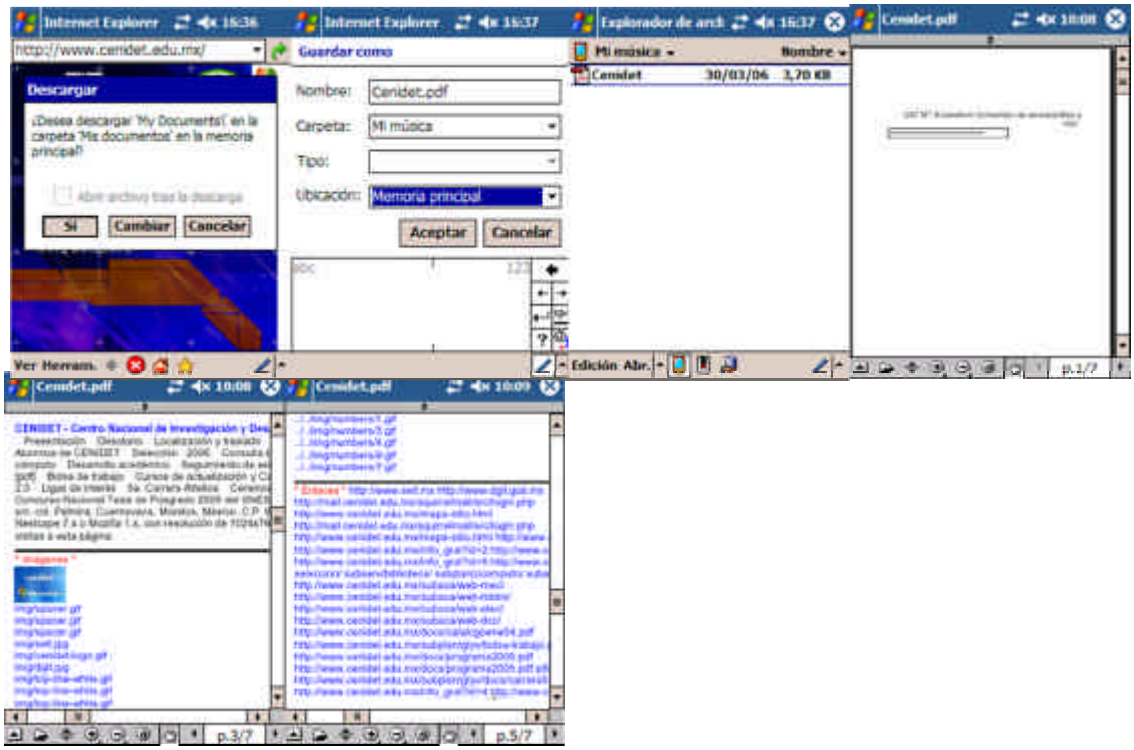


Figura 4.22. Transcodificación de contenidos Web a PDF.

En la figura 4.23 se muestra el resultado de la transcodificación de una página html a formato WML y en la figura 4.24 se muestra el proceso de transformación de una página html a formato html-mp.

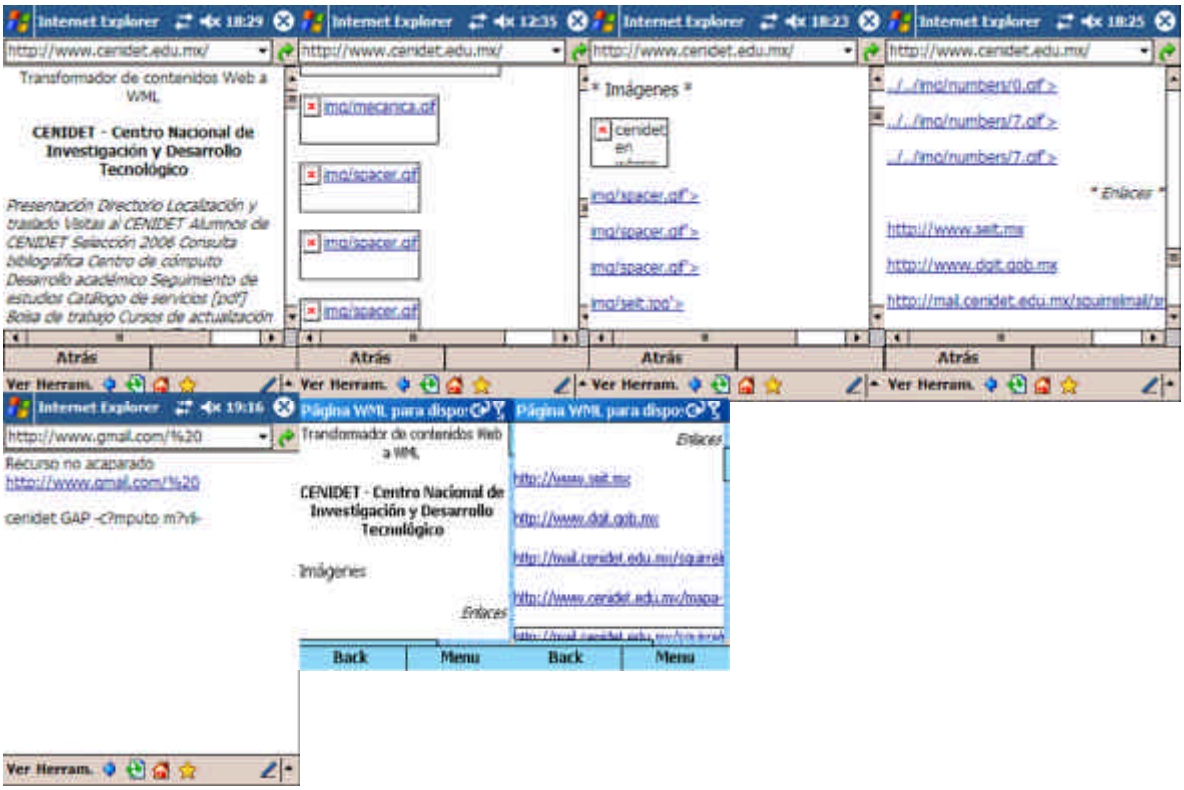


Figura 4.23. Transcodificación de recursos Web a WML.

Figura 4.24. Transcodificación de contenidos Web a XHTML-MP.

En la figura 4.25 se muestra la ejecución del GAP en un dispositivo celular con Windows Mobile, aquí podemos observar que la aplicación es multiplataforma.

Figura 4.25. Ejecución del GAP en un Smartphone con Windows Mobile: "SmartGAP".

Capítulo 5

Trabajos Relacionados

En el capítulo cinco se describen los trabajos relacionados con esta tesis, las características de cada uno, se da una visión general de los trabajos de investigación y desarrollo además de describir las diferencias con esta tesis.

5.1 Trabajos relacionados

En esta sección se presentan los resultados de la investigación que se realizó de los trabajos relacionados con esta tesis. Deacuerdo a los resultados de esta investigación se determinó que los trabajos analizados reúnen la funcionalidad requerida por los ambientes de cómputo móvil, en relación al soporte de desconexión, modelos de interacción y garantías transaccionales. Sin embargo, se identifico que son arquitecturas que en algunos casos proporcionan únicamente soporte a un requerimiento específico de los entornos de cómputo móvil, de los tres que se abordan en esta tesis, por lo tanto la conclusión de esta investigación de trabajos relacionados es la siguiente:

- En resumen, no se detectaron trabajos que incluyeran esquemas de acaparamiento de sitios Web aplicando minería de uso, lo cual es una de las principales aportaciones de este trabajo de investigación. El esquema de acaparamiento propuesto en este trabajo de tesis difiere de los trabajos relacionados, ya que en ninguno de estos se propone el almacenamiento local de un subconjunto de componentes de un sitio Web en dispositivos inalámbricos.

En los siguientes párrafos mostramos una descripción de los trabajos relacionados con esta tesis.

En [22] se presenta la idea de una base de datos distribuida construida completamente de componentes móviles. Debido a que las desconexiones estarán frecuentemente presentes en tales ambientes, se desarrolló un procedimiento de desconexión y reconexión que permite procesamiento normal en los componentes conectados. Se propuso la idea de base de datos distribuida móvil la cual soporta la operación de un equipo de trabajo móvil. Se desarrolló un protocolo basado en propagación epidémica para administrar tal sistema, e integrado con un procedimiento de desconexión planeada. La aproximación epidémica es especialmente apropiada para desconexiones ya que el registro de eventos mantiene el registro de todos los eventos necesarios para la ejecución aun cuando algunos miembros no estén disponibles. Se asumió que cualquier falla será detectada y corregida rápidamente, por lo que esos problemas causados por fallas en la red pueden ser ignoradas. Esto no es real, y una posible solución es usar quórum en lugar de requerir un voto desde todos los sitios. Recientemente se expandió la aproximación epidémica para incorporar quórum, de esta manera se incrementó el grado de tolerancia a fallas y desconexiones permitidas. Finalmente este protocolo permite al nodo desconectado acceso a

datos en modo sólo lectura. Se está trabajando en un protocolo que permita a un nodo solicitar privilegios de lectura/escritura en los datos cuando se encuentren desconectados.

En [1] se presenta una arquitectura para bases de datos móviles. El trabajo se enfoca en los aspectos de los servidores de bases de datos que necesitan ser rediseñados para facilitar el almacenamiento y la reintegración. El modelo que se considera es una modificación del modelo computacional cliente/servidor. Así como en el modelo tradicional cliente-servidor cada cliente tiene recursos limitados comparados con el servidor (espacio en disco y poder de cómputo). Un servidor de bases de datos centralizado resuelve solicitudes desde clientes. A los clientes acaparadores se les permite almacenar datos en sus discos locales creando réplicas locales. Cuando un cliente acaparador se encuentra conectado, existe un ancho de banda alto y un enlace físico entre ellos, las consultas son resueltas por el servidor. Cuando un cliente se desconecta, un servidor local en el cliente usa lo almacenado para resolver las consultas. Así cuando el cliente se reconecta, los servidores locales reconcilian su réplica con la copia del servidor mediante reintegración y actualizaciones locales.

En [13] se presenta una arquitectura Web no interactivo (Web&). La arquitectura incorpora soporte para clientes no conectados y heterogéneos, transacciones vía una interfaz uniforme de servidor, y estado persistente del cliente. La arquitectura del Web& incluye 1) un motor proxy el cual toma las solicitudes del cliente y las ejecuta en la Web, 2) un directorio que refuerza una estructura en la Web y define protocolos para usarlo, y 3) un grupo de servidores proxy que proveen otros componentes con una interfaz uniforme de la Web. Se ha desarrollado un prototipo de Web& basado en Java, JDBC y XML para cada uno de sus componentes. En este trabajo se propone una nueva arquitectura que le permite a los usuarios realizar tareas en el WWW en un modo no interactivo. Esta arquitectura apunta los problemas creados por las búsquedas interactivas en la Web. El objetivo es proveer un usuario no interactivo con una experiencia en línea y no sufrir frustraciones del desempeño interactivo en tareas repetitivas. Se logró esto interponiendo una capa intermedia de proxy entre el cliente y el servidor que permite el reemplazo de la interacción síncrona de cliente-servidor con una interacción asíncrona entre el cliente y su capa, moviendo la sincronía entre esta capa y el servidor.

En [37] se definió un nuevo modelo llamado Programación Asistida Móvil (MAP) para el desarrollo y ejecución de aplicaciones de comunicación en redes a gran escala de computadoras heterogéneas. El objetivo del modelo es mejorar la habilidad de las aplicaciones de comunicación para que desarrollen acciones complejas en redes a gran escala mediante el acercamiento de los datos distribuidos, dispersos en servidores. Los asistentes MAP son programas interpretados de

alto nivel que pueden moverse entre nodos, crear clones y reportar resultados. Su ejecución es asíncrona y persistente para permitir al cliente desconectarse y sobrevivir a fallas de nodos. Se ha implementado el modelo MAP usando el ambiente de los servidores de información WWW y el lenguaje de programación Scheme. Los asistentes MAP son programas Scheme cuyos estados de ejecución se pueden guardar, transferir a un servidor WWW remoto y restaurar.

En [8] se presentó una aplicación (Alycta) mediante la cual se ofrece un acceso eficiente al Internet a través de equipos portátiles e inalámbricos. La forma de hacer eficiente el acceso es mediante 1) el delegado de actividades a agentes móviles, y 2) destilación de contenido en datos específicos como imágenes, sonido y vídeo entre otros. Alycta se ejecuta en una computadora móvil y presenta una interfaz en Java que permite al usuario introducir una búsqueda a un gran volumen de datos específicos aceptables tales como sonido, video e imágenes. En el documento se propuso una aplicación que ofrece soporte basado en agentes móviles para el acceso al WWW a través de GSM (Global System for Mobile Comunications). Para experimentar con esta técnica, se diseño e implementó Alycta, una aplicación que usa agentes móviles y MAP (Mobile Assistant Programming). Corriendo en un host móvil, Alycta puede delegar operaciones que consumen tiempo tales como descarga de documentos y destilación de sus contenidos a un nodo en la red.

La extracción de patrones de navegación a partir de datos Web ha sido utilizada en múltiples sistemas. Existen proyectos como WebSIFT [51], WUM [52] [53], SpeedTracer [54] y los trabajos de Shahabi [55] que enfocan sus investigaciones a la minería de uso Web en forma general sin enfocarse a una de las sub-categorías.

El proyecto SpeedTracer [54] de IBM se basa en la metodología propuesta por [56]. El sistema Web Utilization Miner (WUM) [52] es capaz de generar una base de datos y acceder a ella mediante un lenguaje tipo SQL llamado MINT, este lenguaje acepta parámetros específicos de minería de datos como soporte y confianza para descubrir patrones frecuentes que resultan del algoritmo de minería de reglas de asociación. Shahabi [55] tienen uno de los pocos sistemas de minería de uso Web que cuentan con recolección de datos desde el nivel del cliente.

El sistema WebPersonalizer [57] cae dentro de la clasificación de los sistemas que en base a preferencias del usuario, perfiles y patrones de navegación crean un sitio Web a la medida del usuario.

WebWatcher [58], SiteHelper [59], Letizia [60] y los trabajos de Yan [61], están concentrados en proveer sitios Web personalizados basándose en la información del uso Web.

Yan [18] descubre grupos de usuarios que tienen patrones de acceso similares. El sistema propuesto en [18] consiste de un módulo fuera de línea que busca grupos de usuarios y de un módulo en línea el cual es responsable de la generación de páginas Web a la medida del usuario. El proyecto SiteHelper lleva a cabo un aprendizaje acerca de las preferencias de los usuarios mediante una búsqueda de páginas accedidas. En base a una lista de palabras clave en cada página localizada los usuarios son clasificados en un grupo. WebWatcher inicia basándose en una pequeña descripción de los usuarios y sus preferencias. Cada petición a los recursos de un sitio Web es encaminada a través del servidor proxy para facilitar el seguimiento de las sesiones a través de múltiples sitios Web y de esta forma marcar cada link interesante. Letizia [60] es una gente del lado del cliente que busca entre los Web visitados por el usuario páginas similares a las que están en los bookmarks del usuario. El sistema WebPersonalizer [57] crea grupos de páginas recomendadas para grupos de usuarios basándose en sus preferencias.

Almeida [62] propone un modelo para predecir y localizar páginas solicitadas a un servidor Web por un usuario en particular o un grupo de usuarios accediendo desde el mismo servidor proxy, también aborda el problema del uso incremental de contenidos dinámicos que reduce los beneficios de la tecnología del catching.

Los proyectos sobre sitios Web adaptativos [63], [64], están enfocados a cambiar automáticamente la estructura de un sitio Web basado en el uso de patrones descubiertos a partir de los log de los servidores Web. La agrupación de páginas es usada para determinar que páginas deben ser ligadas directamente.

La información acerca de cómo los usuarios están usando un sitio Web es crítica para la inteligencia de negocios o marketing. Buchner [2], [65] ha desarrollado un proyecto para descubrir conocimiento con el objeto de impactar en las dediciones de marketing desde los datos de los archivos log. Son cuatro los distintos pasos que define en el ciclo de vida de las relaciones del cliente que pueden ser apoyadas por sus técnicas para el descubrimiento del conocimiento: Atracción del cliente, retención del cliente, ventas cruzadas y salida de clientes. Existen varios productos comerciales tales como SurfAid [70], Accrue [67], NetGenesis [69], Aria [66], Hitlist [68] y WebTrends [71] que proveen análisis de tráfico que ayudan principalmente en la toma de decisiones concernientes a la inteligencia de negocios. Además del uso de las estadísticas, Accrue, NetGenesis y Aria están diseñados para analizar eventos relacionados con el comercio electrónico, tales como compras de productos y porcentajes de clicks sobre anuncios. Han [72] a cargado logs de servidores Web dentro de estructuras de cubos de datos con el objeto de aplicar

técnicas de minería de datos. Su sistema conocido como WebLogMiner ha sido usado para descubrir reglas de asociación, clasificación y análisis de series de tiempo.

Existen trabajos como el de Pitkow [73], [74], que se enfocan en caracterización del uso de sitios Web, cabe destacar que entre minería de uso Web y caracterización del uso Web existe una gran diferencia. En el trabajo de Pitkow se modifico el navegador Web XMosaic con el objeto de que este recolectara datos acerca de las estrategias que siguen los usuarios para navegar por un sitio en particular. El proyecto genera estadísticas sobre eventos del lado del cliente tales como clicks sobre recursos, botones de atrás, recarga, adelante, archivos guardados y bookmarks.

En resumen los trabajos relacionados con esta tesis se distribuyen en la Figura 5.1 de acuerdo al área de aplicación de las mismas, como se describe en la sección 3.3. Esta tesis se ubica en la clase de aplicaciones de minería de uso y de estructura Web

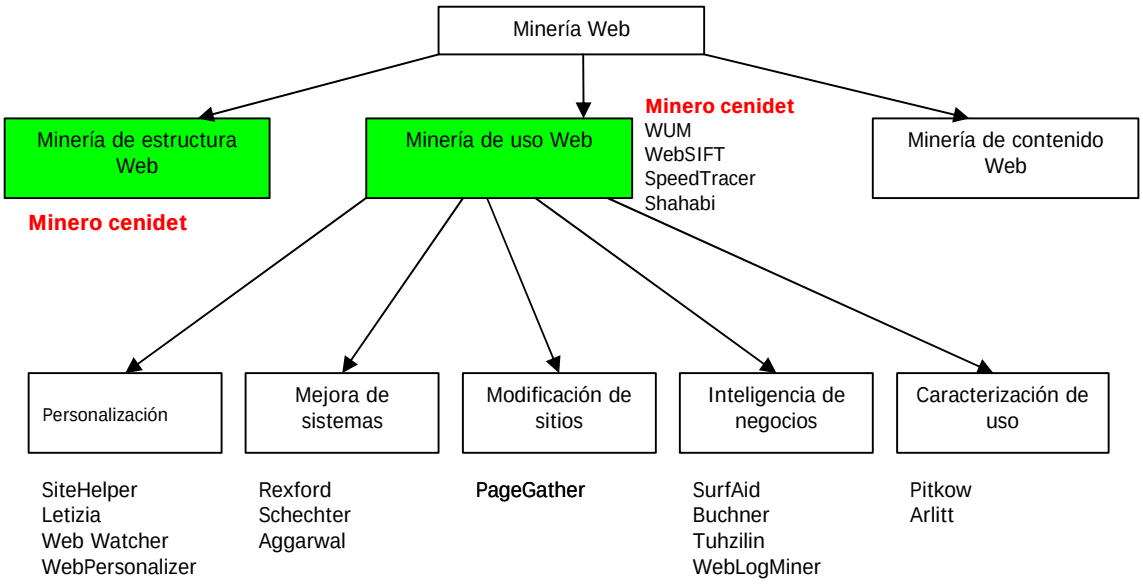


Figura 5.1 Clasificación de proyectos sobre minería Web

5.2. Tabla comparativa de trabajos relacionados con minería de uso Web

Ahora se muestra un concentrado de información, comparando las tecnologías analizadas y el prototipo de generación de patrones y acaparamiento desarrollado en esta tesis, esta información se presenta en la tabla 5.12.

Proyecto	Soporte a operaciones en modo desconexión	Generación de patrones de acceso	Acaparamiento de sitios Web
Ficus	Sí	No	No
Coda	Sí	Sí	No
TXAgent	Sí	No	No
Desconexiones planeadas	Sí	No	No
MAP	Sí	No	No
Alycta	Sí	No	No
Web&	Sí	No	No
Bases de datos Móviles	Sí	Sí	No
CRAS	Sí	No	No
Esquema propuesto	Sí	Sí	Sí

Tabla 5.1 Comparativa de los trabajos relacionados.

5.3 Transformación de páginas Web para dispositivos con pantallas pequeñas.

Hay varias soluciones a este problema de acceso a la Web mediante múltiples plataformas. Una de éstas, la más simplista y frecuente, es simplemente que los usuarios con dispositivos PDA restrinjan el acceso a la Web sólo a páginas diseñadas para pantallas pequeñas o aprovechar lo mejor de las páginas grandes en sus dispositivos.

Trabajos recientes basados en servidores proxy (Berkeley Pythia proxy [11] y Transend [17]) proporcionan compresión dinámica de contenidos de Web para mejorar la velocidad de transmisión en dispositivos inalámbricos. Otra solución más simple es diseñar páginas de Web ad-hoc, es decir, múltiples versiones de la misma página, lo cual implica una labor intensa, costosa y prohibitiva para la gran cantidad de páginas de Web disponibles en la WWW. A continuación se describen algunas de las soluciones analizadas.

Es un sistema de navegación creado por la compañía AvantGo [5], cuyo objetivo es enviar contenidos de Web a PCs de bolsillo. En realidad no son más que páginas de Web diseñadas con algunas características especiales.

Los requerimientos para tener acceso al servicio de AvantGo son los siguientes:

- Este servicio se puede configurar en múltiples plataformas, así que se requiere de un dispositivo portátil como Palm, dispositivos con Windows CE o móviles WAP.

- Una PC de escritorio con una sincronización configurada a través de ActiveSync[1] y conexión a Internet disponible.
- Una cuenta en AvantGo.

Los usuarios de AvantGo pueden manejar las configuraciones de sus cuentas, lo cual incluye, configuración de canal, administración de claves de acceso, bitácoras de sincronización y software actualizado de descarga. Esta aplicación permite la configuración de canales personalizados que se descargan desde la conexión con la PC cada vez que se sincroniza con el PDA. Esto posibilita volcar datos al dispositivo y acceder a ellos offline hasta que se vuelva a sincronizar éste.

Su funcionamiento es sencillo, tan sólo se descarga el software correspondiente al dispositivo portátil a utilizar, se instala siguiendo los pasos que se indican a través de la Web, debiendo tener sincronizada la PC de bolsillo en todo momento para poder realizar la configuración adecuadamente. Una vez que se lleve a cabo la configuración y el proceso de sincronización haya finalizado, se podrá tener acceso a la información sobre el dispositivo móvil.

Las aplicaciones de Web Clipping [24] (extracción de información de las páginas Web) para Palm OS 3.5, permiten acceder a la información que más interesa de forma rápida y eficiente, a través de una conexión inalámbrica que evita la pérdida de tiempo que suponen la complejidad de los gráficos o la información no requerida. Hay disponibles muchas aplicaciones de Web Clipping que proporcionan noticias, cotizaciones de bolsa, horarios de vuelos, mapas de carreteras, partes meteorológicos y muchos otros datos de interés.

Para entender cómo funcionan, se han analizado los componentes que integran este tipo de aplicaciones: PQA y Web Clipping.

Una PQA (Palm Query Application), es un tipo especial de aplicación para los dispositivos Palm VII que le permite al usuario interactuar inalámbricamente con el contenido de Web. Una PQA lleva el contenido estático en una aplicación que puede ser instalada dentro de un dispositivo Palm VII. Los enlaces en un documento PQA pueden referirse a otras páginas dentro de la aplicación, a documentos o scripts residentes en un servidor de Web disponible. Las páginas que

[1] ActiveSync. Software que hace posible la comunicación entre una computadora personal y un dispositivo PDA, permitiendo la transferencia de información de una a otra.

no se encuentran dentro de la PQA, naturalmente, son resultado de una solicitud de HTTP inalámbrica.

Finalmente, Web Clipping es el resultado de las solicitudes enviadas desde la PQA al servidor de Internet. Tanto la Palm Query Application como Web Clipping, son escritos en HTML.

jTranscoder [36] es un convertidor de HTML. Este permite convertir fácilmente archivos de HTML en archivos de WML, visibles en teléfonos celulares habilitados para la Web, y archivos Web Clipping en Palm VII Pilots. Se puede traducir un solo archivo de HTML a la vez, mediante la selección de dicho recurso accesible desde el sistema de archivos de la PC, o traducir un sitio de Web completo mediante la especificación de la URL de dicho sitio. jTranscoder guarda el resultado de la traducción dentro de archivos.

Cuando se están convirtiendo los archivos de HTML al formato de Web Clipping, se podrá utilizar la herramienta Webclipping Builder de Palm Inc. para construir una aplicación PQA de los archivos resultantes e instalarlos dentro de la Palm Pilot. Por otro lado, cuando se trata de convertir un sitio de Web completo, jTranscoder provee un listado de los enlaces que se tradujeron, para informar acerca del directorio y la estructura de archivos del resultado de la conversión y los enlaces de HTML que se han visitado durante la transformación.

5.4 Tabla comparativa entre el mecanismo de transformación de páginas Web

Ahora se muestra un concentrado de información, comparando las tecnologías analizadas y el prototipo Transformador desarrollado en esta tesis, esta información se presenta en la tabla 5.12.

Tabla 5.2 Comparativa de los trabajos relacionados.

Aplicación	Proceso de Sincronización	Transformación en Línea	Transformación Completa de Página	Transformación Almacenada	Tipo de Servicio
AvantGo	Si	No	No	No	Tarifa
Web Clipping	Si	No	No	No	Tarifa
jTranscoder	Si	No	No	No	Plataforma Comercial
Transformador	No	Si	No	Si	Libre

Capítulo 6

Conclusiones

El capítulo seis presenta las conclusiones a las que se llego durante el desarrollo de esta investigación. El capítulo concluye dando sugerencias de trabajos futuros.

6. Conclusiones

El objetivo de esta tesis fue evaluar la factibilidad de implementar una arquitectura basada en gestores de acceso a Internet que permite a las aplicaciones cliente/servidor adaptarse de manera transparente a la dinámica de los escenarios de cómputo móvil, esta arquitectura proporciona servicios de gestión de conexiones asíncronas no interactivas y servicios de precarga y reformateo de páginas Web para dispositivos móviles heterogéneos lo cual permite garantizar que los usuarios tengan acceso a la Web en cualquier lugar, en todo momento y desde cualquier dispositivo con capacidad de conexión a Internet.

Para definir este objetivo fue necesario plateando preguntas de investigación las cuales nos permiten describir las aportaciones al área de conocimiento en la que desarrollamos esta tesis, por lo que las conclusiones de esta investigación son las siguientes:

- **Gestión de conexiones**. ¿Se puede aplicar de manera transparente un modelo de interacción asíncrono no interactivo en las aplicaciones cliente/servidor actuales? La conclusión a la que se llego es que es factible aplicar de manera transparente servicios intermediarios para modificar el esquema de interacción tradicional de las arquitecturas cliente servidor, esto se logro implementando una arquitectura de procesos gestores de cuatro niveles. Para sustituir de manera transparente el modelo de interacción síncrono de las aplicaciones cliente que utilizamos de manera común fue necesario implementar un proceso gestor que se ejecuta en el dispositivo móvil, este gestor tiene la función de un servidor proxy local por el cual pasan todas las solicitudes de las aplicaciones cliente, lo cual permite mantener el modelo de interacción síncrono interactivo, una vez que el gestor recibe la solicitud de la aplicación cliente este la envía al gestor externo utilizando un modelo de interacción asíncrono no interactivo, es aquí donde se logra romper el esquema de interacción síncrono tradicional y se obtiene como beneficio inmediato la reducción en los tiempos de conexión, lo cual impacta de manera directa en servicios de conexión tarifados por tiempo. En la figura 3.1 se muestra la ubicación de los gestores que se propuso en esta tesis para implementar el modelo de interacción asíncrono no interactivo. En esta figura se puede observar la ubicación del gestor local, el cual se ejecuta en el dispositivo móvil, que puede ser un teléfono celular, una pocket pc o una computadora portátil tradicional, es decir, el gestor local es multiplataforma ya que solo se requiere de una maquina virtual para ejecutarlo en cualquier dispositivo y el tamaño de la aplicación es adecuado para ejecutarse en dispositivos con recursos limitados en poder de procesamiento y en memoria RAM. En resumen, los resultados obtenidos con la implementación de una arquitectura basada en gestores nos permiten concluir que es factible sustituir de manera

transparente el modelo de interacción de las aplicaciones cliente-servidor actuales sin necesidad de modificar su esquema de interacción original. Otro aspecto importante del modelo de interacción que se propone en esta tesis es el mercado potencial de clientes que se beneficiarían con este modelo, por ejemplo en México hay cerca de cuarenta millones de usuarios de celulares a los cuales se les puede proporcionar la opción de acceder a Internet en cualquier momento y en cualquier lugar.

- **Acaparamiento de páginas Web**. ¿Se puede precargar un subconjunto de páginas de un sitio Web en un dispositivo móvil en función de patrones de uso del sitio? Los resultados que se obtuvieron en esta tesis permiten concluir que es factible el acaparamiento de sitios Web en dispositivos móviles. Para esto se implementó una herramienta de minería de uso de la Web, la cual utiliza algoritmos para extraer patrones de navegación, estos patrones permiten identificar páginas de un sitio Web que están relacionadas entre si, y con esto generar un mapa de navegación del sitio Web que únicamente incluya las páginas que tienen mayor probabilidad de ser visitadas de acuerdo a los patrones de navegación de ese sitio en particular. En conclusión, de acuerdo a las pruebas que se realizaron se pudo observar que el tamaño de un sitio Web se puede reducir hasta en un 35%. El proceso de acaparamiento se logra implementando un servicio de replicación local en el dispositivo cliente, este proceso de copiar (replicar) páginas Web en el dispositivo cliente lo controlan el proceso gestor local, que se ejecuta el dispositivo móvil (celular, PDA o laptop) y el proceso gestor servidor que se ejecuta generalmente en una PC que funciona como pasarela de los cliente móviles. Las páginas Web acaparadas localmente en el dispositivo móvil se guardan en una cache, similar a la que manejan los navegadores Web como Netscape o Explorer, con este esquema de acaparamiento se logro reducir hasta en un 80% el número de solicitudes de una aplicación cliente, esto significa, que con las paginas replicadas de manera local, hay un 20% de probabilidad de que el cliente solicite una página que no se encuentra en la cache de acaparamiento, lo cual reduce de manera significativa las solicitudes de conexiones desde un dispositivo móvil, por ejemplo si se utiliza un celular con una conexión tarifada por tiempo, el cliente solo estaría conectado el tiempo necesario para bajar el sitio Web que se copiará de manera local en su dispositivo. Otra ventaja del esquema de acaparamiento que se propuso en esta tesis es que antes de realizar el acaparamiento de sitio Web en el dispositivo móvil se debe realizar un proceso de compresión lo cual reduce hasta en un 86% el tamaño original de los datos. Es importante aclara que esta es una de las principales aportaciones de esta tesis ya que hasta este momento no se han encontrado referencias de trabajos que propongan el acaparamiento de un subconjunto de paginas de sitio Web en dispositivos móviles.

- **Transformación de páginas Web**. ¿Es posible reformatear una página Web de acuerdo a las características del dispositivo que la solicita? De acuerdo a las pruebas realizadas en esta tesis se demostró la factibilidad de reformatear el contenido de una página Web de acuerdo a las características del dispositivo móvil. Para logar esta transformación es necesario identificar el perfil del dispositivo, pare esto se analizan las cabeceras del protocolo http, de donde se extrae información de la cabecera user-agent de donde se puede identificar atributos del dispositivo como sistema operativo, microprocesador y resolución de la pantalla, con esta información se puede reformatear la página Web y adaptarla a las características del dispositivo. Para esta tesis se propuso el reformateo en varis formatos lo cual da la posibilidad de visualizar el contenido de una página Web en dispositivos heterogéneos.

6.1 Aportaciones

Las principales contribuciones de esta tesis para el área de cómputo móvil, son las siguientes:

- En lo referente a los tiempos de conexión para acceder a recursos Web acaparados en el dispositivo móvil se tiene lo siguiente:
  - Se mejoró un 85% el tiempo de acceso gracias a los recursos Web precargados de manera local en la cache del dispositivo móvil
  - Se redujo la cantidad de solicitudes entre cliente y servidor en un 80%, ya que se replica un subconjunto del sitio Web en el dispositivo móvil por lo que la probabilidad de solicitar una página del sitio no precargada en el dispositivo es del 20%.
- En lo referente al tamaño de los recursos replicados en el dispositivo móvil:
  - El proceso de eliminación o recorte de páginas de un sitio Web reduce en un 35% el tamaño del sitio,
  - la transformación de las páginas que se replican en el dispositivo móvil reduce hasta un 34% el tamaño del recurso y
  - la compresión de un sitio Web previamente recortado y transformado reduce hasta en un 86% el tamaño total del sitio que será replicado en el dispositivo móvil.

- Replanteamiento del modelo de interacción cliente/servidor tradicional al cual denominamos *modelo asíncrono no interactivo con soporte de operaciones en modo desconexión*.

- Análisis de la problemática relacionada a los escenarios de cómputo móvil, este análisis nos permitió identificar de manera específica las tres áreas que consideramos en esta

tesis, manejo de desconexiones en arquitecturas cliente/servidor, acaparamiento de páginas Web y el reformateo de páginas Web para dispositivos de cómputo no convencionales.

- Diseño de un modelo de interacción para las arquitecturas cliente/servidor tradicionales que se denominó *modelo de interacción asíncrono no interactivo* con el cual se logró proporcionar soporte a los eventos de desconexión. Este modelo permite que aplicaciones cliente/servidor tradicionales se adapten a cualquier entorno de cómputo móvil sin necesidad de modificar ningún aspecto arquitectónico ni de interacción original.

- Implementación de servicios para la generación de patrones de navegación aplicando Minería de uso de la Web.

- Evaluación comparativa de algoritmos de generación de patrones de navegación, específicamente se analizaron algoritmos de minería de reglas de asociación y agrupamiento.

- Diseño e implementación de estrategias de acaparamiento de páginas Web en dispositivos de cómputo móvil mediante la identificación de patrones de navegación de sitios de la Web previamente analizados mediante algoritmos de minería de uso de la Web implementados en esta tesis.

- Transformación de contenidos Web para dispositivos de cómputo móvil no convencionales. Se identifica el dispositivo en el momento en que realiza la solicitud y se reformatea la página Web en función del perfil del dispositivo, además de que se almacena en una cache transcodificada para posteriores solicitudes de dispositivos con el mismo perfil.

- La metodología de solución para una aplicación de minería de datos de uso Web. La metodología de solución descrita en el capítulo 3 detalla cada uno de los pasos necesarios que se intervienen en la construcción de una herramienta, esta metodología puede servir como pauta para otras implementaciones que tengan como objetivo la extracción de conocimiento a partir de bitácoras de accesos a recurso Web.

- La interfaz gráfica del minero. La construcción de una interfaz amigable es una de los aportaciones de este trabajo ya que de los proyectos relacionados el más cercano proporciona una interfaz poco amigable y que requiere de conocimientos especializados

para realizar la minería, de tal manera que se consiguió crear una interfaz para la cual no sea necesario tener amplios conocimientos sobre minería de datos, incluso con el simple echo de tener la definición de soporte y confianza, el usuario puede definir el nivel de interés que deseé encontrar en los patrones.

- El análisis gráfico de resultados. Mediante la construcción del recolector de estructuras Web y el visor de estructuras Web, el usuario puede realizar de una forma clara el análisis de sus resultados permitiendo la inspección directa dentro de las estructuras Web.

- Creación de sesiones mediante un método heurístico. La calidad de las sesiones de usuario localizadas dentro de los archivos log, se incrementa cuando son localizadas mediante el método heurístico, lo que provoca que los resultados de la minería sean de mayor relevancia.

- El uso de un manejador de base de datos. Con el uso de un manejador de base de datos, es posible trabajar con grandes cantidades de información, lo cual es una limitante en equipos con escasos recursos de hardware.

6.2 Trabajos futuros

El tema de minería de datos de uso Web es un área de investigación que puede aportar grandes beneficios al comercio electrónico, al diseño y rediseño de estructuras de sitios Web, mejora de sistemas como servidores Web y servidores Proxy, así como para establecer medidas de seguridad contra intrusos, sin embargo, la información recolectada es escasa y en algunos casos puede implicar violaciones a la privacidad del usuario.

De lo anterior podemos recomendar que en los trabajos futuros se aborden los siguientes aspectos.

- Recolección de datos directamente desde los dispositivos de los usuarios con el objetivo de saber como es que un usuario hace uso de cierto recurso Web.
- Extracción de información a partir de de datos almacenados sobre un enfoque de procesamiento analítico en línea (OLAP).

- Implementación de algoritmos para complementación de rutas que pueden ayudar a eliminar el problema de los caches tanto en los dispositivos de los usuarios como en los servidores Proxy.
- Implementación de técnicas de agrupamiento tanto de páginas Web como de visitantes de un sitio Web basando en preferencias y perfiles de usuarios.
- Implementación de técnicas para detección de intrusos, fraudes y robo de información a partir de minería de datos. Esto se puede lograr con otros algoritmos de minería de datos ya que una vez teniendo los datos preprocesados es posible aplicarse casi cualquier algoritmo de minería de datos.

Referencias

[1] B. R. Badrinath, S.H. Phatak, “An Architecture for Mobile Databases”, Dept. of Computer Science, *Rutgers University*. 1998.

[2] A.G. Buncher, M. Baumgarten, S.S. Anand, M.D. Mulvenna, y J.G. Hughes. “Navigation pattern discovery from Internet data”. In WEBKDD, San Diego, CA, 1999.

[3] B. R. Badrinath, S. Hemant Phatak**, “**Conflict Resolution and Reconciliation in Disconnected Databases”, *Proc. of Mobility in Databases and Distributed Systems (MDDS*), Florence, Italy, Sep.'99. http://www.cs.rutgers.edu/~phatak/mdds.ps

[4] I. Stanoi, D. Agrawal, A. El Abbadi, S.H. Phatak, B.R. Badrinath, “Data Warehousing Alternatives for Mobile Environments”, Dept. of Computer Science, *Rutgers University*, June 1999.

[5] AvantGo, “Mobile Enterprise Software”, http://avantgo.com/frontdoor/index.html.

[6] R. Cooley, Pang-Ning Tan. Discovery of Interesting “Usage Petterns from Web Data”. Departament of Computer Science and Engineering University of Minnesota. 1999.

[7] S. Hemant Phatak, B.R. Badrinath, “Multiversion Reconciliation for Mobile Databases”, *Proc. of International Conference on Data Engineering (ICDE)*, Sydney, Australia, Mar.'99. http://www.cs.rutgers.edu/~phatak/icde.ps

[8] X. Delord, S. Perret, A. Duda, “Efficient Mobile Access to the WWW over GSM”, *INRIA, France and University of California, Berkeley*, June 1997.

[10] G. Kuenning, G. J. Popek and P. Reiher**, “**An Analysis of Trace Data for Predictive File Caching in Mobile Computing”, *Proceedings of the USENIX Summer Conference*, 1994, pages 291-303.

[11] Pythia Proxy. www.cs.berkely.edu/~gribble/glomop_html, 1997.

[12] Shirish Hemant Phatak, B.R. Badrinath, “Data Partitioning for Disconnected Client Server Databases”, *Proc. of International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'99)*, Seattle, Washington, Aug.'99. http://www.cs.rutgers.edu/~phatak/mobide.ps

[13] S.H. Phatak, V. Esakki, B. R. Badrinath and L. Iftode**, “**Web&: An Architecture for Non-Interactive Web”, *Technical Report DCS-TR-405, Department of Computer Science, Rutgers University*, November, 1999. http://www.cs.rutgers.edu/~ohatak/www9/www9.html.

[14] Mona El-Kadi, “Mobile Computing: Issues in Local Data Management”, *Dept. of Computer Science at Old Dominion University*. 1996. http://www.cs.odu.edu/~elkadi/cs871/cs871proj.html

[15] David Ratner, Peter Reiher, and Gerald J. Popek**. “**Replication requirements in mobile environments”**.** *Technical Report CSD-970021, University of California*, Los Angeles, June 1997. ftp://ftp.cs.ucla.edu/tech-report/97-reports/970021.ps.Z

[16] Y. Saygin, O. Ujusoy, A. K. Elmagarmid**, "**Association Rules for Supporting Hoarding in Mobile Computing Environments", *IEEE*, 2000.

[17] Fox, A., S. Gribble, Y. Chawathe, E. Brewer. 1998. Adapting Network and Client Variation Using Infraestructure Proxies: Lessons and Perspectives. IEEE Personal Communications. Vol 5(4) p. 10-19.

[18] N. Pissinou, K. Makki, B. König-Ries, "A Middleware-Based Architecture to Support Transparent Data Access by Mobile Users in Heterogeneous Environments", *Center for Advanced Computer Studies & Center for Telecommunications Studies*, University of Lousiana at Lafayette, 2000.

[20] Woodruff Allison, M. Aoki Paul, "An Investigation of Documents from the World Wide Web", http://www5conf.inria.fr/fich_html/papers/P7/Overview.html, 1996.

[21] Sun microsystems, "Java API for XML Processing (JAXP)", http://java.sun.com/xml/jaxp/index.jsp, 2000.

[22] Holliday J., D. Agrawal, A. El Abbadi, "Exploiting Planned Disconnections in Mobile Environments", *Department of Computer Science, University of California*, at Santa Barbara, 2000.

[23] A. Murphy, "Algorithm Development in the Mobile Environment", Department *of Computer Science, Washington University*, 1999.

[24] PalmR, "¿qué es Webclipping?", http://palmr.com/argentina/movilPQA.asp, 2000.

[25] Geoffrey H. Kuenning and gerald J. Popek, "Automated Hoarding for Mobile Computers"*, In procceding of tha ACM Symposium on Operating Systems Principles*, St Malo, France, 1997

[26] David W. Cheung, Vincent T. Ng, Ada W. Fu, and Yongjian Fu, "Efficient Mining of Association Rules in Distributed Databases"*, Department of Computer Science,University of Hong Kong 1999.*

[27] David W. Cheung, Jiawei Han, Vincent T. Ng., Ada W. Fu, Yongijan Fu, "A Fast Algorithm for Mining Association Rules", *The University of Hong Kong* 1999.

[28] Xinfeng Ye, Jhon A Keane, "Mining associations Rules with Composite Items*" Dept. of Computer Scince, University of Auckland New Zelanda, Dept. of Computation UMIST Manchester UK,* 2000*.*

[29] Yücel Saygin, Özgür Ulusoy, Exploting "Data Mining Techniques for Broadcasting Data in Mobile Computing Environments"*. Dept. of Computing Engineering and Informatión Scince , Bilkent University, Turkey, 2000.*

[30] Zaki, M.J.; Parthasarathy, S.; Wei Li; Ogihara, M. "Evaluation of sampling for data mining of association rules", *Dept. of Comput. Sci., Rochester Univ., NY, USA*. 1999.

[31] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. "Web Usage Mining: Discovery and Application of Usage Patterns from Web Data". *SIGKDD Explorations*, 1(2):12-23. Enero 2000.

[32] Behzad Mortazavi-Asl. "Discovering and mining user Web-page traversal patterns". Simon Fraser University. 1999.

[33] M. Satyanarayanan, “Fundamental Challenges in Mobile Computing”, *School of Computer Science, Carnegie Mellon University*, 1998

[34] Rakesh Agrawal, Ramakrishnan Srikant, “Fast Algorithms for Mining Association Rules”, IBM Almaden Research Center, San Jose CA, USA, 1994.

[35] Jaideep Srivastava, R. Cooley. “Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data”. Department of Computer Science and Engineering. University of Minnesota, Minneapolis, USA.

[36] jTranscoder, ftp://ftp.componentsource.com/Componen/ILGPFAQ/demo/Manual/jtranscoderdoc.ZIP

[37] S. Perret, A. Duda, “MAP: Mobile Assistant Programming for Large Scale Communications Networks”, *Broadcast Technical Report, Esprit Basic Research Project 6360*, ISSN 1350-2042, 1995.

[40] Web Log Explorer. www.exaccttrend.com

[41] Raymond Kosala, Hendrik Blockeel, “Web Mining”, Department of Computer Science, Katholieke Universiteit Leuven, Belgium. 2002.

[51] R. Cooley, Pang-Ning Tan, Jaideep Srivastava. “WebSIFT: The Web Site Information Filter System”. Departament of Computer Science. University of Minnesota. Junio 1999.

[52] Myra Spiliopoulou y Lukas C. Faulstich. ”WUM: A Web utilization miner”. En EDBT Workshop WebDB98. Valencia, Spain. 1998.

[53] Myra Spiliopoulou, Carsten Pohle y Lukas C Faulstich. ”Improving the effectiveness of a web site with usage mining”. En WEBKDD. San Diego, CA. 1999.

[54] Kun-Lung Wu, Philip S Yu y Allen Ballman. SpeedTracer: A web usage mining and analysis tool. IBM Systems Journal, 37(1).1998.

[55] Cyrus Shahabi, Ami M. Zarkesh, Jafar Adibi y Vishal Shah. “Knowledge discovery from users web–page navigation”. Workshop on Research Issues in Data Engineering. Birmingham, England. 1997.

[56] Robert Cooley, Bamshad Mobasher y Jaideep Srivastava. “Web mining: Information and pattern discovery on the world wide web”. International Conference on Tools with Artificial Intelligence. pages 558-567. Newport Beach. 1997.

[57] Bamshad Mobasher, Robert Cooley y Jaideep Srivastava. “Creating adaptative web sites through usage-based clustering of URL’s”. Knowledge and Data Engineering Workshop. 1999.

[58] T. Joachims, D. Freitag y T. Mitchell. Webwatcher: A tour guide for world wide web. XV International Conference on Artificial Intelligence. Nagoya Japón. 1997.

[59] D.S.W. Ngu and X. Wu. Sitehelper: A localized agent that helps incremental exploration of the World Wide Web. VI International World Wide Web conference. Santa Clara, CA. 1997.

[60] H. Liberman. Letizia: An agent that assist web browsing. International Joint Conference on Artificial Intelligence. Montreal, Canada. 1995.

[61] T. Yan, M. Jacobsen, H. Garcia Molina y U. Dayal. From user access patterns to dynamic hypertext linking. V International World Wide Web Conference. Paris, Francia. 1996.

[62] Virgilio Almeida, Azer Bestavros, Mark Crovella y Adriana de Oliveira. Characterizing reference locality in the www. Technical Report TR-96-11, Boston University. 1996.

[63] Mike Perkowitz y Oren Etzioni. Adaptative Web sites: Automatically synthesizing web page. XV National Conference on Artificial Intelligence. Madison. 1998.

[64] Mike Perkowitz y Oren Etzioni. Adaptative Web sites: Conceptual cluster mining. XVI International Joint Conference on Artificial Intelligence. Estocolmo Suecia. 1999.

[65] Alex Buchner y Maurice D. Mulvenna. Discovering Internet marketing intelligence through online analytical web usage mining. SIGMOD páginas 54-61. 1998.

[66] Andromedia Aria. http://www.andromedia.com. 2005

[67] Accrue. http://www.accrue.com. 2005

[68] HitList. http://www.marketwave.com. 2005

[69] NetGenesis. http://www.netgenesis.com. 2005

[70] SurfAid. http://surfaid.dfw.ibm.com. 2005

[71] WebTrends Log analyzer. http://www.webtrands.com. 2005-06-27

[72] O. R. Zaiane, M. Xin y J. Han. Discovering Web access patterns and trends by applying olap and data mining technology on web logs. Advances in Digital Libraries. Páginas 19-29. Santa Barbara, CA. 1998.

[73] L. Catledge y J. Pitkow. Characterizing browsing behaviors on the world on the World Wide Web. Computer Networks and ISDN Systems. 1995.

[74] James Pitkow. Search of reliable usage data on the www. VI international World Wide Web Conference. Páginas 451-463. Santa Clara, CA. 1997.

Anexo A

Caso Práctico de la generación de patrones

A. Pruebas sobre el dominio cenidet.edu.mx

En esta sección se presenta un análisis completo del proceso de minería para el dominio cenidet.edu.mx, se muestran estadísticas sobre los datos de entrada los cuales representan los archivos log generados por el servidor Web durante los meses de agosto y septiembre del 2004.

Las gráficas obtenidas a partir de las tablas estadísticas de los meses de agosto y septiembre del 2004 para el dominio cenidet.edu.mx.

Se analizan las reglas de asociación localizadas durante el proceso de minería de cada mes y se presentan algunas de las reglas de asociación más interesantes.

A.1 Estadísticas para el mes de agosto del 2004

En esta sección se presentan estadísticas generales obtenidas a partir de las bitácoras generadas por el servidor Web del dominio cenidet.edu.mx durante el mes de agosto del 2004.

Todos los datos estadísticos que se muestran en esta sección fueron obtenidos por la herramienta Web Log Explorer Estándar Edition 2.84.

En algunas tablas de esta sección se muestran valores referentes a porcentajes, dichos porcentajes fueron obtenidos basándose en el total de la muestra, valor que se da al final de cada tabla. En la tabla A.1 se muestran datos generales sobre visitantes, hits[1] y promedios de visitantes[2].

Tabla A.1 Estadísticas generales para el mes de agosto del 2004.

Fechas de inicio y fin de almacenamiento	01/Agosto/2004 - 29/Agosto/2004
Número de archivo	4
Tamaño total de los archivos	69 MB
Número total de días	29
Total de hits	361693
Hits por día	12472
Promedio de páginas visitadas por día	1459

[1] Un acceso, una petición al servidor de un fichero.

[2] Internautas entra en una página; básicamente, eso es una visita.

Total de visitantes	21530
Promedio de visitantes por día	742
Direcciones IP únicas	11277
Tamaño de la transmisión	2.33 GB

En la tabla A.2 se muestran estadísticas relacionadas con las actividades de los usuarios sobre las páginas del sitio cenidet.edu.mx para el mes de agosto del 2004. Se presentan las 21 páginas mas solicitas así como el número de visitantes por página.

Tabla A.2. Páginas más visitadas durante el mes de agosto del 2004.

	Hits		Visitantes	
Página(Archivo)	Número	%	Número	%
/	8221	24.25	4279	17.17
/subaca/web-mktro/definicion.html	927	2.73	822	3.30
/subaca/web-mktro/guias.html	701	2.07	675	2.71
/subaca/web-mktro/	673	1.99	522	2.10
/seleccion/	616	1.82	513	2.06
/subaca/electron/	580	1.71	515	2.07
/seleccion/aceptados.php	553	1.63	475	1.91
/subaca/web-dcc/	512	1.51	460	1.85
/directorio.html	474	1.40	344	1.38
/subplan/ccomputo/	421	1.24	362	1.45
/subserv/biblioteca/menu.html	382	1.13	289	1.16
/bolsatrabajo.html	381	1.12	373	1.50
/subaca/web-mec/	359	1.06	242	0.97
/subserv/biblioteca/	354	1.04	285	1.14
/subserv/siceweb/	294	0.87	268	1.08
/subaca/web-mktro/posgrado.html	289	0.85	250	1.00
/subplan/comu-eve/junio/junio.html	260	0.77	29	0.12
/localizacion.html	253	0.75	242	0.97
/subaca/electron/jefedep.htm	236	0.70	202	0.81
/seleccion/info_deptos.html	235	0.69	210	0.84
/subaca/web-dda/	228	0.67	182	0.73
Otras	16948		13377	
Total	33897		24916	

En la tabla A.3 se muestran datos estadísticos basados en direcciones IP. Se muestran las 21 direcciones IP más activas dentro del mes de agosto del 2004. Esta claro que las direcciones IP con más actividad son aquellas que pertenecen a la red local del cenidet.

Tabla A.3. Direcciones IP más activas durante el mes de agosto del 2004.

		Hits		**Páginas**	
Host	**País**	**Número**	**%**	**Número**	**%**
192.168.1.23	N/A	4597	10.85	353	1.36

192.168.1.16	N/A	1817	4.29	133	0.51
200.65.129.25	México	616	1.45	247	0.95
200.23.91.222	México	265	0.63	169	0.65
217.73.164.106	Rumania	257	0.61	83	0.32
200.77.144.246	México	249	0.59	133	0.51
200.23.51.1	México	248	0.59	128	0.49
192.168.1.50	N/A	242	0.57	25	0.10
128.30.52.13	Estados Unidos	241	0.57	79	0.30
24.68.27.27	Canada	237	0.56	53	0.20
192.168.90.1	N/A	234	0.55	45	0.17
192.168.254.5	N/A	231	0.55	28	0.11
192.168.240.4	N/A	171	0.40	34	0.13
200.95.150.254	México	152	0.36	66	0.25
192.168.100.10	N/A	152	0.36	56	0.22
200.34.128.107	México	150	0.35	61	0.24
148.244.102.1	México	143	0.34	70	0.27
200.33.30.18	México	143	0.34	109	0.42
192.100.180.250	México	143	0.34	102	0.39
200.65.0.9	México	138	0.33	68	0.26
148.224.17.100	México	132	0.31	51	0.20
Otros		31687		23842	

En la tabla A.4 se presentan datos estadísticos sobre los tipos de archivo más solicitados.

Tabla A.4. Tipos de archivos más visitados durante el mes de agosto del 2004

	Hits		Visitantes	
Tipo de archivo	**Número**	**%**	**Número**	**%**
.html	29174	86.07	7838	77.24
.htm	4005	11.82	1809	17.83
.php	691	2.04	482	4.75
.exe	22	0.06	14	0.14
.ZIP	5	0.01	5	0.05

En la tabla A.5 se presentan las 21 páginas del dominio cenidet.edu.mx que los visitantes más utilizan como entrada al dominio, es decir estas son consideradas páginas de estrada al dominio por ser las páginas que en primera instancia los visitantes solicitan.

Tabla A.5. Páginas de entrada al sitio Web durante agosto del 2004.

	Hits		Visitantes	
Punto de entrada al sitio	**Número**	**%**	**Número**	**%**
/index.html	4066	43.26	4066	43.26
/subaca/web-mktro/definicion.html	647	6.88	647	6.88
/subaca/web-mktro/guias.html	518	5.51	518	5.51

/seleccion/aceptados.php	242	2.57	242	2.57
/subplan/ccomputo/ index.html	241	2.56	241	2.56
/seleccion/ index.html	136	1.45	136	1.45
/subaca/web-mktro/asigna.html	109	1.16	109	1.16
/bolsatrabajo.html	105	1.12	105	1.12
/localizacion.html	101	1.07	101	1.07
/subaca/web-dcc/Web-sd/LabSisDis/DescripEspSD.html	101	1.07	101	1.07
/subaca/web-mktro/ index.html	81	0.86	81	0.86
/subaca/web-dcc/web-sd/LabSisDis/TesisTerm.html	78	0.83	78	0.83
/subaca/web-mktro/posgrado.html	59	0.63	59	0.63
/subserv/biblioteca/ index.html	58	0.62	58	0.62
/subaca/web-dcc/web-dcc/Becas.html	50	0.53	50	0.53
/subaca/web-dcc/web-dcc/CursosActualizacion.html	45	0.48	45	0.48
/seleccion/fax.html	43	0.46	43	0.46
/subaca/web-mec/proyecto.html	43	0.46	43	0.46
/electron/bolio.htm	42	0.45	42	0.45
/subaca/electron/cdcdvisa.htm	41	0.44	41	0.44
/subaca/web-dcc/web-dcc/TemariosCursos/TemarioTeoComp.html	38	0.40	38	0.40
Otras	2555		2555	
Total	9399		9399	

A.2 Proceso minería para agosto del 2004

En la tabla A.6 se detallan los parámetros de ejecución para la carga y limpieza de los archivos log del mes de agosto del 2004.

Tabla A.6. Parámetros de limpieza para el mes de agosto del 2004.

Extensiones eliminadas	gif,jpg,ico,png,bmp,dib,jpeg,jpe,jfif,tif,tiff,jpeg,mp3,mpg, cab,css.
Tiempo (segundos).	163

En la tabla A.7 se presentan resultados obtenidos para los diferentes casos de sesionización. Se detallan los tiempos de ejecución y los resultados obtenidos.

Tabla A.7. Proceso de sesionización con diferentes parámetros.

Sesionización	No. Sesiones	No. Sesiones útiles	Tiempo Proceso
10 minutos	18000	5111	1254 segundos
15 minutos	17000	4805	1331 segundos
15 peticiones	12530	4836	564 segundos
Heurística	36685	3193	5372 segundos

A continuación se muestra un ejemplo de como se pueden interpretar los datos de la tabla A.7. En el proceso de sesionización que recibió el parámetro de 10 minutos como tiempo máximo

de duración de una sesión de usuario, se localizaron 18,000 sesiones de usuario, de la cuales sólo 5111 son útiles para el proceso de minería, (las sesiones útiles son aquellas que incluyen más de un recurso o página Web diferentes). El tiempo que tardó la herramienta en procesar esto fue de 1,254 segundos (aproximadamente 20 minutos).

En la figura A.1 se observa una gráfica donde se destacan los altos costos de procesamiento para la opción heurística, pero también se observa que es la opción que arroja el mayor número de sesiones; sin embargo, de ese número elevado de sesiones, solo una pequeña porción resulta útil para el proceso de minería y esto se debe a que la mayoría de las sesiones están integradas por una sola petición al sitio Web y para que una sesión sea considerada como útil para el proceso de minería, esta debe estar conformada por al menos 2 peticiones a recursos diferentes.

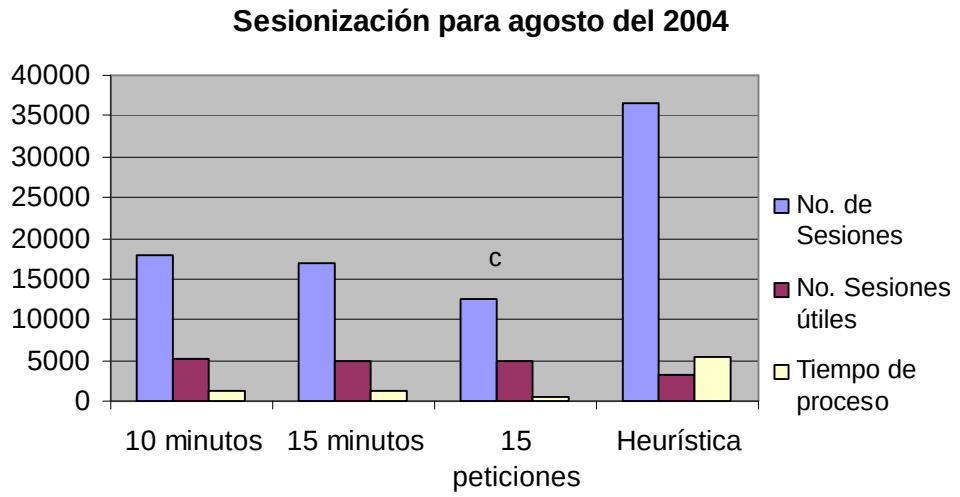


Figura A.1. Gráfica par el proceso de sesionización, (agosto 2004).

En la tabla A.8 se presentan los resultados obtenidos durante el proceso de minería aplicado a cada método de sesionización. Se detallan los tiempos de ejecución y los resultados obtenidos.

Tabla A.8. Minería de reglas de asociación, (agosto del 2004).

Sesionización	Soporte	Confianza	Reglas	Tiempo
10 minutos	20	20	0	76 Segundos
10minutos	10	50	0	4 segundos
10 minutos	8	50	3	5 Segundos
15 minutos	20	20	0	15 Segundos
15 minutos	10	50	0	4 Segundos
15 minutos	8	50	4	5 Segundos
Heurística	20	20	0	10 Segundos
Heurística	10	50	4	5 Segundos

Heurística	8	50	13	6 Segundos
15 Peticiones	20	20	0	14 Segundos
15 Peticiones	10	50	0	5 Segundos
15 Peticiones	8	50	4	6 Segundos

A los resultados del proceso de sesionización basado en 10 minutos se les aplicó minería de reglas de asociación utilizando un 20% de soporte y un 20% de confianza teniendo como resultado cero reglas de asociación. La herramienta tardo 76 segundos en realizar este proceso. En la tabla A.9 se presenta la regla más significativa que se obtuvo con cada método de sesionización aplicado al mes de agosto del 2004.

Tabla A.9. Reglas más significativas para cada método de sesionización, (agosto 2004).

Método	Regla	Soporte	Confianza
10 Minutos	[/subaca/electron/index.html] ---->[/index.html]	8.921933%	86.69202%
15 Minutos	[/subaca/electron/index.html] ---->[/index.html]	9.510926%	88.56589%
Heurística	[/subaca/web-dcc/index.html] ---->[/index.html]	11.243345%	85.07109%
15 Peticiones	[/subaca/electron/index.html] ---->[/index.html]	9.0363935%	86.193293%

Hay que destacar que los valores de las columnas de Soporte y Confianza son los valores obtenidos para cada regla, es decir, el proceso de minería aplicado a los resultados de la sesionización basada en 10 minutos localizó la regla interesante:

[/subaca/electron/index.html] → [/index.html]

La cual se localizó durante un proceso de minería que buscaba reglas con soporte mayor o igual a 8% y una confianza mayor o igual a 50%, sin embargo la regla superó los valores mínimos introducidos por el usuario logrando obtener un 9.51% de Soporte y un 86.69% en el segundo renglón. Se dice que la regla es interesante en este caso, por el hecho de haber superado los valores de interés mínimos del usuario, en este caso, los valores de interés mínimos para el usuario fueron de 8% de soporte y 50% de confianza.

La información que podemos extraer de la regla, indica que el 8.9% de las sesiones de usuarios incluyen las páginas /subaca/electron/index.html e /index.html. La regla también indica que cuando un visitante accede a la página /subaca/electron/index.html existe un 86.69% de probabilidades de que acceda posteriormente a la página index.html.

En la tabla 16 se puede observar que el método de sesionización heurístico es el único que arroja una regla diferente, a diferencia de los métodos restantes que arrojan la misma regla en los

3 casos. Además, los valores de interés de la regla localizada en el método heurístico es la que tiene el interés más alto. En la figura A.2 se muestra una gráfica de los procesos de minería para el mes de agosto del 2004, de esta gráfica se puede destacar que el método heurístico es el que arroja el número de reglas más grande. En la primera ejecución de las tres que se realizaron para cada opción de sesionización, el tiempo es mayor que en los dos restantes, esto se debe a que en la primera ejecución, el sistema tiene que cargar en memoria las transacciones para posteriormente aplicar el algoritmo.

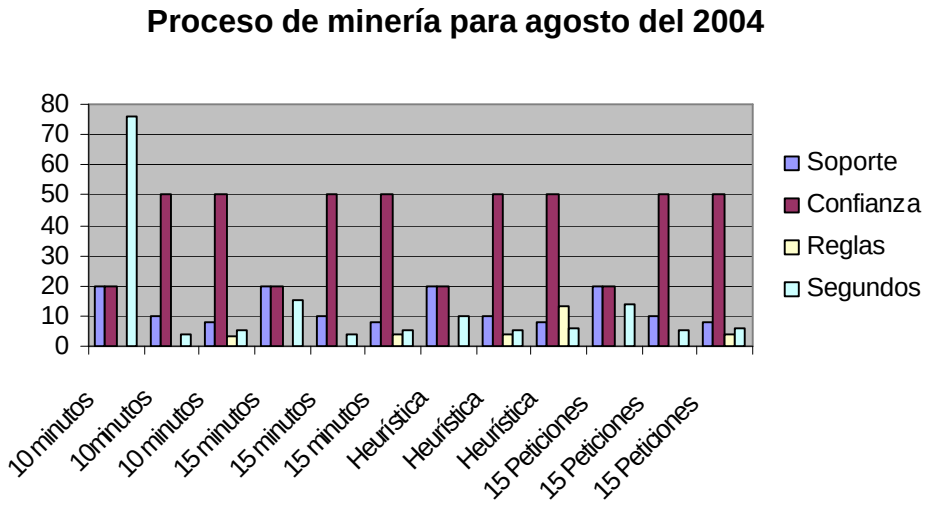


Figura A.2. Gráfica para el proceso de minería, (agosto 2004).

De la tabla A.8 se puede observar que los únicos valores de soporte y confianza que permitieron localizar reglas de asociación en todos los casos fueron los siguientes:

Soporte = 8%, Confianza = 50%

Con el objeto de detectar el comportamiento de la herramienta en casos donde el soporte de las reglas es demasiado bajo se realizaron pruebas utilizando un 2% para el valor del soporte.

En la tabla A.10 se muestra el número de reglas generadas en cada caso.

Tabla A.10. Reglas generadas con 2% de soporte.

Sesionización	Número de reglas
10 Minutos	300
15 Minutos	330
Heurística	436
15 Peticiones	258

En la tabla A.11 se muestran solo algunas reglas de la gran cantidad de reglas localizadas sobre las pruebas realizadas de la tabla 17. Estas reglas incluyen a las más interesantes y algunas de interés más bajo y fueron escogidas al azar.

Tabla A.11. Reglas localizadas en el mes de agosto del 2004.

Método	Regla	Soporte	Confianza
10 Minutos	[/noticias.html]--->[/index.html]	3.18%	88.10%
10 Minutos	[/cgi-bin/biblioteca/biblioteca.pl]---> [/index.html]	4.69%	63.15%
10 Minutos	[/subaca/web-dcc/web-dcc/CursosActualizacion.html]---> [/index.html]	3.07%	84.40%
15 Minutos	[/seleccion/info_tramites.html]---> [/seleccion/aceptados.php]	2.33%	70%
15 Minutos	[/seleccion/avisos.html]---> [/seleccion/index.html]	2.37%	78.62%
15 Minutos	[/index.html, /subaca/electron/iemces.htm, /subaca/electron/jefedep.htm]---> [/subaca/electron/index.html]	2.66%	100%
Heurística	[/subaca/web-ose/pubose/index.html]---> [/index.html]	2.97%	71.96%
Heurística	[/subplan/planeacion/index.html]---> [/index.html]	2.53%	78.64%
Heurística	[/subplan/ccomputo/index.html]---> [/index.html]	2.91%	72.65%
15 Peticiones	[/seleccion/aceptados.php]---> [/seleccion/index.html]	4.69%	63.94%
15 Peticiones	[/seleccion/info_tramites.html]---> [/seleccion/aceptados.php]	2.17	64.81%
15 Peticiones	[/seleccion/info_becas.html]---> [/seleccion/aceptados.php]	2.02%	51.04