Marco de Referencia para la Construcción de Aplicaciones de Comercio Electrónico Móvil en Países en vía de Desarrollo

Anexos

Francisco Orlando Martínez Pabón

Director: Mag. Oscar Mauricio Caicedo Rendón

Universidad del Cauca
Instituto de Postgrados en Electrónica y Telecomunicaciones
Maestría en Ingeniería, Área Ingeniería Telemática
Departamento de Telemática
Línea de Investigación en Servicios Avanzados de Telecomunicaciones
Popayán, Octubre de 2008

Contenido

Anexo G

Lista de Figuras

Lista de Tablas

Anexo A

W3C Mobile Web Initiative

El W3C es un organismo altamente reconocido en el mundo empresarial por sus estándares en cuando a generación, construcción y uso de contenido Web; recientemente, ha definido una iniciativa conocida como la Mobile Web Initiative (MWI), cuyo objetivo primordial es garantizar que el usuario pueda extraer el mayor provecho de su dispositivo y de la Web mientras navega. La MWI define una serie de documentos que analizan diferentes aspectos de la problemática desatada debido a la gran variedad de dispositivos móviles que acceden a la Web (W3C, 2008). De este conjunto de documentos se incluye uno titulado "Buenas Prácticas de la Web Móvil" (Rabin & McCathieNevile, 2006) el cual ha sido la base fundamental del presente trabajo y contiene una serie de recomendaciones que buscan dar pautas claras para permitir la convergencia entre la Web convencional y la Web móvil. Estas recomendaciones involucran a las empresas generadoras de contenido, desarrolladores de aplicaciones Web, así como los desarrolladores de aplicaciones de adaptación de contenido. En otras palabras se podría decir que el documento especifica las buenas prácticas para la correcta entrega de contenidos Web a los dispositivos móviles. A continuación se explicará los aspectos más relevantes encontrados en los documentos de la MWI haciendo especial énfasis en el documento "Buenas Prácticas de la Web Móvil".

1.1. Caracterización de la MWI

El W3C ha definido una serie de categorías para agrupar los problemas que se presentan y dar alternativas de solución a través de MWI. A continuación se describe la actual problemática clasificada en distintos aspectos y la solución que se pretende.

1.1.1. Presentación

Problema:
Actualmente las páginas aprovechan las capacidades de los navegadores de escritorio, sin embargo la experiencia del usuario móvil en estas páginas es muy pobre y poco práctica, ya que por las bajas resoluciones soportadas en los dispositivos muchos contenidos se pierden o son de difícil acceso.

Solución:
Se pretende entonces aprovechar al máximo las capacidades de los navegadores de bajas prestaciones.

1.1.2. Entrada

Problema:
Un problema común de la navegación móvil es la dificultad al introducir texto como las URL[1] extensas y textos con diferentes signos de puntuación.

Solución:
Las recomendaciones pretenden dar facilidades de entrada de datos a los usuarios limitados con teclados telefónicos.

[1] URL: Uniform Resource Locator (Localizador Uniforme de Recursos)

1.1.3. Costo del ancho de banda

Problema:
En los dispositivos móviles el costo de la transferencia de datos es usualmente mayor y en muchas ocasiones se transmiten datos que son inútiles pues el dispositivo no está en la capacidad de presentarlos.

Solución:
Se pretende eliminar la transferencia de cabeceras y código inoficioso por la red móvil.

1.1.4. Objetivos del usuario

Problema:
Los usuarios móviles usualmente tienen otros intereses al visitar las páginas Web, su preferencia está en los contenidos breves y muy concretos que responden a su necesidad inmediata; contrario a los usuarios Web de escritorio que acostumbran a descargar contenidos pesados y documentos voluminosos.

Solución:
Dar pautas para los diseñadores y generadores de contenido de tal forma que ayuden al usuario a obtener la información que realmente busca, de forma inmediata.

1.1.5. Publicidad

Problema:
La publicidad en los dispositivos móviles es indeseable, debido a las bajas resoluciones de pantalla e incapacidades de los dispositivos, como la incapacidad de manejar pop-ups[2].

Solución:
Eliminación de este tipo de contenido no deseado.

1.1.6. Limitaciones de los dispositivos

Problema:
Además de las limitaciones ya nombradas, son muchas las limitaciones que afectan la usabilidad de la Web móvil, por ejemplo la falta de soporte para plugins o scripts en los exploradores. Muchas páginas en la Web utilizan Javascript con el fin de mejorar el aspecto de sus páginas y procesar formularios antes de ser enviados al servidor. En muchos de estos casos, dicho código es indispensable para la funcionalidad de la página provocando graves limitaciones en dispositivos que no interpreten código Javascript. Los plugins en la Web son usados principalmente para reproducir contenido multimedia como videos, música y animaciones desde los navegadores. Los diseñadores usan frecuentemente las animaciones como acceso a las páginas o como disparadores de una funcionalidad de la página, esto provoca que sea necesario ingresar desde un navegador que soporte plugins para acceder a todos los servicios del portal.

Solución:
Eliminar la dependencia de tecnologías que limiten el acceso a los dispositivos de bajas prestaciones para hacer uso de los servicios prestados por un portal web.

1.1.7. Ventajas de la Web móvil

Problema:
La Web móvil tiene un gran número de ventajas, se puede acceder en cualquier momento a la página favorita y explorar en ella información del lugar exacto desde donde se realiza la consulta. Otra ventaja fundamental es que la cantidad de usuarios que pueden acceder a la Web móvil sobrepasa con creces a los de la Web de escritorio.

[2] Pop-Up: Ventana Emergente.

Actualmente, este potencial es desperdiciado por un gran número de portales, perdiendo la atención de los usuarios móviles.

Solución:
Obtener el mayor provecho de estas ventajas por medio de las mejoras en presentación, ofreciendo mecanismos de personalización de las páginas Web y utilizando al máximo la información de las características del dispositivo usado, así como su ubicación geográfica.

1.2. Definiciones básicas

Antes de enunciar las recomendaciones dadas en la MWI, es necesario aclarar ciertos conceptos que permitirán un entendimiento más claro y contextualizado.

1.2.1. Contexto De Entrega

Define las capacidades del dispositivo, su entorno (ancho de banda, geo-posición, etc.) y las preferencias de usuario (Lewis, 2005). Su principal funcionalidad es brindar información para realizar una correcta adaptación del contenido y su presentación. En muchas ocasiones es necesario saber cuál es el contexto de entrega, sobre todo para páginas de navegación compleja, en las cuales se presentan imágenes y objetos de distintos tipos o formatos. Actualmente existen varias implementaciones del contexto de entrega, como CCPP, UAPROF o Deli[3], basadas en XML.

Existe un contexto de entrega mínimo que se ha definido para tener una experiencia agradable al visitar un sitio Web desde un dispositivo móvil; a éste se le conoce también como contexto de entrega por defecto y se define así:

- Ancho usable de la pantalla 120 pixeles.
- Soporte XHTML Basic 1.1.
- Codificación del carácter UTF-8[4] [UTF-8].
- Formato de la imagen JPEG[5]. GIF 89a.[6]
- Peso total máximo de la página 20 kilobytes.
- Colores 256 colores, mínimo.
- Hoja del estilo nivel 1 del CSS.
- HTTP/1.0 o más reciente HTTP1.1.
- Ninguna capacidad para scripting del lado del cliente.

1.2.2. Web unificada

El concepto de la *Web unificada* se refiere al hecho de tener un único contenido y los usuarios puedan acceder a él desde cualquier dispositivo, teniendo una experiencia agradable, satisfactoria y aprovechando las ventajas de cada dispositivo. Esto no significa que toda la información tenga exactamente la misma representación en la variedad de dispositivos sino que la representación de la información variará de acuerdo al contexto de entrega.

Es importante aclarar que muchos servicios tendrán un especial atractivo para un cliente móvil (servicios de localización), mientras que otros servicios tendrán especial atractivo para los clientes Web convencionales (Imágenes de alta calidad, textos extensos).

1.2.3. Adaptación de contenido

Debido a diferencias marcadas en las capacidades de los dispositivos, es necesario realizar una adaptación de contenido, es decir alterar la presentación del mismo y parte de él de tal forma que el usuario tenga una mejor experiencia al adquirir la información. Un ejemplo frecuente de adaptación de contenidos es la reducción de

[3] Cada uno de estas implementaciones serán estudiadas profundamente en el capítulo 4.
[4] UTF-8: 8-bit Unicode Transformation Format
[5] JPEG: Joint Photographic Experts Group .
[6] GIF 89a: Versión de Compuserve GIF o Graphics Interchange Format.

imágenes; en ella se altera el tamaño original para evitar sobrecostos en redes de bajas capacidades y mostrar apropiadamente la imagen en el dispositivo desde el cual se solicita.

Existen muchas maneras de realizar la adaptación de contenidos, algunas de menor complejidad debido a que realizan una adaptación estática, mientras que otras se tornan complejas debido a que adaptan el contenido dinámicamente. La adaptación estática tiene en cuenta una fracción menor del contexto de entrega y en la mayoría de los casos solo utiliza información cómo la gama del dispositivo. Por el contrario la adaptación dinámica hace uso de cada una de las características del contexto de entrega obteniendo una adaptación más consecuente con las capacidades del dispositivo y de su entorno actual.

Existen tres modelos de adaptación de contenidos:

1) *Adaptación en el servidor (Server)*: En este modelo el servidor de contenidos Web es quien realiza la adaptación de su propio contenido para que pueda ser visualizado correctamente desde la mayor gama de dispositivos. Este tipo de adaptación se realiza de manera transparente para el usuario y no es necesario que se realice una modificación del dispositivo o de su navegador. Usualmente esta adaptación es estática, es decir presenta una versión diferente según el dispositivo que realice la petición. Su principal desventaja consiste en los costos agregados que genera la construcción de distintas versiones para los diferentes tipos de dispositivos y su ventaja radica en la adaptación de alta calidad utilizando al máximo los recursos de cada dispositivo.
2) *Adaptación intermedia (In-Network):* Este tipo de adaptación es realizado usualmente por un proxy y suele necesitar la configuración del dispositivo. El proxy o servidor intermedio contiene un programa especializado para realizar la adaptación de contenidos de forma dinámica. Tiene como ventaja la centralización del proceso de adaptación, disminuyendo los costos en la construcción de prototipos para diferentes gamas de dispositivos y su desventaja es la disminución en la calidad de la adaptación
3) *Adaptación en el cliente (Client):* En el navegador del dispositivo se realiza una modificación o actualización con el fin de que este procese los objetos Web (imágenes, texto, scripts, animaciones, etc.) y sean correctamente adaptados. Las desventajas más notorias de este tipo de adaptación es su baja calidad debido a las limitadas capacidades de procesamiento de la mayoría de dispositivos.
4) *Adaptación combinada*: Cualquier combinación de los tres modelos de adaptación básicos anteriormente descritos se conoce como adaptación combinada y busca aprovechar las ventajas de cada uno de los modelos propuestos anteriormente. Un ejemplo de adaptación combinada es el navegador mini-Opera (Goldman, 2006), el cual realiza adaptación en el cliente e intermedia.

En la primera versión de las recomendaciones propuestas por MWI se ha supuesto que la adaptación es del lado del servidor, por su mayor simplicidad, pero se espera que en futuras fases se tenga en cuenta los otros tipos de adaptación.

1.3. Buenas prácticas planteadas por la MWI

A continuación se realizará una breve descripción de cada una de las recomendaciones dadas por el W3C a través de MWI siguiendo la clasificación publicada en su documento “Buenas Prácticas de la Web Móvil”.

1.3.1. Sobre el comportamiento

Temática consistente con la URL

Esto significa que el usuario al acceder a una página Web desde cualquier dispositivo debe encontrar la misma información. Es el caso de que un usuario introduzca la URL: http://www.eltiempo.com.co en su computador de escritorio y seguido a ello introduzca la misma URL en su teléfono celular, ambos dispositivos deben mostrar como resultado de la petición los titulares de la fecha, posiblemente con un formato y presentación distintos.

Explotar las capacidades del dispositivo

Es necesario que se tenga en cuenta el contexto de entrega para aprovechar de la mejor manera las capacidades de cada dispositivo. La adaptación de imágenes debe realizarse de tal manera que no desperdicie espacio en pantalla, ni permita que se descargue un contenido con mayor calidad que la que se puede visualizar.

Trabajo alrededor de Las implementaciones deficientes

Debido a las implementaciones defectuosas en algunos dispositivos (browser con defectos) es necesario que en algunos casos los proveedores de contenido pasen por alto algunas de las buenas prácticas, con el fin de alcanzar el mayor número de dispositivos posibles. Algunos exploradores envían en la cabecera HTTP Accept los caracteres "*/*" indicando que soportan todo tipo de archivos, sin embargo algunos formatos de imágenes son ignorados o mostrados defectuosamente.

Pruebas

Cualquier página Web debe ser probada en el mayor número de dispositivos y en sus respectivos emuladores. Hay que tener en cuenta que en muchos casos el emulador suministrado por los fabricantes no se comporta de igual manera que el dispositivo, por tanto se recomienda que se pruebe directamente en el dispositivo teniendo en cuenta las versiones del software.

1.3.2. Navegación y enlaces

Estas recomendaciones fueron desglosadas en la sección 3.2.3.1 Servicios de Conocimiento en la monografía.

1.3.3. Diseño de la página y contenido

Contenido de la página

Se debe garantizar que el contenido es adecuado para mostrarlo en un dispositivo móvil; debe evitarse el contenido multimedia de alta calidad, puesto que no podrán ser mostrados en la mayoría de dispositivos. Usar lenguaje claro y simple. El contenido de la página debe limitarse a lo que el usuario ha solicitado, las páginas con mucha publicidad provocan confusión y ocasionan que se pierda el objetivo de la consulta. En la Figura 1 se muestra un ejemplo de exceso de publicidad, lo cual puede ocasionar confusiones al acceder la página desde un dispositivo móvil.

Figura 1 Exceso de publicidad en la página

Tamaño de la página

Si se crean páginas muy grandes se incrementa innecesariamente el tiempo de descarga de las mismas; por otro lado, algunos dispositivos móviles limitan el tamaño de la página a visualizar, lo cual implica que si la esta es demasiado corta, entonces el usuario tendrá que navegar por muchas páginas para obtener una información de importancia. En lo posible, el contenido de la página Web no puede exceder los 10Kb y agregando las imágenes no puede exceder de los 20Kb.

Desplazamiento (Scrolling)

El desplazamiento de la página debe ser en una sola dirección. Es recomendable ubicar en otra página diferente las imágenes u objetos que sobrepasen los 120 pixeles de ancho. En la Figura 2 se muestra un ejemplo de una imagen de gran tamaño que no podrá ser mostrada completamente en muchos dispositivos móviles. Por esta razón, es recomendable mostrar inicialmente una versión de menor tamaño y si el usuario desea observar la imagen con mayor detalle darle la posibilidad por medio de un hipervínculo de observar la imagen original.

Figura 2 Imagen que excede los 120 pixeles de ancho en la página www.nasa.gov

Barras de navegación

Se debe asegurar que el contenido principal de la página se ubique en la parte superior, sobre cualquier contenido considerado como adicional. En la página del diario el Tiempo (www.eltiempo.com), se encuentra un buen ejemplo de la distribución de las barras de navegación, puesto que la mayoría de las barras de navegación se encuentran después del contenido principal (Figura 3).

Gráficos

No deben utilizarse imágenes para espaciar y se debe evitar el uso de gráficos de 1 pixel o transparentes con el objeto de tener una posición absoluta. En lo posible, se deben evitar las imágenes que tengan una resolución muy alta o una profundidad de colores pronunciada, ya que la mayoría de dispositivos no podrán desplegarlas correctamente, al tiempo que se consume un mayor ancho de banda.

Color

Es necesario que el contraste entre las letras y el fondo sea apreciable, pues algunos dispositivos tienen una menor definición de colores y pueden llegar a disminuir el contraste de la página, lo cual hace ilegible el texto de la misma. Se debe evitar también que el despliegue de la información dependa de los colores, puesto que existen dispositivos que soportan un número limitado de ello. Adicionalmente, es conveniente evitar el uso del color azul y el morado para las letras, ya que el texto puede confundirse con los hipervínculos.

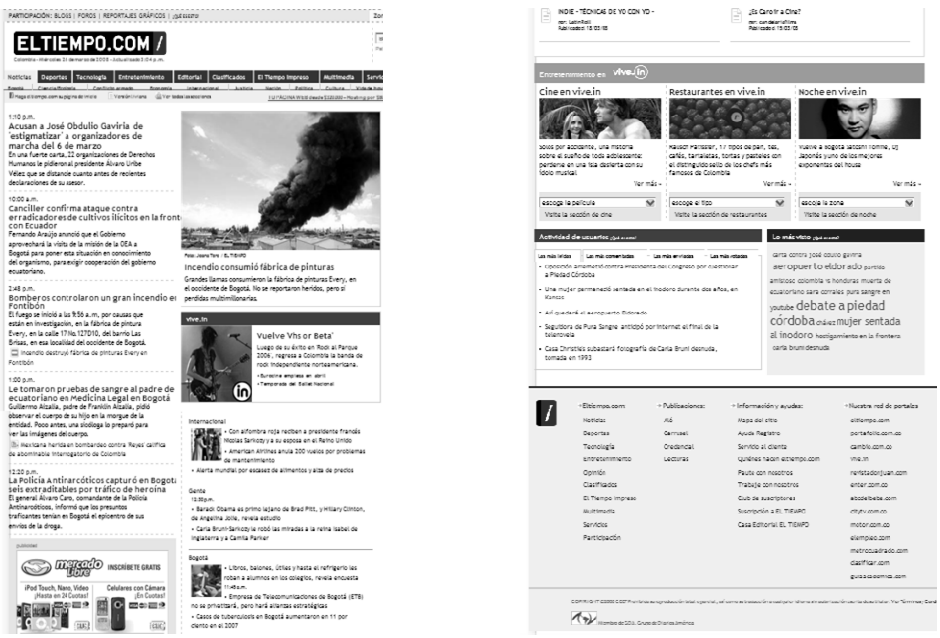


Figura 3 Página www.eltiempo.co. (a) Contenido central ubicado en la parte superior. (b) Contenido complementario ubicado al final de la página.

Imágenes de fondo

Las imágenes de fondo usadas indiscriminadamente pueden causar que el contenido de la página no pueda ser leído por el usuario. Debe asegurarse que el contenido de la página se pueda leer sin imágenes de fondo, puesto que algunos dispositivos no soportan esta característica. En la Figura 4, el texto encerrado en la elipse es ilegible a causa del mal uso de las imágenes de fondo.

Figura 4 Uso incorrecto de imágenes de fondo

1.3.4. Definición de la página

Estas recomendaciones fueron desglosadas en la sección 3.2.3.1 Servicios de Conocimiento en la monografía.

1.3.5. Interfaces de entrada

Entrada por teclado

Es recomendable que el número de teclas que el usuario debe presionar sea mínimo, debido a las limitaciones del dispositivo en sus interfaces de entrada. En este sentido, es necesario disminuir al máximo los campos de texto y usar listas, botones de selección y entradas de texto restrictivas, frecuentemente. Es una buena práctica, colocar una entrada por defecto.

Orden en las tabulaciones

Algunos usuarios usan la tecla “tab” para movilizarse por la página. Para permitir un buen uso de esta facilidad, se debe crear un orden lógico para viajar entre los diferentes enlaces, objetos y controles. El orden se establece usando el atributo “tabindex”.

Etiquetas para los controles de formularios

Es recomendable usar el elemento *label* en HTML o su equivalente en otros lenguajes, con el fin de guiar al usuario en la introducción de información en un formulario. Debe asegurarse que el campo y la etiqueta correspondiente estén lo más cerca posible, para que cuando se realice un proceso de adaptación no se desliguen.

Anexo B

Especificación de la plataforma OneWeb

2.1. Introducción

Uno de los factores indispensables para consolidar la Web móvil es brindar al usuario una información transparente e independiente del tipo de dispositivo, desde el cual accede. Por lo tanto, el objetivo no es ofrecer un contenido limitado con el fin de garantizar el acceso a todos los dispositivos sino por el contrario, tratar de aprovechar las características particulares de cada dispositivo para brindarle al usuario la mejor experiencia posible. Siguiendo esta filosofía, el W3C ha definido la Mobile Web Initiative, un conjunto de recomendaciones para crear y habilitar un acceso adecuado a contenidos Web, desde dispositivos móviles.

OneWeb es una plataforma de adaptación automática de contenidos Web para dispositivos móviles, capaz de reconocer las capacidades de los terminales de acceso para ajustar la información de las páginas siguiendo las recomendaciones definidas por la Mobile Web Initiative - MWI.

2.2. Selección de recomendaciones de la MWI

Algunas de las recomendaciones de la MWI están dirigidas a aspectos de diseño que se deben tener en cuenta durante la construcción de las páginas por parte de los desarrolladores de los portales; por esta razón, este tipo de directrices no fueron tenidas en cuenta como parte del proceso de adaptación realizado por OneWeb. A continuación se describe el proceso de selección de estas recomendaciones.

2.2.1. Recomendaciones no implementadas

Trabajo alrededor de las implementaciones deficientes: la plataforma implementada tiene en cuenta con rigor las recomendaciones del W3C, por tanto no se tienen en cuenta aquellos dispositivos que tengan defectos en la implementación de los estándares; no obstante, las pruebas realizadas sobre diversos dispositivos arrojaron resultados satisfactorios.

Pruebas: la realización de pruebas no es una operación de la plataforma como parte del proceso de adaptación, sin embargo durante el desarrollo de esta se realizaron diferentes pruebas en un gran número de dispositivos, de variadas capacidades.

URL de los puntos de acceso: la URL del punto de acceso no es modificada por la plataforma, y es el dato de entrada que la plataforma usa para realizar la adaptación de contenidos.

Estructura balanceada: el correcto balance de la página recae sobre el diseñador del sitio Web. Sin embargo, por medio de los procesos adaptativos que realiza la plataforma, dicha característica se mejora. Al realizar la segmentación indexada y la transformada de elisión selectiva, permite que las páginas sean más cortas y el usuario encuentre con mayor facilidad la información requerida.

Mecanismos de navegación: la plataforma no introduce ningún método de navegación adicional al que incluya la página, por ende esta recomendación compromete al diseñador de la página Web. El código HTML no tiene información referente al método de navegación que se está usando, sino que esta se encuentra mezclada con la

información o contenido de la página Web; por eta razón, las plataformas software no pueden alterar los métodos de navegación, sin alterar la información de la página Web produciendo efectos impredecibles.

Contenido de la página: la plataforma descarga los contenidos que el dispositivo soporta y se realiza la adaptación de imágenes, con el fin de garantizar que serán mostradas correctamente; sin embargo no garantiza un lenguaje claro y simple, pues esto esta tarea es responsabilidad de quien genera los contenidos.

Imágenes de fondo: en ninguna de las fuentes que conforman el contexto de entrega (WURFL, UAProf y Cabeceras HTTP) se encuentra información acerca del soporte de las imágenes de fondo; debido a esta limitante, la plataforma no puede garantizar que el contenido sea observable en los dispositivos que no muestran imágenes de fondo.

Título: el diseñador de la página Web, es el responsable de garantizar que el título de la página describa correctamente el contenido de la misma. La plataforma no añade ningún tipo de información y conserva el título creado originalmente. En el caso de las páginas que contienen marcos, se titula la página adaptada mediante el título del primer marco.

Elementos estructurales: OneWeb hace uso de los elementos estructurales para analizar las diferentes secciones de la página. Sin embargo, es labor del diseñador incluir el mayor número posible de estos elementos, para obtener mejores resultados en la adaptación. Se incluye esta recomendación dentro de las no implementadas puesto que la plataforma no incluye nuevos elementos estructurales.

Objetos no-textuales: actualmente no existen librerías para la adaptación de objetos no-textuales como objetos Java, Flash, etc. Este tipo de archivos complejos de procesar y de difícil adaptación.

Unidades de medida: el diseñador de la página Web debe indicar las medidas con las cuales desea que se muestren los objetos. La plataforma respeta estas medidas mientras sea desplegable en el dispositivo y no se introduzca una barra de navegación horizontal; de lo contrario se realiza una adaptación del objeto con las medidas del despliegue.

Cookies: esta recomendación está dirigida especialmente a los desarrolladores de las aplicaciones Web, puesto que el procesamiento de cookies es labor interna de la aplicación en el servidor y del navegador del cliente final, por tanto la plataforma es transparente a este proceso y transmite directamente toda la información relacionada con las cookies.

Fuentes: el número de fuentes que presenta la página Web es conservado por la plataforma, debido a que no existe en el código HTML información que asegure la pertinencia o no de un cambio de fuente.

Entrada por teclado: la plataforma no altera los tipos de entrada de datos diseñados originalmente. Simplemente, los campos de texto son adaptados al tamaño de la pantalla.

Etiquetas para los controles de formularios: si el elemento *label* no es introducido por el diseñador de la página, no existe suficiente información para asociar el texto a un campo de texto determinado; por lo tanto, esta recomendación debe ser acatada por el diseñador Web, para garantizar un mejor proceso de adaptación.

2.2.2. Recomendaciones implementadas

Temática consistente con la URL: las diferentes adaptaciones que genera la plataforma para cada uno de los dispositivos, no altera de ninguna forma el contenido, tan solo altera su presentación; por tanto al acceder a cierta URL la plataforma siempre entregará la misma información.

Explotar las capacidades del dispositivo: la plataforma realiza una adaptación dinámica, ya que se obtiene un contexto de entrega que incluye las capacidades de cada dispositivo como paso previo al proceso de adaptación

Barras de navegación: se ha implementado un proceso que desplaza los menús al final de la página y se asegura que el contenido principal se encuentre en la parte central. En primer lugar, se clasifica las secciones en dos categorías: barra de navegación y contenido; acto seguido, se ubica la primera barra de navegación en la parte superior de la página y el resto de éstas al final; por último se aplica la transformada de elisión. Los criterios que se tienen en cuenta para la clasificación de las secciones de la página se definen de la siguiente manera:

*A= ((número de caracteres en los hipervínculos de la sección) / (número caracteres de la sección))*100*
*B= ((número de caracteres en los hipervínculos de la sección)/ (número de caracteres en el total hipervínculos))*100*

El valor de A debe superar el 60%, mientras que el valor de B debe ser menor al 90% para que la sección sea clasificada como una barra de navegación. El valor de A mide el número de hipervínculos que tiene una sección; si el contenido de ésta tiene un alto porcentaje de hipervínculos, se considera entonces una barra de navegación. El valor de B indica el tamaño de la sección y es limitado para evitar que casi gran parte del contenido sea clasificado como una barra de navegación.

Teclas de acceso: todos los hipervínculos con etiquetas *H0, H1, H2* y *H3* se les agrega un atributo numérico *accesskey*, asignando un número menor a los hipervínculos que se encuentran en la parte superior de la página y uno mayor a los posteriores.

Identificación del enlace de destino: la plataforma introduce el tamaño en Kb de los hipervínculos que están identificados con las etiquetas *H0, H1, H2* y *H3*, por ser considerados hipervínculos de mayor relevancia para el usuario. No se ha realizado este procedimiento con todos los hipervínculos pues esto introduce un retardo significativo en la respuesta del servidor.

Mapas de imágenes: en caso que el dispositivo solo disponga de un teclado numérico, se añade un hipervínculo por cada área del mapa de imágenes; de esta forma, se garantiza que se muestra la imagen y se puede acceder a los destinos del mapa.

Recarga, redirección y ventanas emergentes: se eliminan las etiquetas de recarga y redirección; el atributo *target* de los hipervínculos es fijado en el valor *_self,* evitando la generación de ventanas emergentes. En otras palabras, se garantiza que la descarga del contenido Web es activada por el usuario y se realiza en la misma ventana.

Tamaño de la página: la página original se fracciona en pequeñas subpáginas, aprovechando la máxima capacidad de alojamiento del dispositivo. Para conocer la máxima capacidad del dispositivo se usa como primera instancia la información de WURLF, más exactamente el atributo *max_deck_size*; en caso que esta información no esté disponible se usa el atributo *WmlDeckSize* de UAProf. Parte de este procedimiento, se conoce como transformada de segmentación indexada como se explicó en la monografía. Si definitivamente no es posible obtener la máxima capacidad que soporta el dispositivo, se limita el tamaño a 10Kb como lo indica la recomendación.

Desplazamiento: se destruyen las estructuras que provocan un desplazamiento horizontal es decir, se eliminan las etiquetas * *, *<table>*, *<td>* y *<tr>*, más no se elimina su contenido. La plataforma realiza adaptación de imágenes en formato y tamaño de acuerdo a las capacidades del dispositivo.

Gráficos: la plataforma adapta la profundidad de colores de acuerdo a las capacidades del dispositivo; además, se eliminan los gráficos que tengan un tamaño de 1 pixel usados comúnmente para espaciar. La profundidad de color del dispositivo es obtenida inicialmente de la característica *color* incluida en WURFL; si ésta no está disponible, entonces se usa la característica de UAProf, *BitsPerPixel*. Si en forma definitiva no es posible obtener la profundidad de color, el valor por defecto es de 8 bits, es decir 256 colores.

Color: la plataforma revisa las hojas estilo, con el fin de evitar colores similares a los hipervínculos en las fuentes. Si el color de una fuente se encuentra dentro del rango de los siguientes colores *RGB(0,0,FF hex), RGB(80 hex,0,0)* y *RGB(80 hex,0,80 hex),* éste es convertido al color negro. Lo relativo al contraste y el uso de colores como información, es responsabilidad del diseñador Web.

Marcos: se ha agregado el contenido de cada uno de los marcos en forma secuencial, eliminándolos por completo, dejando su contenido intacto. Esto se logra recuperando las cabeceras y cuerpos de cada uno de los marcos que componen la página y unificándolos en un solo contenido HTML.

Tablas: como se explicó anteriormente, las tablas son eliminadas con el fin de obtener una página con desplazamiento único.

Tamaño de la imagen: la plataforma ajusta el tamaño de las imágenes de acuerdo a las dimensiones de su pantalla. Esta características se extrae de la propiedad *resolution_width* de WURFL, o alternativamente la propiedad *ScreenSize* de UAProf; si la característica no se encuentra disponible, se establece un tamaño por defecto de 120 pixeles.

Etiquetas válidas: las etiquetas no válidas se eliminan durante la fase de pre-procesamiento, donde se corrigen varios errores relacionados con la sintaxis de la página.

Hojas de estilo: el W3C recomienda que las etiquetas <*font*> sean eliminadas. En su lugar, éstas son convertidas en estilos en cascada, se unifican todos en una hoja de estilos externa y por último se reducen las hojas de estilo, eliminando estilos propios de etiquetas que no se encuentren en la página. Previamente a este proceso, la plataforma comprueba si el dispositivo soporta hojas de estilo, a través de la característica *CcppAccept* de UAProf. Si esta información no está disponible, se asume que el dispositivo soporta hojas de estilo en cascada por defecto.

*Reducción del tamaño del código***:** la plataforma elimina espacios en blanco innecesarios; por otro lado, el proceso empleado para disminuir las hojas de estilo reduce significativamente el tamaño de las páginas.

Tipos de contenido soportados: las imágenes u objetos que no son soportados no son descargados. La información de los tipos de contenidos soportados por el dispositivo, es obtenida de las tres fuentes de información como se muestra en la Tabla 1. Por defecto, se asume que el dispositivo soporta imágenes jpg y hojas de estilos en cascada.

Fuente	Propiedad
HTTP Headers	Accept
UAProf	CcppAccept
WURFL	wbmp bmp gif jpg png tiff html_web_3_2 html_wi_imode_compact_generic wml_1_1

Tabla 1 Características usadas en el contexto de entrega para los tipos de contenido

Codificación de caracteres: la plataforma cambia la codificación utilizada a la preferida del dispositivo. Esta última se obtiene de la cabecera *accept-charset* del protocolo HTTP y es complementada con la propiedad *CcppAccept-Charset* de UAProf. Si la información no se encuentra disponible, por defecto se asume la codificación UTF-8.

Mensajes de error: cuando se presentan inconvenientes al cargar la página solicitada, la plataforma realiza una redirección a una página de error que indica brevemente las causas de la falla. Estas páginas incluyen navegación de retorno a la página anterior y al formulario inicial de la plataforma.

*Caché headers***:** la plataforma envía la cabecera *Cache-Control* fijada en *private*, indicando que se puede almacenar en caché, teniendo en cuenta que cada archivo está asignado a un dispositivo en particular. De esta manera, se disminuye la descarga de contenidos invariantes, al tiempo que se mantiene la personalización del contenido en cada dispositivo.

Orden en las tabulaciones: se define un orden a las tabulaciones para facilitar el desplazamiento por los diferentes hipervínculos de acuerdo al nivel de encabezado (*H1, H2,…*) y al orden en el que se encuentran en la página. Esta operación no incluye objetos y controles, puesto que no existen elementos estructurales HTML que permitan conocer la relevancia de estos elementos en el contenido para ajustar un orden lógico en las tabulaciones.

2.3. Casos de uso

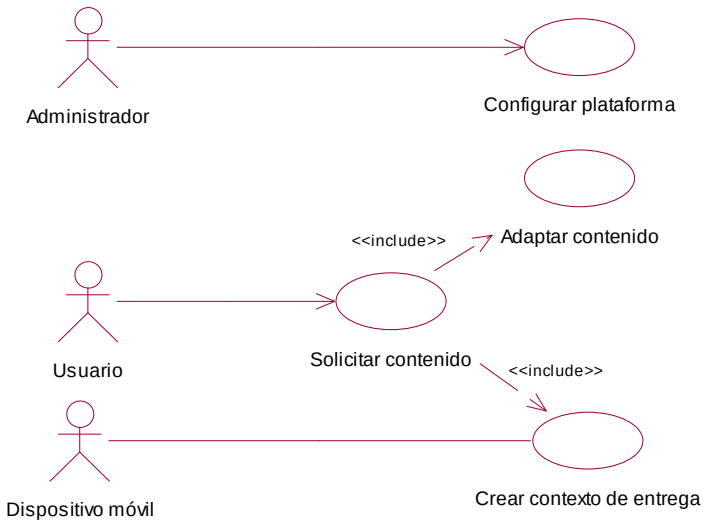


Figura 5 Diagrama de casos de uso OneWeb

Información General	
Caso de uso:	**Configurar plataforma**
Actores:	Administrador
Propósito:	Establecer parámetros específicos al comportamiento de la plataforma.
Tipo:	Primario.
Precondiciones	
- El administrador debe tener acceso al sistema de archivos de la plataforma y tener permisos para modificación de archivos	

Flujo Principal
- El administrador ingresa al archivo config.xml, ubicado en la raíz del sistema de archivos donde la plataforma ha sido instalada o desplegada. - Se modifica el archivo estableciendo los parámetros deseados de comportamiento (Ver diagrama de componentes).

Flujos de Excepción
Ninguno

Información General	
Caso de uso:	**Solicitar contenido**
Actores:	Usuario
Propósito:	Obtener una versión adaptada del contenido Web de acuerdo a las capacidades del dispositivo
Tipo:	Primario.
Precondiciones	
Ninguna	

Flujo Principal
- El usuario ingresa la URL donde se encuentra el contenido a adaptar. - La información es enviada al servidor - Se crea el contexto de entrega - La plataforma procede a adaptar el contenido. - Se entrega al dispositivo la versión adaptada del contenido.

Flujos de Excepción
E1: Error al solicitar el contenido. - Se despliega una página indicando que no existe el contenido solicitado o se presentó un error en la petición. *E2: El tipo de contenido no es soportado por el dispositivo* - Se despliega una página informando que el contenido solicitado no puede ser adaptado por la plataforma a las capacidades del dispositivo.

Información General	
Caso de uso:	**Adaptar contenido**
Actores:	Usuario
Propósito:	Obtener una versión adaptada del contenido solicitado de acuerdo al contexto de entrega.
Tipo:	Primario.
Precondiciones	
- Debe haberse creado el contexto de entrega de acuerdo a las capacidades del dispositivo.	

Flujo Principal
- Se descarga del contenido original - Se ejecuta la fase de pre-procesamiento - Se ejecuta la fase de reestructuración de la página - Se aplican las recomendaciones MWI.

Flujos de Excepción
E1: Error al solicitar el contenido. - Se despliega una página indicando que no existe el contenido solicitado o se presentó un error en la petición. *E2: El tipo de contenido no es soportado por el dispositivo* - Se despliega una página informando que el contenido solicitado no puede ser adaptado por la plataforma a las capacidades del dispositivo.

Información General	
Caso de uso:	**Crear contexto de entrega**
Actores:	Usuario
Propósito:	Crear un contexto de entrega basado en las capacidades del dispositivo del usuario.
Tipo:	Primario.
Precondiciones	
Ninguna.	

Flujo Principal
- Se obtienen las cabeceras HTTP. - Se obtiene la información del dispositivo disponible en WURFL y UAProf. - Se creación el objeto que describe las capacidades del dispositivo.

Flujos de Excepción
Ninguno

2.4. Descripción de clases

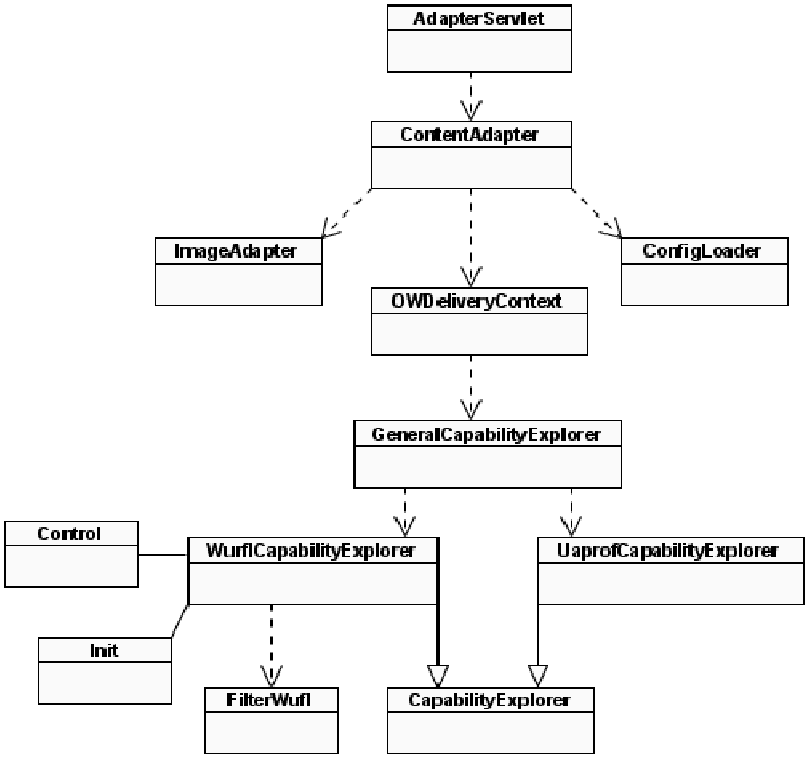


Figura 6 Diagrama de clases OneWeb

El desarrollo de la plataforma OneWeb está basado en el patrón de diseño MVC[7]. En la capa de modelo se encuentra el acceso a la biblioteca WURFL, archivos de configuración, repositorios y vocabularios de UAProf. En la capa de control se encuentran las clases encargadas de recibir la petición y coordinar el proceso de adaptación. Por último en la capa de vista se encuentran las plantillas de mensajes de error y subpáginas. A continuación se realiza una breve descripción de cada una de las clases.

- *AdapterServlet*: servlet que atiende las peticiones provenientes de los dispositivos móviles, descarga el contenido Web original y dirige la petición a la clase ContentAdapter para controlar los procesos de adaptación.

- *ContentAdapter*: clase encargada de realizar los procesos de adaptación realizados por la plataforma, con la excepción de la adaptación de las imágenes. Esta clase coordina el acceso a la configuración de la plataforma, así como la obtención del contexto de entrega.

- *ImageAdapter*: clase encargada del procesamiento de imágenes. Descarga, redimensiona y cambia la profundidad de color de acuerdo a las características definidas en el contexto de entrega.

- *ConfigLoader*: clase encargada de cargar la configuración de la plataforma desde el archivo *config.xml*.

- *OWDeliveryContext*: clase encargada de obtener el contexto de entrega, utilizando las tres fuentes de información.

[7] MVC: Model View Controller (Modelo Vista Controlador).

- *GeneralCapabilityExplorer*: clase que explora las capacidades del dispositivo, coordinando el proceso de consulta a la biblioteca WURFL o repositorios UAProf.

- *WurflCapabilityExplorer*: clase encargada de explorar las capacidades de un dispositivo obteniendo la información en la biblioteca WURFL.

- *UaprofCapabilityExplorer*: clase encargada de explorar las capacidades de un dispositivo móvil a partir de la información disponible en los repositorios UAProf.

- *Control*: clase encargada de controlar el proceso de conexión a las bibliotecas WURFL.

- *Init*: se encarga de cargar en RAM el archivo "wurfl.xml" que contiene la información de la biblioteca WURFL.

- *FilterWurfl*: se encarga de explorar las capacidades de un dispositivo en la biblioteca WURFL a partir de la información disponible en la cabecera HTTP, "User-Agent".

- *CapabilityExplorer*: interfaz que define la estructura de los exploradores de capacidades para dispositivos móviles.

2.5. Diagramas de secuencia

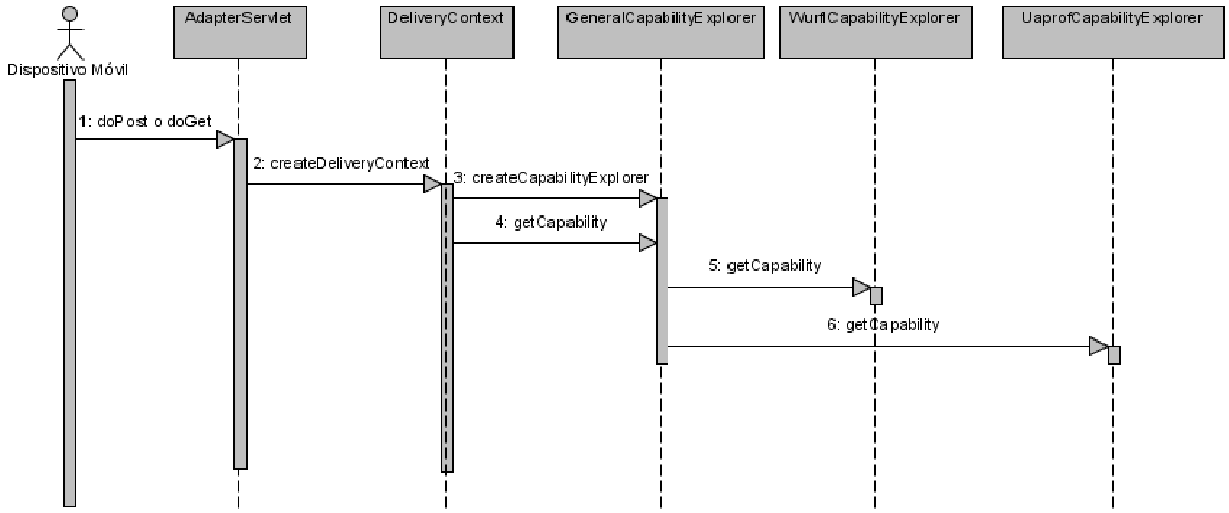


Figura 7 Diagrama de secuencia Crear contexto de entrega

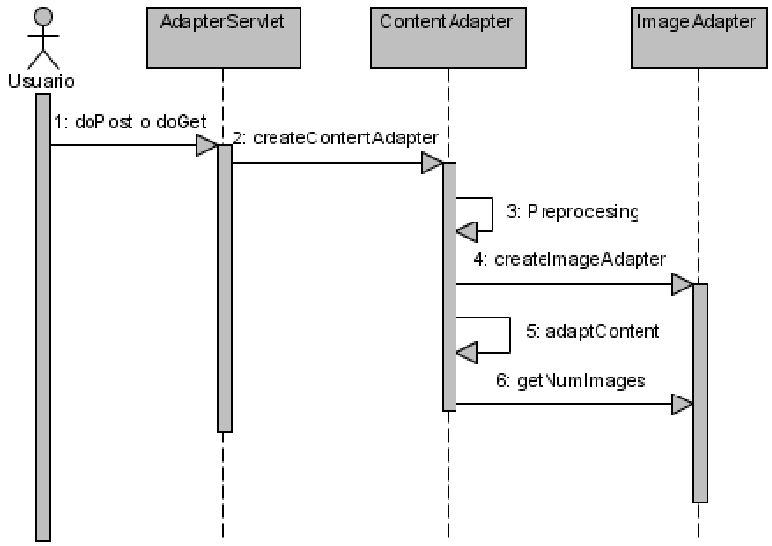


Figura 8 Diagrama de secuencia Adaptar contenido

2.6. Diagrama de componentes

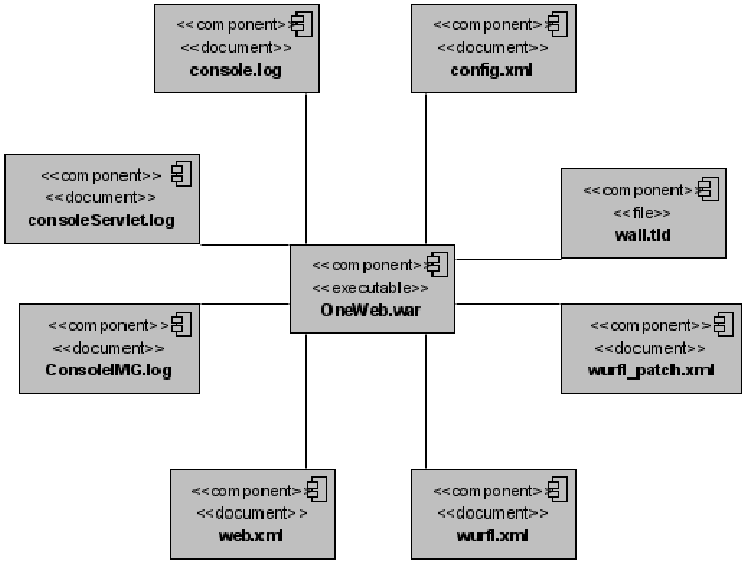


Figura 9 Diagrama de componentes OneWeb

- *console.log*: almacena los mensajes de consola y errores durante el proceso de adaptación.
- *consoleServlet.log*: almacena los mensajes de consola y errores propios del servidor.
- *consoleIMG.log*: contiene los mensajes de consola y errores de los procesos de adaptación de imágenes.
- *config.xml:* archivo de configuración de la plataforma.
- *wall.tld*: archivo que contiene una librería de tags propios de WALL.
- *web.xml*: define el contexto web de la plataforma.

- *wurfl.xml:* contiene las capacidades de los dispositivos, obtenidas a partir de la biblioteca WURFL.
- *wurfl_patch.xml*: información adicional sobre los dispositivos disponibles en WURFL.
- *OneWeb.war*: archivo de despliegue de la plataforma sobre el servidor Apache Tomcat.

A continuación se desglosan los propiedades del archivo *config.xml*, que sirven de base para la configuración de algunos parámetros de la plataforma:

- *downloadLinks:* corresponde a un valor booleano. True indica que los enlaces etiquetados con encabezados serán descargados para obtener su peso en Kb. Esta operación introduce un retardo significativo en la entrega de la respuesta de la plataforma.
- *minPageSize*: valor entero que indica el número de bytes mínimos que puede tener una subpágina.
- *thresholdLinksVsCharsInf*: valor que determina el mínimo porcentaje para la división del número de caracteres en los hipervínculos sobre número de caracteres en la sección. Si una sección de la página Web tiene una relación con un valor mayor a este umbral, se considera una barra de navegación.
- *thresholdLinksVsCharsUp*: propiedad que indica el máximo valor porcentual para la división del número de enlaces en la sección sobre número total de enlaces. Si el valor de esta relación en una sección es menor a este umbral, solo entonces la sección se considera una barra de navegación.
- *minImageDownloads*: indica numéricamente el menor número de imágenes que la plataforma debe descargar antes de mostrar el contenido; esto permite que el usuario visualice parte del contenido mientras el proceso de adaptación de imágenes continúa para las secciones siguientes, mejorando el tiempo de respuesta. Si se introduce un numero considerablemente grande (ej. 50), el retardo en entregar la respuesta adaptada será significativo; igualmente un valor muy bajo ocasiona que la respuesta entregada no muestre imágenes, puesto que éstas se encuentran en proceso de descarga. Las pruebas experimentales permitieron fijar un valor por defecto de 5.

2.7. Pruebas de la plataforma

En la sección Pruebas de la plataforma OneWeb de la vista de infraestructura en la monografía, fueron descritos algunos resultados experimentales relacionados con el rendimiento de OneWeb en cuanto al tamaño del contenido descargado y el tiempo de procesamiento del proceso total de adaptación para cada una de las veinte páginas de prueba. A continuación se muestran algunos resultados que desglosan los tiempos de procesamiento empleados por OneWeb en cada una de las fases involucradas en el proceso de adaptación.

2.7.1. Descripción de los procesos de adaptación sometidos a medición

A continuación se describen los criterios de medición de tiempo que se tuvieron en cuenta y se relacionan con los procesos de adaptación correspondientes a cada una de las fases definidas por OneWeb:

Reconocimiento de dispositivo

Recuperación en las cabeceras y cookies: es el tiempo en que la aplicación tarda en almacenar en memoria las cabeceras HTTP y de almacenar las cookies para retransmitirlas en la respuesta final.

Creación contexto de entrega: hace referencia al tiempo que tarda la plataforma en recolectar todas las capacidades del dispositivo desde donde se solicita el contenido.

Preprocesamiento

Descarga del contenido original de la página: tiempo que tarda la aplicación en obtener el contenido a adaptar.

Eliminación de tildes: tiempo empleado en la eliminación de tildes para evitar errores de codificación.

Pre-procesamiento: tiempo empleado por la plataforma para completar la depuración del código, para que este pueda ser adaptado más fácilmente.

Reestructuración de la página y aplicación de las políticas MWI del W3C

Transformada de elisión y segmentación indexada: tiempo empleado por la plataforma para realizar un resumen de las barras de navegación y dividir la página en una serie de sub páginas que cumplen con las recomendaciones de la WMI del W3C.

Adaptación del mínimo de imágenes: tiempo que tarda la plataforma en procesar el número mínimo de imágenes especificado en la configuración, por defecto cinco.

Adaptación de imágenes: tiempo que tarda la plataforma en procesar la totalidad de imágenes incluidas en el contenido.

Creación hoja de estilos externa: tiempo que tarda la plataforma en recolectar los estilos que incluye la página y almacenarlos en un archivo externo.

Eliminación de espacios: tiempo empleado en eliminar los espacios innecesarios en la página.

Asignación de teclas de acceso e información de enlace: tiempo empleado en la asignación de las etiquetas acceskey y la obtención del peso de los enlaces más relevantes de la página.

2.7.2. Resultados del tiempo de procesamiento

Los números en el eje x de las gráficas hacen referencia a cada una de las páginas probadas, las cuales se relacionan nuevamente a continuación (se han resaltado los sitios de comercio electrónico):

1. http://www.google.com/search?q=%22w3c%22
2. http://www.unicauca.edu.co
3. http://www.eltiempo.com.co
4. http://europa.eu.int/eures/home.jsp?lang=es
5. http://www.nytimes.com
6. http://www.mercadolibre.com.co
7. http://www.cnn.com
8. http://www.latimes.com
9. http://www.altavista.com/web/results?itag=ody&q=w3c&kgs=1&kls=0
10. http://www.hotbot.com/?query=w3c&ps=&loc=searchbox&tab=web&mode=search&currProv=ask
11. http://www.yahoo.com
12. http://www.nasa.gov
13. http://www.gnu.org
14. http://computers.ebay.com/
15. http://es.wikipedia.org/wiki/E-Commerce
16. http://www.comprastop.com/?nodo=301185
17. http://www.amazon.com/
18. http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/frames/ej_rw_cl.htm
19. http://www.webestilo.com/html/ejem/ej13.html

20. http://www.desarrollocristiano.com

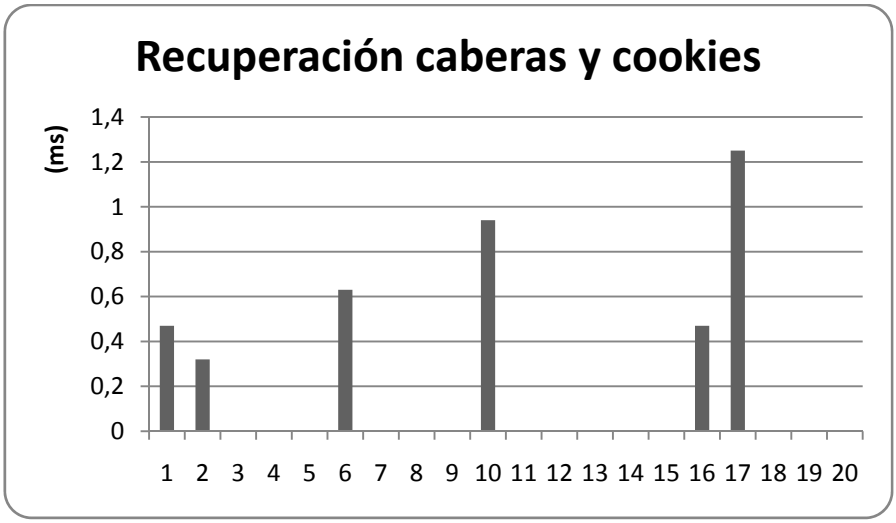


Figura 10 Recuperación de cabeceras y cookies

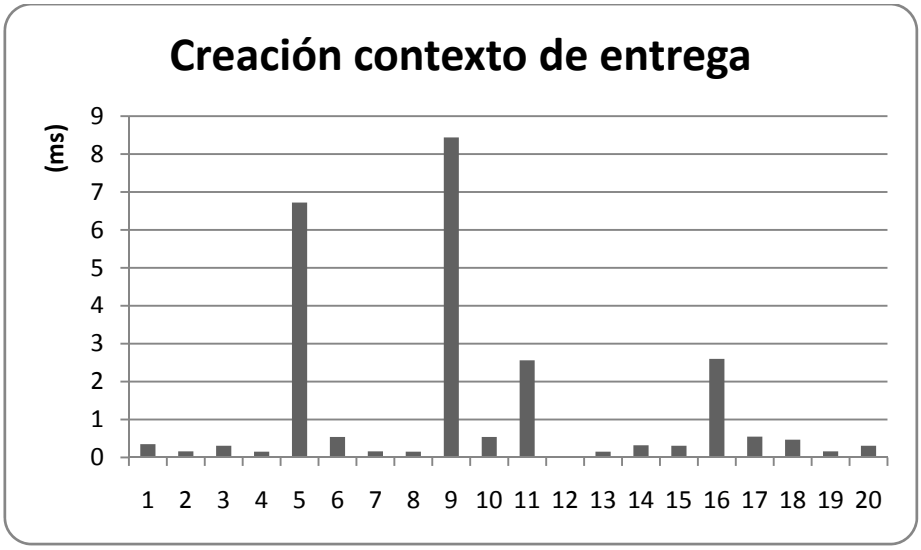


Figura 11 Creación contexto de entrega

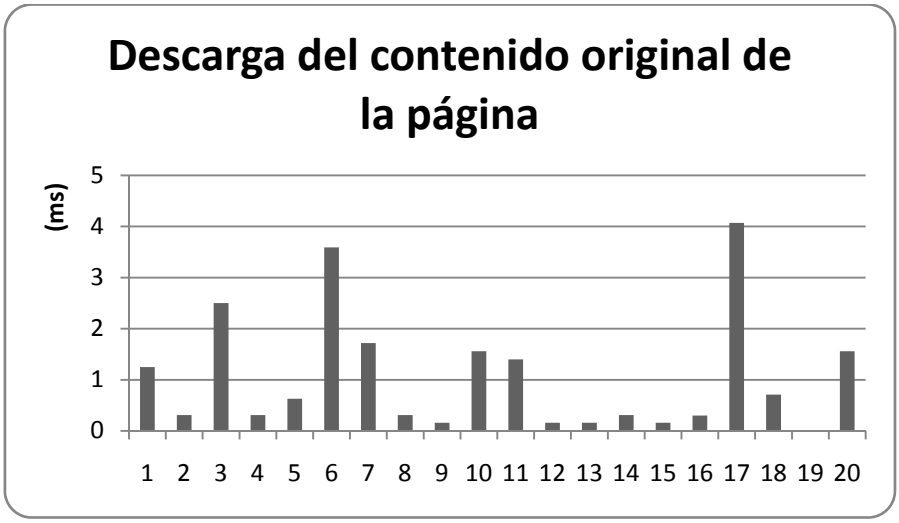


Figura 12 Descarga del contenido original de la página

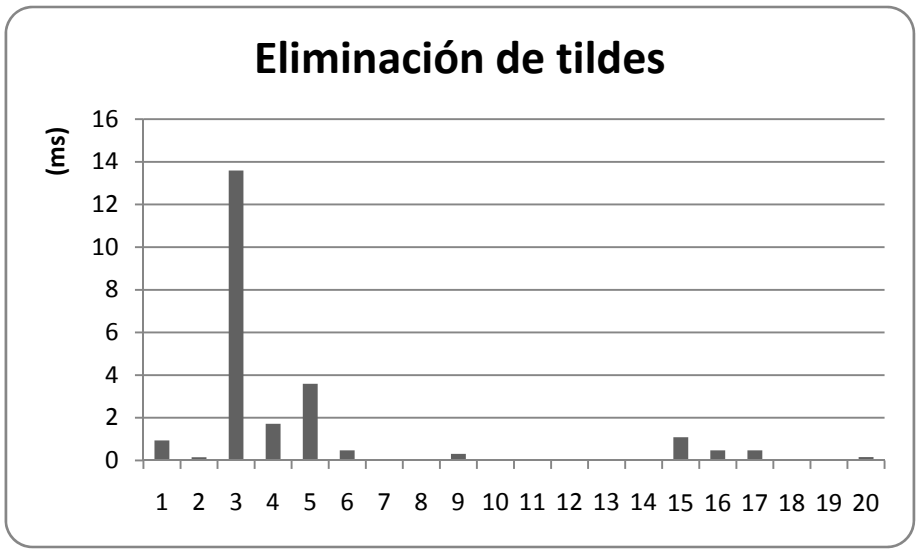


Figura 13 Eliminación de tildes

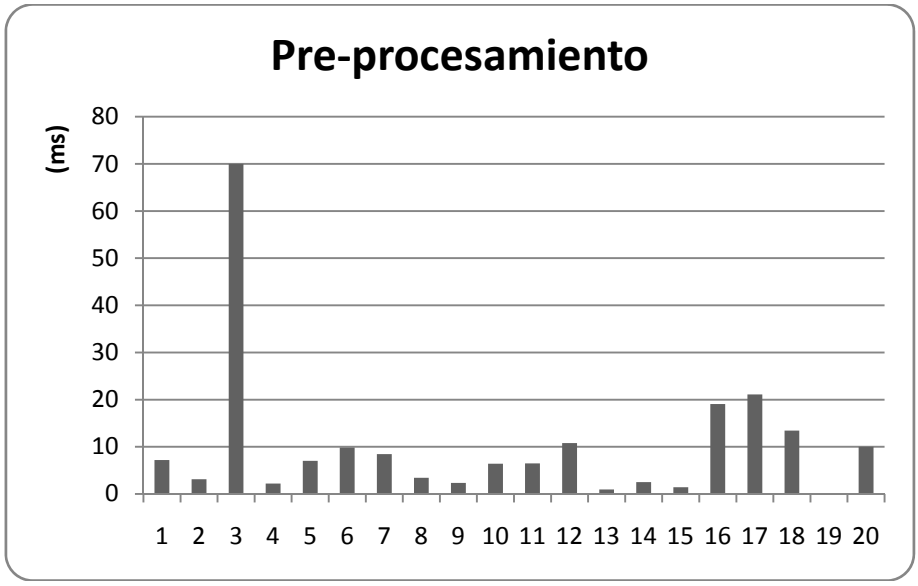


Figura 14 Pre-procesamiento

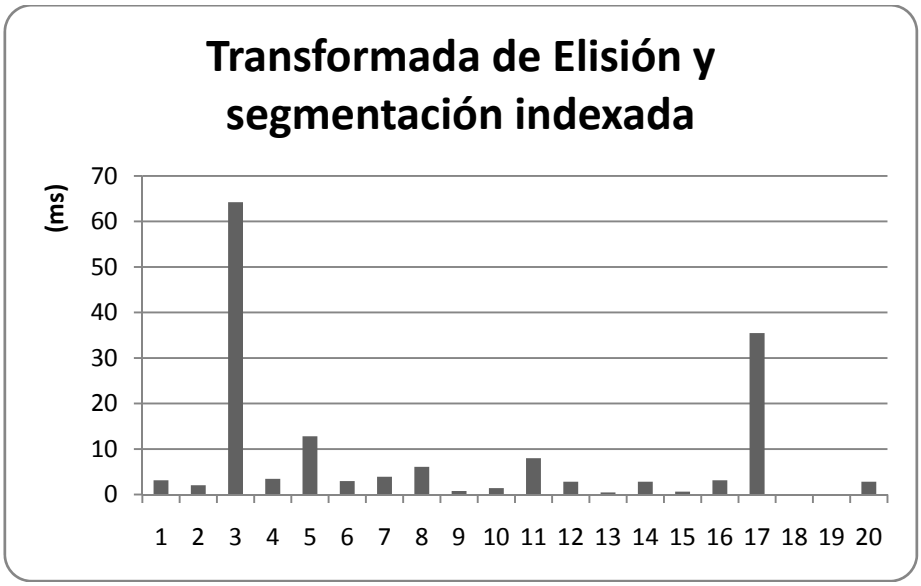


Figura 15 Transformada de Elisión y segmentación indexada

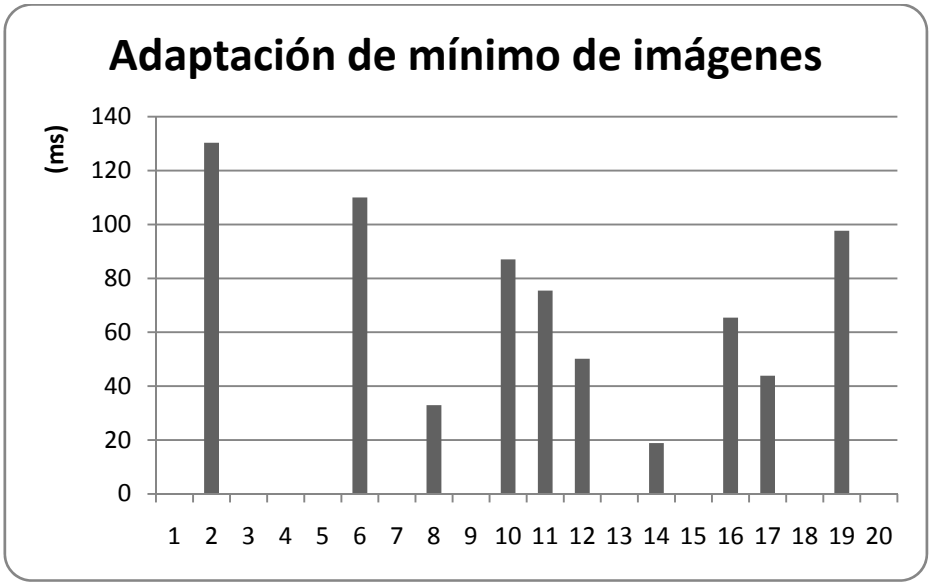


Figura 16 Adaptación del mínimo de imágenes

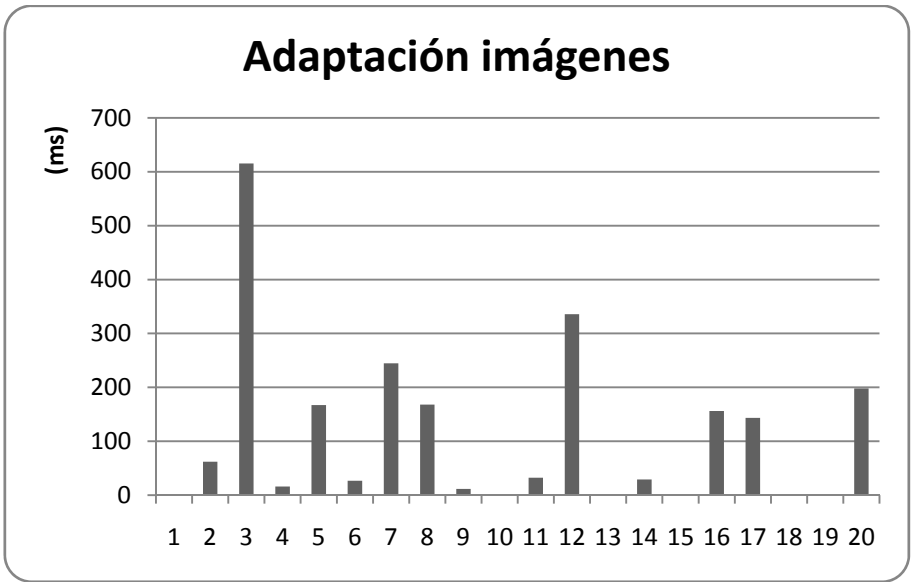


Figura 17 Adaptación de imágenes

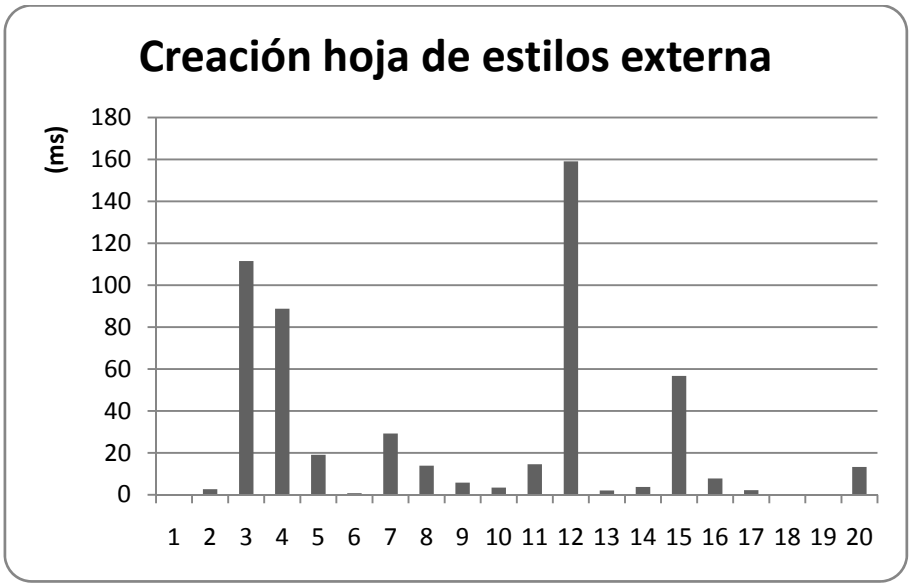


Figura 18 Creación hoja de estilos externa

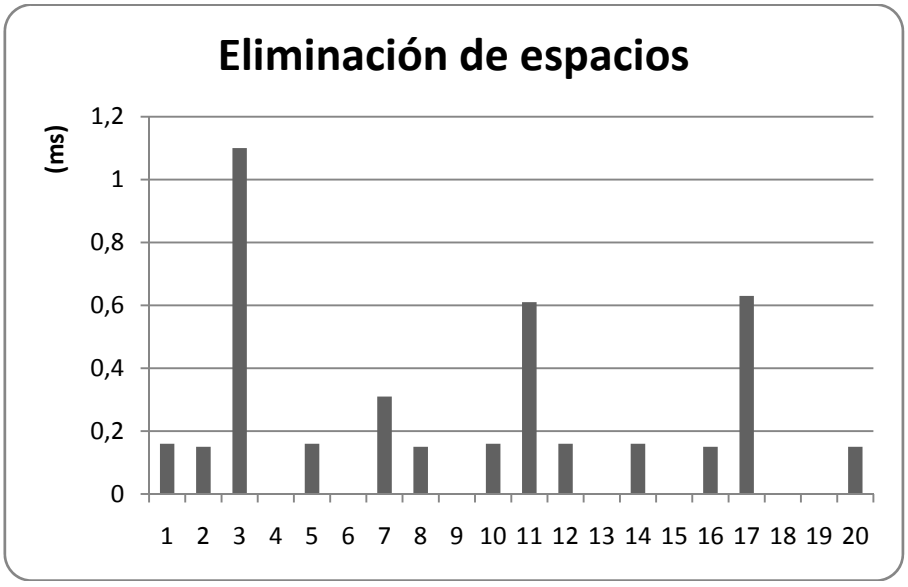


Figura 19 Eliminación de espacios

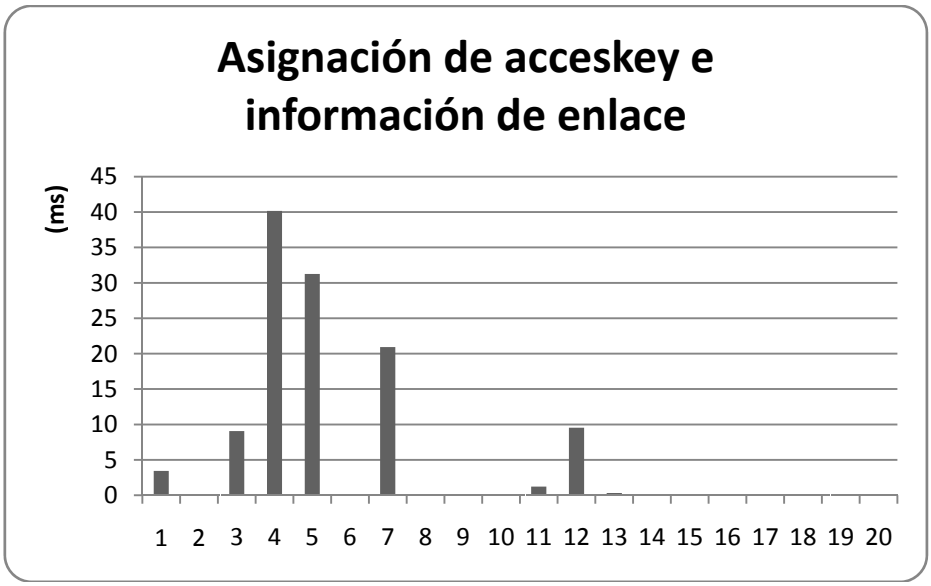


Figura 20 Asignación de teclas de acceso y recuperación de información del enlace

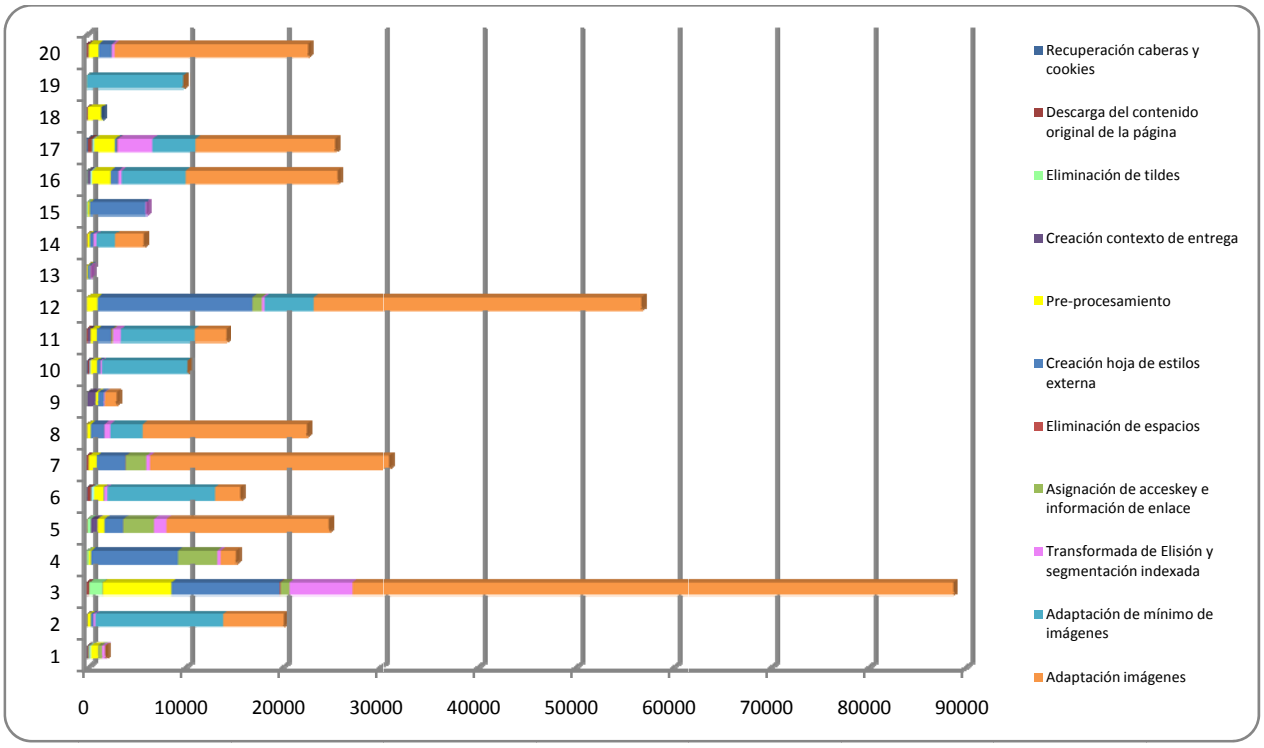


Figura 21 Gráfica resumen tiempo de procesamiento OneWeb

Anexo C

Análisis de alternativas de adaptación de contenido Web para dispositivos móviles

3.1. Patente "Sistema y proceso de adaptación de contenido Web[8]"

Esta patente muestra un procedimiento para adaptar contenidos Web en diferentes dispositivos (Scott, Chua, & NG, 2005). En primera instancia, plantea la posibilidad de fraccionar el contenido de una página Web en varias sub-páginas de menor tamaño, con el fin de ser desplegadas correctamente en dispositivos de bajas capacidades como teléfonos móviles. Incluye capacidades de reconocimiento de dispositivos, a través de un caché que almacena perfiles identificados previamente. El proceso de adaptación de contenido está basado en heurísticas que analizan los elementos estructurales de la página (tablas, enlaces, formularios, etc), realizan una representación en árbol de la página y la ajustan de acuerdo a unos patrones de despliegue preestablecidos.

3.2. WebAlchemist[9]

Es el prototipo de un sistema de adaptación Web, que convierte automáticamente una página HTML en una secuencia equivalente de sub-páginas, que pueden ser visualizadas en un dispositivo de mano (Yonghyun, Changwoo, Jihong, & Sungkwon, 2001). WebAlchemist se basa en un conjunto de heurísticas de adaptación de HTML gestionadas por un módulo administrador de adaptación. Con el fin de hacer frente a dificultades inherentes al proceso de adaptación, como reformar las complejas estructuras de algunas páginas, se desarrollan procedimientos que extraen información semántica a partir de la información sintáctica es decir, se extrae la relevancia de una sección de la página teniendo en cuenta características tales como ancho de la tabla, el tamaño de la letra y la hoja de estilo en cascada.

La Figura 22 muestra un ejemplo de adaptación de contenidos Web a través de WebAlchemist.

3.3. Mobile Adapter

Mobile Adapter usa un modelo híbrido basado en las capacidades de un cliente móvil y un proxy para realizar la adaptación de contenidos (Viana, Teixeira, Cavalcante, & Andrade, 2005). El proxy es responsable de recibir las peticiones y coordinar la adaptación de la respuesta del servidor. Se dispone de un módulo de reconocimiento de dispositivo que construye el contexto de entrega a partir de la información brindada por UAProf, descrita en CCPP. El cliente que debe ser instalado en el dispositivo móvil, está diseñado para las plataformas Java ME y SuperWaba.

La Figura 23 muestra un ejemplo de las capacidades de adaptación de Mobile Adapter, enfocadas principalmente al ajuste de imágenes en el contexto de una aplicación denominada Fotoblog.

[8] Web Content Adaptation Process And System

[9] Web Transcoding System for Mobile Web Access in Handheld Devices

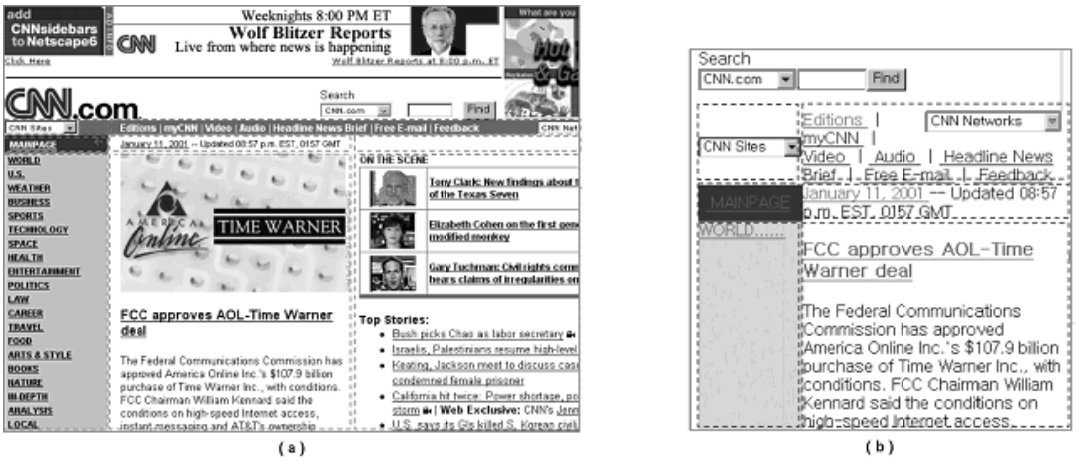


Figura 22 Ejemplo de adaptación WebAlchemist (a) Página Web CNN (b) Adaptación obtenida para un dispositivo de mano.

Figura 23 Ejemplo de adaptación Mobile Adapter

3.4. Google Mobile

Esta aplicación hace parte del selecto conjunto que hoy ha ampliado el portafolio de servicios de Google, a través de plataformas como Gmail, Google Maps o Google Earth. Google Mobile es un sistema que se accede a través de la URL http://www.google.com/m, y fue creado con el objeto de facilitar la navegación en la Web cuando se ingresa desde un dispositivo móvil (Google, 2008). A través de un conjunto de *transcoders*[10], se reconocen las capacidades del dispositivo y se realiza el proceso de adaptación, a través de la ejecución de los siguientes procedimientos: 1) transformada de elisión selectiva, 2) transformación de segmentación indexada, 3) cambio de codificación HTML a WML (cuando se requiera), 3) adaptación del formato de codificación y 4) reducción de imágenes. Estas tareas fueron descritas en la sección *3.2.4.2 Módulo de adaptación de contenido* de la monografía.

La Figura 24 muestra la página de la Universidad del Cauca adaptada por Google Mobile.

[10] Transcoder: adaptador

Figura 24 Página Universidad del Cauca adaptada por el servicio de Google Mobile

3.5. AOL Mobile Search

AOL[11] ofrece una plataforma semejante a Google Mobile que habilita la búsqueda y adaptación de sitios Web para dispositivos móviles (Netimperative, 2006). En esencia, realiza las mismas tareas de adaptación analizadas en el caso anterior, pero agrega dos procedimientos relacionados con la *inclusión de un sistema de navegación rápida* y una fase de *reordenamiento de la página*. El primero, consiste en la generación de un conjunto de enlaces que permiten un acceso más ágil a ciertas secciones de la página como barras de navegación, formularios de búsqueda, formularios de ingreso, o títulos. El segundo, ubica las secciones que se consideran de mayor relevancia al inicio de la página. No obstante, estos procesos no siempre son exactos, lo cual puede generar resultados inesperados en el proceso de adaptación.

La Figura 25 muestra la página de la Universidad del Cauca adaptada por Google Mobile.

Figura 25 Página Universidad del Cauca adaptada por AOL Mobile Search

3.6. Yahoo Mobile

Yahoo ofrece un gran número de servicios para dispositivos móviles, los cuales se pueden acceder directamente desde la Web o a través de un aplicativo Java ME (Yahoo, 2008). De manera similar al servicio de Google, Yahoo Mobile permite buscar contenido en la Web, al tiempo que facilita la adaptación del mismo para dispositivos móviles. En términos generales, soporta los procesos de reducción de imágenes, adaptación del formato de

[11] America OnLine

codificación y transformación de segmentación indexada; estos procesos son descritos en la sección *3.2.4.2 Módulo de adaptación de contenido* de la monografía.

La Figura 26 muestra la página de la Universidad del Cauca adaptada por Yahoo Mobile.

Figura 26 Página Universidad del Cauca adaptada por Yahoo Mobile

3.7. Análisis comparativo

De acuerdo a las características de cada una de las plataformas de adaptación descritas, es posible concluir que no existe un estándar en cuanto a los procedimientos que habilitan la adaptación de contenidos Web para dispositivos móviles. Sin embargo, se puede establecer que ciertos procedimientos son indispensables para obtener una mejor experiencia de navegación desde este tipo de terminales y son comunes a todas las plataformas estudiadas: 1) redimensionamiento de imágenes, 2) adaptación del formato de codificación y 3) transformación de segmentación indexada. Aunque de alguna manera todas las plataformas realizan estas operaciones, ninguna de ellas lo hace de manera idéntica y se obtienen resultados diferentes en cada una de estas etapas.

En este orden de ideas, para establecer un primer paso de análisis comparativo sobre las propuestas estudiadas, se toman en cuenta seis criterios relacionados con los procesos de adaptación, que complementan los tres procedimientos señalados anteriormente y que son comunes a todas las plataformas:

- *Adaptación Intermedia*: indica si se usa un modelo de adaptación basado en proxy (ver sección 1.3.3 del Anexo A).
- *Adaptación en el cliente*: indica si se usa un modelo de adaptación en el cliente (ver sección 1.3.3 del Anexo A).
- *Conserva hojas de estilo*: indica si la plataforma mantiene la hoja de estilos del contenido original.
- *Menú de navegación rápida*: indica si se crea un menú para facilitar la navegación por las diferentes secciones de la página adaptada.
- *Transformada de elisión*: indica si se transforman secciones de texto o imágenes en hipervínculos descriptivos que representan la sección.
- *Reordenamiento de las secciones*: indica si la plataforma reordena según un criterio de relevancia propio, las diferentes secciones del contenido Web.

La Tabla 2 resume los resultados de este primer paso de análisis, el cual favorece a la plataforma de AOL.

Como segundo paso de análisis comparativo, se toman en cuenta algunas características particulares de cada plataforma. La propuesta “Web Alquemist” obtiene resultados muy satisfactorios; sin embargo fue desarrollada para ser utilizada principalmente en computadoras de mano y no en equipos de menores capacidades como

teléfonos celulares. Por otro lado, aunque la propuesta "Mobile Adapter" ofrece buenos resultados, requiere la instalación de un aplicativo en el cliente, lo cual reduce la gama de dispositivos a la cual está dirigida.

La plataforma "Yahoo Mobile" tiene varias limitantes dentro de las que se encuentran la falta de soporte para el procesamiento de marcos y mapas de imágenes, al definir procesos de adaptación más simples con respecto a las otras alternativas. La plataforma "AOL Mobile Search", añade procedimientos como la inclusión de menús de navegación rápida y reordenamiento de las secciones, pero no ejecuta transformada de elisión. Por otro lado, según los resultados de la exploración, la plataforma desarrollada por Google exhibe un grado de desarrollo mayor y ofrece una experiencia de navegación superior; sin embargo, entre sus limitaciones se encuentra la eliminación de las hojas de estilo propias de la página.

Propuesta	Adaptación Intermedia	Adaptación en el cliente	Conserva hojas de estilo	Menú de navegación rápida	Transformada de elisión	Reordenamiento de las secciones
Web Alquemist	☑				☑	
Patente "Web Content Adaptation Process And System"	☑		☑			
Mobile Adapter	☑	☑	☑			
Google Mobile	☑				☑	
AOL Mobile Search	☑		☑	☑		☑
Yahoo Mobile	☑		☑			

Tabla 2 Comparación plataformas de adaptación de contenidos Web

Finalmente, la Tabla 3 muestra las recomendaciones de la W3C MWI que son cumplidas por cada plataforma de adaptación. Las directrices que no aparecen en la tabla, se cumplen en las diferentes propuestas o corresponden a recomendaciones a nivel de diseño de las páginas que no se tienen en cuenta como punto de comparación en los procesos de adaptación. En términos generales, es evidente que las plataformas estudiadas al no estar basadas en la MWI, no satisfacen muchas de las recomendaciones planteadas por la iniciativa.

Sumando las conclusiones de los puntos analizados, las plataformas de Google y AOL muestran una mayor madurez en el proceso de adaptación de contenidos Web para dispositivos móviles.

Recomendación	Web Alquemist	Patente "Web Content Adaptation Process And System"	Mobile Adapter	Google Mobile	AOL Mobile Search	Yahoo Mobile
Explotar las capacidades del dispositivo		☑	☑	☑	☑	☑
Barras de Navegación				☑	☑	
Teclas de acceso		☑	☑	☑	☑	☑
Identificación del enlace de destino						
Mapas de imágenes		☑				
Recarga, redirección y ventanas emergentes.						
Gráficos			☑			
Color			☑	☑		
Marcos			☑	☑		
Hojas de estilo	☑	☑	☑		☑	☑

Tabla 3 Recomendaciones MWI cumplidas por las plataformas analizadas

Anexo D

Sistemas Criptográficos

La seguridad en el escenario móvil ha tenido una evolución continua y especialmente importante en los últimos años. Desde que ingresó al mercado la telefonía celular digital GSM[12] a inicios de los años 90, se ha puesto especial atención en los mecanismos para garantizar la seguridad tanto en la comunicaciones de voz como en la transferencia de datos, esta última, empleada por las aplicaciones; varios esfuerzos en este sentido, han tratado de migrar las técnicas de seguridad de las redes fijas hacia las redes móviles. En la actualidad, los mecanismos de seguridad basados en cifrado son los más efectivos y comienzan a ser utilizados ampliamente en las redes móviles gracias a tecnologías como Java Card y SATSA[13].

4.1 Cifrado simétrico

El cifrado simétrico (Eudmednet, 2006) es aquel que emplea la misma clave tanto para cifrar como para descifrar los datos. Su gran inconveniente, es que para ser empleado en comunicaciones, la clave debe estar disponible tanto en el emisor como en el receptor, lo cual adiciona un aspecto a considerar y es cómo transmitir la clave de forma segura. Los algoritmos de cifrado simétrico más utilizados son DES[14], Triple DES y AES[15]. El cifrado simétrico tiene la ventaja de ser más rápido y eficiente que el cifrado asimétrico, pero este último es más seguro.

Usualmente la calidad del cifrado se mide por la cantidad de esfuerzo que se necesita para averiguar las claves. En la Tabla 4 se listan los algoritmos de cifrado simétrico más utilizados con la longitud correspondiente de sus claves.

Cifrado	Longitud de Clave
RC2[16]	128 bits
RC4	128 bits
DES	64 bits
Triple DES (2 Keys)	128 bits
Triple DES (3 Keys)	192 bits
AES	128, 192 ó 256 bits

Tabla 4 Tipos de cifrado simétrico

4.1.1 Cifrado DES

El cifrado DES fue hasta hace unos años el estándar más utilizado para cifrado simétrico. Se utiliza una clave de 64 bits, lo cual significa que utilizando el tipo de ataque más simple, fuerza bruta, se tendría un total de 2^{64} posibilidades en el proceso de ensayo y error (Eudmednet, 2006). Con un computador capaz de hacer mil millones de operaciones por segundo, tomaría más de 584 años probar la totalidad de las claves. Sin embargo, con el paso

[12] GSM: Global System for Mobile Communications
[13] SATSA: Security And Trust Services API
[14] DES: Data Encryption Standard
[15] AES: Advanced Encryption Standard
[16] RC: Rivest Cipher

del tiempo se han diseñado ataques más sofisticados alternos al de fuerza bruta, capaces de descifrar la clave en mucho menos tiempo.

4.1.2 Cifrado 3DES

3DES o Triple DES aparece como una evolución de DES. Este consiste en cifrar la información con una clave DES, luego descifrarla con otra clave DES y finalmente esta información se vuelve a cifrar con otra clave. Por eso una clave Triple DES es generalmente la unión de tres claves DES. Utilizando el ataque de fuerza bruta y una clave de 192 bits, se tendrían 2^{192} posibilidades. Le tomaría 199^{39} años a un computador capaz de hacer mil millones de operaciones por segundo, probar la totalidad de las claves (Eudmednet, 2006).

4.2 Cifrado asimétrico

Este se basa en la infraestructura de clave pública PKI[17] (BlueHackTeam, 2005), cuyos orígenes se remonta a finales de los años 70; en este esquema se propone la idea de disponer de una pareja de claves para las operaciones criptográficas, una pública, conocida por todos, y otra privada, sólo conocida por el usuario a quien le es asignada. Un mensaje puede ser cifrado por cualquier persona usando la clave pública, ya que es conocida abiertamente, aunque sólo el poseedor de la clave privada podrá descifrarlo. Recíprocamente, un mensaje cifrado con la clave privada sólo puede ser cifrado por su poseedor, mientras que puede ser descifrado por cualquiera que conozca la clave pública.

Estas propiedades de la criptografía asimétrica o también llamada de clave pública, brinda mayor fortaleza a los esquemas de seguridad para prestar servicios como la *autenticación de usuarios* (para asegurar la identidad del usuario, ya que sólo éste puede conocer su clave privada, minimizando así el riesgo de suplantación), el *no repudio* (para impedir que una vez emitido un mensaje el emisor se retracte o niegue haberlo enviado), la *integridad de la información* (para prevenir la modificación deliberada o accidental de los datos, durante su transporte, almacenamiento o manipulación), y el acuerdo de claves secretas para garantizar la *confidencialidad de la información intercambiada.*

4.2.1 Clave pública y clave privada

Como fue señalado anteriormente, la novedad fundamental de los algoritmos asimétricos con respecto a la criptografía simétrica es que las claves no son únicas, sino que forman pares; buscan en general plantear al atacante problemas matemáticos difíciles de resolver. El más popular de estos algoritmos por su sencillez es RSA[18] (Talavera, 2005). Los algoritmos asimétricos poseen dos claves diferentes, una pública y otra privada; como fue señalado anteriormente, una de ellas se emplea para codificar, mientras que la otra se usa para decodificar. Los datos cifrados con una clave no pueden ser descifrados con la misma, sino con su pareja. La Tabla 5 lista algunos algoritmos de cifrado asimétrico con la longitud respectiva de sus claves.

Cifrado	Longitud de Claves
RSA	512, 736, 768, 896, 1024, 1280, 1536, 1984 ó 2048 bits
DSA[19]	512, 768 ó 1024 bits
ECDSA[20]	128 ó 192 bits

Tabla 5 Tipos de cifrado asimétrico

Los algoritmos asimétricos emplean generalmente longitudes de clave superiores con respecto a los simétricos. Por ejemplo, mientras que para los algoritmos simétricos se considera segura una clave de 128 bits, para los

[17] PKI: Public Key Infraestructure
[18] RSA: Rivest, Shamir y Adleman
[19] DSA: Digital Signature Algorithm
[20] ECDSA: Elliptic Curve Digital Signature Algorithm

asimétricos se recomiendan claves de al menos 1024 bits. Sin embargo, la complejidad de cálculo que requieren estos últimos los hace considerablemente más lentos. En el escenario móvil, desde hace varios años las tarjetas inteligentes han desempeñado un papel importante debido a que su hardware es capaz de generar claves asimétricas, alojando en su interior la clave privada. Esta característica ofrece la posibilidad de trasladar las claves de forma segura al interactuar con otros sistemas PKI.

4.2.2 Cifrado RSA

El cifrado RSA llamado así en honor a sus inventores Rivest-Shamir-Adleman, es actualmente uno de los algoritmos más utilizado a nivel mundial debido a su relativa sencillez de implementación y gran seguridad (Talavera, 2005). Sin embargo, es importante destacar que RSA se utiliza generalmente para cifrar datos pequeños, ya que computacionalmente es complejo, lo cual redunda en un tiempo superior para realizar los procesos. Por esta razón, frecuentemente se acostumbra a utilizar una combinación de cifrado simétrico con cifrado RSA, a la cual se le conoce como *Cifrado híbrido*. A continuación, se muestra un ejemplo al respecto.

Para cifrar un mensaje, primero se cifra con una clave simétrica del tipo 3DES (192 bits), que es un proceso muy rápido, y después se cifra dicha clave (simétrica) con una clave asimétrica pública (larga) del tipo RSA (1024 bits). Con este procedimiento, se consigue que sólo el poseedor de la clave privada (asimétrica) pueda descifrar la clave simétrica (rápida) que le permitirá descifrar el mensaje. En cada mensaje, la clave simétrica utilizada es diferente por lo que si un atacante es capaz de descubrir la clave simétrica, solo es válida para ese mensaje y no para los restantes. La Figura 27 muestra paso a paso los procesos descritos anteriormente:

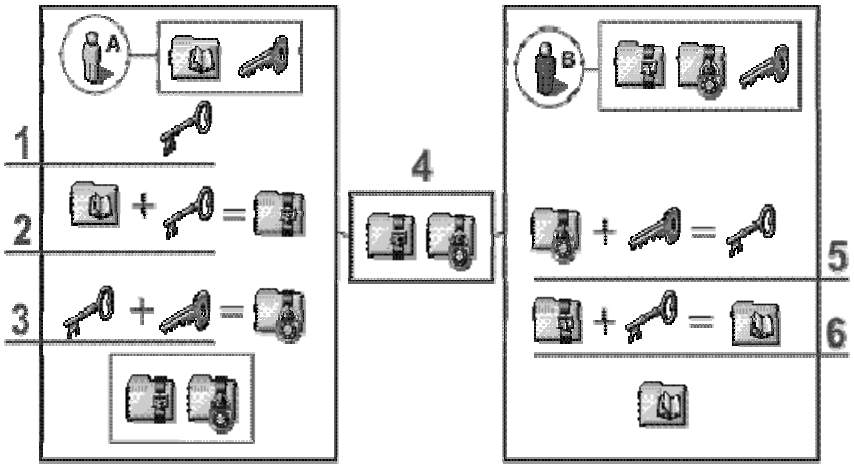


Figura 27 Pasos de un cifrado híbrido

A. El usuario A tiene la clave pública del destinatario B y el mensaje sin cifrar.

1) Se genera una clave simétrica de 192 bits (3DES por ejemplo).
2) Se cifra el mensaje con la clave simétrica.
3) Se cifra la clave simétrica con la clave pública del destinatario (asimétrica RSA de 1024 bits, por ejemplo).
4) Se envía el mensaje cifrado y la clave cifrada.

B. El destinatario tiene su clave privada; recibe el mensaje y la clave cifrada.

5) Se descifra la clave simétrica de 192 bits (3DES) con su clave privada de 1024 bits.
6) Finalmente, se descifra el mensaje con la clave simétrica que lo cifró originalmente.

4.3 Firma digital

La finalidad de la firma digital (Talavera, 2005) es garantizar la autoría de la misma y la integridad de los datos. Un mensaje firmado no está cifrado y es completamente legible; haciendo una analogía, básicamente es como una postal firmada.

4.3.1 Generación de la firma

El primer paso consiste en generar un hash del mensaje a firmar (message digest). El hash básicamente es un resumen del mensaje original, que se obtiene a partir de una función que lleva el mismo nombre (IETF, 1999). Una función hash presenta las siguientes características:

- Es irreversible, es decir que a partir del resultado de la función hash no es posible obtener el documento original.
- Un ligero cambio en el documento original (una letra por ejemplo), genera un resultado completamente distinto.

Las funciones hash más utilizadas son MD2, MD4, MD5 y SHA, las cuales entregan resultados entre 128 y 160 bits. La Figura 28 muestra un ejemplo de generación de un hash.

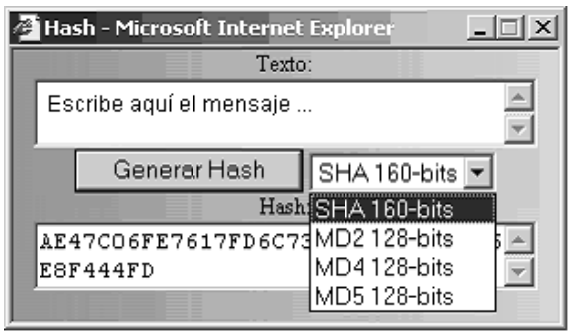


Figura 28 Generación de un hash

El siguiente paso, consiste en cifrar el hash generado con una clave asimétrica, la clave privada. Finalmente se envía al destinatario el mensaje, la firma y un certificado digital del emisor; este último concepto será explicado más adelante.

4.3.2 Verificación de la firma

Para comprobar una firma, ésta se descifra con la clave pública disponible en el certificado digital y se obtiene un hash. Luego se le calcula el hash del mensaje a comprobar y finalmente se compara con el hash anterior. Si estos coinciden, se garantiza que el mensaje fue firmado por el titular de la clave privada (ya que se ha podido descifrar con la pública) y que el documento no ha sido modificado (ya que los hash coinciden). Si cualquiera de las condiciones anteriores no se cumple, se produce una "Rotura de firma" (IETF, 1999). La Figura 29 describe paso a paso el proceso generación y verificación de la firma digital.

A. El usuario "A" dispone del mensaje, su certificado (con su clave pública), y su clave privada.

1) Se genera el hash del mensaje (con la función SHA por ejemplo).
2) Se cifra el hash con la clave privada y se obtiene la firma.
3) Se envía el mensaje, la firma, y una copia del certificado.

B. El destinatario recibe el mensaje, pero se es necesario verificar que éste no ha sido modificado en el camino.

4) Se comprueba el certificado y se descifra la firma con la clave pública disponible en el mismo; este procedimiento verifica que la firma haya sido cifrada con la clave privada respectiva, garantizando la autoría, y se obtiene un hash.
5) Se genera el hash del mensaje recibido.
6) Se comparan los dos hash obtenidos. Si los hash coinciden, la integridad de la información está garantizada.

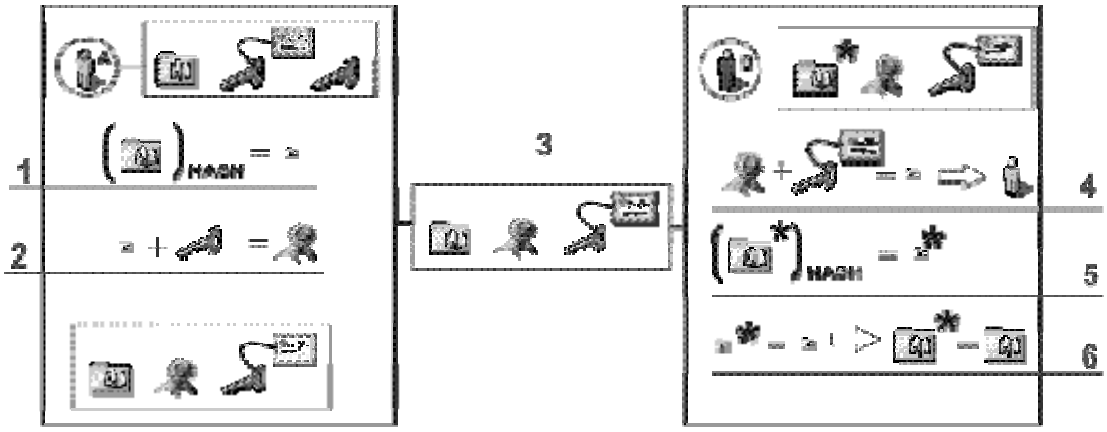


Figura 29 Generación y verificación de la firma digital

4.4 Certificado digital

La principal finalidad de los certificados digitales es asegurar que efectivamente la clave pública que se recibe en un momento dado es de la persona correcta y no de un suplantador (Eurologic, 2005) (Talens-oliag, 2003). Para asegurar la identidad del emisor es necesaria la intervención de una Autoridad Certificadora *CA*, la cual, entre otras funciones, se encarga de la emisión de estos certificados. El certificado digital es un archivo electrónico de aproximadamente 2 Kb, que contiene un conjunto de datos, que vinculan una clave pública con la identidad de una persona física o jurídica (empresa, servidor Web, etc.), de manera que se puede verificar que efectivamente pertenece a quién dice poseerla (Talens-oliag, 2003).

El formato de certificado X.509 (IETF, 1999) es el más común y extendido en la actualidad. Estos certificados se estructuran de forma jerárquica, de tal forma que se puede verificar la autenticidad de un certificado comprobando la firma de la autoridad que lo emitió, que a su vez tendrá otro certificado expedido por otra autoridad de rango superior. De esta forma se asciende en la jerarquía hasta llegar al nivel más alto, que deberá estar ocupado por un certificado que goce de la confianza de toda la comunidad.

El mecanismo convencional para la generación de un certificado digital, consiste en la creación de una pareja de claves asimétricas, pública y privada, y un identificador con los datos del solicitante; posteriormente, la clave pública se envía junto con un identificador a la autoridad certificadora, con fines de autenticación; una vez este proceso se ha llevado a cabo de manera exitosa, se envía el certificado digital, que no es otra cosa que la clave pública y el identificador, firmados con la clave privada de la CA. Las claves asimétricas para los certificados se generan en el equipo del usuario, ya que toda la infraestructura PKI se basa en que sólo el propietario de la pareja de claves posee la clave privada. El encargado de generar dichas claves es el Proveedor de servicios criptográficos (CPS[21]) de cada usuario. Cuando se accede a las páginas de emisión de certificados (online) de una autoridad certificadora, se llama al registro de su sistema en busca de los CPS disponibles, para que el usuario seleccione uno de ellos. En este instante, es posible seleccionar si el certificado se almacenará en un disco duro o en un soporte criptográfico externo como llaves USB, tarjetas inteligentes, etc. En el momento de la instalación de estos dispositivos, se añaden al sistema los CPS que los gestionan.

[21] CPS: Certification Practice Statement

La Figura 30 muestra un ejemplo de certificado digital. Como se puede observar, este certificado es válido durante un año y es emitido por la autoridad certificadora SECURITY ZUTANEZ para el usuario Jaimito cuya clave pública RSA se observa en la Figura 31. Todos los datos están firmados por la AC usando la función hash MD5 y su clave privada RSA.

issuer: C=ES ST=L=Barcelona O=SECURITY ZUTANEZ OU=Division de certificados CN=Fulano Menganez Email=Fulano@fulanez.es subject: C=ES ST=O=OU=CN=Jaimito Email=Jaimito@jaimito serial:15
Certificate: Data: Version: 1 (0x0) Serial Number: 21 (0x15) Signature Algorithm: md5WithRSAEncryption Issuer: C=ES ST=L=Barcelona O=SECURITY ZUTANEZ OU=Division de certificados CN=Fulano Menganez Email=Fulano@fulanez.es Validity Not Before: Nov 18 15:15:31 1998 GMT Not After : Nov 13 15:15:31 1999 GMT Subject: C=ES, ST=, O=, OU=, CN=Jaimito Email=Jaimito@jaimito Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (1024 bit) Modulus (1024 bit): 00:9e:74:de:c9:1a:6b:f4:fe:d1:04:30:58:7e:8b: 51:7a:98:23:e9:45:a9:c2:a7:7c:f8:f8:b5:9a:a2: ea:c1:99:68:ba:f7:c3:d8:06:05:1b:6a:47:a1:44: 5c:2c:a6:e0:4b:6f:ce:02:c4:06:32:20:34:be:13: 97:39:a3:aa:6f:2f:41:a7:bc:14:c8:f3:0c:ad:9d: 09:63:8a:f5:eb:60:5b:06:a6:01:fb:1a:07:b2:c6: 39:48:bb:b7:00:56:4e:20:6d:87:3f:67:0b:2f:f4: b0:5f:74:7f:90:6b:b4:47:6f:56:1a:b5:c5:42:54: 9b:e5:e3:00:e2:4f:e3:14:47 Exponent: 65537 (0x10001) Signature Algorithm: md5WithRSAEncryption 3b:2b:e9:ff:48:48:35:ab:30:5c:e2:d1:88:c9:29:8b:bc:09: b2:58:80:17:9c:e7:08:0a:7d:8a:5e:46:a8:83:3b:ee:84:de: 62:e3:ea:51:cb:92:bc:fa:db:90:bd:cd:9f:25:d4:4a:48:63: ac:b8:93:f9:dc:9c:cf:ef:fd:45
- -----BEGIN CERTIFICATE----- MIICOzCCAeUCARUwDQYJKoZIhvcNAQEEBQAwgaYxCzAJBgNVBAYTAkVTMRIwEAYD
VQQIEwlDYXRhbHVueWExDDAKBgNVBAcTA0JjbjEVMBMGA1UEChMMU0VDVVJJVFkg
QkNOMRowGAYDVQQLExFzZWNjaW8gZCdlbXByZXNlczEdMBsGA1UEAxMURGF2aWQg
R3VlcnJlcm8gVmlkYWwxIzAhBgkqhkiG9w0BCQEWFGd1ZXJyZXJvQGdyZWMudXBj
LmVzMB4XDTk4MTExODE1MTUzMVoXDTk5MTExMzE1MTUzMVowZjELMAkGA1UEBhMC
RVMxCTAHBgNVBAgTADEJMAcGA1UEChMAMQkwBwYDVQQLEwAxGDAWBgNVBAMUD0Nh
bHZpbiAmIEhvYmJlczEcMBoGCSqGSIb3DQEJARYNY2FsdmluQGhvYmJlczCBnzAN
BgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAnnTeyRpr9P7RBDBYfotRepgj6UWpwqd8
+Pi1mqLqwZlouvfD2AYFG2pHoURcLKbgS2/OAsQGMiA0vhOXOaOqby9Bp7wUyPMM
rZ0JY4r162BbBqYB+xoHssY5SLu3AFZOIG2HP2cLL/SwX3R/kGu0R29WGrXFQlSb
5eMA4k/jFEcCAwEAATANBgkqhkiG9w0BAQQFAANBADsr6f9ISDWrMFzi0YjJKYu8
CbJYgBec5wgKfYpeRqiDO+6E3mLj6lHLkrz625C9zZ8l1EpIY6y4k/ncnM/v/UU= - -----END CERTIFICATE-----

Figura 30 Ejemplo de un certificado digital

Exponente: 65537
modulo: 111271955529718274977020001053287254971153574325639486465242654911453961403088231010544304032158540188499185504478881755061645893205889184340440484177173313682979482908132499473623983651771075446109365198267065678811090107152632592388889101510157610404623906744451048525264576885364836810773621503974118471

Figura 31 Ejemplo de una clave pública

Para verificar que el certificado es correcto se debería obtener el certificado digital emitido para dicha AC por una segunda AC de nivel superior; el proceso se repite ascendiendo en la jerarquía hasta llegar a las llamadas autoridades raíz que firman sus propios certificados, como VeriSign.

4.5 Infraestructura de clave pública (PKI)

Como se explicó anteriormente, en la criptografía asimétrica se manejan dos claves, una pública y otra privada, por lo cual se requiere una infraestructura adecuada que administre todas las claves públicas de la red, de tal forma que no solo estén disponibles para los demás usuarios sino que se garantice la autenticidad de cada una de ellas. Para conseguir este objetivo nace PKI, una infraestructura en la cual se distribuyen las claves públicas mediante

certificados, que son firmados y avalados por una CA (Almenarez, 2005). Además de esta Autoridad, PKI está compuesta por: un depósito de certificados, un sistema de revocación de certificados, un sistema de generación de copias de seguridad y recuperación de llaves, soporte para no-repudio, actualización automática de llaves, administración del historial de llaves y capacidad de seguimiento del tiempo de creación y modificación de un documento.

En este sentido, es claro que la función principal de la CA es avalar si una llave pública pertenece al usuario que la posee. Para este propósito, la CA emplea certificados digitales como fue explicado anteriormente, como el medio formal para garantizar la autenticidad de dicha clave. La administración de los certificados digitales en la red, incluye varias actividades como generar, firmar, distribuir y revocar certificados, generar una lista CRL[22] con los certificados revocados, autenticar usuarios, y procesar solicitudes de certificados (Caicedo, 2007).

[22] CRL: Certificate Revocation List

Anexo E

Especificación de la plataforma P3SIM

5.1 Introducción

La Plataforma de Seguridad para Servicios móviles basada en SIM - P3SIM, brinda a los desarrolladores de aplicaciones móviles sobre redes de telefonía móvil de 2.5 y 3G, las facilidades necesarias para implementar un acceso seguro a servicios a través de las capacidades del módulo SIM del dispositivo. En términos generales, la plataforma permite:

- Un acceso seguro a servicios móviles basado en parámetros SIM.
- Cifrar o descifrar información tanto con algoritmos simétricos como asimétricos.
- Gestionar (fijar, recalcular u obtener) de forma segura en la SIM claves simétricas y asimétricas.
- Manejo de funciones Hash, con el objetivo de generar y verificar firmas digitales.

5.2 Casos de uso

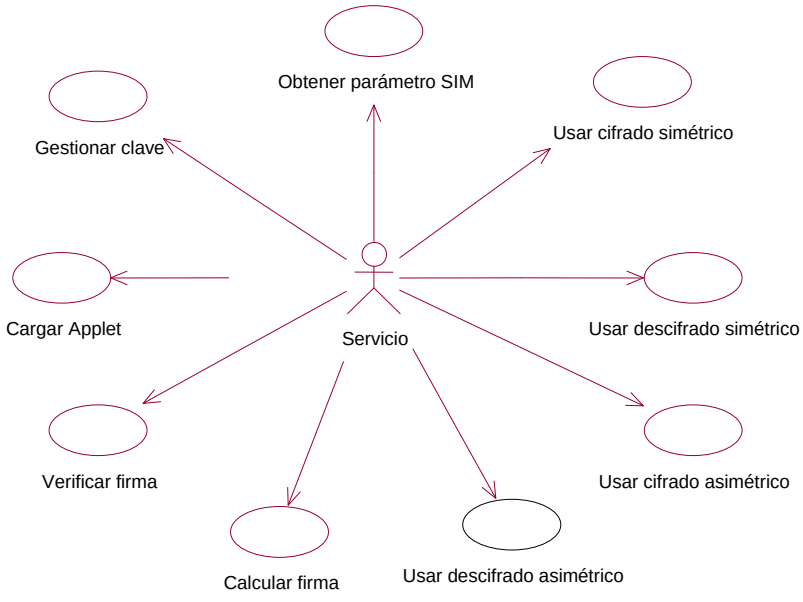


Figura 32 Diagrama de casos de uso P3SIM

Información General	
Caso de uso:	**Cargar Applet[23] SIM**
Actores:	Servicio
Propósito:	Permitir que el servicio instale un Applet en la tarjeta SIM.
Resumen:	El servicio envía a la plataforma un archivo que representa un Applet el cual será cargado en la SIM. Una vez se carga el applet, se procede a inicializarlo.
Tipo:	Primario.
Precondiciones	
- El archivo contiene APDU[24] que deben cumplir con el estándar GSM 03.48.	

Flujo Principal
- El servicio le indica a la plataforma cual es el archivo que contiene los APDU para que posteriormente la plataforma las envíe a la SIM. - La plataforma trata de seleccionar el Applet. - Si la selección es exitosa, la plataforma envía al servicio el mensaje de instalación exitosa.

Flujos de Excepción
E1: El Applet no pudo ser seleccionado. - La plataforma encontró inconvenientes al cargar al Applet en la SIM. - La plataforma envía al servicio el mensaje de error respectivo.

Información General	
Caso de uso:	**Obtener parámetro SIM**
Actores:	Servicio
Propósito:	Permitir que el servicio obtenga un parámetro SIM.
Resumen:	La plataforma envía al Applet en la SIM, un comando APDU para obtener el parámetro que especificó el servicio; la SIM retorna el parámetro a la plataforma.
Tipo:	Primario.
Precondiciones	
- El applet debe estar instalado en la tarjeta SIM.	

Flujo Principal
- El servicio especifica a la plataforma que parámetro SIM quiere obtener. - La plataforma solicita el parámetro específico al Applet en la SIM - La SIM retorna el parámetro a la plataforma. - La plataforma envía el parámetro al servicio.

Flujos de Excepción
E1: El parámetro no existe. - El parámetro no está disponible en la SIM. (Algunos parámetros son opcionales para los fabricantes) - La plataforma le envía al servicio el respectivo mensaje de error.

Información General	
Caso de uso:	**Gestionar clave**
Actores:	Servicio
Propósito:	Permitir que el servicio almacene, actualice u obtenga una clave en la tarjeta SIM.
Resumen:	El servicio puede crear una clave, obtenerla, actualizarla o almacenarla. La plataforma realiza el procedimiento sobre la tarjeta SIM.
Tipo:	Primario.
Precondiciones	
- El applet debe estar instalado en la tarjeta SIM.	

[23] Applet: Aplicación Java Card

[24] APDU: Application Protocol Data Unit

Flujo Principal
- El servicio especifica lo que quiere hacer: generar, obtener o almacenar una clave. Subflujos S1, S2 ó S3. - Se le informa al servicio del éxito o fracaso de la operación.

Subflujos
S1: Generar una clave - La plataforma le ordena a la tarjeta SIM que genere el tipo de clave que el servicio especifica. - La clave se almacena en la SIM. *S2: Obtener una clave* - El servicio le informa a la plataforma qué clave en particular quiere obtener. - La plataforma obtiene la clave desde la tarjeta SIM y la retorna al servicio. *S3: Almacenar una clave.* - El servicio envía a la plataforma una nueva clave, para que sea almacenada en la tarjeta SIM. El servicio debe informar las características de la clave que ha enviado: si es simétrica o asimétrica (pública o privada) y el algoritmo de cifrado para el que es utilizada.

Flujos de Excepción
E1: La Clave no es válida. - La plataforma no soporta el tipo de clave especificada por el servicio. - No se almacena ni se actualiza ninguna clave

Información General	
Caso de uso:	**Verificar firma**
Actores:	Servicio
Propósito:	Garantizar la integridad de la información y el no repudio.
Resumen:	El servicio obtiene la información con su respectiva firma digital. Este caso de uso permite verificar la validez de dicha firma digital con respecto a la información recibida.
Tipo:	Primario.
Precondiciones	
- El applet debe estar instalado en la tarjeta SIM.	

Flujo Principal
- El servicio le pasa a la plataforma tres parámetros: la información recibida, la firma digital y la clave pública de la entidad que envió la información. - La plataforma descifra con la clave pública la firma digital, obteniendo un hash. - Se calcula el hash de la información recibida. - Se comparan los dos hash. - Si los hash son iguales, se le informa al servicio de la validez de la firma. Si no son iguales se le informa lo contrario.

Flujos de Excepción
E1: No se puede descifrar la firma digital. - La plataforma lanza una excepción debido a tres posibilidades: la firma no es válida, la clave pública no es válida, el algoritmo con que se obtuvo la firma no es soportado.

Información General	
Caso de uso:	**Calcular firma**
Actores:	Servicio
Propósito:	Este caso de uso permite calcular una firma digital.
Resumen:	El servicio debe especificar los datos de entrada (información) y el tipo de algoritmo usado para obtener la firma. Internamente, la plataforma también hará uso de las claves asimétricas del usuario.
Tipo:	Primario.
Precondiciones	

- El applet debe estar instalado en la tarjeta SIM.
- Deben estar almacenadas la clave pública y privada del usuario.

Flujo Principal

- La plataforma calcula el hash a la información proporcionada por el servicio.
- La plataforma envía a la SIM el hash para que la tarjeta lo cifre con la clave privada del usuario. La SIM retorna a la plataforma el hash cifrado.
- La plataforma obtiene la clave pública del usuario, la cual está almacenada en la SIM.
- La plataforma retorna al servicio el hash cifrado y la clave pública del usuario.

Flujos de Excepción

E1: El algoritmo para generar la firma no está soportado.

- Se le informa al servicio, del fracaso al generar la firma con dicho algoritmo.

Información General

Caso de uso:	**Usar descifrado asimétrico**
Actores:	Servicio
Propósito:	Permitirle conocer al servicio si la información que le llegó no ha sido modificada. Dependiendo si fue cifrada con una llave pública o una privada, permite garantizar que el usuario es el destino real o que la entidad que lo envió es realmente quien dice ser.
Resumen:	El servicio le envía a la plataforma la información cifrada, el algoritmo utilizado y con qué clave asimétrica quiere que la descifre.
Tipo:	Primario.

Precondiciones

- El applet debe estar instalado en la tarjeta SIM.
- Debe estar almacenada la clave privada del usuario, en el caso correspondiente.

Flujo Principal

- El servicio le envía a la plataforma la información cifrada, el nombre del algoritmo y como quiere que la descifre. Subflujos S1ó S2.
- La plataforma le retorna al servicio la información descifrada.

Subflujos

S1: Descifrar con la clave privada del usuario

- La plataforma ordena a la tarjeta SIM que descifre la información con la clave privada del usuario.
- La SIM retorna la información descifrada a la plataforma.

S2: Descifrar con la clave pública de una entidad conocida

- La plataforma almacena la clave pública de la entidad conocida en la tarjeta SIM.
- La plataforma ordena a la tarjeta SIM que descifre la información con la clave pública de la entidad.
- La SIM le retorna la información descifrada a la plataforma.

Flujos de Excepción

E1: No se puede descifrar la información.

- La plataforma lanza una excepción debido a tres posibilidades: la información ha sido modificada, la clave no es válida, el algoritmo no es soportado

Información General

Caso de uso:	**Usar cifrado asimétrico**
Actores:	Servicio
Propósito:	Le permite al servicio cifrar información con la clave pública o privada del usuario, o cifrar con la clave pública de una entidad.
Resumen:	El servicio le envía a la plataforma la información para ser cifrada con alguna de las claves asimétricas almacenadas en la SIM.
Tipo:	Primario.

Precondiciones
- El applet debe estar instalado en la tarjeta SIM. - Debe estar almacenada la clave asimétrica correspondiente.

Flujo Principal
- El servicio le envía a la plataforma la información a cifrar, el nombre del algoritmo y la clave con la cual se desea cifrar. - La tarjeta SIM cifra la información con el algoritmo y la clave especificada. La SIM le retorna a la plataforma la información cifrada. - La plataforma le retorna al servicio la información cifrada.

Flujos de Excepción
E1: No se soporta el algoritmo. - La plataforma lanza una excepción debido a que no puede cifrar con el algoritmo especificado.

Información General	
Caso de uso:	**Usar cifrado simétrico**
Actores:	Servicio
Propósito:	Permitirle al servicio cifrar información con una clave simétrica. Así se logra garantizar la integridad y confidencialidad de la información.
Resumen:	El servicio le envía a la plataforma la información a cifrar, el algoritmo utilizado y con qué clave simétrica quiere que la cifre.
Tipo:	Primario.
Precondiciones	
- El applet debe estar instalado en la tarjeta SIM. - Debe estar almacenada la clave simétrica correspondiente.	

Flujo Principal
- El servicio le envía a la plataforma la información, el nombre del algoritmo y especifica la clave con la cual se desea cifrar. - La plataforma obtiene de la tarjeta SIM la clave respectiva. Subflujos S1ó S2. - La plataforma cifra la información con el algoritmo respectivo. - La plataforma envía al servicio la información cifrada.

Subflujos
S1: Obtener la clave simétrica del usuario - La plataforma pide a la tarjeta SIM la clave simétrica del usuario. - La SIM envía la clave a la plataforma. *S2: Obtener la otra clave simétrica* - La plataforma pide a la tarjeta SIM la otra clave simétrica. - La SIM envía la clave a la plataforma.

Flujos de Excepción
E1: No se puede cifrar la información. - La plataforma lanza una excepción debido a que el algoritmo no es soportado

Información General	
Caso de uso:	**Usar descifrado simétrico**
Actores:	Servicio
Propósito:	Permite al servicio conocer si la información ha sido modificada, garantizando la integridad y confidencialidad de la información.
Resumen:	El servicio le envía a la plataforma la información cifrada, el algoritmo utilizado y especifica la clave simétrica con la cual se desea descifrar.
Tipo:	Primario.
Precondiciones	

- El applet debe estar instalado en la tarjeta SIM. - Debe estar almacenada la clave simétrica correspondiente.

Flujo Principal
- El servicio le envía a la plataforma la información cifrada, el nombre del algoritmo y especifica la clave con la cual se desea descifrar. - La plataforma obtiene de la tarjeta SIM la clave respectiva. Subflujos S1ó S2. - La plataforma descifra la información cifrada con el algoritmo respectivo. - La plataforma envía al servicio la información descifrada.

Subflujos
S1: Obtener la clave simétrica del usuario - La plataforma solicita a la tarjeta SIM la clave simétrica del usuario. - La SIM envía la clave a la plataforma. *S2: Obtener la segunda clave simétrica* - La plataforma solicita a la tarjeta SIM la segunda clave simétrica. - La SIM envía la clave a la plataforma.

Flujos de Excepción
E1: No se puede descifrar la información. - La plataforma lanza una excepción debido a que el algoritmo no es soportado

5.3 Descripción de clases

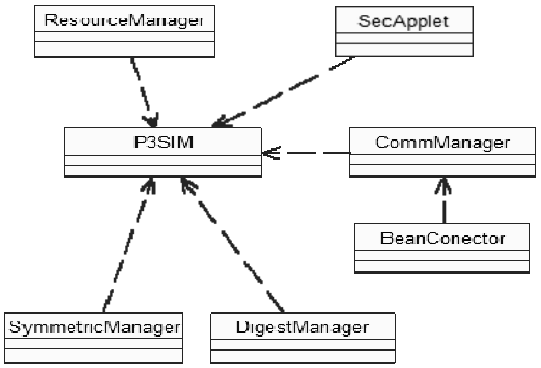


Figura 33 Diagrama de clases P3SIM

- *SECApplet*: Es una clase Java Card que implementa todas las facilidades criptográficas y SAT (SIM Application Toolkit) que serán utilizadas, como por ejemplo generación y almacenamiento de claves, cifrado asimétrico y el acceso a parámetros GSM. Esta clase puede ser modificada para ampliar las capacidades del framework, como fue planteado en los objetivos para la construcción de la plataforma.

- *P3SIM*: Es la clase principal de la plataforma, provee los métodos estáticos que pueden ser invocados por los desarrolladores que hagan uso del Framework.

- *ResourceManager*: Clase encargada del manejo de los recursos necesarios para instalar el SECApplet en la tarjeta SIM.

- *CommManager*: Clase encargada del manejo de las comunicaciones entre la clase P3SIM y el Applet de seguridad SECApplet instalado en la tarjeta SIM del móvil. Para realizar sus tareas se soporta en la clase BeanConnector.

- *BeanConnector*: Clase que realiza los procesos de comunicación con el Applet de seguridad a bajo nivel, manipula los datos y los bytes necesarios para la generación de las command APDU (Application Protocol Data Units) con las cuales se comunica P3SIM y el Applet de seguridad instalado en la tarjeta SIM.

- *SymmetricManager*: Clase encargada del manejo de la criptografía simétrica, posee métodos necesarios para el cifrado y descifrado utilizando el algoritmo DES.

- *DigestManager*: Clase encargada de la generación de MessageDiggest utilizando el algoritmo SHA-1, además permite la comparación de dos MessageDiggest a fin de verificar la validez de una firma.

5.4 Diagramas de secuencia

Caso de uso: Cargar Applet

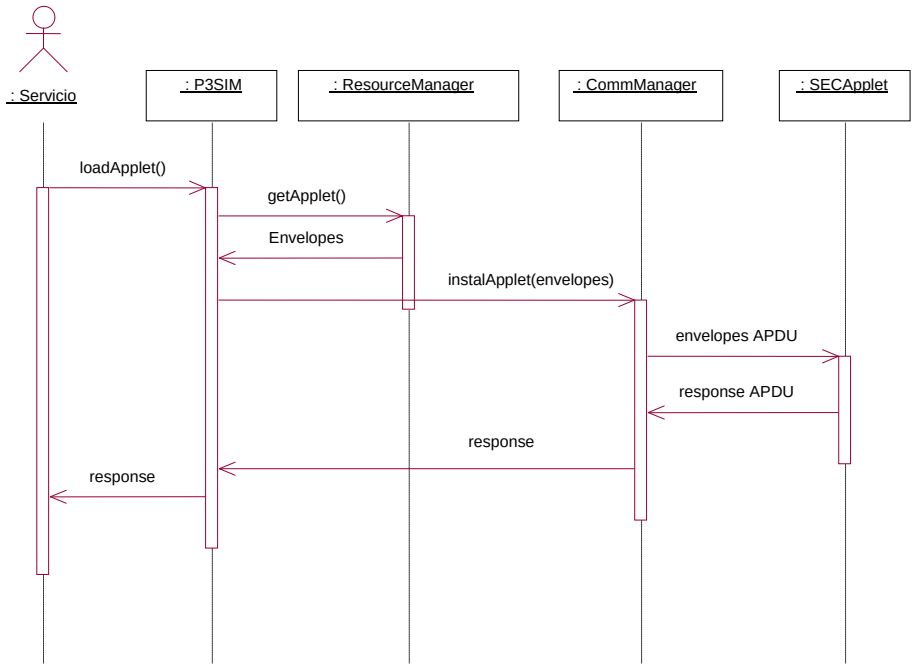


Figura 34 Diagrama de secuencia Cargar Applet

Caso de uso: Generar firma

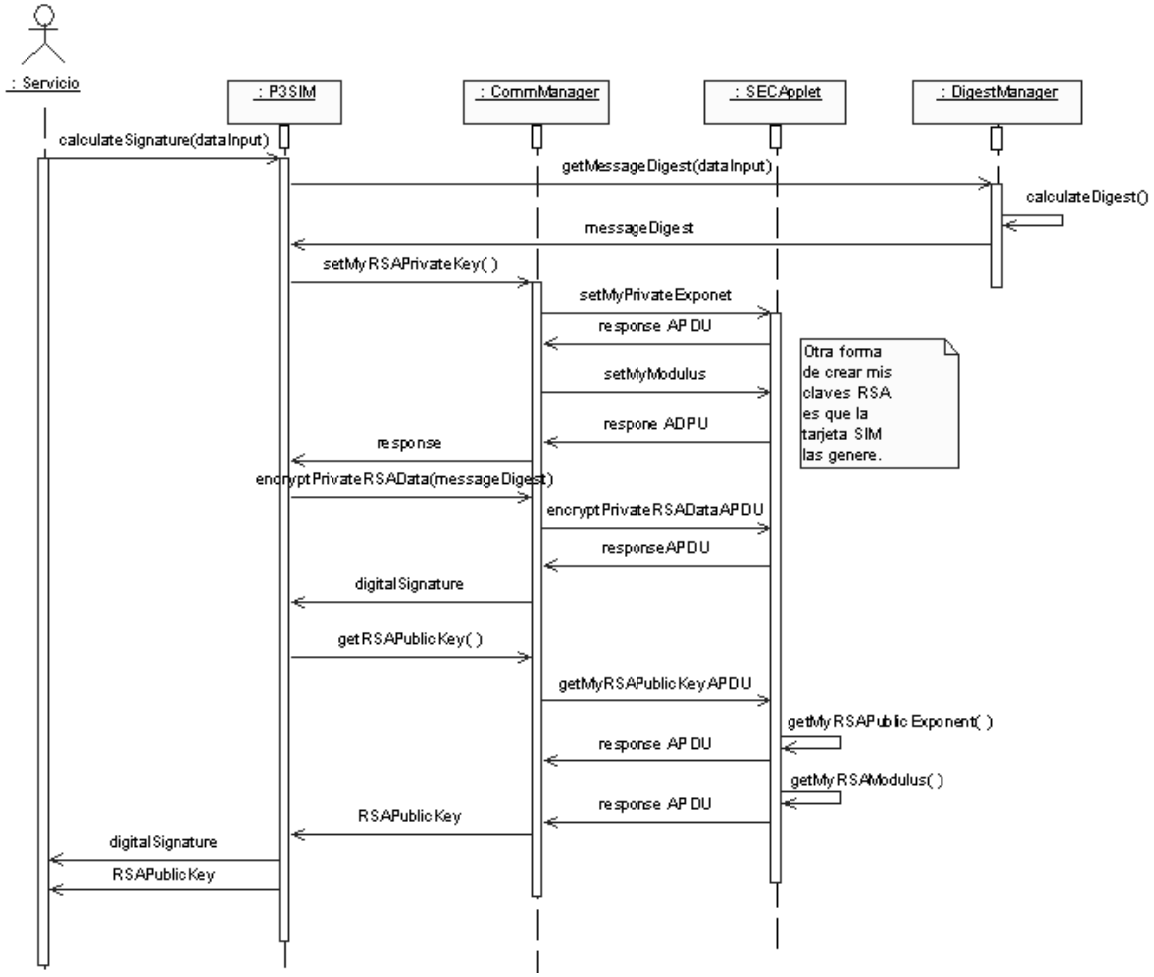


Figura 35 Diagrama de secuencia Generar firma

Caso de uso: Gestionar clave

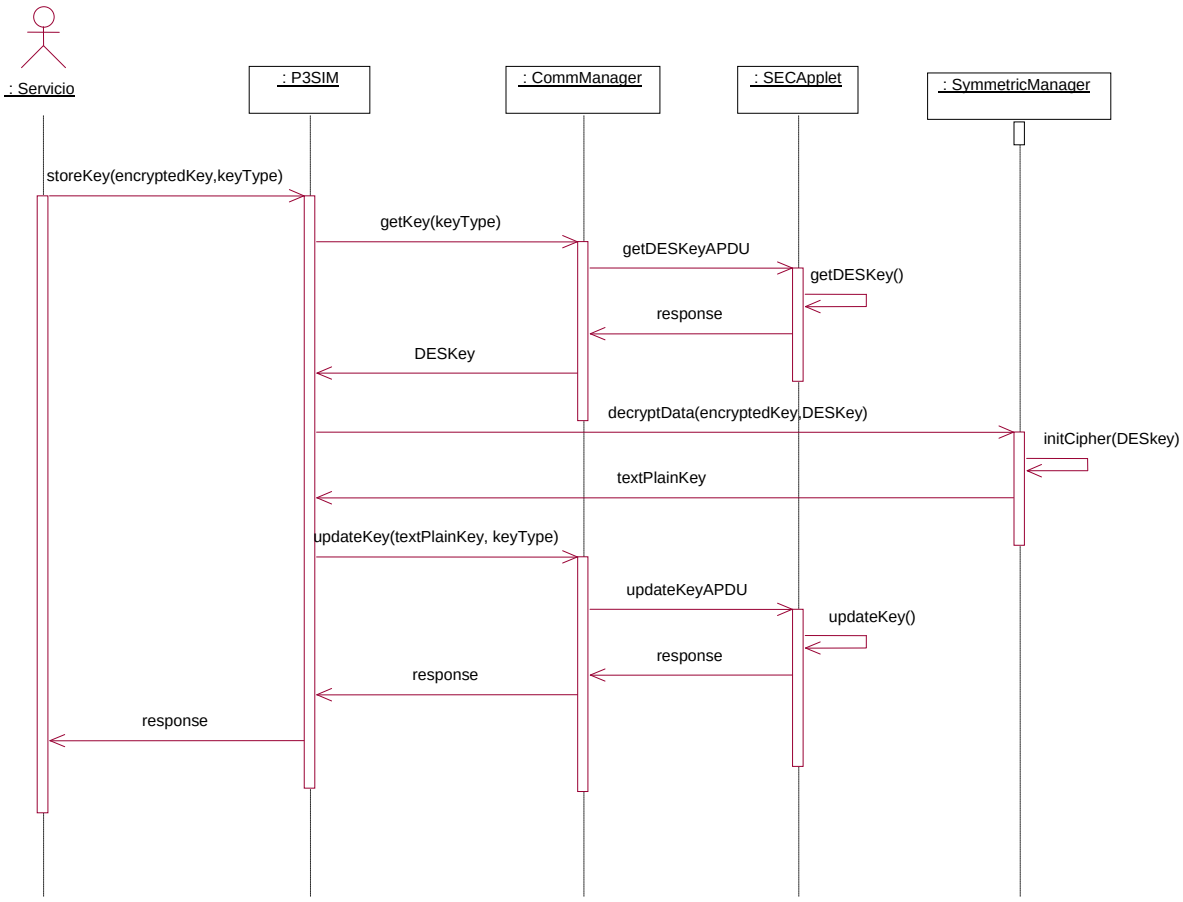


Figura 36 Diagrama de secuencia Gestionar clave

Caso de uso: Obtener parámetro SIM

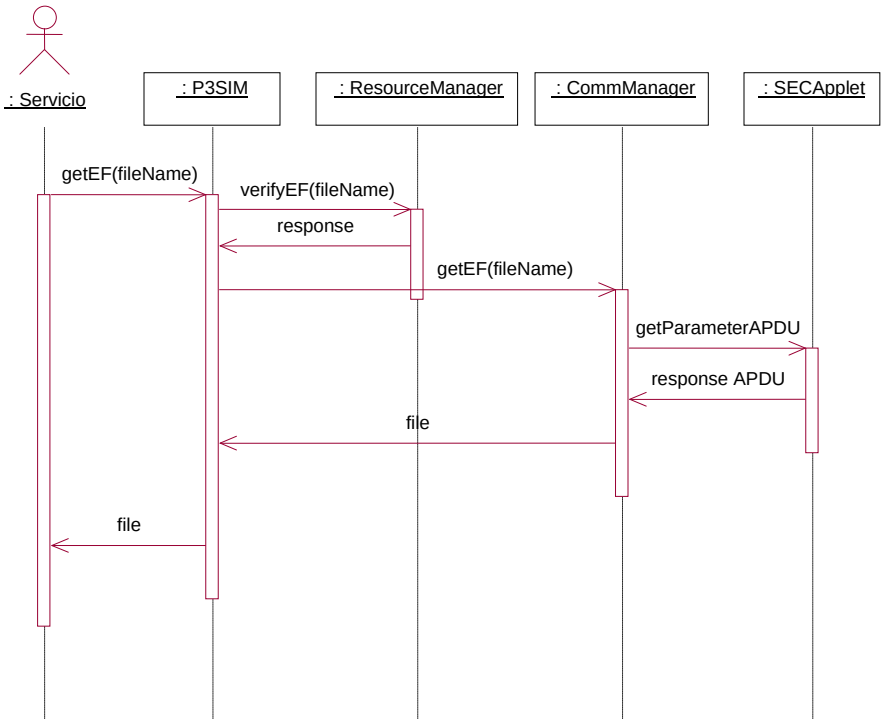


Figura 37 Diagrama de secuencia Obtener parámetro SIM

Caso de uso: Usar cifrado asimétrico

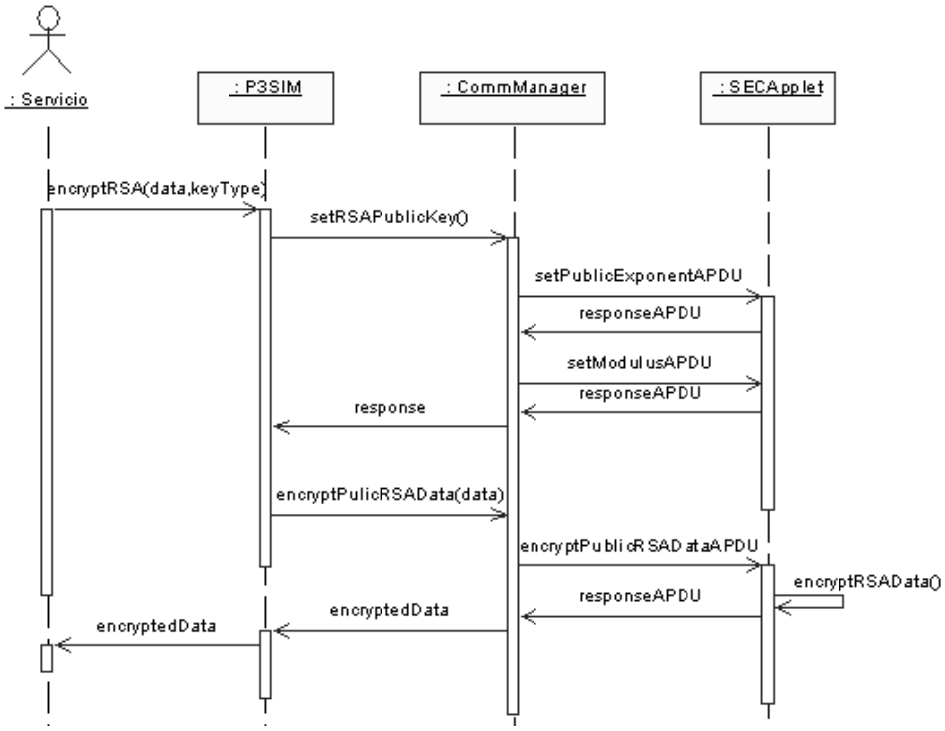


Figura 38 Diagrama de secuencia Usar cifrado asimétrico

Caso de uso: Usar cifrado simétrico

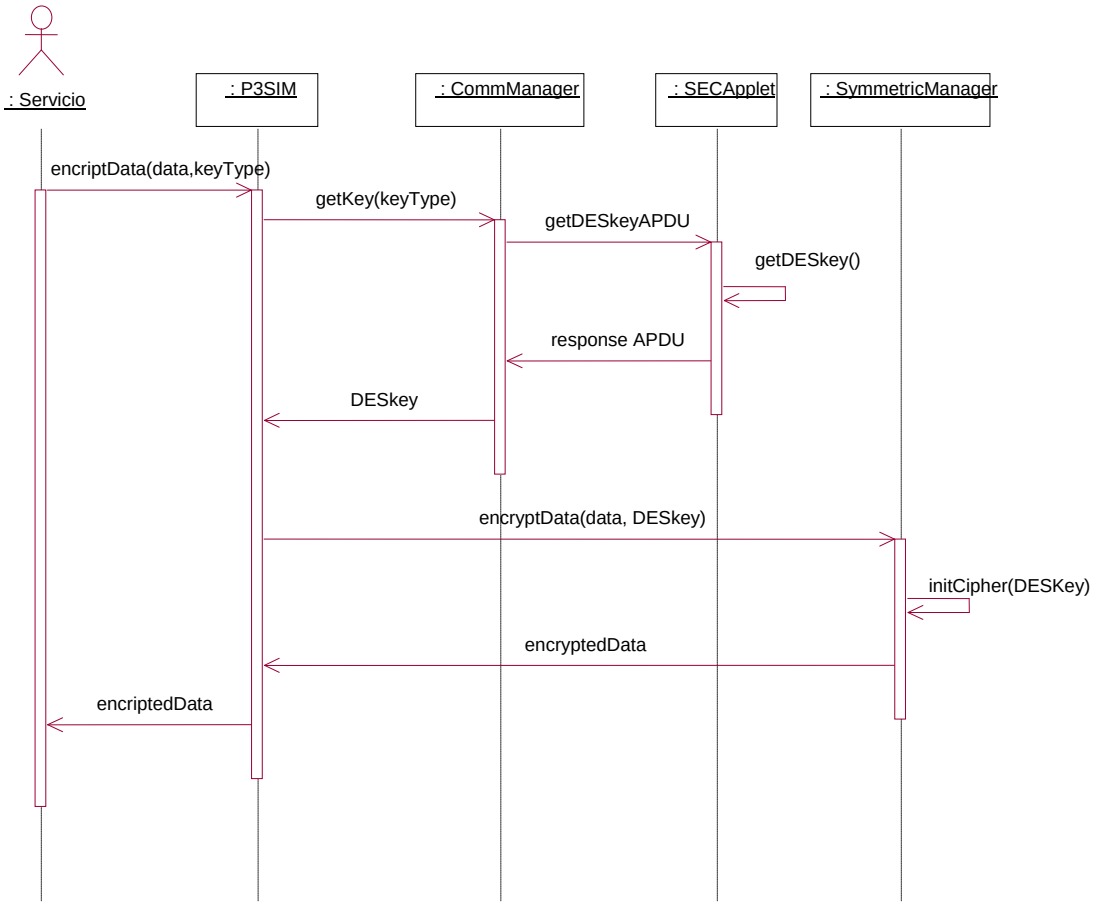


Figura 39 Diagrama de secuencia Usar cifrado simétrico

Caso de uso: Usar descifrado asimétrico

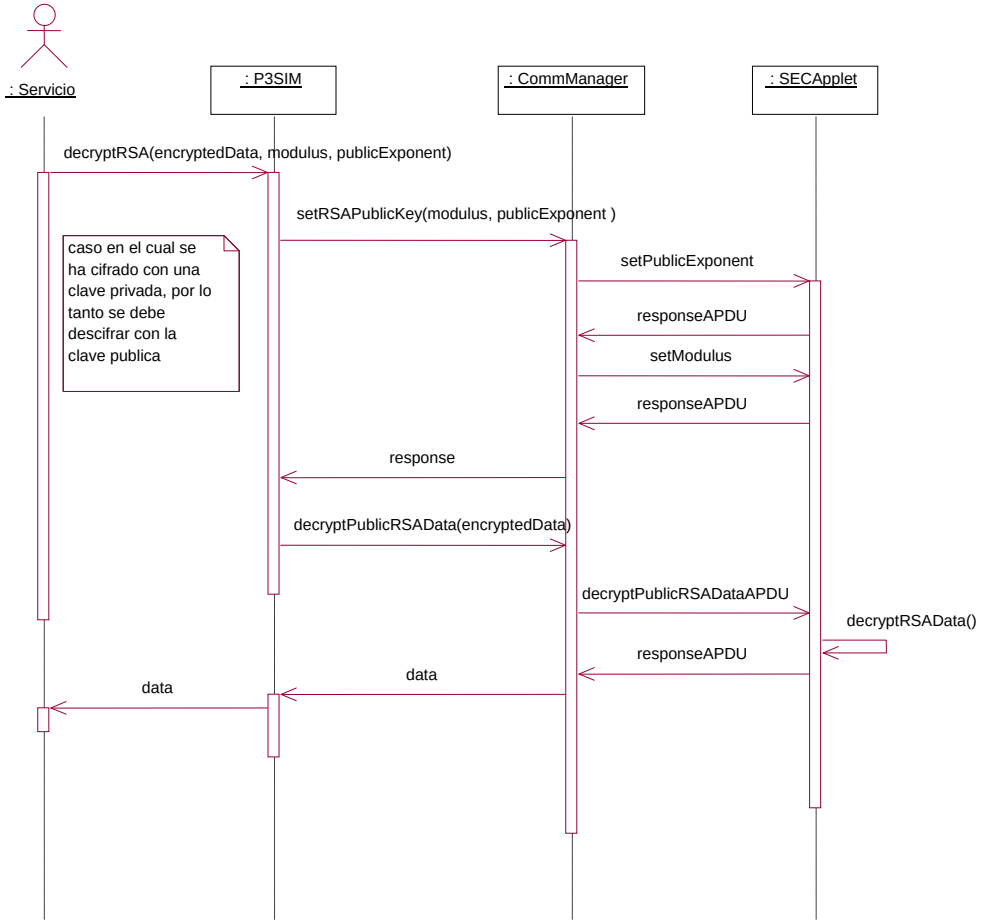


Figura 40 Diagrama de secuencia Usar descifrado asimétrico

Caso de uso: Usar descifrado simétrico

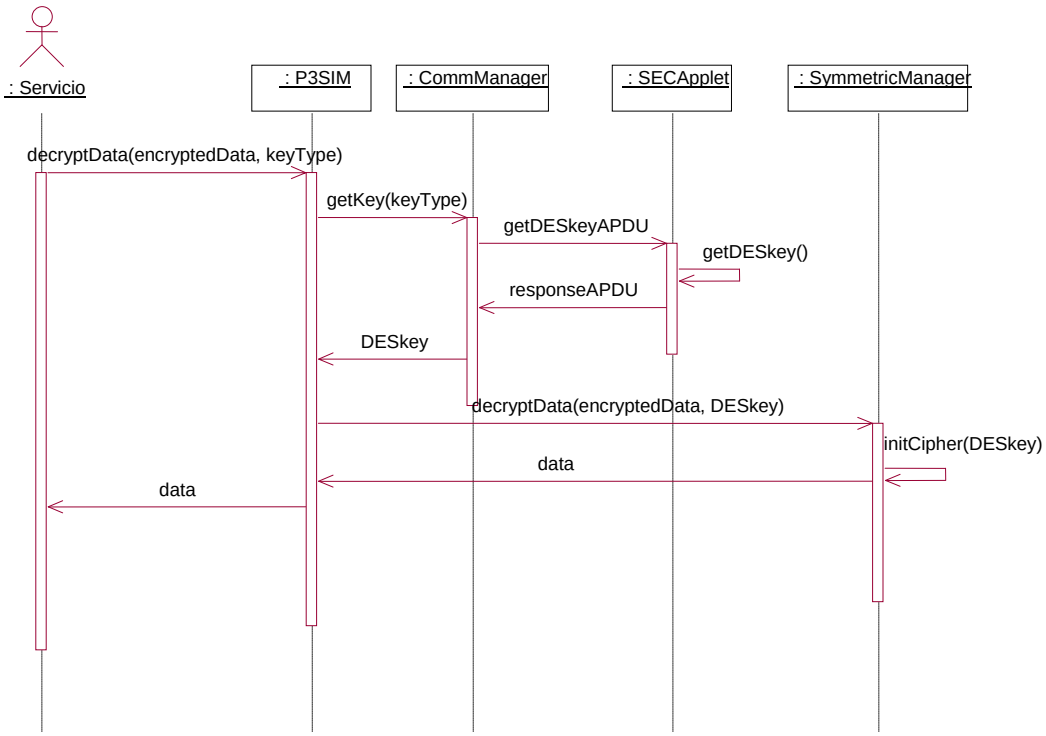


Figura 41 Diagrama de secuencia Usar descifrado simétrico

Caso de uso: Verificar firma

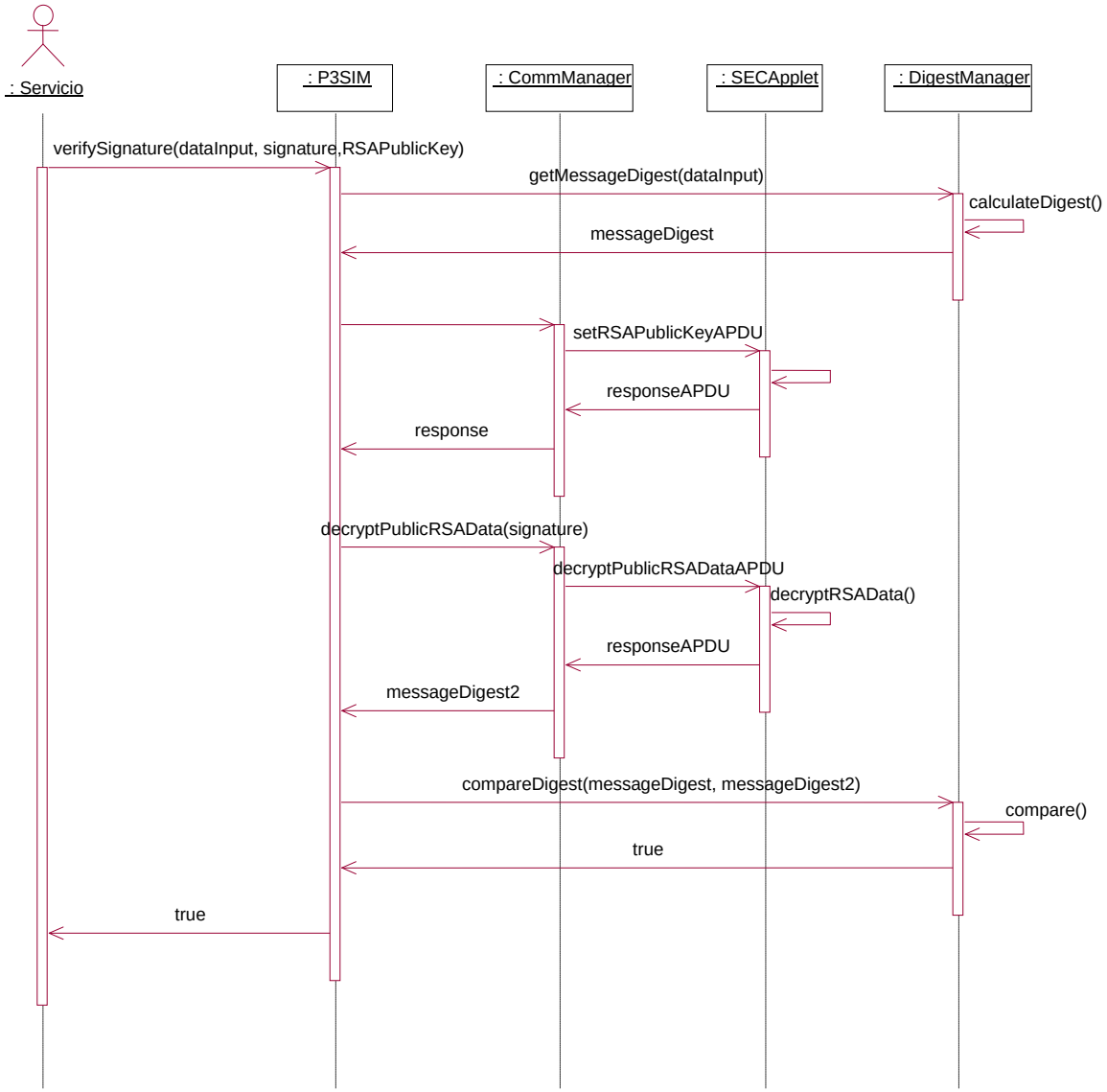


Figura 42 Diagrama de secuencia Verificar firma

5.5 Pruebas de la plataforma

5.5.1 Definición del archivo de compilación

Como parte de los servicios de la plataforma, se creó un archivo XML para el entorno de desarrollo Eclipse, que permite compilar, empaquetar y generar los scripts de los Applets Java Card de una forma muy simple. Las capacidades del archivo de compilación, hacen uso de Java Card Kit 2.2.1 y JSDK 1.4.x. A continuación se muestra su contenido.

```
<?xml version="1.0" encoding="UTF-8"?>
<project default="makeAll" name="JavaCard Builder" basedir=".">

 <property name="scr" value="src" />
 <property name="classes" value="classes" />
 <property name="script" value="script"/>
 <property name="package" value="SECApplet"/>
 <property name="AID" value="0xa0:0x00:0x00:0x00:0x62:0x01:0x0d 1.0"/>
 <property name="Applet"
  value="0xa0:0x00:0x00:0x00:0x62:0x01:0x0d:0x01 ${package}.SECApplet"/>

 <property name="JC_HOME" value="C:\javacard\java_card_kit-2_2_1"/>

<target name="clean" description="clean project directories">
 <delete dir="${classes}" />
</target>

<target name="compile" depends="clean" description="Compile code">
 <mkdir dir="${classes}"/>
 <javac srcdir="${scr}" destdir="${classes}" debug="yes" verbose="yes"/>
</target>
<target name="converter" depends="compile" description="Generate EXP, CAP files">
 <exec executable="${JC_HOME}/bin/converter.bat">
                <arg line="-classdir ${classes}" />
                <arg line="-exportpath ${JC_HOME}/api_export_files"/>
                <arg line="-out EXP JCA CAP"/>
                <arg line="-applet ${Applet}"/>
                <arg line="${package} ${AID}"/>
 </exec>
<!--<copydir dest="javacard" src="${classes}/${package}/javacard"></copydir>-->
 <copy todir="javacard" overwrite="yes">
        <fileset dir="${classes}/${package}/javacard"></fileset>
 </copy>
 <delete dir="${classes}/${package}/javacard" />
</target>
<target name="scriptgen" depends="converter" description="Gener the script file">
 <mkdir dir="${script}"/>
 <exec executable="${JC_HOME}/bin/scriptgen.bat">
        <arg line="-o script/${package}_script.txt"/>
        <arg line="javacard/${package}.cap"/>
 </exec>
</target>

<target name="makeAll" depends="scriptgen" description="make all targets"/>
</project>
```

Como se puede observar, inicialmente se definen las propiedades y luego se realizan cuatro tareas: La primera es limpiar el directorio, la segunda es compilar el Applet, la tercera es convertir los .class en un paquete .cap, y la cuarta es generar el script que contiene los APDU que crearán el Applet en la implementación de referencia de Java Card, conocida como “cref”. Cada tarea es dependiente del resultado de la tarea que la antecede.

5.5.2 Pruebas de unidad y de sistema

La descripción de las pruebas de unidad y del sistema fue realizada en la sección *Pruebas de la Plataforma P3SIM* de la Vista de Infraestructura en la monografía.

Anexo F

Vista Comunidad – Cuestionario Entrevista

P1. ¿Dónde residen los artesanos y productores de CORSEDA?

P2. ¿Dónde reside el personal encargado de la logística y administración de CORSEDA?

P3. ¿Debe desplazarse a otras zonas frecuentemente para cumplir con sus labores en la corporación?

P4. ¿Qué nivel de formación con respecto al manejo del PC y teléfonos móviles tienen los miembros de CORSEDA?

P5. ¿Qué tipo de actividades realiza el personal vinculado a CORSEDA en la corporación?

P6. ¿Hace uso de los servicios del sistema financiero (ej. cuentas bancarias, tarjetas de crédito)?

P7. ¿Con qué frecuencia utiliza Internet? ¿Para qué lo usa?

P8. ¿Qué ventajas percibe en el uso de Internet?

P9. ¿Tiene un teléfono móvil? En caso afirmativo, indique el modelo y el tipo de plan en el cual se encuentra activado (Prepago o Postpago).

P10. ¿Qué factores lo motivaron para adquirir un teléfono móvil?

P11. ¿Para qué usa el teléfono móvil frecuentemente?

P12. ¿Existe una buena cobertura por parte del operador de telefonía móvil en las zonas donde realiza su trabajo?

P13. ¿Cuánto dinero invierte mensualmente en servicios de telefonía móvil (ej. recarga del teléfono)?

P14. ¿Ha navegado en Internet a través de su teléfono móvil? En caso negativo, explique las razones por las cuales no lo ha hecho.

P15. ¿Cómo ha sido su experiencia en la navegación a través de su teléfono móvil? (Velocidad experimentada, acceso al contenido y navegación sobre el mismo).

P16. ¿Comparte su teléfono móvil con otra persona en el sitio de trabajo?
P17. ¿Generalmente toma la iniciativa para iniciar las llamadas desde su teléfono o prefiere recibirlas?

P18. ¿Usualmente envía mensajes de texto? ¿Qué tipo de mensajes envía? (ej. Familiares, amistad, concursos)

P19. ¿Lee usualmente los mensajes de texto que llegan a su teléfono móvil?

P20. ¿Considera conveniente acceder a los servicios de LINK ALL a través de su teléfono móvil?

P21. ¿Qué servicios de la plataforma LINK ALL actual considera como los más útiles para su labor?

P22. ¿De los servicios mencionados anteriormente, cuáles considera de mayor utilidad para ser accedidos desde su teléfono cuando se encuentre fuera de su sitio de trabajo?

Anexo G

Recomendaciones de la Mobile Web Initiative aplicadas al piloto en el Caso de Estudio

A continuación se muestra la aplicación de las recomendaciones de la W3C Mobile Web Intitiative, sobre el piloto construido en el caso de estudio descrito en el capítulo 4 de la monografía.

1. Temática consistente con la URL

En la Figura 43 se muestra como desde tres dispositivos diferentes se accede al mismo contenido a través de la URL de acceso al piloto implementado.

2. Explotar las capacidades del dispositivo

En las diferentes imágenes tomadas del piloto construido, se puede observar claramente como el sitio adaptado aprovecha las capacidades de cada dispositivo. La Figura 44 compara las imágenes en el encabezado de la página principal. Cada imagen es adaptada convenientemente, tratando de aprovechar al máximo las capacidades del dispositivo de acuerdo a su contexto de entrega.

3. Trabajo Alrededor De Las Implementaciones Deficientes

Se ralizaron algunas ajustes con respecto a las implementaciones deficientes. Por ejemplo, al utilizar el emulador OpenWave UP.Simulator 4.1, las cabeceras HTTP confirman el soporte para código HTML, aunque en la práctica, sólo brinda soporte para código WML. En este caso, la priorización de la información obtenida de las fuentes UAProf y WURFL, ayudan a solucionar el problema.

4. Pruebas

En las diferentes imágenes tomadas del piloto construido, puede constatarse la variedad de pruebas realizadas sobre varios dispositivos, incluyendo emuladores y dispositivos reales:

Teléfonos	Nokia 5200, 3220 (Nokia, Nokia device specifications, 2008); Sony Ericsson W200 (SonyEricsson, 2008); Motorola V3 (Motorola, 2008); Siemens A56i (SmartGSM, 2008)
Emuladores	Openwave Phone Simulator (OpenWave, 2005), UP SDK (Phone.com, 1999), Nokia S40 SDK (Nokia, 2007), Nokia S60 SDK (Nokia, 2007), WinWAP (WinwapTechnologies, 2008)

5. URL De Los Puntos De Acceso

Para efectos de demostración del piloto, no fue posible adquirir un dominio como tal para facilitar el acceso al mismo. Sin embargo, se tuvo la posibilidad de contar con una dirección IP real a partir de la cual se estructuró la URL http://190.5.195.48/linkall/ como punto de acceso; como se puede observar, la dirección es suficientemente corta y no contiene el nombre de la página.

Figura 43 Página de acceso desde varios dispositivos

6. Barras De Navegación

La plataforma OneWeb ubica una barra de navegación en la parte superior de la página y el resto de barras al final de la misma. Es preciso tener en cuenta que la plataforma realiza una abreviatura de la barra de navegación, a través de la transformada de Elisión. El resultado se puede observar en la Figura 45.

7. Estructura Balanceada

La Figura 46 muestra la estructura de las barras de navegación del piloto construido. Se puede concluir claramente que éstas son cortas y expresan claridad en su contenido. Por otro lado, las pruebas realizadas permiten determinar un grado de profundidad adecuado, en el cual los usuarios no requieren más de tres saltos para acceder a una funcionalidad específica.

8. Mecanismos De Navegación

En el piloto construido, se utilizaron las barras de navegación de manera recurrente como mecanismo de navegación. No se utilizó el mecanismo “drill-down”, ya que el contenido de las páginas es muy breve en términos generales.

Figura 44 Adaptación dinámica de imágenes

Figura 45 Barras de navegación ubicadas en la parte superior e inferior de la página

9. Teclas de acceso

OneWeb agrega teclas de acceso a los enlaces de mayor relevancia, como se muestra en la Figura 47. Las teclas de acceso se indican entre paréntesis; por ejemplo, para acceder al contenido del enlace “artesanía”, se debe presionar la tecla “1”.

10. Identificación del enlace de destino

En la Figura 47 se puede observar cómo la plataforma introduce información sobre el enlace, para aquellos de mayor relevancia; entre corchetes se muestra el peso en kilobytes de la página destino.

11. Mapas De Imágenes

No se emplearon mapas de imágenes en la construcción del piloto.

12. Recarga, redirección y ventanas emergentes

En el piloto construido no se utilizaron etiquetas para recargas automáticas, redirecciones o ventanas emergentes.

Figura 46 Estructura balanceada

Figura 47 Enlaces relevantes de la barra de navegación

13. Contenido de enlaces externos

En el piloto construido la mayoría de las imágenes son ubicadas en el servidor local, aunque existe flexibilidad para que los usuarios puedan ingresar sus imágenes desde otros sitios. Por otro lado, los estilos y el código Java Script son ubicados en archivos externos. En la Figura 48 se muestra el segmento de código que invoca la única hoja de estilos externa.

14. Contenido De La Página

El piloto construido no incluye ningún tipo de publicidad, ni la plataforma añade contenido de este tipo. Por otro lado todo, el contenido de las páginas es claro y adaptable en una amplia gama de dispositivos.

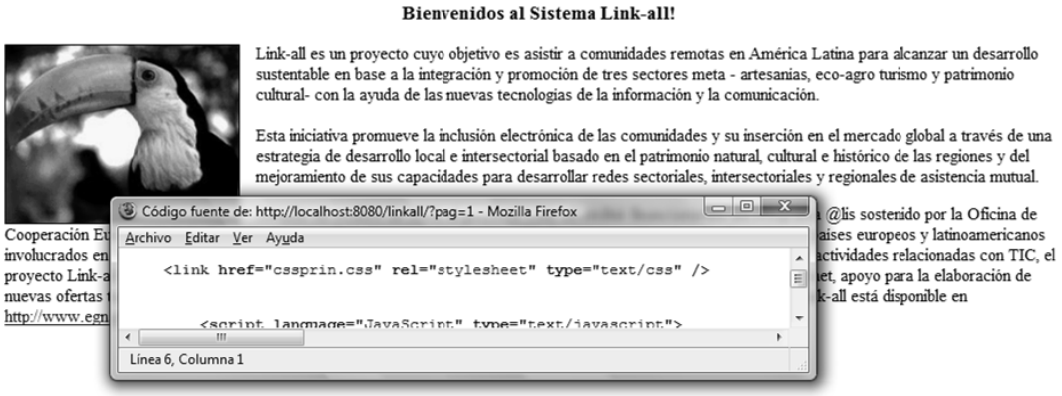


Figura 48 Invocación a la hoja de estilos externa

15. Tamaño De La Página

El tamaño de las páginas desplegadas es calculado por la plataforma de acuerdo a las capacidades de cada dispositivo, eliminando la posibilidad de que exista una sobrecarga de memoria. Por otro lado, el tamaño de una página sin adaptar en el piloto construido, no supera los 10kb.

16. Desplazamiento (Scrolling)

Gracias a la adaptación de las imágenes en correspondencia con el ancho de la pantalla del dispositivo, se omite la necesidad de un desplazamiento horizontal por parte del usuario. En la Figura 49 se muestra una comparación entre la página original y la página adaptada, con respecto a la necesidad de desplazamiento.

Página original Página adaptada

Figura 49 Desplazamiento horizontal página original y página adaptada

17. Gráficos

El piloto construido no usa gráficos para establecimiento de espacios.

18. Color

Los colores de los textos fueron seleccionados de tal forma que contrastaran con el fondo y en ningún caso, los colores de las fuentes representan información indispensable para el usuario.

19. Imágenes De Fondo

No se usaron imágenes de fondo en el piloto construido, para evitar sobrecarga y conservar la legibilidad del texto.

20. Título

Los títulos usados describen brevemente el contenido de la página. En la Figura 50 se muestran algunos ejemplos.

Figura 50 Títulos descriptivos

21. Marcos

El piloto construido no utiliza marcos o frames; en su lugar, divide la página a través de etiquetas * .*

22. Elementos Estructurales

Los enlaces considerados de mayor relevancia fueron marcados mediante las etiquetas *<H1>, <H2>* y *<H3>* como se muestra en la Figura 51.

23. Tablas

No se implementaron tablas en el piloto construido.

24. Objetos no-textuales

Los únicos objetos no textuales utilizados son las imágenes; para cada una de ellas, se incluye un texto alternativo que será desplegado cuando ésta con pueda ser cargada por el dispositivo. Se omiten los eventos Java Script “onmouse” y “onkey”.

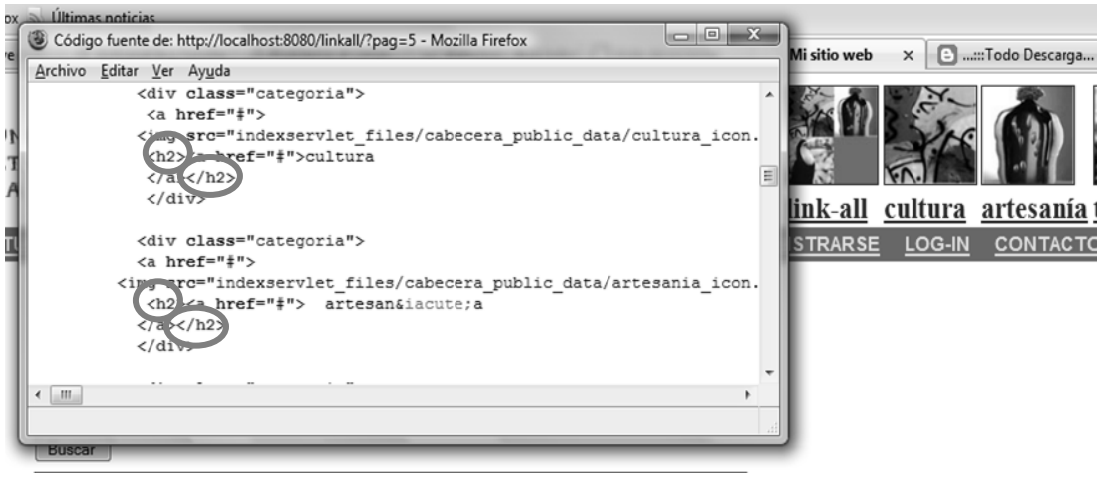


Figura 51 Ejemplo de elementos estructurales

25. Tamaño de la imagen

En todas las imágenes del piloto, fueron incluidas sus propiedades de ancho y su alto, con el fin de facilitar el proceso de adaptación a cualquier dispositivo. La plataforma realiza una reducción del tamaño en el servidor disminuyendo los bytes transmitidos. La Figura 52 muestra un segmento de código donde se fija el ancho y alto de varias imágenes.

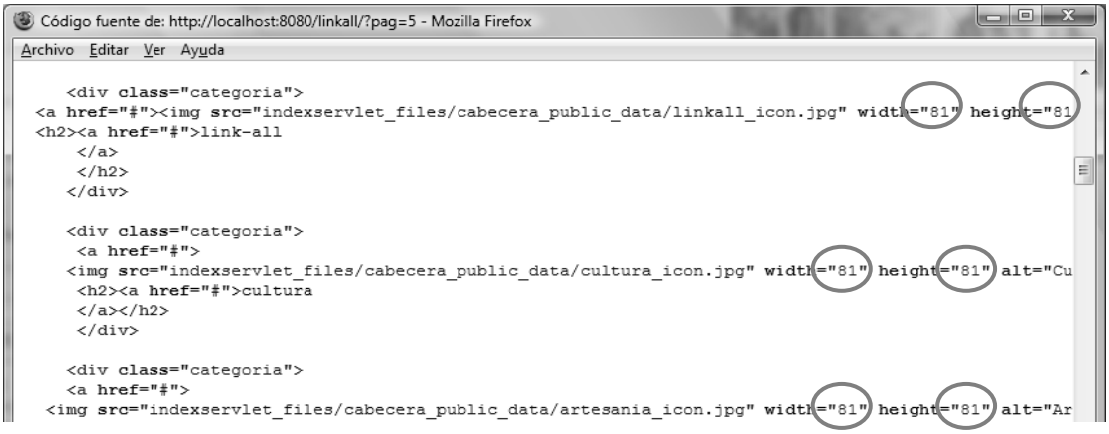


Figura 52 Sección de código etiquetas IMG.

26. Etiquetas Válidas

Las páginas del piloto construido aprobaron con éxito la validación realizada por la herramienta del W3C, como se muestra en la Figura 53.

27. Unidades de medida

Se emplearon medidas relativas para referirse a aquellas que podían ser adaptadas por el navegador del dispositivo. En la Figura 54 se muestra un fragmento del archivo de estilos que muestra el uso de las medidas relativas.

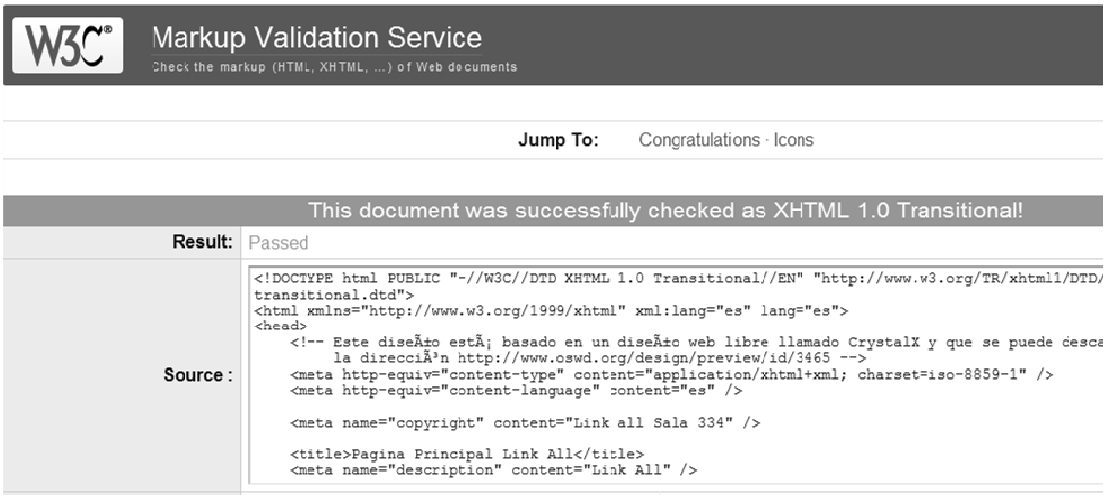


Figura 53 Página del servicio de validación del W3C

```
#menu {
  float:left;
 width: 16%;
  background-color:#ff4500;

}



#contenido {

  float: right;
   width:78%;
  padding:0.5em;
  background-color:#FFFFFF;
```

Figura 54 Uso de las medidas relativas

28. Hojas de estilo

La plataforma OneWeb dispone de un archivo de estilos, siguiendo las recomendaciones de la MWI; sin embargo, la eliminación de las hojas de estilo no evita que el contenido pueda ser desplegado. En la Figura 55 se muestra la página principal del piloto construido, eliminando las hojas de estilo. Por otro lado, la plataforma elimina aquellos estilos que no son soportados por el dispositivo o están relacionados únicamente con elementos estructurales, durante el proceso de adaptación.

29. Reducción del tamaño del código

El piloto construido evita el uso de espacios en blanco innecesarios, utilizando las etiquetas de espaciado *
* y *.*

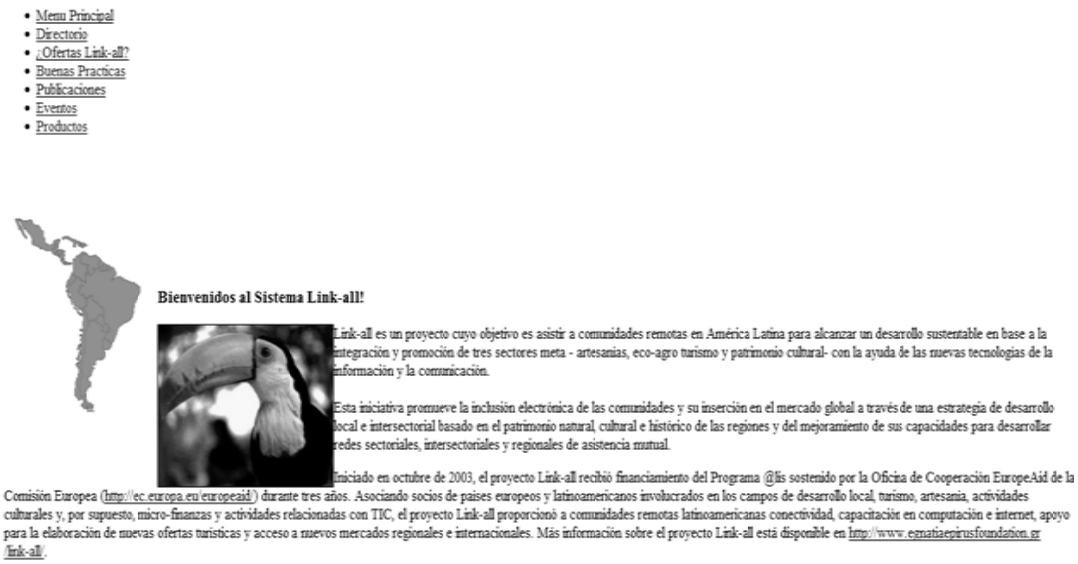


Figura 55 Página principal sin hoja de estilos.

30. Tipos de contenido soportados

La plataforma realiza una adaptación de todos los contenidos según sean las capacidades del dispositivo móvil. La siguiente figura muestra una imagen originalmente en formato jpg convertida en wbmp para un dispositivo que sólo soporta WML.

Figura 56 Imagen en formato jpg adaptada a wbmp

31. Codificación De Caracteres

El piloto construido utiliza la codificación de caracteres ISO-8859-1. Sin embargo la plataforma OneWeb, realiza la adaptación a la codificación de caracteres preferida por cada dispositivo.

32. Mensajes de error

La plataforma OneWeb dispone de algunas páginas de error cuando ocurren excepciones; en ellas se incluye un sistema de navegación como lo muestra la Figura 57. Cuando se trata de un error asociado a la lógica del piloto, se despliega una página de error.

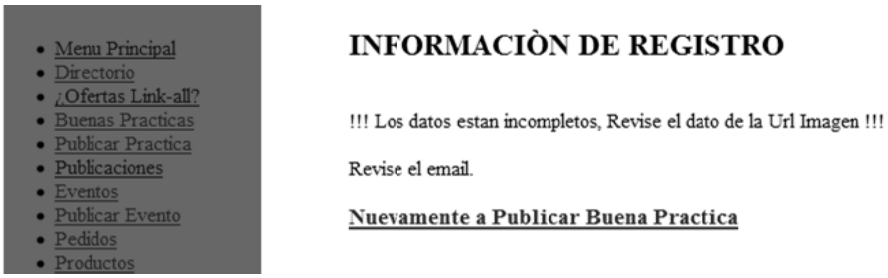


Figura 57 Página de error

33. Cookies

El piloto construido mantiene las sesiones mediante cookies y en el caso de no ser soportadas por el dispositivo, se permite la asociación de los datos de la sesión a la URL.

34. Caché Headers

La plataforma asigna a la cabecera HTTP *Cache-control* el valor "private", de tal forma que se almacene el contenido en caché sólo para el dispositivo que accede actualmente.

35. Fuentes

El piloto construido mantiene un número reducido de tamaños y tipos de fuentes; se uso el grupo de fuentes *Arial, Verdana, Comic Sans MS* y *Sans-Serif*.

36. Entradas Por Teclado

Se procura utilizar las listas de selección en los casos donde se admite para el piloto construido, con el objeto de reducir la introducción de texto desde el teclado del dispositivo. La Figura 58 muestra un ejemplo de la aplicación de esta práctica.

37. Orden en las tabulaciones

Gran parte de las páginas que componen el piloto construido, brindan la facilidad de desplazamiento lógico y ágil por parte del usuario. La Figura 59 muestra un fragmento de código que ilustra esta capacidad, al permitir que el enlace "LINK-ALL" sea seleccionado al presionar la tecla tab una vez.

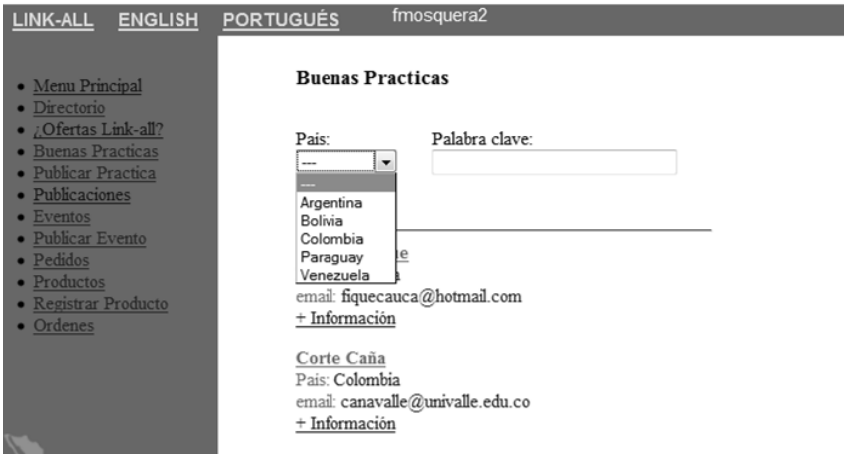


Figura 58 Uso de las listas de selección

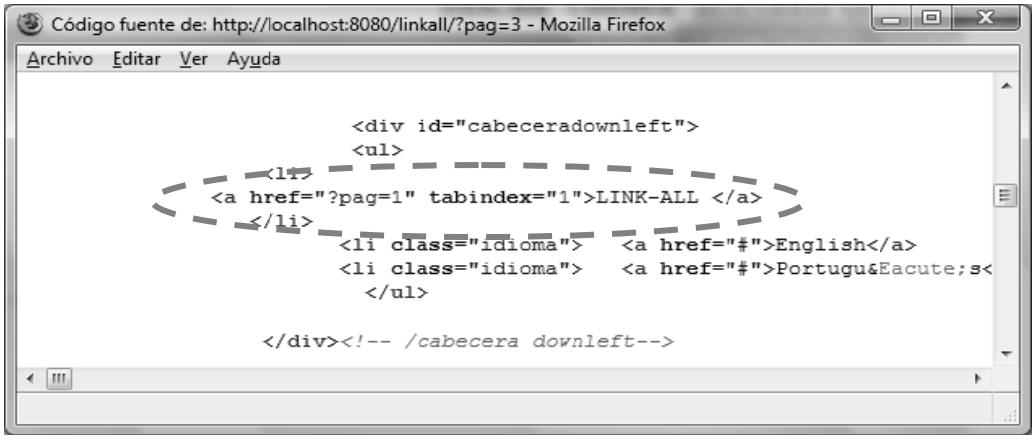


Figura 59 Uso del atributo tabindex

38. Etiquetas para los controles

Se asignaron las etiquetas a los controles mediante el uso de la etiqueta <*label*>, con el fin de garantizar la cohesión durante el proceso de adaptación de contenido. La muestra el caso de la publicación de una nueva práctica, en el cual se busca garantizar que el texto “Nombre” que esta sobre el control de entrada de texto, siempre se ubique junto a este control.

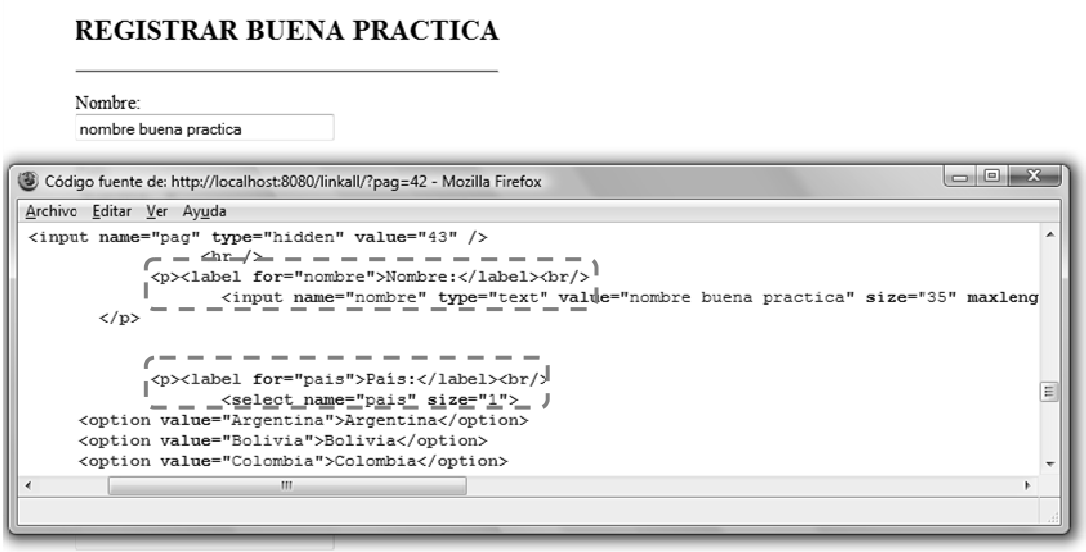


Figura 60 Uso de las etiquetas <label>

Anexo H

Especificación de la plataforma MERCURIO

Arquitectura para la Provisión Segura de Servicios en Redes de Telefonía Móvil (Mercurio)

Oscar Mauricio Caicedo Rendón, Diana Cerón Imbachí, Diego Ivan Chamorro Salas, Francisco Orlando Martínez Pabón, Javier Alexander Hurtado Guaca
Grupo de Ingeniería Telemática, Universidad del Cauca, Popayán, Colombia
{omcaicedo, dceron, dchamorro, fomarti, javhur}@unicauca.edu.co

Abstract **-- La telefonía móvil y las redes de datos inalámbricas han evolucionado en beneficio de las necesidades de los usuarios, permitiéndoles actualmente acceder a diversos servicios de contenido y aplicaciones para realizar sus tareas cotidianas, de trabajo, negocios, entre otras, de una forma más fácil y contando con la disponibilidad de acceso a la información en cualquier momento y lugar. Sin embargo, a estas ventajas se superponen problemas de seguridad que día a día se incrementan, poniendo en peligro la información de los usuarios; por esta razón existe la necesidad de proporcionar un acceso seguro a los servicios móviles que permita garantizar la integridad, confiabilidad y legitimidad de dicha información. Para lograrlo, este artículo propone una arquitectura, basada en los cuatro pilares de una comunicación segura (autenticidad, confidencialidad, integridad y no repudio), que permite la provisión segura de servicios en redes de telefonía móvil, y plantea para su implementación el uso de Java, Java ME, XML y los Servicios Web como tecnologías de soporte.**

Index terms **– Arquitectura para Seguridad, Cifrar, Comunicaciones Seguras, Firmar, PKI, Servicios Móviles, XML Digital Signature, XML Encryption, WS-Security.**

1. INTRODUCCIÓN

Hoy, existe una gran variedad de servicios desplegados sobre sistemas de 2.5 y 3G y una alta gama de dispositivos personales avanzados que permiten a los usuarios de redes móviles conectarse en cualquier momento y lugar, gozando de la movilidad y facilidad de mantener siempre a la mano la información que necesitan. Sin embargo, los servicios soportados en redes de telefonía móvil al necesitar como medio de acceso un entorno inalámbrico traen consigo los problemas de seguridad relacionados a las características de éste, lo cual implica por ejemplo, que la información se desproteja más que en medios cableados [1].

De otro lado, con los servicios móviles el usuario intercambia diferente tipo de información, ya sea personal, empresarial o comercial, cuando realiza transacciones bajo un esquema de comercio electrónico móvil (M-Commerce). En este último caso, se maneja información que requiere distintos niveles de seguridad, como números de tarjeta de crédito o débito, datos personales, claves de acceso, etc. El tipo de información intercambiada, es un atractivo para hackers, espías y personas maliciosas, lo cual, obliga a que los investigadores, las empresas generadoras de tecnología y las prestadoras de servicios móviles dediquen gran parte de sus esfuerzos a garantizar y generar mecanismos confiables de seguridad para la prestación y acceso a los servicios móviles, de forma tal que se mantenga segura, confidencial e integra la información crítica de usuario que se expone en las redes móviles y se mejoren los mecanismos de AAA (Authentication, Authorization, Accounting) en el acceso y uso de este tipo de servicios.

Otro problema en los servicios móviles es la dificultad que se tiene en brindar soluciones robustas de seguridad basadas en tecnologías empleadas en sistemas cableados tradicionales, debido a las bajas capacidades de la mayoría de dispositivos móviles existentes (principalmente en lo relacionado a memoria y poder de procesamiento, aunque se vienen disminuyendo estas limitaciones), haciendo que el dispositivo móvil sea uno de los puntos más débiles en la seguridad de un sistema; en este punto es importante recordar que la seguridad de todo sistema es tan fuerte como la de su punto más débil [2].

Con el fin de minimizar los problemas de seguridad presentes en los servicios móviles y garantizar los

pilares de una comunicación segura [3]: autenticación, autorización, confidencialidad, integridad y no repudio, ofreciendo un acceso seguro extremo a extremo (de medio nivel) a los servicios móviles, el grupo de interés en desarrollo de aplicaciones móviles e inalámbricas (W@PColombia) perteneciente al Grupo de Ingeniería Telemática (GIT) de la Universidad del Cauca diseñó e implementó la plataforma Mercurio, haciendo uso de conceptos y tecnologías afines a la seguridad computacional.

A continuación, para presentar Mercurio, en la sección 2, se definen aspectos generales, problemas y niveles de seguridad en servicios móviles, en la sección 3, se describen las tecnologías orientadas a garantizar la seguridad en los servicios móviles, en la sección 4 se realiza la descripción en sí de Mercurio y finalmente en la sección 5 se exponen las conclusiones.

2. Seguridad en Servicios Móviles

2.1. Aspectos Generales

La seguridad es un concepto asociado a la certeza, falta de riesgo o contingencia. Entonces, conviene aclarar que no siendo posible la certeza absoluta, el elemento de riesgo esta siempre presente, independiente de las medidas que se tomen. En adelante se entenderá que la seguridad informática es un conjunto de técnicas encaminadas a obtener altos niveles de seguridad en los sistemas informáticos y de comunicación. Además, la seguridad informática precisa de un nivel organizativo, por lo que se dice que [4]: Sistema de Seguridad = Tecnología + Organización.

Los servicios basados en la movilidad brindan ciertas ventajas sobre los tradicionales, permitiendo por ejemplo agilizar y facilitar algunos procedimientos que pueden ser más dispendiosos si se realizan de forma convencional. A pesar de las ventajas que los servicios móviles traen consigo, se encuentran expuestos a violaciones de seguridad, lo cual constituye un gran desafío para todos los actores de este mercado, requiriendo la utilización de técnicas y tecnologías para evitar estas eventualidades. En este sentido, es importante tener en cuenta que por un número de razones, dictadas por el mercado, las medidas de seguridad asociadas con la transferencia de fondos entre bancos internacionales, no son las apropiadas para ver los cortos de una película. Por ejemplo, muchas medidas de seguridad sobrecargaran las aplicaciones de bajo nivel con complejidades innecesarias, lo cual obstaculiza su uso espontáneo. En contraste, transacciones o descargas de alto nivel que impliquen un valor monetario significativo requerirán medidas de seguridad más fuertes, incluso sacrificando que la ejecución de la transacción dure un poco más. En este caso, un leve retardo es un precio pequeño a pagar por una transacción segura.

Los aspectos mencionados en los párrafos anteriores hacen que los ASPs (Application Service Providers), y por tanto los desarrolladores, deban encontrar un balance entre las expectativas de los usuarios hacia la experiencia con la multimedia que proporcionan las aplicaciones y el nivel de seguridad apropiado en las mismas, el cual puede ser [5]:

- De Nivel bajo. Cuando la información importante o personal no esta en peligro o cuando el valor de una transacción es bastante bajo. En este caso, la seguridad del servicio móvil se puede salvaguardar adecuadamente con técnicas simples de cifrado y PKI (Public Key Infrastructure).
- De Nivel medio. Aplicaciones que almacenan alguna información personal tal como número de licencia de conducción, tarjeta crédito o el pasaporte en un dispositivo móvil. En este caso, la seguridad del servicio móvil se puede salvaguardar con técnicas fuertes de cifrado y PKI.
- De Nivel Alto. Las aplicaciones que se incluyen en el final del espectro de seguridad, son aquellas que implican transacciones monetarias muy grandes, acceso a VPNs (Virtual Private Networks), aplicaciones móviles de oficina que busquen por ejemplo la protección de software muy valioso como copyright de archivos de video o audio. En este caso, la seguridad se puede salvaguardar utilizando módulos dedicados Hardware/Software de tipo SIM (Subscriber Identity Module), WIM (Wireless Identity Module), o Smart Card.

2.2. Problemas de Seguridad en Servicios Móviles

En los últimos años, el nivel de seguridad de las empresas y usuarios ha mejorado considerablemente. Cada vez, se vigila más: la conexión a Internet, la existencia de herramientas de seguridad en las máquinas instaladas en la red, la formación de los usuarios, etc. Sin embargo, al mismo tiempo que se avanza en este sentido, se ve que la tecnología también lo hace pero en una dirección que puede, sin duda, entrar en conflicto con las políticas de seguridad clásicas: la movilidad [6]. En este sentido, existen problemas de seguridad que permiten la transformación, detección o borrado de la información contenida en un dispositivo móvil, tales como virus electrónicos, robo o pérdida de dispositivos, y problemas con Bluetooth entre otros. Sin embargo, éstos no son la única amenaza para la prestación de servicios móviles de forma segura, existen otros aspectos, como autenticidad, confidencialidad, integridad y no repudio, que deben ser tratados en el proceso de transferencia de información electrónica pues son los cuatro pilares de una comunicación

segura, los cuales deben manejarse adecuadamente para evitar que se conviertan en problemas debilitadores de la seguridad en la comunicación.

- Autenticidad [3]. Todas las entidades participantes en una transacción deben estar perfecta y debidamente identificadas antes de comenzar la misma, para evitar la transferencia de datos confidenciales a una persona o entidad no deseada, que pueda hacer uso malintencionado de los mismos.
- Confidencialidad [3]. Los datos enviados en una comunicación no deben poder ser leídos por persona distinta al destinatario final, si ocurre esto, el espía no debe poder entender el mensaje enviado.
- Integridad [3]. Es necesario asegurar que los datos enviados en una comunicación lleguen sin modificaciones (íntegros), a su destino final. Lo cual significa que la información no ha sido alterada, borrada, reordenada, copiada, etc.
- No repudio [3]. Se debe asegurar que una vez enviado un mensaje con datos importantes o confidenciales el destinatario de los mismos no pueda negar haberlos recibido, o en el caso del emisor éste no pueda negar el envío.
- Virus en teléfonos móviles. Aunque se ha dado el caso de ataques y virus en PCs de mano, no puede decirse que el problema sea generalizado. Sin embargo, siempre existe la posibilidad de recibir un virus en un dispositivo móvil, pero en estos momentos, la probabilidad es mínima, aunque creciente.
- Bluejacking [7]. Es una forma simple de mandar mensajes con textos personalizados a cualquier dispositivo Bluetooth sin pedir permiso.
- Bluesnarfing [7]. Cuando un teléfono está en modo "visible" (o sea que otros dispositivos bluetooth lo pueden detectar), es posible conectarse al dispositivo sin que el usuario se entere, y tener acceso a datos del mismo como la agenda, el calendario, el IMEI (International Mobile Equipment Identity), lo cual permitiría clonar el teléfono, por ejemplo.
- Robo o pérdida de dispositivos. Consiste en que los dispositivos móviles pueden ser perdidos o robados fácilmente. Con sólo disponer de un identificador y contraseña válido el ladrón podría realizar compras, ya que es frecuente que en el propio móvil o el servidor remoto se almacenen perfiles de usuario con información sobre las tarjetas de crédito y otros datos de validación, a fin de evitar que el usuario se vea forzado a introducir una cantidad grande o considerable de datos.

3. TECNOLOGÍAS BASE

La implementación de medios seguros para la transferencia y acceso a información, se ha conseguido generalmente con el uso de sistemas basados en la criptografía, firmas digitales y certificados digitales, los cuales pueden combinarse para brindar mayor robustez a la seguridad, a continuación, se abordan brevemente las tecnologías utilizadas para la definición de Mercurio.

3.1. Criptografía

La criptografía provee un conjunto de técnicas para codificar mensajes de forma tal que éstos puedan ser almacenados y transmitidos en forma segura. Permite la transformación de textos legibles en secuencias de caracteres no legibles y viceversa, utilizando algoritmos y funciones matemáticas muy complejas.

Los algoritmos de cifrado actuales se agrupan en dos grandes clases: simétrico [9] y asimétrico [9], de acuerdo al proceso de cifrado y descifrado. Sin embargo, independientemente del tipo de cifrado utilizado, la seguridad en la criptografía depende de [8]: la longitud de la clave y que ésta sea realmente secreta (evidente), que el algoritmo no sea invertible, es decir si se conoce cómo funciona, no se pueda revertir el proceso sin la llave, que el algoritmo no tenga puertas traseras y no permita descifrar todo el texto si se conoce el contenido de una parte.

3.2. Infraestructura de Clave Pública (PKI)

Los esquemas basados en llaves públicas presentan un pequeño inconveniente: Se puede garantizar la autenticidad del emisor, solamente si se ha tratado antes con la persona que envía el mensaje, pues, en una primera comunicación no hay manera de comprobar sí quien está hablando es quien dice serlo. Para la solución de este inconveniente se creó la PKI [10], el cual es un sistema que entrega certificados digitales y llaves criptográficas, que permiten la seguridad en transacciones económicas, financieras y de intercambio de información sensible entre personas relativamente desconocidas.

3.3. Firma Digital

Desde un punto de vista estructural, la firma digital es una simple cadena o secuencia de caracteres que se adjunta al final del cuerpo del mensaje (por lo tanto, un mensaje firmado es completamente legible). Éste concepto está adquiriendo gran importancia en los actuales sistemas de autenticación, pues la firma electrónica a nivel legal constituye en la mayoría de los casos una prueba indudable de autoría, semejante a la firma tradicional de puño y letra, pues, tiene las mismas propiedades que la firma convencional; existe un método de firma, uno de verificación y es posible asociar un número único a cada persona o entidad [11].

3.4. Certificado Digital

Es un archivo electrónico que contiene un conjunto de datos formateados bajo el estándar X.509v3,

detallado en la RFC2459 [12], el cual vincula una clave pública con la identidad de una persona física o jurídica, de manera que se puede verificar que efectivamente ésta pertenece a quién dice poseerla, todo bajo el aval de una CA (Certification Authority). Los certificados digitales, debido a su naturaleza (estar basados en el uso de claves) y al papel que desempeñan, no son documentos imperecederos, es decir, tienen estipulado un periodo de validez (típicamente un año) en el cual deben ser renovados, además, pueden ser revocados anticipadamente en ciertos supuestos; por ejemplo, en el caso de que la clave privada, que debe permanecer secreta, haya pasado a ser conocida por terceras personas no autorizadas para usarla.

3.5. Single Sign-On (SSO)

Las aplicaciones móviles a menudo interactúan con múltiples servidores de aplicaciones, quienes brindan información idónea para desplegar en pantallas personalizadas sobre el dispositivo de usuario. Sin embargo, cada servidor puede tener sus propios protocolos de autenticación de usuarios, lo que constituye uno de los principales inconvenientes relativos a la usabilidad de las mismas ya que los usuarios deben autenticarse manualmente en cada uno de ellos [13]. Una de las formas de combatir este problema es mediante SSO, el cual se refiere al acceso a múltiples recursos o servicios por medio de un único proceso de ingreso [14]. El principal objetivo de una arquitectura que implementa SSO es transferir la funcionalidad y complejidad de todos los componentes de seguridad a un solo servicio, permitiendo que los usuarios realicen el proceso de ingreso al sistema una sola vez, a pesar de que continúen interactuando con múltiples componentes del mismo.

3.6 Seguridad en Servicios Web

Actualmente, el esquema de seguridad más común disponible para los Servicios Web es SSL (Secure Sockets Layer) sobre HTTP. A pesar de esta popularidad utilizar SSL con Servicios Web tiene algunas limitaciones. En primer lugar, SSL esta diseñado para proveer seguridad punto a punto, la cual no es adecuada para Servicios Web porque estos necesitan seguridad extremo a extremo, donde múltiples nodos intermediarios pueden existir entre dos puntos finales. En segundo lugar, SSL asegura comunicaciones a nivel de transporte y no a nivel de mensaje. Como resultado, los mensajes son protegidos únicamente mientras viajan sobre el medio de trasporte. Finalmente, SSL no proporciona una forma de firmar y cifrar partes de una comunicación; por ejemplo, si se tiene una larga orden de compra sobre un documento XML, sólo se requerirá firmar y cifrar el elemento tarjeta de crédito, y no todo el documento.

La TI (Technology Industry) ha definido diferentes esquemas de seguridad para subsanar los problemas mencionados, entre los cuales destacan: XML Digital Signature [15], que define un esquema en XML para la captura del resultado de una operación de firma digital aplicada a datos arbitrarios, XML Encryption [16] que proporciona seguridad “extremo a extremo” a las aplicaciones de intercambio de datos estructurados y WSS (WS-security) [17] el cual define un grupo estándar de extensiones SOAP (Simple Object Access Protocol), que pueden utilizarse para implantar la integridad y confidencialidad en las aplicaciones que hacen uso de Servicios Web.

4. MERCURIO

En busca de solucionar los problemas mencionados, se diseñó e implementó Mercurio, una plataforma para acceder de forma segura a servicios móviles, escalable y modular que puede ser replicable y utilizada en complemento con otras plataformas de servicios. Entre sus características más importantes se encuentran:

- Mercurio se centra en obtener un nivel medio de seguridad en los servicios móviles para lo cual abarca un conjunto conceptos básicos (autenticación, autorización, confidencialidad, integridad y no repudio), implementa SSO para la autenticación y autorización de los usuarios y tecnologías de lado del dispositivo móvil como cifrado asimétrico, firmas y certificados digitales y almacenamiento seguro con cifrado simétrico para la realización de transacciones seguras; todo ello, buscando realizar aportes en la definición de esquemas de seguridad que hagan viable el comercio móvil no solo a las grandes entidades comerciales sino también a las medianas y pequeñas.
- Mercurio, a diferencia de otras plataformas de prestación de servicios móviles seguros dedicadas a definir la seguridad en clientes móviles delgados (implementados generalmente en dispositivos móviles que no soportan un lenguaje de programación), trabaja fuertemente en la definición de los esquemas y tecnologías de seguridad disponibles en los clientes móviles gruesos (implementados en dispositivos móviles de alta gamma o de nueva generación que soportan al menos un lenguaje de programación) de forma tal que hace mejor uso de ellos al aprovechar al máximo esas capacidades. Además, al trabajar en conjunción con clientes Java ME cubre el más grande amplio rango del espectro de teléfonos móviles existentes en el mundo.
- Mercurio al basarse en Servicios Web realiza una implementación de SOA (Service Oriented Architecture) que le permite desacoplar sus

funcionalidades y ofrecerlas como servicios al interior o exterior de ella.

A continuación se describe la arquitectura de Mercurio desde diferentes vistas para lograr mayor detalle en la descripción y facilitar el entendimiento de la misma.

4.1. Arquitectura

4.1.1. Vista Modular

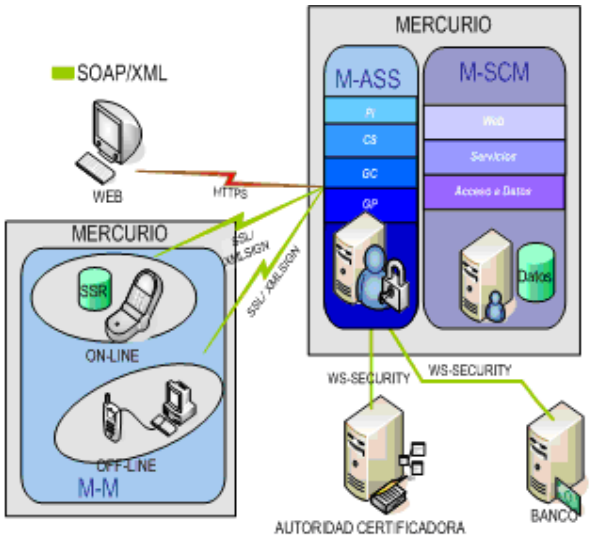


Figura 1: Arquitectura de Mercurio

La arquitectura de referencia de Mercurio se presenta en la Figura 1, en la cual se identifican claramente tres sistemas: el primero denominado M-ASS (Mercurio - Acceso Seguro a Servicios), orientado a proporcionar todos los mecanismos y políticas necesarias para garantizar el acceso seguro a los servicios de la plataforma de comercio móvil; el segundo llamado M-SCM (Mercurio Servicios de Comercio Móvil), encargado de proporcionar los servicios de prueba de la plataforma; y M-M (Mercurio - Móvil), aplicación móvil definida para utilizar los servicios de Mercurio, de forma segura y desde un dispositivo móvil conectado. A continuación se describe cada uno de los componentes mencionados.

4.1.1.1. M-ASS

Constituye el núcleo de seguridad de la plataforma Mercurio y se divide básicamente en 5 componentes, ver Figura 1.

- PI (Proveedor de Identidad). Encargado de gestionar la autenticación y autorización del usuario a los servicios de prueba de la plataforma, utilizando fuertes mecanismos de seguridad. Para ello, el PI basa su funcionamiento en la arquitectura SSO, que proporciona el intercambio de Tokens de seguridad basados en XML, permitiendo una mayor interoperabilidad. Además, tiene soporte para la verificación de certificados de usuario y firmas digitales.
- CS (Comunicaciones Seguras). Incluye acceso a Servicios Web sobre conexiones seguras para aquellos servicios que lo requieran.
- GC (Gestión de Certificados). La plataforma basa gran parte de sus funciones de seguridad en una infraestructura de clave publica, por lo cual es necesario la gestión de certificados y una entidad de confianza o CA. Este componente crea el par de llaves de usuario (pública y privada) y las peticiones de certificados que son enviadas a la CA para la obtención de los correspondientes certificados digitales.
- GP (Gestión de Pagos). Componente que permite realizar una conexión directa con una entidad bancaria, facilitando al cliente móvil realizar pagos en una transacción de comercio electrónico, sin intervenir en los datos críticos de usuario, garantizando cifrado asimétrico extremo a extremo.
- Entidades Externas. Son necesarias para el correcto funcionamiento de Mercurio.
  - Banco. Se utiliza cuando el servicio proporcionado por Mercurio involucra un sistema de pago. Utiliza un sistema de llaves públicas y comunicación segura sobre Servicios Web con el fin de garantizar que los datos de usuario, como el número de la tarjeta de crédito, sean legibles únicamente por la entidad bancaria.
  - La CA. Se encarga de proveer certificados de usuario tanto a los consumidores, como a los demás componentes de la plataforma que los requieran, además mantiene actualizadas las listas de revocación de certificados digitales, las cuales son indispensables para verificar la validez de una transacción.

4.1.1.2. M-M.

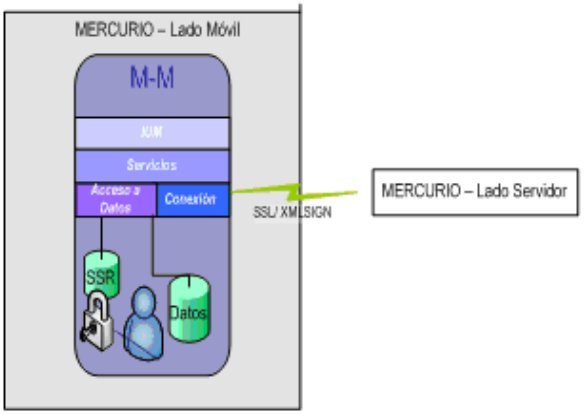


Figura 2: Arquitectura de M-M

Es el dispositivo móvil que sirve de mediador entre el usuario y el sistema, a través de interfaces gráficas.

Además, posee un componente para la gestión de seguridad en sus transacciones, el cual incluye, soporte para firma digital y cifrado de datos, y un SRS (Secure Record Store) para el almacenamiento de certificados y llaves necesarias con el fin de garantizar la autenticación, la autorización y el no repudio.

La Figura 2 ilustra la arquitectura de M-M, la cual se compone fundamentalmente de dos módulos, el dispositivo hardware y el agente software de usuario móvil que permite utilizar los servicios de Mercurio. El componente software esta constituido por:

- IUM (Interfaz de Usuario Móvil). Como su nombre lo indica, constituye la capa de aplicación más cercana al usuario del dispositivo móvil. En Mercurio, se encarga de la presentación de todos los elementos gráficos necesarios para el uso de los servicios de comercio móvil implementados en M-SCM. Constituye la Vista de un patrón MV implementado con Java ME.
- Servicios. Representa tanto al Control (de lógica de negocio y navegación entre las pantallas de la IUM) como al Modelo (lógica de negocio) del patrón MV mencionado en el ítem anterior. En este caso, se encarga de prestar todos los servicios de comercio móvil con acceso On-Line, a través de la capa de conexión. Además, soporta la gestión de seguridad en las transacciones, incluyendo soporte para firma digital y cifrado de información.
- Acceso a Datos. Es el componente Software necesario para gestionar tanto datos confidenciales (número de tarjeta de crédito, clave, certificados y llaves digitales necesarias para garantizar la autenticación, la autorización y el no repudio), como datos comunes (no confidenciales – logo de la aplicación, dirección del servidor de Mercurio….), almacenados en el dispositivo móvil.
- Conexión. Permite utilizar servicios de Mercurio On-Line a partir del uso de conexiones seguras basadas en SSL y XML Digital Signature.

4.1.1.3. M-SCM

Provee los servicios de la plataforma, es decir, se encarga de entregar contenidos al servicio móvil, únicamente si el proceso de autenticación y autorización del cliente con el M-ASS ha sido exitoso. La Figura 1 ilustra la arquitectura general de M-SCM. A continuación se describe cada uno de sus componentes arquitectónicos.

- Capa de Datos. Se encarga de realizar todos los procesos de gestión y comunicación con los datos persistentes y esta constituida por dos módulos, el primero correspondiente al sistema de almacenamiento de datos, compuesto por dos bases de datos fundamentales para el sistema. La BD_Global, que almacena los datos de la lógica del negocio y la BD_Administrativa que almacena los datos relacionados con la gestión y seguridad de la plataforma; el segundo módulo corresponde a la entidad de precios, que permite la obtención de valores de referencia para los productos que se ingresan a Mercurio. Esta entidad no es implementada en este proyecto, pero si entra a formar parte de la capa de datos de Mercurio en forma de fuente de datos externa.
- Capa de Acceso a Datos. Como su nombre lo indica, permite acceder a la capa de datos para que estos sean utilizados desde la capa de servicios de la plataforma de comercio móvil. Esta capa se considera como la base que da soporte a la lógica del negocio, ya que sobre ella se construyen todos los servicios que Mercurio presta. El acceso a datos en Mercurio se realiza a través del Mapeo Objeto Relacional (MOR) de la capa de datos, lo cual permite que el sistema sea altamente portable y fácilmente adaptable a cualquier cambio de repositorio de datos, ya que el mismo es independiente del DBMS (Data Base Management System), permitiendo manipular las tablas de una base de datos bajo el concepto de orientación a objetos.
- Capa de Servicios. Se encarga de gestionar toda la lógica de negocio de los diferentes servicios y soportar todas las transacciones desde y hacia el cliente Web y/o Móvil, dando soporte a aplicaciones On-Line. La definición de los servicios implementados en Mercurio se describen brevemente en el apartado 4.2.2.3. Sin embargo, desde ya es importante mencionar que la implementación de esta capa utiliza los conceptos de programación basada en componentes y patrones de diseño Java, que permiten a Mercurio ser portable y escalable.
- Capa Web. Es responsable de brindar a los usuarios acceso a los servicios vía Web (utilizando un navegador), es decir, es la capa que tiene una relación directa con el usuario atendiendo sus diferentes peticiones y transacciones, brindando un acceso práctico, rápido y seguro a las aplicaciones. Permitiendo entre otras cosas, realizar procesos de gestión y actualización.

4.1.2. Vista de Diseño

4.1.2.1. M-ASS

La Figura 3 ilustra los componentes de diseño de M-ASS, a continuación se describen brevemente.

- Nivel de Aplicación
  - Cliente Web. Representa el navegador a partir del cual se tiene acceso la aplicación de administración de seguridad de M-ASS.
  - Acceso a Servicios. Contiene las clases necesarias para implementar el acceso seguro a los servicios de Mercurio, a través de la implementación de SSO. La integridad del token de seguridad que

permite la validación única del usuario utiliza XML Digital Signature.

- Seguridad ASS. Componente que garantiza la seguridad para el manejo, almacenamiento y transmisión de información desde y hacia los clientes Web o móviles. Por lo tanto, implementa el CS, GP, y el GC, utilizando WSS, XML Encryption y PKI.
- M/SCM. Representa los servicios de prueba de la plataforma.
- M/M. Representa los clientes móviles de la plataforma.

▪ Nivel de Mediación de Seguridad

- JWSDP (Java Web Service Developer Package). Herramienta integrada para construir, probar y desplegar Servicios Web seguros compatibles con la especificación WSA (Web Service API) de J2ME.
- XWS-Security. Librería que contiene las clases de implementación de la especificación WSS.
- XStream. Librería que contiene las clases necesarias para serializar objetos a XML y viceversa.
- TSIK. API Java que contiene las clases que permiten simplificar el desarrollo de aplicaciones confiables. Soporta XML Signature y XML Encryption.
- JCE. Conjunto de paquetes que proporcionan un framework para la generación de llaves, cifrado simétrico y asimétrico, entre otros.

▪ Nivel de Servicios de Red

- SOAP. Protocolo bajo el auspicio de la W3C (World Wide Web Consortium). Utilizado por los Servicios Web para el intercambio de datos XML.
- SSL. Protocolo desarrollado por Netscape, que se convirtió en el método elegido para asegurar transmisiones de datos por Internet bajo una estructura Cliente Servidor.
- WS-Security. Es uno de los primeros estándares de Servicios Web desarrollado por OASIS (Organization for the Advancement of Structured Information Standards) para soportar, integrar y unificar múltiples modelos, mecanismos y tecnologías de seguridad.
- JBoss Server. Servidor de aplicaciones Java; contiene la parte servidora de Mercurio, agrupando los EJB de lógica de negocio y todas la interfaces JSP y clases de lógica de presentación (Action y Forms) de acceso Web a los servicios de administración de Mercurio
- J2EEDK. Representa la API J2EE (Java 2 Enterprise Edition).
- Firewall. Elemento software utilizado en las redes para prevenir algunos tipos de comunicaciones prohibidas por las políticas de seguridad.

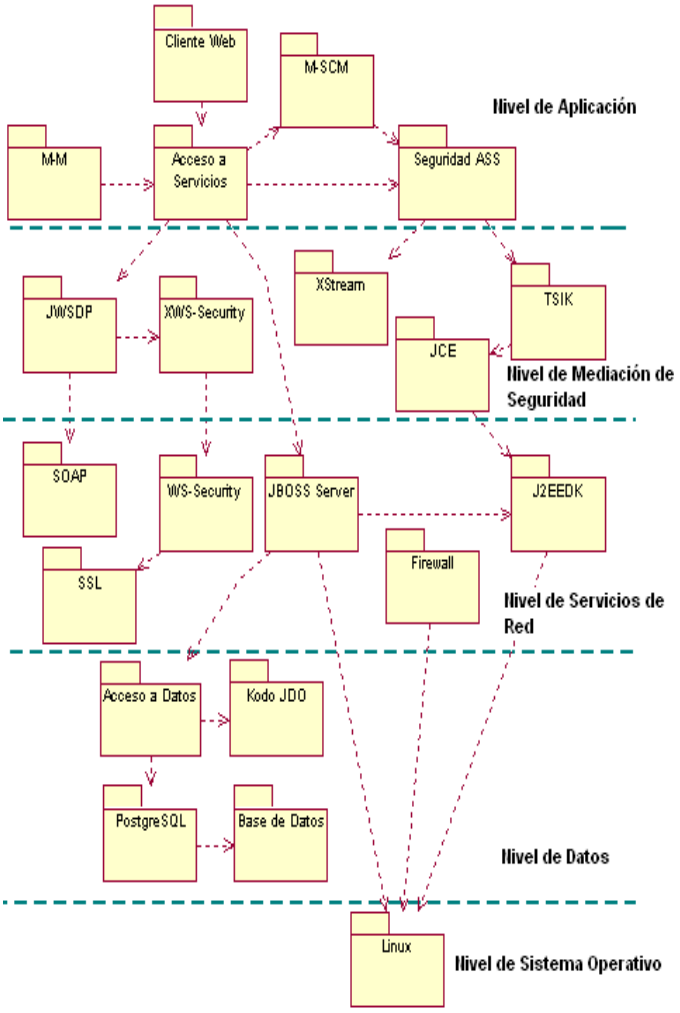


Figura 3: M-ASS / Diagrama de paquetes

▪ Nivel de Datos

- Acceso a Datos. Agrupa las clases que permiten la interacción del sistema con los datos persistentes almacenados en la BD_Administrativa, realizando un mapeo objeto-relacional basado en JDO (Java Data Object).
- Kodo JDO. Es una implementación de la especificación JDO, creada por Sun Microsystems para la persistencia transparente de objetos Java sobre cualquier base de datos transaccional.
- PostgreSQL Es el sistema gestor de base de datos utilizado para realizar la persistencia de los datos manejados por Mercurio.
- Base Datos. Representa la BD_Administrativa que almacena la información de los aspectos de gestión y seguridad requeridos por Mercurio.

▪ Nivel de Sistema Operativo

- Linux. Sistema operativo base para el despliegue de los diferentes módulos de la plataforma.

4.1.2.2. M-M

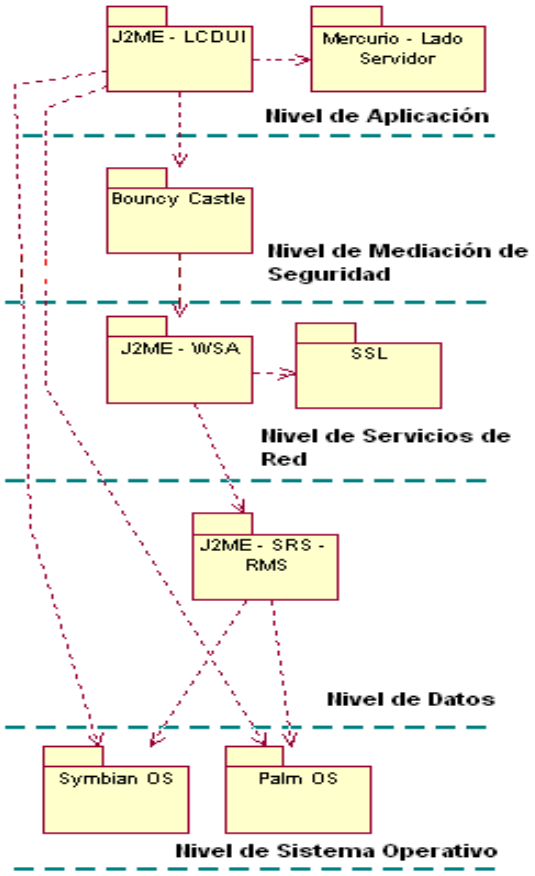


Figura 4: M-M / Diagrama de paquetes

La Figura 4 ilustra los componentes de diseño de M-M, a continuación se describen brevemente.

- Nivel de Aplicación
  - J2ME/LCDUI (Liquid Crystal Display User Interface). Contiene las clases correspondientes a la lógica de negocio, el control y la IUM, utilizando el patrón MV. Su implantación se realizó utilizando MIDP (Mobile Information Device Profile) 2.0.
  - Mercurio Lado Servidor. Representa tanto los servicios de seguridad como los de usuario final proporcionados por Mercurio.
- Nivel de Mediación de Seguridad
  - Bouncy Castle (BC). Contiene las clases que implementan un proveedor JCE, para operaciones de firma y cifrado de datos. Para ello utiliza la API BC [18] para J2ME.
- Nivel de Servicios de Red
  - J2ME/WSA. Contiene las clases que permiten consumir servicios Web desde un dispositivo móvil que soporta J2ME [19].
  - SSL. Descrito en el ítem anterior.
- Nivel de Datos
  - J2ME/SRS/RMS. Representa la BD_Móvil y las clases que permiten crear en J2ME repositorios de datos seguros o convencionales, basados en SRS o RMS respectivamente.
- Nivel de Sistema Operativo
  - Symbian OS. Sistema operativo líder en el segmento de la telefonía móvil.
  - Palm OS. Sistema operativo para PDAs (Personal Digital Assitant).

4.1.2.3. M-SCM

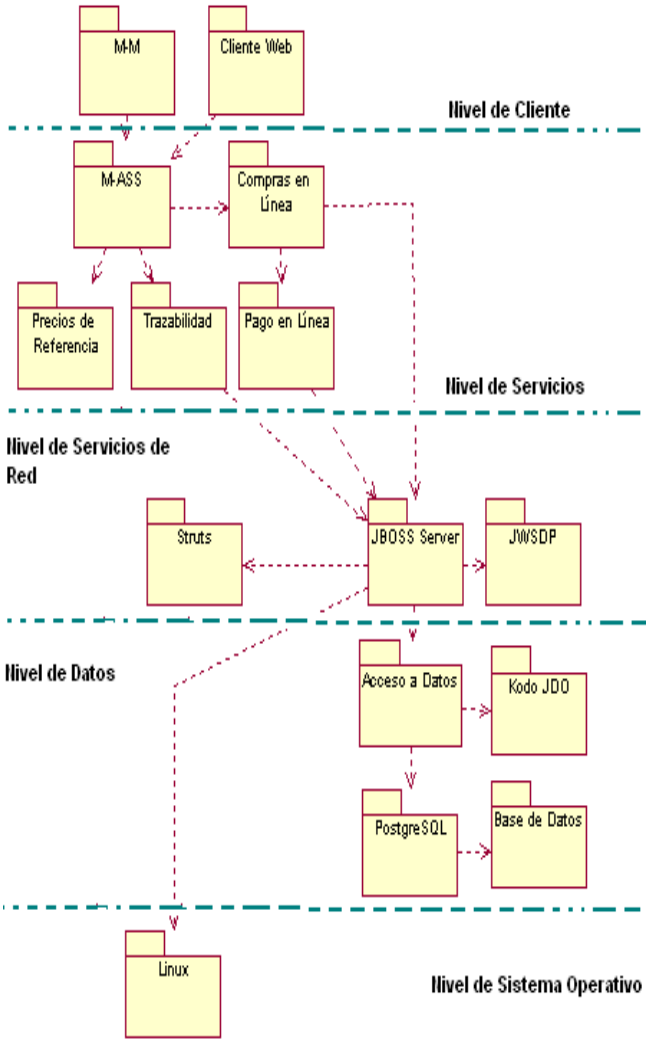


Figura 5: M-SCM / Diagrama de paquetes

La Figura 5 ilustra los componentes de diseño de M-SCM, a continuación se describen brevemente.

- Nivel de Cliente (Web/Móvil)
  - Cliente Web. Representa al usuario Web de la plataforma Mercurio, que básicamente realizará sobre ella procesos administrativos.

- M/M. Representa al cliente móvil que utiliza los servicios de Mercurio.

▪ Nivel de Servicios
- M/ASS. Representa el servicio de seguridad implementado en Mercurio. De esta forma, se abre la posibilidad para que este sea utilizado por otras plataformas de servicios orientados al comercio móvil.
- Compras en Línea. Contiene las clases que permiten gestionar las peticiones de compra de los clientes móviles.
- Pagos en línea. Agrupa las clases correspondientes para realizar transacciones bancarias.
- Trazabilidad. Es el servicio que permite a un usuario móvil obtener el historial del producto que va a comprar, el cual incluye el proceso completo de fabricación del mismo.
- Precios de Referencia. Servicio externo utilizado para obtener valores de referencia para los productos ofrecidos a través de Mercurio.

▪ Nivel de Servicios de Red
- JBOSS Server y JWSDP. Descritos con anterioridad
- Struts. Para lograr una arquitectura escalable basada en el desacople de las capas, Mercurio utiliza el patrón MVC (Model – View - Controller) implementado en este WAF (Web Application Framework). Este proyecto utiliza Struts para realizar la aplicación de administración de los servicios prestados por Mercurio

▪ Nivel de Datos
- Acceso a Datos. Agrupa las clases que permiten la interacción del sistema con los datos persistentes almacenados en la BD_Global.
- Kodo JDO y PostgreSQL. Descritos con anterioridad.
- Base Datos. Representa la BD_Global que almacena la información necesaria para prestar los diferentes servicios de Mercurio.

▪ Nivel de Sistema Operativo
- Linux. Descrito con anterioridad.

4.2. Servicio de Prueba

Con el fin de validar las funcionalidades y aportes de Mercurio, se implementó el servicio Compra en Línea, habilitado para un dispositivo móvil con soporte Java ME lo cual requiere el acceso a algunas de las funcionalidades proporcionadas por Mercurio como el catálogo de productos y carrito de compras. Este servicio se escogió como elemento de validación porque requiere características de seguridad de nivel medio y una forma de pago en transacciones comerciales. La funcionalidad del servicio de comercio móvil radica en que el consumidor haciendo uso de su tarjeta de crédito paga en línea productos que desea adquirir, y Mercurio se encarga de proveer seguridad extremo a extremo.

En la figura 6 se presenta un esquema paso a paso del servicio implementado para validación de Mercurio, donde cada número y flecha indica la acción ejecutada al hacer uso completo del servicio de comercio móvil. Sin embargo, antes de utilizar el servicio es necesario realizar el proceso de ingreso a la plataforma.

▪ Proceso de Ingreso.
1. El usuario ingresa sus datos de acceso (login y password) y selecciona la opción validar.
2. El sistema cifra el login y password utilizando BC.
3. El sistema envía la información por un canal de seguridad (construido con SSL+XML Digital Signature).
4. El sistema verifica que la información de acceso suministrada por el usuario, corresponda con los datos almacenados en la BD_Administrativa.
5. El sistema busca en la BD_Administrativa y obtiene información sobre los permisos que tiene el usuario para acceder a los servicios disponibles en Mercurio, las políticas de seguridad del sistema, el tiempo de caducidad establecido para cada servicio de Mercurio, cuando el mismo está inactivo.
6. El sistema arma un Token de Seguridad con los permisos y el tiempo de caducidad obtenidos y adiciona al Token un tiempo límite de permiso al usuario para acceder a los servicios.
7. El sistema adiciona al Token la fecha y hora en la que el usuario solicita autenticación.
8. El sistema firma digitalmente el Token de seguridad con la llave privada del Proveedor de Identidad.
9. El sistema devuelve el Token de seguridad firmado digitalmente. En este instante Mercurio da por autenticado al usuario.
10. El usuario autenticado selecciona el servicio al que desea ingresar.
11. El sistema valida el token de seguridad del usuario autenticado, comprobando su integridad y posteriormente su autenticidad, verificando que la firma que trae el Token de seguridad coincida con la firma digital del Proveedor de Identidad.
12. Mercurio obtiene el resultado de validación. En este momento el usuario es autorizado a utilizar el servicio elegido.
13. El sistema presenta las opciones del servicio solicitado. En este caso el catálogo de los productos disponibles.

▪ Uso del Servicio.
1. Ver catálogo, línea y productos.

2. Adicionar y eliminar productos al carro de compras.
3. Buscar elementos de seguridad en el dispositivo móvil.
4. Obtener elementos de seguridad. Son necesarios el certificado digital del cliente, su llave privada y el certificado digital de la entidad bancaria.
5. Crear llaves y CSR (Certificate Signing Request) de usuario.
6. Solicitar elementos de seguridad.
7. Devolver elementos de seguridad.
8. Entrega elementos de seguridad.
9. Cifrar número de tarjeta de crédito. El uso de los elementos de seguridad mencionados en el ítem 15 y el cifrado de la información de la tarjeta de crédito permite garantizar que esos datos únicamente son conocidos por el consumidor y la entidad bancaria durante todo el proceso de la transacción (confidencialidad), sin importar que los mismos pasen por Mercurio; garantizando seguridad extremo a extremo.
10. Comprar.
11. Transferir fondos.
12. Devolver comprobante.
13. Pasar el comprobante.
14. Firmar el recibo. Para terminar con el proceso de venta, tanto el consumidor como el vendedor reciben un comprobante de transacción firmado digitalmente (integridad), con el cual ninguno de los dos implicados puede negar haber intervenido en la transacción comercial (no repudio).
15. Almacenar recibo firmado.
16. Nota: Todo el intercambio de información se realiza a través de canales seguros.

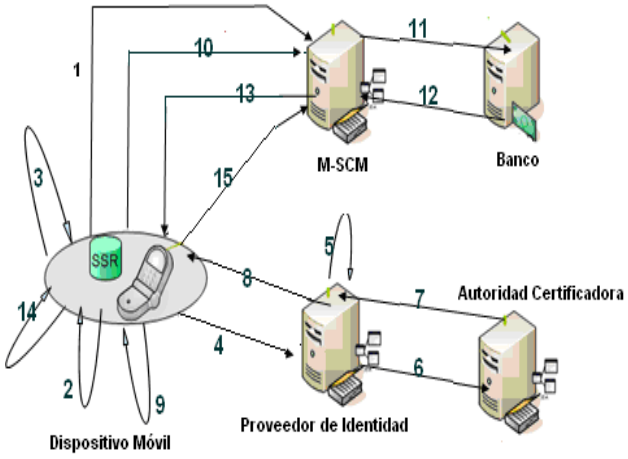


Figura 6: Servicio de validación de Mercurio

5. CONCLUSIONES

- La seguridad en servicios móviles es un tema nuevo, que en el mundo está tomando gran importancia. Justamente por ello, existen muchos aspectos por definir.
- Mercurio ataca todos los pilares fundamentales de la seguridad, lo cual permite crear un modelo de protección extremo a extremo.
- La definición modular de Mercurio y la implementación del M-ASS como un servicio de la misma permiten que el esquema de seguridad definido sea replicable o utilizado en conjunción con otras plataformas de servicios móviles.
- Mercurio garantiza seguridad de nivel medio para incrementarlo sería necesario, en primera instancia, cambiar la forma de implementación de la BD-Movil; pasándola de SRS/RMS a un firmware tipo SIM/USIM o Smartcards.
- Mercurio utiliza BC para la implementación del JCE en detrimento de SATSA (Security and Trust Service API) debido a que en el mercado existen muy pocos celulares que soporten esta especificación. Sin embargo, dada la modularidad de la implementación de M-M, su sistema de conexión puede ser enriquecido con un módulo de soporte a SATSA, de ser necesario.
- La creación de una plataforma como la planteada permitirá incrementar la confianza en los sistemas de comercio móvil, facilitando la incursión de más personas a estos nuevos servicios que marcan un paso importante en la evolución de la sociedad moderna.
- Como un trabajo futuro se plantea la necesidad de representar las políticas de seguridad tanto en el cliente móvil como en el servidor con estándares como XACML (eXtensible Access Control Markup Language) [20].

6. REFERENCIAS

[1] F. J. Chamorro y Otros. "Arquitectura de Internet Móvil". Telefónica Investigación y Desarrollo. Marzo de 2001. Disponible: http://lauca.usach.cl/~lsanchez/comprimido/arqmovil.zip.

[2] McGraw, G, et.al. (2002). The Weakest Link, "Software security principles" http://www-128.ibm.com/developerworks/security/library/s-link.html.

[3] L. Moreno. "Transacciones Seguras". Departamento de Diseño Web de BJS Software. Disponible: http://www.htmlweb.net/seguridad/ssl/ssl_4.html

[4] EumedNet. "Criptografía de clave asimétrica". Universidad de Málaga. Disponible: http://www.eumed.net/cursecon/ecoinet/seguridad/asimetrica.htm

[5] Texas Instruments. "Reducing the Security Threats to 2.5G and 3G Wireless Applications". Enero de 2002. Disponible: http://focus.ti.com/pdfs/vf/wireless/securitywhitepaper.pdf.

[6] Belt Ibérica. "Los Sistemas Informáticos Móviles, un Peligro para la Seguridad Corporativa". Enero de 2005. Disponible: http://www.belt.es/noticias/2005/enero/13/sist_informaticos.htm.

[7] Überbin I/A. "Seguridad en Bluetooth". Abril de 2004. Disponible:

http://mobile.uberbin.net/archivos/seguridad/seguridad-en-bluetooth.php.

[8] Universidad de Navarra. “Nociones sobre criptografía y firma digital”. Disponible: http://www.unav.es/SI/servicios/seguridad/certifica8.html

[9] E. Ortiz. “Certificados Digitales”. Noviembre de 2003. Disponible: http://www2.sharesafe.net/sharesafe/TuturialPKI2.asp?men2=none

[10] R. Gonzáles. “PKI y Software Libre”. 2005. Disponible: http://www.criptored.upm.es/descarga/PKI_SoftwareLibre.zip.

[11] J. R. Aguire. “Seguridad Informática y Criptografía”. Universidad Politécnica de Madrid. Marzo de 2005. Disponible: http://www.criptored.upm.es/guiateoria/gt_m001a.htm

[12] RFC 2459. “Internet X.509 Public Key Infraestructure Certificate and CLR Profile”. IETF. 1999. Disponible: www.ietf.org/rfc/rfc2459.txt

[13] Consejo Superior de Investigación Científica. “Transacciones Electrónicas Seguras”. Diciembre de 2000. Disponible: http://www.iec.csic.es/criptonomicon/comercio/set.html.

[14] M. Juntao. Yuan. “Data Security in Mobile Java Applications”. Diciembre de 2002. Disponible: www.javaworld.com/javaworld/jw-12-2002/jw-1220-wireless.html

[15] W3C. “W3C Recommendation: XML-Signature Syntax and Processing”. Febrero de 2002. Disponible: http://www.w3.org/TR/xmldsig-core/

[16] W3C. “W3C Recommendation: XML Encryption Syntax and Processing”. Diciembre de 2002. Disponible: http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/Overview

[17] OASIS. “OASIS Standard: Web Services Security (WS-Security)”. Marzo de 2004. Disponible: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[18] Bouncy Castle. “Bouncy Castle API”. Septiembre de 2005. Disponible: http://www.bouncycastle.org/documentation.html

[19] O.M. Caicedo, F.O. Martinez, M.J.Gómez, J.A Hurtado. "Architectures for Web Services Access from Mobile Devices".La Web2005. Printed by IEEE press. Buenos Aires, Noviembre de 2005.

[20] OASIS. “OASIS Standard: eXtensible Access Control Markup Language”. Febrero de 2005. Disponible: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#XACML20.

Anexo I

Especificación de la plataforma SEUS

Propuesta de arquitectura para facturación y pago por proximidad de servicios ubicuos en el contexto colombiano

Milton Ausecha Penagos
mausecha@unicauca.edu.co

Javier Imbús Guzmán
jimbus@unicauca.edu.co

Zeida Solarte
zsolarte@unicauca.edu.co

Oscar Caicedo
omcaicedo@unicauca.edu.co

Grupo de Ingeniería Telemática, Universidad del Cauca, Popayán, Colombia

Fecha de recepción: 19-10-2007 Fecha de selección: 18-04-2008 Fecha de aceptación: 15-01-2008

ABSTRACT

Ubiquitous services pretend to interact proactively with the user, proposing solutions to some problems. Those services must to be billed in a secure and efficient form in order to guarantee confidence between customer new services tha fulfill the customer's expectation. For this reason, ubiquitous computing is a new research area that looks for the provisión of services in a transparent way for the user in mobile environments. However, ubiquitous services have the problem of a non-existent protocol that fulfils all the mobility, identity and context requirements of such services. This paper describes the development of a ubiquitous services pilot that pretends to fulfill such requirements.

KEY WORDS

Ubiquitous services, Service discovery, Bluetooh, mobility, web services.

RESUMEN

Se busca que los nuevos servicios lleguen al usuario en cualquier lugar y hora, de forma transparente y brindando la posibilidad de acceder a los mismos mediante diferentes dispositivos y tecnologías de acceso. Los servicios ubicuos buscan adelantarse a las acciones del usuario para no solo esperar su intervención, sino también para proponerle soluciones a sus problemas y ayudarlo proactivamente con sus tareas. Estos servicios deben ser facturados de forma segura y eficiente para que los establecimien-

tos comerciales puedan brindar un servicio con la certeza de no perder capital debido a fallas en los procesos de facturación y el cliente pueda confiar en que se le facturará únicamente la cantidad correspondiente a los productos y servicios adquiridos, por esto se hace necesario proponer una arquitectura que permita garantizar las operaciones de pago y facturación brindando un alto nivel de seguridad que repercuta en la confianza del usuario y contribuya de forma considerable a la masificación del servicio. Por ello el grupo WapColombia y el GIT (Grupo de Ingeniería Telemática) de la Universidad del Cauca proponen una arquitectura que se adapta a las condiciones del contexto colombiano, para de esta forma contribuir en la construcción de una realidad, que cada día es más cercana gracias a la llegada al país de nuevas tecnologías, como NFC y RFID, que en conjunto con otras ya existentes como certificados digitales, algoritmos de cifrado, entre otros, permitirán innovar para garantizar seguridad en las transacciones financieras.

PALABRAS CLAVE

Computación ubicua, RFID, NFC, entidad certificadora, arquitectura, encriptación.

Clasificación Colciencias: Tipo 1

1. INTRODUCCIÓN

En la actualidad el concepto de computación ubicua es muy popular en países como Japón y Suiza, y día a día despierta mayor interés en los profesionales de las telecomunicaciones de todo el mundo.[1] El término Ubicuidad en el entorno de las telecomunicaciones fue definido por Mark Weiser, quien lo acuñó por primera vez en su artículo titulado: "El computador para el siglo 21" en el Scientific American Ubicomp en 1991,[2] en este artículo se proyectaba un ambiente dotado de un conjunto de recursos, los cuales brindarían no solo capacidades de comunicación a los usuarios mediante el acceso a múltiples dispositivos en cualquier lugar y a cualquier hora, sino que además lo harían de forma transparente y personalizada, gracias a un conocimiento previo del contexto de usuario.[1]

De otro lado, en los últimos veinte años el comercio electrónico móvil ha crecido explosivamente gracias al desarrollo de Internet, tecnologías de comunicaciones inalámbricas y dispositivos móviles, desempeñando un papel cada día más importante en nuestras vidas.[3] Al mismo tiempo la tecnología RFID (Radio Frequency IDentification), en donde un lector se comunica con una etiqueta por medio de radio frecuencia, ha despertado gran interés tanto en la industria como en la academia,[3] por lo cual en un futuro cercano todos los servicios de datos móviles estarán integrados a tecnologías de este tipo, y si además se considera que los servicios de datos contribuirán con el 70%-80% de las ganancias de los Proveedores de Servicios de Telecomunicaciones, se tiene que el futuro de los servicios basados en RFID es prometedor, especialmente el Servicio de Pago Móvil, que será la principal aplicación de RFID en el área de las telecomunicaciones.[4]

Aunque el pago desde dispositivos móviles es un campo incipiente a nivel nacional, con la llegada de nuevas tecnologías al país, entidades financieras, comerciantes y usuarios empiezan a manifestar interés por estas nuevas formas de pago, con lo cual entes reguladores como el Ministerio de Comunicaciones afrontan retos legislativos, que ya se empiezan a abordar con la creación del Decreto 2870 de 2007, que tiene en cuenta las diferentes áreas donde es posible una convergencia: los servicios, equipos terminales, redes o medios de transmisión, y mercados. Este decreto de convergencia abre el camino para que personas o empresas que no cuentan con grandes recursos económicos, participen en el mercado de las telecomunicaciones, el cual cada vez es más dinámico y competitivo, especialmente en el área de los servicios.

Los servicios de pago móvil hacen parte de la convergencia descrita anteriormente, por lo cual el planteamiento de una arquitectura para un sistema de pago que garantice transacciones seguras se convierte en una necesidad para la sociedad colombiana, la cual debe ser abordada teniendo en cuenta las limitaciones tecnológicas, de regulación y los retos sociales que implica el desarrollo de una aplicación de este tipo en el contexto nacional, además ya se han sentado en el país las bases legales para el uso de la factura digital con el decreto 1929 que la define como un documento equivalente a la factura

física en papel que soporta las transacciones de bienes y servicios y que para efectos fiscales su expedición, entrega, aceptación, conservación y exhibición, debe hacerse en un formato electrónico de conservación y la tecnología de información autorizados.[5] Por estas razones el Grupo de Ingeniería Telemática de la Universidad del Cauca está llevando a cabo los primeros pilotos como un acercamiento a una implementación real, y prepara gradualmente a los usuarios para la llegada de las nuevas tecnologías móviles.

El presente documento está estructurado de la siguiente manera, inicialmente se presenta un Marco Conceptual donde se explican las características de los servicios ubicuos y se definen las propiedades de un sistema de pago en un entorno ubicuo. Luego se presenta la arquitectura propuesta y se describe el proceso de pago llevado a cabo por un cliente al hacer uso del sistema; después se especifica la función de cada una de las entidades, que hacen parte de la arquitectura por medio de un caso de estudio y se compara frente a otras formas de pago implementadas en el mundo.

Finalmente se presentan las conclusiones y trabajos futuros que surgen al proponer esta arquitectura.

2. MARCO CONCEPTUAL

A. Servicios ubicuos

La esencia de la computación ubicua es la creación de ambientes saturados con capacidades de computación y comunicación, que se integran a la vida de las personas. Este entorno implica retos, algunos de los cuales ya han sido afrontados en el proceso de maduración que han sufrido sus dos antecesores: Los Sistemas Distribuidos y la Computación Móvil.[6]

Las características más relevantes de los sistemas distribuidos que se retoman en la computación ubicua son:[6] comunicación remota, tolerancia a fallos, alta disponibilidad, acceso a información remota y seguridad. Mientras que de la computación móvil se retoman características como: sistemas de redes móviles, acceso móvil a información, gestión adaptativa de recursos y sensibilidad de la locación.

Sin embargo, con la computación ubicua aparecen retos nuevos, como uso efectivo de espacios inteligentes, invisibilidad, escalabilidad localizada y enmascaramiento de desigualdades tecnológicas del entorno.[6]

Los servicios de comunicación en la vida diaria llegarán a ser más personalizados, y las capacidades que hagan uso del contexto y de las preferencias del usuario serán cada vez más importantes. Por esto se hacen necesarias herramientas y entidades en la red que permitan un fácil acceso a los servicios, un rendimiento y funcionamiento óptimos, pero sobre todo que garanticen comunicaciones seguras y confiables.[7] Desde este punto de vista se puede dar una definición más precisa de los servicios ubicuos:[7] Un entorno ubicuo comprende una infraestructura de red y un conjunto de recursos, que brindan capacidades de comunicación a los usuarios mediante el acceso a múltiples dispositivos en cualquier lugar y a cualquier hora, de forma transparente y personalizada, gracias a un conocimiento previo del contexto

de usuario, garantizando siempre la seguridad y confiabilidad de la comunicación, tanto a los clientes como al proveedor del servicio.

B. Servicio de pago en entornos ubicuos

En el modelo de comercio móvil clásico, la gente accede a internet usando su dispositivo móvil para seleccionar sus productos y ordenarlos en línea.

Pero muchos pagos no se efectúan de esta forma. En lugar de buscar la información del producto en internet antes de su elección, las personas interactúan con su ambiente o el medio que las rodea.[3] Normalmente cuando una persona desea obtener información de los productos cercanos, incluyendo información detallada del artículo y del vendedor, recurre a un conocido o a un aviso publicitario, con lo cual puede obtener los siguientes resultados: la persona no obtiene la información que necesita, así que recurre a otros medios como internet; otras personas le pueden brindar información sobre el producto, pero ésta no es útil ya que puede estar desactualizada; y finalmente, algunas personas le pueden brindar información útil acerca de un producto, pero ésta no es determinante para saber si está tomando la mejor opción.[3] Pero la selección del artículo es solo el primer paso para su compra, que inicia el siguiente proceso:

- Selección del artículo en la página Web
- Adición del artículo a la cuenta del usuario, si no se tiene se crea una con la información del comprador.
- Si la cuenta no tiene el respaldo de una tarjeta de crédito o cuenta bancaria, el usuario debe consignar el valor del artículo en la cuenta del vendedor.
- Si la cuenta tiene el respaldo de una tarjeta de crédito o cuenta bancaria, el usuario debe autorizar el traspaso de dinero a la cuenta del vendedor.
- El usuario recibe una confirmación de la transacción, que es generalmente una factura enviada a su casa.

En un entorno ubicuo, una vez seleccionado el producto, se realiza su pago y facturación, al igual que en cualquier transacción, pero con algunas características especiales:

- Se crea una cuenta de usuario con los datos y gustos personales.
- El sistema le informa al usuario la existencia de un artículo de su interés, por medio de una tecnología inalámbrica.
- El usuario observa los detalles del artículo desde su teléfono móvil y lo adiciona a su cuenta.
- El usuario realiza el pago por medio de una tecnología de campo cercano.
- El usuario recibe una confirmación de la transacción por medio de un mensaje de texto, en su teléfono móvil, y la factura en su correo electrónico.

Las interacciones de computación ubicua son típicamente espontáneas y de breve duración, con la posibilidad de ser iniciadas sin intervención de los usuarios e involucrando numerosos servicios dispersos, geográficamente confiables y no confiables. De acuerdo con las características anteriores se identifican los siguien-

tes requerimientos en un sistema de pago de computación ubicua: espontaneidad, eficiencia, seguridad, privacidad, flexibilidad, usabilidad y despliegue.[8]

Espontaneidad: La espontaneidad es una característica inherente y deseable de las interacciones ubicuas, que establece que los sistemas de computación ubicua deben ser diseñados con la suposición de que un grupo de usuarios participantes es altamente dinámico e impredecible. En términos de sistemas de pago, esto significa que es altamente improbable que los individuos entren en relaciones de larga duración con los diferentes proveedores de servicio que puedan encontrar.[8]

Eficiencia: Cuando se refiere a pagos que impliquen transacciones de medio o alto valor, la eficiencia de los sistemas de pago de computación ubicua está relacionada con la confianza que debe existir entre los usuarios y los proveedores de servicio. Cuando muchos pagos pequeños están implicados, es importante que el proceso del pago sea ligero y eficiente; caracterizado por la baja transferencia de datos y bajos costos tanto en lo computacional como en lo financiero.[8]

Seguridad: Claramente la seguridad es una consideración importante en cualquier sistema de pago y debe ser utilizada para evitar fraudes tales como hurto, falsificación de dinero, y evasión del pago. Los ambientes de computación ubicua, que brindan servicios a cualquier hora y en cualquier lugar, son más vulnerables a violaciones de seguridad que ambientes controlados que pueden ser asegurados físicamente. Los problemas de seguridad también se presentan debido a la falta de establecimiento de relaciones de confianza entre los usuarios finales y los proveedores de servicio.[8]

Privacidad: Muchos sistemas de pago como las tarjetas de crédito y las transferencias de dinero requieren que los usuarios faciliten información personal como nombre y números de cuenta durante la transacción. Se puede dar el caso en el que los usuarios no desean divulgar esta información pero necesitan pagar por los servicios. Además, sin los mecanismos adecuados para proteger la privacidad, la información de pago podría ser combinada con información del contexto que proporciona detalles de las actividades de los usuarios.

Flexibilidad: Los sistemas ubicuos deben seguir el principio de la volatilidad y no deben asumir ninguna configuración específica de red, de dispositivos y/o de usuarios, con lo cual se pueden identificar dos casos especiales: operación desconectada e indisponibilidad del dispositivo. La operación desconectada es un modo de operación que permite a los clientes continuar accediendo a servicios durante fallas temporales de un depósito de datos compartido o de una conexión de red. En el segundo caso, es importante que las interacciones de los usuarios con los ambientes de computación ubicua no dependan exclusivamente de un dispositivo móvil, ya que existe la posibilidad de que un cliente no tenga su dispositivo móvil cerca, pero todavía necesite hacer uso del servicio de pago.[8]

Funcionalidad: Se refiere al grado de comodidad y a la utilidad percibida por los usuarios, en el momento de

hacer uso del sistema de pago ubicuo, es un aspecto crucial si se tiene en cuenta el gran número de transacciones que una persona podría realizar durante el curso de un día normal. Por ejemplo, los usuarios esperarán un nivel de servicio comparable al existente con las tarjetas de crédito y las cuentas bancarias. Problemas como confianza, responsabilidad, contabilidad y aseguramiento se deben tratar adecuadamente para que los usuarios acepten el sistema. La mayoría de los usuarios no aceptaría un sistema que permita que se lleven a cabo transacciones financieras sin su intervención; al mismo tiempo, no es práctica la participación del cliente en todas las transacciones especialmente cuando estas son de bajo valor. Por lo tanto, los diseñadores de los sistemas de pago se enfrentan al reto de balancear estas necesidades contradictorias.[8]

Despliegue: Para tener éxito, el sistema de pago de computación ubicua se debe poder emplear a gran escala. En términos prácticos, debe soportar tanto nuevos servicios como los ya existentes.[8]

Anteriormente se definieron las características más importantes que debe tener cualquier servicio de pago ubicuo, y teniendo en cuenta que el servicio de facturación y pago desarrollado pretende ser un servicio con estas características, a continuación se relacionan cada una de ellas con la solución propuesta.

El servicio de pago implementado es espontáneo, ya que los usuarios lo pueden iniciar en cualquier momento, en alguno de los puntos de venta POS, siempre y cuando ya esté registrado en el sistema y tenga una cuenta de usuario. El proceso de pago es corto debido a las propiedades de las tecnologías y herramientas empleadas, por lo cual es poco probable que el cliente entable una relación de larga duración con el sistema en el momento del pago. Los bajos costos de operabilidad y el empleo de herramientas de libre acceso incrementan la eficiencia del servicio.

En un sistema de pago, la seguridad en las transacciones y la privacidad de la información suministrada por los clientes, son dos aspectos trascendentales para generar confianza tanto en los usuarios como en el proveedor del servicio. En el sistema se hace uso de la infraestructura de llave pública, que permite realizar la autenticación de los usuarios y cada una de las entidades que conforman el sistema mediante el empleo de credenciales, además de garantizar la integridad de la información con el cifrado de la misma, para lo cual se requiere el uso de certificados digitales emitidos por autoridades certificadoras.

La naturaleza ubicua del servicio de facturación y pago desarrollado lo hace un sistema flexible frente a otras formas de pago tradicionales, en donde la usabilidad del sistema es un aspecto determinado en gran parte por la experiencia de cada individuo al momento de utilizar el servicio, por lo cual es algo difícil de percibir por los desarrolladores. Pero si se tienen en cuenta el grado de aceptación de las tarjetas de crédito por parte de los usuarios y las mejoras que un sistema de pago móvil presenta frente a estas formas de pago tradicionales, se puede garantizar en gran medida la funcionalidad que representa la solución propuesta para las personas.

Al no ser necesario llevar consigo una tarjeta ni digitar claves en equipos extraños se mejora la experiencia del usuario final que percibe el servicio como fácil de utilizar.

El sistema propuesto ha sido desarrollado sobre plataformas robustas y con herramientas de libre acceso, que lo hacen no solo escalable ante un aumento en el número de usuarios, sino integrable a soluciones y sistemas similares.

C. RFID y NFC

RFID y NFC son tecnologías que merecen especial atención en la computación ubicua. RFID se refiere a cualquier sistema que permita la transmisión de números de identificación sobre radio, obteniendo una identificación automática con capacidades de almacenamiento y recuperación remota de datos, mediante el uso de dispositivos llamados etiquetas.[9] Un sistema RFID se compone de dos partes fundamentales: la etiqueta (transponder) y el lector. Una etiqueta RFID es un pequeño objeto que puede ser adherido o incorporado en un producto, animal o persona, que está compuesto de un microcontrolador, una antena (cableada o impresa con tinta de carbón conductivo) que habilita la recepción y la respuesta a solicitudes de radio-frecuencia desde un transmisor/receptor (tranceiver) RFID, y un material de encapsulación de polímero que envuelve la antena y el microcontrolador.[9] El lector es quien inicia el proceso de identificación al generar un campo RF en una frecuencia específica, definida por un sistema en particular, con lo cual causa una diferencia de voltaje por medio de un acoplamiento capacitivo o inductivo.[9] La etiqueta detecta este cambio y después de un proceso de autenticación opcional, gracias a un mecanismo de respuesta del lector, responde transmitiendo la identificación que posee.[9]

NFC es una tecnología que involucra Identificación de Radio Frecuencia (RFID) de corto alcance y posibilita la transferencia de datos entre un dispositivo móvil y un sistema de servicios, por simple contacto entre aquél y una placa NFC.[10] NFC se diferencia de las demás tecnologías RFID o de proximidad en que su distancia de funcionamiento es corta y depende del diseño de la etiqueta y del lector, pero, generalmente, el radio de acción es muy pequeño; esto es una ventaja a la hora de atender servicios que implican una necesaria privacidad, como es el caso de un proceso de facturación y pago. Además, NFC va más allá de RFID en el sentido que es un protocolo simétrico, donde los lectores pueden leer de etiquetas y de otros lectores directamente, es decir, se pasa de transferir datos en un solo sentido a un proceso bidireccional.[10]

3. ARQUITECTURA DE PAGO Y FACTURACIÓN

Una arquitectura para pago y facturación de servicios ubicuos debe estar estructurada de forma tal que sus diferentes componentes permitan satisfacer las necesidades de los usuarios desde el punto de vista del servicio prestado, pero al mismo tiempo debe garantizar la protección de la información a intercambiar, por lo cual es necesario hacer énfasis en el manejo en la seguridad de cada bloque y sus conexiones, ya que un sistema es tan seguro como su elemento más inseguro.

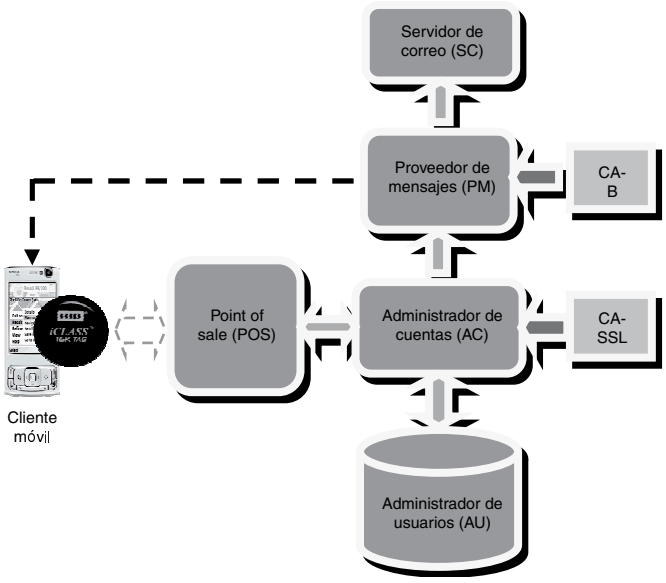


Figura 1. Arquitectura para pago y facturación de servicios móviles ubicuos

La arquitectura propuesta (Figura 1) tiene los siguientes componentes: un Cliente Móvil, que permite al usuario realizar pagos, un POS (Point of Sale) capaz de recibir pagos de móviles habilitados con interfaces de contacto cercano, un Administrador de Cuentas (AC) que es el núcleo de la arquitectura y le recibe todas las peticiones de transacción, un Administrador de Usuarios (AU), que almacena toda la información concerniente a las cuentas de los diferentes usuarios del sistema, principalmente las credenciales de cada uno de ellos que son generadas por el AC, un Proveedor de Mensajes (PM) encargado de enviar mensajes de información al móvil seguro del usuario, y finalmente un servidor de correo que envía la factura digital a la cuenta de correo del usuario.

Se tienen además dos entidades certificadoras, CA – SSL, que se encarga de certificar al Administrador de Usuarios y CA – B (Certificate Authority - Billing), que certifica al proveedor de mensajes y en especial le proporciona un certificado para firmar digitalmente las facturas enviadas a las cuentas de correo de los usuarios del servicio.

A. Proceso de pago y facturación

El proceso de pago y facturación se describe a continuación.

El usuario accede al servicio mediante su teléfono móvil, el cual como requisito debe tener dos interfaces de comunicación, la primera de tipo contactless o por contacto cercano, que es necesaria al momento de efectuar el pago; la segunda una interfaz inalámbrica que le permita acceder a una red de área amplia con el fin de recibir los mensajes de confirmación cuando la transacción sea exitosa.

Con la interfaz de contacto cercano que puede ser una etiqueta RFID o

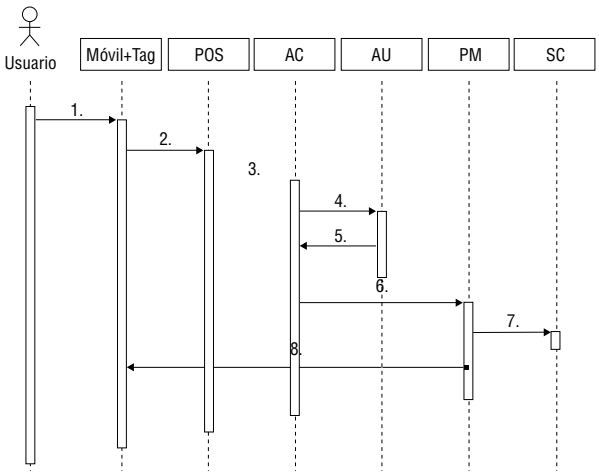
un elemento NFC, el móvil se puede comunicar con el POS de forma confiable, al realizarse autenticación mutua y cifrada de la información transmitida, para de esta forma distribuir la seguridad y no dejarla como responsabilidad única de un servidor central.

Para obtener la información acerca de la capacidad de pago del usuario, el POS debe comunicarse de forma segura con el Administrador de Cuentas, el cual se comunica con la mayoría de los bloques de la arquitectura, el AU debe estar certificado por una CA (Certification Authority) para poder cifrar todos los mensajes de la comunicación. El AC se comunica directamente con el Administrador de Usuarios que maneja la información de los usuarios activos y su correspondiente capacidad de pago.

El Administrador de Cuentas debe notificarle al usuario el resultado de la transacción a través del PM, que debe contar con acceso a la interfaz inalámbrica del móvil. El PM también se encarga de realizar el envío de la factura electrónica, la cual debe estar firmada digitalmente por una segunda CA, que se ha denominado CA-B (CA – Billing), esta prueba de compra digital se envía al correo electrónico que el usuario proporciona al crear su cuenta. El proceso completo es ilustrado en el diagrama de la Figura 2.

B. Creación de cuenta

Para el proceso de creación de cuenta se añade un nuevo componente a la arquitectura general, denominado Punto de Creación de Cuentas (PCC). Este componente permite realizar



1. Selección de compra
2. Envío de credenciales
3. Envío de información de compra
4. Petición de validación de información
5. Validación de transacción
6. Solicitud de mensaje de información
7. Envío de factura digital
8. Envío de mensaje de información

Figura 2. Diagrama de secuencia del proceso de pago y facturación

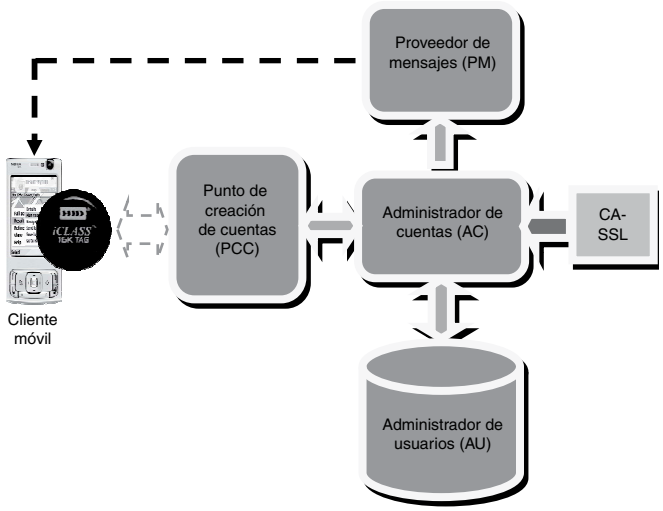


Figura 3. Arquitectura para creación de cuenta

operaciones como creación de cuenta para un nuevo usuario, bloqueo de cuenta en caso de presentarse un reporte de robo del dispositivo móvil y eliminación de cuentas.

El usuario debe crear una cuenta para usar el sistema, en este proceso se asigna una etiqueta RFID a su teléfono móvil, en la que se graban mediante una conexión segura los datos de cuenta necesarios para realizar el proceso de pago. El resultado de este proceso se envía en un mensaje a través de la otra interfaz inalámbrica del dispositivo móvil.

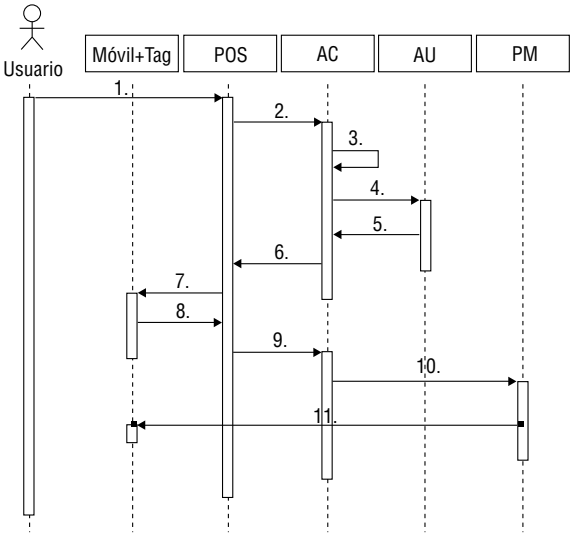
El proceso de grabación en la etiqueta se realiza mediante el PCC, este se comunica con el AC, como se muestra en la Figura 3, y le hace una petición de creación de cuenta, el AC crea uno o varios identificadores para el nuevo usuario y los envía al AU para que sean registrados.

Al obtener un resultado satisfactorio de la operación de registro, el AC envía esta información al PCC para que los datos de cuenta sean grabados en la etiqueta del dispositivo móvil. Toda la información transferida durante este proceso se cifra, y si es necesario se refuerza la seguridad con autenticación mutua y firma digital sobre los datos transmitidos. Además el AC debe estar certificado por una entidad confiable, como Verisign, Thawte o Global Trust que son empresas muy conocidas y dedicadas a la comercialización de varios tipos de certificados. Finalmente el PM se encarga de hacer llegar un mensaje al usuario para darle la bienvenida al sistema de pago o en caso de algún inconveniente informarle que no fue posible crear su cuenta y qué debe hacer para poder acceder al servicio, tal como se aprecia en el diagrama de la Figura 4.

4. CASO DE ESTUDIO

A. Implantación de un servicio de pago por proximidad en un ambiente ubicuo.

A continuación se describen los detalles de implantación de cada



1. Solicitud de creación de cuenta
2. Envío de información del usuario
3. Creación de identificadores
4. Registro de identificadores
5. Confirmación de registro
6. Envío de identificadores
7. Grabación de etiqueta
8. Confirmación
9. Confirmación
10. Petición de envío de mensaje de bienvenida al sistema
11. Envío de mensaje de bienvenida

Figura 4. Diagrama de secuencia del proceso de creación de cuenta

una de las partes de la arquitectura propuesta.

Teléfono Móvil

El teléfono móvil utilizado por el usuario para acceder al servicio debe contar con una interfaz de contacto cercano para efectuar el pago; existen dos opciones para cubrir este aspecto: emplear una etiqueta RFID o usar un teléfono NFC. Según la que se utilice cambia el modelo de negocio.

Al usar un teléfono NFC, tal como el Nokia 6131 NFC, se puede consultar la información que se encuentra en la etiqueta dentro del teléfono mediante un *Trusted Midlet* que se comunica con un *Java Card Applet* instalado en el elemento seguro NFC.[11] El *applet* nunca inicia una comunicación, solamente espera a que un *midlet* que debe ser seguro, lo haga mediante comandos APDU (Application Protocol -Data Unit); únicamente un *midlet* firmado con un *code-signing certificate* es considerado seguro, con lo cual es asociado al dominio *trusted third party* del teléfono y puede acceder a las API restringidas de java.

Empresas como Thawte y Verising cobran alrededor de US500 por un certificado con vigencia anual. Adicional a esto es necesario tener en

cuenta que el elemento seguro NFC puede ser accesible únicamente mediante llaves de autenticación, por lo cual, para efectuar un desarrollo debe realizarse una operación de desbloqueo del elemento seguro NFC. En el caso del teléfono de Nokia, existe un Servicio de Desbloqueo mediante una midlet que agrega llaves de autenticación conocidas, con el fin de permitir el proceso de desarrollo de aplicaciones.[11]

La segunda opción que se tiene para la interfaz de contacto cercano es usar etiquetas RFID, que tienen características muy adecuadas para la implementación de la arquitectura, como son:

- Comunicaciones confiables a alta velocidad, sin arriesgar la seguridad de los datos.
- Nivel elevado de seguridad con autenticación mutua, codificación de datos, y llaves diversificadas de 64-bit para permitir la lectura/escritura.
- Suficiente memoria de lectura/escritura como para almacenar varias plantillas biométricas.
- Sistema de archivos separados para garantizar seguridad, lo que permite implementar numerosas aplicaciones.
- Cumplimiento de los estándares ISO 15693[12] y 14443B[13] para las comunicaciones sin contacto.

Toda la transmisión de datos por radiofrecuencia entre la tarjeta y el lector se codifica con un algoritmo seguro que normalmente pertenece al fabricante del lector, no obstante también se utilizan técnicas de cifrado estándar de la industria para reducir el riesgo de robo o manipulación de información. Para más seguridad aún, los datos de la tarjeta también pueden protegerse con encriptación DES (Data Encryption Standard) o triple DES, y el tiempo necesario para llevar a cabo las operaciones de encriptación se mantiene bajo, de forma que las transacciones se realizan en menos de 100 milisegundos, en el caso de una aplicación típica segura de billetera electrónica.[14]

POS

Se debe implementar con un lector capaz de establecer una comunicación segura con la etiqueta RFID que garantice un nivel de seguridad adecuado. Básicamente el punto de venta o pago está compuesto por un lector y un computador conectado a este, en el cual debe haberse instalado una aplicación que utilice el API proporcionado por el fabricante del lector. Algunas opciones disponibles son lectores de fabricantes como Omnikey y HID Global que ofrece lectores de la serie iCLASS, de los cuales se eligió el RW100[15] como unidad lectora. En este punto la seguridad se basa en autenticación mutua entre las etiquetas y el equipo de lectura, además de la encriptación de la información mediante cifrado triple DES, que se puede realizar mediante el SDK que proporciona el fabricante escrito en Visual Basic para plataforma Windows, pero debido a que se busca el mayor nivel de seguridad posible en la estación de pago, se optó por trabajar con la distribución Debian Etch de Linux y desarrollar directamente sobre esta plataforma mediante NetBeans IDE. En el POS debe ir instalada una aplicación cliente que es básicamente

una aplicación de escritorio escrita en Java, con acceso al protocolo serial definido por el fabricante (ya que este protocolo es independiente de la plataforma) mediante JNI y además con capacidades de comunicación con un servicio web Java. Esta aplicación se utilizará para establecer una comunicación segura con el lector ya que se puede implementar autenticación mutua entre ellos y así los identificadores de un usuario que han sido almacenados en la etiqueta asociada a su móvil y posteriormente enviar esta información al AC mediante la utilización de un protocolo seguro como HTTPS (HTTP Over SSL).

Administrador de cuentas (AC)

El Administrador de Cuentas es un bloque fundamental de la arquitectura ya que interviene con la mayoría de los bloques definidos y debe establecer una comunicación segura con cada uno de ellos, además debe ser lo suficientemente flexible como para poder interactuar con aplicaciones de escritorio, Bases de Datos, Directorios que implementen LDAP[16] (Lightweight Directory Access Protocol) como openLDAP[17] y el proveedor de mensajes definido.

En esencia es un Servicio Web Java, desarrollado con NetBeans IDE que hace uso de Metro,[22] un stack para el desarrollo de Servicios Web seguros y es desplegado sobre GlassFish v2 que es un servidor de aplicaciones gratuito y de código libre, distribuido con la licencia CDDL y la GNU GPL. GlassFish tiene como base al Sun Application Server.[23] El AC tiene la capacidad de atender solicitudes por parte de un cliente, que en este caso es el POS, solicitando verificación de la información leída de la etiqueta asociada a un móvil, o creación de identificadores cuando un nuevo usuario necesita acceder al servicio.

El AC debe conectarse con la base de datos, repositorio o directorio, que administre los usuarios, en este caso se ha elegido un servidor de directorio LDAP. El AC se comunica con él mediante JNDI (Java Naming and Directory Interface), que es la interfaz de la plataforma Java para conexión con servicios de nombres y directorios.[24]

Administrador de usuarios

La información de cada usuario debe ser almacenada y para ello existen diferentes clases de bases de datos y directorios, de forma que cuando un usuario realice un pago se pueda validar su información personal, números de cuenta, permisos, y certificados de forma muy rápida, por lo cual se sugiere una implementación de LDAP, como openLDAP, para realizar la administración de usuarios, ya que permite consultas con tiempos de respuesta muy bajos.

El protocolo LDAP, además de permitir un manejo jerárquico y sistemático en la información de los usuarios, lo cual es muy útil por el modelo del negocio implementado en un sistema de pago, también posibilita varias formas de acceso, como JNDI.

LDAP es adecuado para implementar sistemas de autenticación/autorización centralizada, o para sistemas donde se guarda gran cantidad de registros y se requiere un uso constante de los mismos. Pero se pueden usar métodos alternativos para el proceso de autenticación como Kerbe-

ros v5 o Digest-MD5 que se acoplan perfectamente con openLDAP y dejar este último únicamente para realizar la autorización. Para usar Kerberos es necesario hacer uso de GSSAPI (Generic Security Services Application Programming Interface) en AC, a través de JNDI y realizar la configuración necesaria en openLDAP, además de correr los demonios de Kerberos en Linux.

Proveedor de mensajes

Con el fin de enviar información de servicio a los usuarios se ha definido el PM, que mediante mensajes Wap Push puede notificar al comprador de una transacción efectuada satisfactoriamente o de inconvenientes debido a que el saldo en cuenta no fue suficiente, al momento de realizar la adquisición de un producto o servicio.

Los mensajes pueden enviarse mediante una Gateway SMS o un Push Proxy Gateway tal como Kannel[18] que corre sobre Linux, o APIs de Java como jSMS,[19] que actúan como interfaz entre la red cableada y la red inalámbrica, permitiendo la comunicación entre los componentes arquitecturales soportados en una infraestructura cableada y el dispositivo móvil del usuario.

Gracias a la compatibilidad directa entre el Servicio Web y jSMS, además de las facilidades de configuración que proporciona esta API, es una opción sobresaliente para formar parte en la implementación de un servicio de pago por proximidad.

Servidor de correo

Debido a que la gran mayoría de teléfonos móviles en el país son de gamas media y baja y no están en capacidad de leer documentos como archivos pdf, formato en el cual se podría enviar la factura electrónica firmada, se hace necesario el uso de un servidor de correo, ya sea mediante un API como Java Mail o servicios de correo como Postfix o Sendmail corriendo bajo Linux.

Java Mail posee un diseño universal y abstracto que lo hace más difícil de usar y las clases requeridas (o archivos .jar) son bastante pesadas en términos de espacio requerido en disco. Por consiguiente, si pequeñas aplicaciones o dispositivos embebidos necesitan capacidades de mensajería tal como e-mail o SMS, deben implementar sus propios mecanismos para enviar y recibir correos.[20]

Por su parte, jSMS provee una API pequeña para SMS y correo que fácilmente puede ser adaptada a pequeños dispositivos.

Básicamente se necesita un servicio de correo que permita enviar como archivo adjunto la factura electrónica firmada digitalmente a la cuenta de correo proporcionada por el usuario al crear su cuenta de uso del servicio, también que el servicio de correo se pueda comunicar de forma segura con el Administrador de Usuarios.

Entidades certificadoras

Es necesario garantizar que la información enviada de un módulo a otro sea auténtica e íntegra.

Auténtica en cuanto a que se pueda saber a ciencia cierta que un módulo determinado envió el mensaje y prevenir ataques de suplantación. Íntegra en cuanto a que se garantice que el mensaje no ha sido modificado. Esto se puede garantizar mediante el

uso de PKI (Public Key Infraestructure), la cual implica la generación de un par de llaves (pública y privada) para identificar a un usuario determinado, y el uso de funciones Hash. En PKI se necesita de una autoridad certificadora confiable que emita certificados cuyo contenido sean llaves públicas de las entidades con las que se establece comunicación.[21]

En esta arquitectura se muestran dos tipos de entidades certificadoras. Con CA-SSL se hace referencia a una entidad certificadora externa como Verisign o Thawte, las cuales comercializan certificados SSL que permitirían validar la identidad del Administrador de Cuentas.

El segundo tipo de entidad certificadora es necesaria para generar los certificados que permitan firmar la factura electrónica, las opciones de implementación para este caso son openSSL y entornos gráficos que usan SSL como XCA que funciona tanto con Linux como con Windows.

Otra herramienta muy útil es keytool, que hace parte de la plataforma Java y se encarga de generar parejas de llaves públicas y privadas, certificados autofirmados y almacenes de claves o keystore.

B. Comparación

No existe una clasificación única para los sistemas de pago móvil, debido a los diferentes aspectos que se deben tener en cuenta en este tipo de transacciones, que limitan y caracterizan el sistema. La ubicación del usuario, el monto a pagar, el proveedor del servicio, el instante y el medio utilizado para el pago, son los rasgos más sobresalientes en un sistema de pago y que además permiten su caracterización.[25]

Basándose en los aspectos nombrados en el párrafo anterior se puede definir el tipo de pago propuesto en este artículo; el cual es un sistema POS (Point of Sale) al requerir la presencia del usuario en el punto de venta; las cantidades de dinero que maneja son pequeñas, y lo convierten en un sistema micropago diseñado para ser administrado por un sistema independiente soportado en Entidades Financieras, y basado en tokens que se transfieren del usuario al vendedor en tiempo real.[26]

En todo el mundo han sido planteados diversos sistemas y arquitecturas para pagos móviles, en diferentes escenarios y con características distintas, por lo cual no existe un estándar o Modelo Global para un Sistema de Pago Móvil. Esta falta de normalización se debe a la competencia entre Entidades Financieras y Operadores de Telefonía Móvil por el control del mercado; y a la diversidad de escenarios donde se requiere un sistema de Pago móvil, cada uno con necesidades específicas, que caracterizan y limitan la solución de pago. En la Tabla 1 se comparan dos medios de pago implementados en el mundo, con el sistema de pago propuesto, donde se puede apreciar la dificultad para crear un medio de pago estándar.

En la Tabla 1 se aprecian sólo 3 soluciones de pago de varias que se han implementado en el mundo, y cada una ha sido creada para una necesidad específica en un contexto concreto. Incluso existen soluciones con la misma lógica, pero que emplean tecnologías distintas, por ejemplo

Tabla 1. Sistemas de pago

	LIPSO	PAYBOX	Propuesta
Forma de pago	Cargo a la Factura del teléfono Móvil	Se debita de una cuenta Bancaria o Tarjeta de crédito	Se debita de una cuenta Bancaria
Instante de pago	Pospago	Pago en Tiempo Real	Pago en Tiempo Real
Monto	Micropago	Micropago	Micropago
Ubicación del usuario	Pago desde Internet	Persona a Persona (P2P)	Punto de Venta (POS)
Según el proveedor	Operador Móvil	Proveedor Independiente/ Intermediario	Proveedor Independiente

eco-PAY es una solución similar a PAYBOX pero emplea SMS en lugar de un Servicio IVR (Interactive Voice Response), ya que fue creada para el contexto 11 canadiense, donde los SMSs tienen mayor acogida.25 La gran ventaja del sistema propuesto en este artículo frente a los pilotos y sistemas ya establecidos en el mundo, es que ha sido diseñado para el contexto colombiano de una forma tal que pueda funcionar con herramientas libres y pocos recursos económicos. Este sistema de pago permite llevar a cabo transacciones de poco valor en tiempo real, en situaciones comunes para los colombianos, sin la limitante de estar asociado a un operador móvil específico, además de emplear tecnologías de campo cercano creando un entorno ubicuo para los clientes.

5. CONCLUSIONES Y TRABAJOS FUTUROS

A pesar de las limitaciones tecnológicas que se tienen en Colombia, es posible llevar a cabo las primeras implementaciones para un sistema de facturación y pago ubicuo. En el momento, tecnologías como NFC y RFID no tienen un uso masivo por parte de los usuarios, pero la construcción de un piloto para pago y facturación desempeña un papel importante como un primer acercamiento de ellas a la población colombiana.

La calidad de un proceso de facturación y pago está medida por la seguridad, la privacidad y la confianza que éste refleje hacia los usuarios en el manejo de los datos personales, y especialmente de la información financiera. Es por esto que el papel que desempeñan las entidades certificadoras adquiere gran importancia en la prestación del servicio, para lograr satisfacción en los usuarios finales.

El panorama colombiano es muy alentador para la implementación de servicios de pago y facturación de nueva generación por la acogida que han tenido en los últimos años las tarjetas débito y crédito, y la enorme demanda de teléfonos móviles en el mercado.

El uso de tecnologías como RFID y/o NFC permitirá incrementar los niveles de seguridad en servicios de pago y facturación, para proteger mejor la identidad del usuario y dificultar la copia de su información, al contrario de como hoy en día se presenta con las tarjetas de banda magnética que son fácilmente duplicables.

Existen diversas opciones de implementación para cada uno de los

componentes de la arquitectura propuesta. En el momento de hacer la selección, se debe evaluar cuidadosamente la compatibilidad entre estos ya que muchos de ellos han sido construidos sobre diferentes lenguajes de programación (Java, C#) y diferentes plataformas (Windows, Linux). Además se deben tener en cuenta los costos que implica cada una de las opciones (equipos, licencias de software y certificados digitales), para así implementar la solución más conveniente.

Algunos trabajos futuros son:

- Implementación de la arquitectura propuesta, teniendo en cuenta las limitaciones de las redes móviles actualmente existentes en el país en cuanto a velocidades de acceso GPRS.
- Pruebas de seguridad sobre la implementación obtenida, que permitan validar la escogencia de soluciones particulares y formular soluciones alternativas en caso de ser necesario.
- Construcción de una plataforma para facturación y el pago de servicios ubicuos en ambientes móviles, que defina los protocolos necesarios para realizar la facturación y el pago de servicios ubicuos en ambientes móviles de forma segura y conveniente, y además establezca recomendaciones para la implementación e implantación de la arquitectura y los protocolos definidos en el contexto colombiano.

BIBLIOGRAFÍA

1. Ishii Hiroshi. Bottles: A Transparent Interface as a Tribute to Mark Weiser. IEICE Transactions Inf. And Syst, Vol. E87-D, Número 6. p. 1299-1311. Junio 2004.
2. Weiser Mark. The Computer for the 21st Century. Scientific American Ubicomp. Septiembre 1991.
3. Z Weiping, W Dong, S Huanye, Dep. of Computer Science & Engineering, Shanghai Jiaotong University: Mobile Rfid Technology For Improving M-Commerce. IEEE International Conference on e-Business Engineering. 2005
4. L Wei, Z Chenglin, Z Wei, Z Zheng ,Z Feng, L Xiaoji, F Jielin, K KyungSup: The Gprs Mobile Payment System Based On Rfid. Communication Technology, 2006. ICCT '06. International Conference on. Noviembre 2006.
5. Espaldarazo a la Factura Electrónica. Revista Dinero – Sección Tecnología. p. 66 - 67. Junio de 2007.
6. Satyanarayanan M, Carnegie Mellon University. Pervasive Computing: Vision and Challenges. Personal Communications. p. 10 - 17. Agosto 2001.
7. Sunaga H, Takemoto M, Yamato Y, Yokohata Y: Ubiquitous Life Creation Through Service Composition Technolgies. World Telecommunications Congress 2006 - WTC2006.
8. P Boddupalli, F Al-Bin-Ali, N Davies, A Friday, O Storz, M Wu, Department of Computer Science University of Arizona, Arizona, United States: Payment Support In Ubiquitous Computing Environments. Fifth IEEE Workshop

on Mobile Computing Systems & Applications. 2003.

9. Roussos G. Birkbeck College, University of London. Enabling RFID in Retail. IEEE Computer Magazine, Vol 39. p. 25 – 30. Marzo 2006

10. Anokwa Y, Borriello G, Pering T, Want R, Computer Science and Engineering University of Washington Seattle: A User Interaction Model for NFC Enabled Applications. Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007.

11. Nokia 6131 NFC SDK: User's Guide. Forum Nokia. Julio de 2007.

12. WG8 Working Group 8 ISO/IEC JTC1/SC17. Project Details on: ISO/IEC 15693, Vicinity cards (VICCs). Disponible en: http://www.wg8.gipp.com/sd1.html#15693

13. WG8 Working Group 8 ISO/IEC JTC1/SC17. Project Details on: ISO/IEC 14443, Proximity cards (PICCs). Disponible en: http://www.wg8.gipp.com/sd1.html#14443.

14. Tag iCLASS Datasheet. HID Global. 2007. Disponible en: http://www.hidcorp.com/documents/iclass_tag_ ds_es.pdf

15. iCLASS® RW100, RW300, RW400 Readers Datasheet. HID Global. Abril de 2007. Disponible en: http://www.hidcorp.com/documents/iclass_rw10 0_300_400_ds_en.pdf

16. Lightweight Directory Access Protocol (LDAP): The Protocol. IETF Proposed Standard RFC 4511. Junio de 2006. Disponible en: http://tools.ietf.org/html/rfc4511.

17. The OpenLDAP Project Overview. OpenLDAP Project Foundation. 2007. Disponible en: http://www.openldap.org/project/

18. Kannel: Open Source WAP and SMS Gateway – Overview. The Kannel Group. 2006. Disponible en: http://www.kannel.org/overview.shtml

19. jSMS – Java SMS & MMS API – Overview. Object XP. 2007. Disponible en: http://www.objectxp.com/products/jSMS/

20. JavaMail API Specification. Sun Microsystems, Inc. 2007. Disponible en: http://java.sun.com/products/javamail/reference/ api/index.html

21. Varela Rubén. Criptografía, Una Necesidad Moderna. Revista Digital Universitaria, Vol. 7. Departamento de Seguridad en Cómputo en DGSCA, UNAM. Julio de 2006. Disponible en: http://www.revista.unam.mx/vol.7/num7/art56/j ul_art56.pdf

22. What is Metro? GlassFish Metro. Disponible en: https://metro.dev.java.net/discover/

23. GlassFish. GlassFish Community. Disponible en: https://glassfish.dev.java.net/public/users.html

24. Java Naming and Directory Interface (JNDI). Sun Microsystems, Inc. 2008. Disponible en: http://java.sun.com/products/jndi/

25. Ondrus J, Pigneur Y, INFORGE - Ecole des HEC University of Lausanne, Switzerland: A Disruption Analysis in the Mobile Payment. Proceedings of the 38th Annual Hawaii International Conference on System Sciences. 2005.

26. Vásquez A, Depto, Ing. Eléctrica y Computación, UACJ (Universidad Autónoma de Ciudad Juárez): Estándares de Métodos de Pago por Móvil. Revista Cultura Científica y Tecnológica CULCyT, 2006.

CURRÍCULOS

Zeida María Solarte. Ingeniera Electrónica de la Universidad del Cauca, en 1990. Especialista en Redes y Servicios Telemáticos, por la Universidad del Cauca en 1999. Magíster en Ingeniería, área Telemática, Universidad del Cauca, en 2005. Profesora titular de la Corporación Universitaria Autónoma de Occidente de Cali.

Oscar Mauricio Caicedo. Ingeniero en Electrónica y Telecomunicaciones, Universidad del Cauca, 2001. Especialista en Redes y Servicios Telemáticos, Universidad del Cauca, 2003. Magíster en Ingeniería, área Telemática, Universidad del Cauca, 2006. Coordinador del grupo de interés en desarrollo de aplicaciones móviles e inalámbricas W@PColombia y miembro del grupo de Ingeniería Telemática.

Docente del departamento de Telemática de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

Milton Royers Ausecha Penagos. Estudiante del programa de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, candidato a recibir el título en 2008. Durante el período 2007-2008 se desempeñó como vicepresidente de la rama estudiantil IEEE de la Universidad del Cauca, y coordinador del comité académico del Cuarto Seminario Nacional de Tecnologías Emergentes en Telecomunicaciones TET 2007, evento organizado por este grupo estudiantil en la ciudad de Popayán.

Actualmente se encuentra realizando su trabajo de tesis en servicios de pago y facturación para servicios ubicuos y pertenece al semillero de investigaciones en aplicaciones móviles e inalámbricas W@PColombia en el área de nuevos servicios.

Javier Fernando Imbús Guzmán. Estudiante del programa de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, candidato a recibir el título en 2008. Durante el período 2005-2006 se desempeñó como vicepresidente de la Fundación Pulsos FIET, y organizador de la tercera jornada FIET, evento organizado en la ciudad de Popayán. Actualmente se encuentra realizando su trabajo de tesis en servicios de pago y facturación para servicios ubicuos y pertenece al semillero de investigaciones en aplicaciones móviles e inalámbricas W@PColombia en el área de nuevos servicios.

Referencias

Almenarez, F. (2005). *Arquitectura De Seguridad Para Entornos De Computación Ubicua Abiertos Y Dinámicos.* Tesis doctoral, Universidad Carlos III De Madrid, Madrid.

BlueHackTeam. (2005). *Public Key Infraestructure.* Recuperado el Julio de 2008, de Hacker.net Website: http://foro.elhacker.net/index.php/topic,53455.0.htm

Caicedo, O. (2007). *Plataforma de comercio móvil para el sector artesanal colombiano.* Tesis de Maestría, Universidad del Cauca, Departamento de Telemática.

Eudmednet. (2006). *Clave simétrica.* Recuperado el Junio de 2008, de Eumednet Website: http://www.eumed.net/cursecon/ecoinet/seguridad/clavesimetrica.ppt

Eurologic. (2005). *La Firma Digital.* Recuperado el Enero de 2008, de Eurologic Website: http://www.eurologic.es/conceptos/firmadigital.htm#fechado

Goldman, D. (2006). *Opera Mini 2.0.* Recuperado el Enero de 2008, de Opera Watch Website: http://operawatch.com/news/2006/05/opera-mini-20-released.html

Google. (2008). *Google para móviles*. Recuperado el Enero de 2008, de Google Mobile: http://www.google.com/intl/es_es/mobile/

IETF. (1999). *Internet X.509 Public Key Infraestructure Certificate and CLR Profile.* Recuperado el Noviembre de 2007, de IETF RFC 2459: http://www.ietf.org/rfc/rfc2459.txt

Lewis, R. (Enero de 2005). *Glossary of Terms for Device Independence.* Recuperado el Junio de 2008, de W3C Website: http://www.w3.org/TR/di-gloss/#def-delivery-context-v2

Motorola. (2008). *Motorola Handsets.* Recuperado el Agosto de 2008, de Motorola Website: http://developer.motorola.com/products/handsets/

Netimperative. (2006). *Case Study: AOL Mobile Search.* Recuperado el Enero de 2008, de InfoLogin Website: http://www.infogin.com/doc/press%20releases/Netimperative%208%20May%202006.pdf

Nokia. (2008). *Nokia device specifications.* Recuperado el Agosto de 2008, de Nokia Website: http://www.forum.nokia.com/devices/matrix_all_1.html

Nokia. (2007). *S60 Platform SDKs for Symbian OS, for Java.* Recuperado el Agosto de 2008, de Nokia Website: http://www.forum.nokia.com/info/sw.nokia.com/id/6e772b17-604b-4081-999c-31f1f0dc2dbb/S60_Platform_SDKs_for_Symbian_OS_for_Java.html

Nokia. (2007). *Series 40 Platform SDKs.* Recuperado el Agosto de 2008, de Nokia Website: http://www.forum.nokia.com/info/sw.nokia.com/id/cc48f9a1-f5cf-447b-bdba-c4d41b3d05ce/Series_40_Platform_SDKs.html

OpenWave. (2005). *OpenWave Phone Simulator.* Recuperado el Agosto de 2008, de OpenWave Website: http://developer.openwave.com/dvl/tools_and_sdk/phone_simulator/

Phone.com. (1999). *UP.SDK Getting Started Guide.* Recuperado el Agosto de 2008, de OpenWave Website: http://developer.openwave.com/htmldoc/32w/getstart/

Rabin, J., & McCathieNevile, C. (Noviembre de 2006). *Mobile Web Best Practices 1.0 - Basic Guidelines.* Recuperado el Noviembre de 2007, de W3C Mobile Web Initiative website: http://www.w3.org/TR/mobile-bp/

Scott, S., Chua, H., & NG, S. (2005). *Web Content Adaptation Process And System.* Recuperado el Enero de 2008, de Wipo Website: http://www.wipo.int/pctdb/en/wo.jsp?WO=2005%2F033969&IA=WO2005%2F033969&DISPLAY=STATUS

SmartGSM. (2008). *Características técnicas del celular Siemens A56.* Recuperado el Agosto de 2008, de SmartGSM Website: http://www.smart-gsm.com/siemens-a56.phtml

SonyEricsson. (2008). *Sony Ericsson Phone Galley.* Recuperado el Agosto de 2008, de Sony Ericsson Website: https://developer.sonyericsson.com/device/searchDevice.do?restart=true

Talavera, J. (Septiembre de 2005). *Infraestructura para la Criptografía de Clave Pública.* Recuperado el Mayo de 2008, de Memorias Exposición Tecnológica y Científica (ETyC): http://www.cnc.una.py/cms/cnc/content/pki.pdf

Talens-oliag, S. (2003). *Introducción a los Certificados Digitales.* Recuperado el Enero de 2008, de Infocentre - Universidad de Valencia Website: http://www.uv.es/~sto/articulos/BEI-2003-11/certificados_digitales.html#id2859490

Viana, W., Teixeira, R., Cavalcante, P., & Andrade, R. (2005). *Mobile Adapter: Uma abordagem para a construção de Mobile Application Servers adaptativos utilizando as especificações CC/PP e UAProf.* Recuperado el Marzo de 2008, de Sociedade Brasileira de Computacao Website: http://www.sbc.org.br/bibliotecadigital/download.php?paper=173

W3C. (2008). *Mobile Web Initiative*. Recuperado el Enero de 2008, de W3C WMI Website: http://www.w3.org/Mobile/

WinwapTechnologies. (2008). *Browsing and Messaging Solutions for Mobile Devices.* Recuperado el Agosto de 2008, de Winwap Webiste: http://www.winwap.com/

Yahoo. (2008). *Yahoo Mobile.* Recuperado el Febrero de 2008, de Yahoo Website: http://mobile.yahoo.com/onesearch

Yonghyun, W., Changwoo, J., Jihong, K., & Sungkwon, C. (2001). *WebAlchemist: A Web Transcoding System for Mobile Web Access in Handheld Devices.* Recuperado el Marzo de 2008, de CiteSeer Website: http://citeseer.ist.psu.edu/474666.html