

Adaptación a dispositivos móviles de portales web diseñados para navegadores de escritorio

Ignacio Marín, Rodrigo García, Luis M. González, Julio A. Argüello

Área de Investigación, Desarrollo e Innovación
Fundación CTIC

Parque Científico y Tecnológico
33213 Cabueñes (Gijón)

ignacio.marin@fundacionctic.org
rodrigo.garcia@fundacionctic.org
luis.gonzalez@fundacionctic.org
julio.arguello@fundacionctic.org

M^a Jesús Fernández,
Victor Morlán

Unidad de Gestión Web Municipal
Ayuntamiento de Zaragoza

Plaza del Pilar
50071 Zaragoza

mjferuiz@zaragoza.es
vmorlan@zaragoza.es

Resumen

El siguiente trabajo pretende ofrecer un posible enfoque acerca de cómo abordar el problema de la adaptación de contenidos web a dispositivos móviles, a partir de un contenido que previamente está siendo ofrecido para navegadores web de escritorio (Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari, Konqueror, etc.), con una serie de restricciones en el desarrollo del proyecto que afectarán la estrategia a seguir.

Este trabajo ha sido utilizado como punto de partida para añadir una política de Independencia Web de Dispositivo a la Política Web del Ayuntamiento de Zaragoza y está disponible al público en www.ayto-zaragoza.mobi. La intención del Ayuntamiento es la de permitir que cualquier ciudadano acceda a la información disponible en los portales web dependientes de dicha organización, intentando que ésta se encuentre disponible para cualquier dispositivo con capacidades web, sin restricciones de memoria o capacidades gráficas o de procesamiento. Además, otros de los requisitos impuestos fueron el uso de tecnologías basadas en estándares y en código abierto y de uso libre.

La adaptación de los contenidos web de los portales dependientes del Ayuntamiento para su consumo desde cualquier tipo de cliente web supuso todo un desafío. Este trabajo pretende mostrar cómo se logró superar el mismo y, al mismo tiempo, ilustrar los avances que durante su desarrollo han tenido lugar en el estado de la técnica de las tecnologías relacionadas con la web

móvil (tecnología disponible, estándares en desarrollo y algunos servicios).

1. Antecedentes

En el invierno de 2005, el Ayuntamiento de Zaragoza encargó a Fundación CTIC un estudio relativo al estado de la técnica en torno a tecnologías relacionadas con el desarrollo de la web móvil en aquel instante. La intención de aquel estudio era una propuesta por parte de Fundación CTIC de una ampliación de la política de gestión de la Web Municipal del Ayuntamiento de Zaragoza, añadiéndole requisitos de Independencia Web de Dispositivo. Es importante destacar el hecho de que los nuevos requisitos debían afectar en el menor grado posible a la política y la metodología en vigor. Una vez definida la propuesta en papel, ya en primavera de 2006, se abordó la creación de un prototipo software que definiese la arquitectura básica de un sistema para el servicio de contenidos con requisitos de Independencia de Dispositivo.

2. Estado de la técnica

Previamente, al desarrollo del prototipo en verano de 2006, el panorama de la web móvil se ofrecía como algo verdaderamente heterogéneo. Trabajos como el de la actividad en torno a la Independencia de Dispositivo de W3C, complementado por el grupo de trabajo UAProf [1] de la Open Mobile Alliance (OMA), o el de

WURFL [2] (estándar de facto en la detección de capacidades de dispositivos) se complementaban con dos enfoques diametralmente opuestos para poder acceder a la identidad del cliente web y sus capacidades. Empresas como Volantis, SevenVal, Drutt o MobileAware ofrecían motores de transcodificación de contenidos (bien sea bajo demanda, o de forma estática, previa al arranque del servicio), pero no eran válidos por tratarse de tecnología propietaria.

En el preciso instante en que el trabajo de recolección del estado de la técnica tuvo lugar, despegaba también la Iniciativa de Web Móvil de W3C en forma de dos grupos de trabajo, cuyos objetivos en su primera fase de vida (la que estaba teniendo lugar en el momento de efectuar el estado del arte) se comentan a continuación:

- *Buenas Prácticas para la Web Móvil*: grupo que intentaba ofrecer pautas generales que permitiesen servir contenidos web a todos los dispositivos, con una experiencia correcta para el usuario en términos de usabilidad. También trabajaba en torno a la definición de tests que puedan dirimir si un sitio web soporta estas pautas generales (pruebas *mobileOK*, para la potencial obtención de un sello basado en el resultado de esas pruebas). Fundación CTIC se incorporó a este grupo de trabajo en otoño de 2005 y ha estado presente como miembro activo de este grupo de trabajo.
- *Descripción de dispositivos*: su labor era la de definir los posibles entornos en los que podría ser necesaria la existencia de un repositorio de descripción de dispositivos para, en una fase posterior, disponer de los requisitos necesarios en la definición, diseño e implementación de uno de estos repositorios.

3. Condiciones de partida

El punto de partida establecido por el Ayuntamiento de Zaragoza fue la entrega a Fundación CTIC de sendos documentos internos (“Guía de Estilo de Portal del Ayuntamiento de Zaragoza” y “Metodología de trabajo”) en los que se establecen las pautas generales seguidas por dicha organización. A partir de ambos y de un estudio del formato de las páginas web servidas por el Ayuntamiento, se definió por convenio el uso de un número limitado de plantillas, que los

documentos XHTML servidos a dispositivos de escritorio deberían seguir (para facilitar la adaptación a otros dispositivos). Estas plantillas están definidas como documentos XHTML Strict 1.0 [3] válidos, para simplificar el problema de la adaptación de escritorio a otro tipo de cliente web (dispositivos móviles, principalmente, aunque la solución también podría servir para cualquier tipo de cliente con capacidades limitadas de memoria, procesamiento y gráficos). De esta forma, la adaptación se basará en la conversión del documento XHTML Strict a XHTML Basic 1.0 [4], aplicando una serie de transformaciones que serán comentadas en epígrafes posteriores.

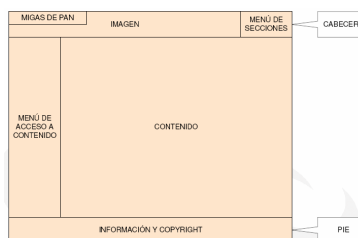


Figura 1. Ejemplo de plantilla de documento XHTML

Además, los contenidos servidos al ciudadano (que hasta entonces presuponían el acceso a los portales web del Ayuntamiento a través de navegadores de escritorio) seguían fielmente los estándares W3C de accesibilidad, ayudando a que el resultado del proceso de adaptación genere también contenido accesible y facilitando el propio proceso en sí.

Finalmente, el modelo de atención a las peticiones del cliente por parte de las aplicaciones web del Ayuntamiento se basa en la creación de aplicaciones web mediante Servlets [5] (aspecto éste crucial en las decisiones tomadas, como se verá más adelante).

4. Enfoque general

La primera decisión tomada fue la elección de la estrategia a seguir dentro de todas las posibilidades ofrecidas por las tecnologías relacionadas con la web móvil, en lo que respecta a la adaptación de contenidos. Debido al corto periodo de tiempo disponible para la elaboración de un prototipo (cuatro meses), un enfoque basado en adaptación con un alto nivel de granularidad

(explotando toda la información que WURFL o UAProf proporcionan) no pareció factible en su momento.

Se decidió por ello preparar el contenido para los dispositivos clientes de forma que, una vez detectado el tipo de dispositivo, el marcado sería servido de la siguiente manera:

- Si se trata de un navegador de escritorio, ofrecer los recursos web tal y como fueron creados originalmente.
- Si se trata de otro tipo de navegador, ofrecer los documentos XHTML adaptados, de forma que sigan las Buenas Prácticas para la Web Móvil [6] (en continua evolución durante el proceso de desarrollo del prototipo), y adaptar las imágenes contenidas en dichos documentos de forma que se adapten a las dimensiones de la pantalla del dispositivo cliente y a un formato gráfico soportado por el mismo.

Para poder detectar el tipo de dispositivo cliente y sus capacidades, y poder modificar la respuesta retornada por el servidor, se implementaron un conjunto de filtros [7] del API Servlet, aplicados sobre los Servlets originales utilizados por el Ayuntamiento para servir los contenidos de sus distintos portales.

5. Detección del dispositivo cliente y descubrimiento de capacidades

Para la detección del dispositivo cliente, se utiliza un filtro que analiza las peticiones HTTP que los clientes efectúan al servidor. Este filtro utiliza las APIs WURFL y JSR-188 [8], que permiten identificar al cliente web, siguiendo dos aproximaciones radicalmente distintas.

Sin extenderse demasiado en la explicación de ambas tecnologías, y remitiendo a la bibliografía adjunta para ambas, básicamente:

- WURFL se basa en un fichero (*wurfl.xml*) que contiene una base de datos de agentes de usuario y las capacidades asociadas a cada uno de ellos (lenguajes de marcado soportados o dimensiones de la pantalla serían algunos ejemplos). Su ventaja principal es que el fichero XML de WURFL puede residir en el propio servidor y puede ser extendido con

nuevos agentes de usuario y capacidades. La principal desventaja es que la información de WURFL es proporcionada por la comunidad de desarrolladores y que la frecuencia de actualización de *wurfl.xml* no está garantizada.

- La implementación de la especificación JSR-188 permite acceder al perfil CC/PP. En este caso, cuando un dispositivo móvil hace una petición al servidor, incluye entre las cabeceras HTTP una o más referencias a documentos RDF que describen las capacidades del dispositivo cliente. Las principales ventajas del uso de esta tecnología son dos: la fiabilidad de la información, proporcionada por los propios fabricantes; y la posibilidad de que los navegadores puedan indicar preferencias de usuario que modifiquen las capacidades declaradas en el perfil (por ejemplo, desactivar el sonido). Las principales desventajas residen en cierta falta de rigor de ciertos fabricantes, generando perfiles incorrectos sintácticamente (que no validan como RDF o como CC/PP) o semánticamente (incluyen información inexacta para los valores de alguna capacidad). Además, la primera vez que un cliente efectúa una petición al servidor, se provocan accesos remotos al documento o documentos RDF que especifican el perfil UAProf del dispositivo.

Se efectuó, teniendo en cuenta lo anterior, una implementación que combina ambas aproximaciones de forma que, ante cada petición, se examina la información que UAProf proporciona acerca del dispositivo cliente. En el caso de que ésta no se encuentre, se buscará entonces por medio de WURFL. Si la información no se encuentra en esta segunda fuente, se supondrá para el cliente web un Contexto de Envío por Defecto (DDC –Default Delivery Context–), tal y como lo definen las Buenas Prácticas para Web Móvil de W3C.

El prototipo ha sido diseñado de forma que el orden de uso de las APIs de detección del dispositivo cliente puede ser modificado. Además, se permite añadir dinámicamente nuevas APIs de búsqueda, así como establecer su precedencia. El acceso a las APIs se ha homogeneizado utilizando una interfaz común que esconde el vocabulario

particular que las distintas tecnologías utilicen para referirse a una misma capacidad.

Esta interfaz común o *vocabulario común* surge como medio para unificar la estrategia de nombrado de las capacidades de dispositivo, ocultando las nomenclaturas específicas de cada tecnología.

El vocabulario común consta de claves y valores. Las claves identifican características (el lenguaje de marcado preferido, el formato de imágenes, el soporte de CSS...) y los valores discriminan las diferentes opciones para cada característica (imágenes en JPG, PNG, GIF, etc.).

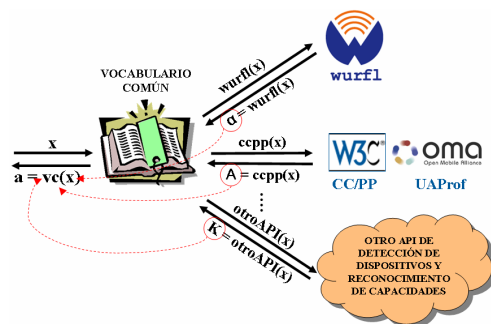


Figura 2. Concepto de vocabulario común

La Figura 2 muestra cómo la búsqueda del valor asociado a una clave x del vocabulario común dispararía una búsqueda de la clave correspondiente en WURFL ($wurfl(x)$) y, en el caso de que el dispositivo cliente disponga de información de esa clave (característica del dispositivo) retornará un valor α , que será traducido como valor a , en el lenguaje de pares clave-valor manejado por el vocabulario común.

Si WURFL no devolviese valor para la clave $wurfl(x)$ (pues no almacena información de la característica x para el dispositivo cliente que hizo una petición al servicio), se buscaría mediante el API JSR-188, consultando esa información al perfil CC/PP del dispositivo. Para ello, la búsqueda de un valor para x en el vocabulario común, dispararía la búsqueda de un valor $ccpp(x)$ en el perfil CC/PP correspondiente.

Este encadenamiento de búsquedas hasta encontrar un valor para una característica determinada de un cliente web podría fácilmente extenderse utilizando cuantas APIs de detección de capacidades de dispositivo se deseen. Si ninguna de las APIs disponibles ofreciese valor

para la característica deseada, se ofrecería un valor por defecto (basado de nuevo en los valores propuestos por el DDC).

Un ejemplo de dispositivo real (Motorola V360) ilustrará el concepto expresado en la Figura 2, con la siguiente sustitución de valores (suponiendo que se desea obtener el número de colores de la pantalla):

- Clave $x = NumColors$ (valor asociado, independientemente de que se obtenga a través de WURFL o de CC/PP, $a = 262144$)
- Clave $wurfl(x) = colors$ (valor asociados $\alpha = 262144$)
- Clave $ccpp(x) = BitsPerPixel$ (valor asociado $A = 18$)

Este ejemplo evidencia además la necesidad de hacer transformaciones en los valores de retorno de las distintas APIs al vocabulario común: tanto el vocabulario común como WURFL expresan el número total de colores que la pantalla del V360 pueden manejar, mientras que en CC/PP se maneja el concepto de profundidad de color o número de bits necesario para expresar la gama de colores representable por la pantalla.

Como es evidente, la cantidad de capacidades susceptibles de ser declaradas para un dispositivo con capacidades web puede llegar a ser enorme, por lo que sólo se ha implementado ese vocabulario común para las capacidades estrictamente necesarias para el caso de uso abordado por el prototipo:

- Dimensiones de la pantalla en pixels.
- Número de colores soportado por la pantalla.
- Formatos de lenguaje de hipertexto soportados
- Formatos gráficos soportados por el navegador.

Una de las posibles mejoras de este módulo sería la posibilidad de hacer que, para distintas capacidades de dispositivo (distintas claves del vocabulario común), se pudiesen consultar las APIs en un orden particular. Por ejemplo, podría ser interesante establecer que, mientras el orden de búsqueda de capacidades sea CC/PP-WURFL, en el caso de una propiedad determinada sea más conveniente invertir el orden.

6. Adaptación del lenguaje de marcado

Como se ha comentado anteriormente, en lo que respecta al lenguaje de marcado, existen dos opciones básicamente: servir los contenidos tal cual han sido confeccionados por la Unidad de Gestión Web Municipal del Ayuntamiento de Zaragoza, cuando se detecta que el cliente web es un navegador de escritorio; o servir contenido basado en las Buenas Prácticas para Web móvil, en el caso contrario.

Aprovechando que *XHTML Basic 1.0* es un subconjunto de *XHTML Strict 1.0* y de *XHTML-MP 1.1*, el proceso de transformación del marcado se simplifica. Puesto que los elementos y atributos de *Basic* utilizados en el marcado generado también forman parte del lenguaje *MP 1.1* [9], [10]–, se sirve el mismo documento pero con el DTD correspondiente al lenguaje de marcado soportado por el dispositivo.

Para permitir esa simplificación, se transforman elementos de *Strict* no existentes en *Basic* en construcciones equivalentes en este último (en el sentido de que permiten expresar la misma información de otra forma). Por ejemplo, el elemento `<legend />`, permitido en *Strict 1.0*, es transformado en un elemento `<p />`, válido tanto en *Basic 1.0* como en *MP 1.1*.

La adaptación del contenido no se limita sólo a una mera traducción de *Strict* a *Basic/MP*, sino que se efectúan una serie de transformaciones adicionales para seguir las distintas pautas fijadas en las Buenas Prácticas para la Web Móvil (como se comenta en los subepígrafos siguientes). Es importante destacar que dichas transformaciones no sólo afectan a la reordenación de los elementos del árbol DOM del documento, sino que implican en ocasiones la división de un documento XHTML (versión para navegadores de escritorio) en varios (versión para navegador móvil). Además, existe la posibilidad de marcar contenido en los documentos XHTML de escritorio, que eviten su transformación por parte del software diseñado. Esto puede ser interesante, por ejemplo, para tablas XHTML sencillas (ver 6.4).

6.1. Reubicación del contenido

Este proceso implica la sustitución de partes del documento, como el “Menú de Acceso a Contenido” que se puede ver en la plantilla de

ejemplo de la Figura 1, por un hiperenlace que lleva al documento en el que se muestra el menú. De esta forma, se evita en cada página accedida todo el marcado de esa parte de la plantilla (menú de navegación central del portal) y sólo se servirá esa información cuando el usuario realmente desee acceder a otras partes del portal. Por otro lado, posteriores accesos implicarían el acceso a una copia local del documento, gracias a la cache del navegador, optimizando el ahorro de consumo de datos y el tiempo de latencia al acceder al menú. Además, esta transformación intenta minimizar elementos de navegación en la parte superior de las páginas servidas (pauta *NAVBAR* de las Buenas Prácticas), para que el usuario acceda más fácilmente a la información central del documento (pauta *CENTRAL_MEANING*).

6.2. Particionado de los documentos

Las páginas se parten para evitar que documentos grandes, diseñados para navegadores de escritorio, puedan agotar la memoria disponible en dispositivos con capacidades limitadas. Además, podría permitir que el usuario pudiese echar un vistazo al documento sin tener que descargárselo entero.

Uno de los mejores ejemplos para ilustrar la necesidad de partir documentos cuando exceden cierto tamaño es el acceso a páginas que permiten al ciudadano acceder al contenido de una disposición legal (como la normativa municipal mostrada en la Figura 3). A pesar de que el tamaño de dicha figura no permita observar el contenido, la barra de scroll de la derecha permite al lector intuir el tamaño del texto. Sucesivas pruebas efectuadas demostraron cómo el acceso a este documento provocaba fallos en navegadores web con capacidades de memoria muy limitadas.

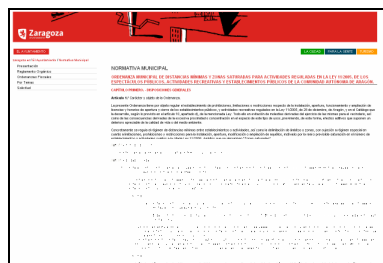


Figura 3. Visualización de una disposición legal

La paginación se hace evitando que el usuario pierda la noción de la parte del documento que está leyendo. Para el caso anterior, el usuario móvil siempre observará, en la parte superior de una página que muestre un fragmento de esa disposición legal, el encabezado con los elementos `<h1/>...<h1/>` en los que se ha estructurado el contenido del texto. En otras palabras, el ciudadano siempre dispondrá en la parte superior del título de la normativa y del capítulo y sección de la misma que se encuentra leyendo (pauta *PAGE_SIZE_USABLE*).

El problema de establecer umbrales adecuados para resolver la partición de un documento no tiene una solución sencilla, pues cada navegador web representa internamente de distinta forma la estructura de un documento XHTML y su contenido (texto e imágenes). Ya que la mayoría de las veces la solución viene dada por el método de prueba y error con dispositivos reales, este es un caso en el que podría ser interesante tener una configuración especial del mecanismo de detección de capacidades, para crear una nueva capacidad en WURFL que indique el umbral para partir documentos a partir de pruebas realizadas con dispositivos y que sólo se consulte a este API.

La solución ofrecida en el prototipo se basa de nuevo en el DDC y establece en 20KB el tamaño de la página (imágenes y texto) para partir el mismo (pauta *PAGE_SIZE_LIMIT*), siendo esta cantidad fácilmente reconfigurable.

Finalmente, cabe destacar otra característica interesante en cuanto al particionado de documentos de gran tamaño, como es la inclusión de elementos de navegabilidad que permitan al usuario avanzar y retroceder a través del documento partido (ver Figura 4).



Figura 4. Elementos de navegación entre fragmentos de un documento partido

Opcionalmente, el prototipo permite ofrecer una serie de enlaces numerados a los distintos fragmentos generados a partir de un documento inicial de gran tamaño (similar a la paginación de resultados de Google). La versión en pruebas en los portales del Ayuntamiento no está configurada para este comportamiento, pues es necesario afinar aún la forma de ofrecer estos enlaces cuando el número de fragmentos es muy grande.

6.3. Eliminación de scripting

El soporte de scripting es muy heterogéneo en los navegadores móviles y hay que evitar que, a causa de su uso, el acceso a una página implique un error del navegador web móvil. Algunos documentos que utilizan scripting de forma intensiva (p.e., validación de formularios en el lado del cliente) funcionarán adecuadamente debido a que los contenidos servidos por la Unidad de Gestión Web Municipal siguen estrictamente las pautas de accesibilidad de W3C. Esto permite que los formularios ofrecidos al ciudadano sean plenamente funcionales incluso cuando las capacidades de ejecución de scripts estén desactivadas en el navegador o cuando el script es eliminado al servirlo a un dispositivo móvil (pauta *OBJECTS_OR_SCRIPT*).

6.4. Transformación de tablas y mapas de imágenes

La adaptación de tablas es un asunto complicado, sobre todo si contienen un gran número de columnas (lo cual es un problema para anchos de pantalla reducidos) o de tablas anidadas (que muchos navegadores móviles no soportan). Ante los distintos casos de tablas muy grandes o complejas encontradas en algunos documentos servidos por el Ayuntamiento, se ha optado por convertir las tablas en listas de definición (mediante el uso adecuado de elementos `<dl/>`, `<dt/>`, `<dd/>`, etc.).

Teniendo en cuenta la semántica de una tabla, en las que se puede preferir que la lista recorra la tabla por filas o por columnas según el tipo de información que esté ofreciendo la tabla, se puede marcar las tablas de forma que se advierta al mecanismo de transformación que se recorra la tabla de una forma o de la otra al transformar (pauta *TABLES_ALTERNATIVE*).

En el caso de los mapas de imágenes, para un dispositivo móvil se ofrecería como un gráfico seguido de la lista de hiperenlaces contenida en el mapa.

6.5. Colaboración con el módulo de adaptación de imágenes

El módulo de adaptación del lenguaje de marcado servido colabora con el de adaptación de imágenes, puesto que, por la pauta *IMAGES_SPECIFY_SIZE* se deben indicar en los atributos *width* y *height* de un elemento ** valores que concuerden con el ancho y alto de la imagen referenciada. Así, se pretende evitar el redimensionamiento de la imagen al tamaño de la pantalla en el lado del cliente.

7. Adaptación de imágenes

Básicamente, este módulo se encarga de convertir las imágenes estáticas disponibles en el servidor, para navegadores de escritorio, a un formato adecuado para el dispositivo cliente (adaptando formato gráfico, dimensiones –redimensionando respecto del ancho de pantalla, respetando la razón de aspecto– y profundidad de color). Está implementado como otro filtro Servlet que consume la información ofrecida por el módulo de detección del dispositivo y adapta las imágenes solicitadas en caso de que no se trate de un cliente web de escritorio.

Al igual que en el caso del acceso a APIs de detección de dispositivos y sus capacidades, se decidió crear un envoltorio, utilizando una interfaz común que hace homogéneas las llamadas al API subyacente de manejo de recursos gráficos.

8. Cache en el servidor

El prototipo implementado utiliza además un sistema de cache persistente que hace que, cuando un cliente pretende acceder a un recurso (documento o imagen) al que ya haya accedido un dispositivo con las mismas características de tamaño de pantalla, profundidad de color y formatos gráficos soportados, se le servirá la copia en cache, evitando una nueva transcodificación.

9. Lista blanca

Entre otros aspectos de usabilidad, se ha configurado un aviso que será mostrado al usuario de un dispositivo móvil cuando acceda a un enlace que le lleve fuera del dominio o dominios gestionados por la Unidad de Gestión Web Municipal del Ayuntamiento de Zaragoza.

El aviso advierte al usuario de que, potencialmente, podría acceder a recursos en formatos no aceptados por el navegador, o cuyo tamaño implicará un gran tráfico de datos y un coste importante para el usuario, e incluso el agotamiento de la memoria del dispositivo.

La configuración de esta funcionalidad se lleva a cabo a través de un archivo (lista blanca, en oposición a las *black lists* utilizadas para evitar fuentes de *spam*), en el que cual se declaran mediante expresiones regulares las direcciones de los servidores que se conoce que tienen instalado el prototipo desarrollado.

10. Prototipado rápido

Se podrían comentar otras características de usabilidad implementadas en el prototipo, todas ellas configurables de forma sencilla a través de archivos de configuración siguiendo el paradigma de prototipado rápido (alta granularidad de sintonización del prototipo, pero con muchos parámetros pendientes de ajustar). Sólo se han comentado las más destacables, por motivos de espacio.

11. Conclusiones

Se ha intentado demostrar cómo se puede abordar la extensión del envío de contenidos web a navegadores con capacidades limitadas a partir del concepto inicial de envío a navegadores web de escritorio. El enfoque de simplificar la solución tratando a todo dispositivo móvil como un mismo DDC y centrar el esfuerzo de transformación de contenidos en el caso del envío de imágenes ha permitido crear un prototipo funcional en un tiempo limitado, ofreciendo tiempo para implementar el soporte de funcionalidades complejas pero necesarias como el particionado de documentos o el uso de cache en el lado del servidor para minimizar transcodificación.

Este servicio ha sido propuesto a W3C como Implementación de Referencia para que las Buenas Prácticas para la Web Móvil puedan pasar

a ser una Recomendación W3C. Se invita al lector a que consulte el informe en [6], ya que indica con detalle todas las buenas prácticas que el prototipo cumple.

El panorama de la web móvil se muestra cada vez más prometedor, desde la perspectiva de estándares y herramientas libres que permitan a cualquier organización o individuo ofrecer contenidos web a cualquier tipo de dispositivo cliente.

Las Buenas Prácticas para la Web Móvil se han consolidado y comunidades de desarrolladores (como la de .mobi, impulsada por los agentes más importantes en la web móvil) invitan a utilizar sus pautas como garantes de una experiencia de usuario apropiada para cualquier dispositivo con capacidades web. Algunos temas en los que el grupo podría centrarse en el futuro más cercano podrían ser pautas para la adaptación de contenidos o Mobile AJAX.

El grupo de trabajo de W3C sobre Descripción de Dispositivos comienza a trabajar en el diseño de un API común para acceso a repositorios de descripciones de dispositivos, con lo que se obtendría un amplio vocabulario común universal.

Nuevos grupos de trabajo aparecen para intentar estandarizar diversos aspectos de la web móvil. En especial, se muestra como muy interesante la convergencia de OMA y W3C en cuanto a lenguaje de marcado propuesto para los dispositivos móviles en el futuro (XHTML Basic 1.1/MP 1.2).

Fundación CTIC se encuentra colaborando *activamente* en varios de los grupos de trabajo mencionados, con la intención de ayudar al desarrollo de estándares para la web móvil (incluyendo casos de uso como el que se trata en este documento). La relación con el Ayuntamiento de Zaragoza permitirá implementar demostradores de todas estas tecnologías, y hará que los ciudadanos y visitantes de la capital de Aragón sean pioneros en el acceso a la información de forma universal, añadiendo a los obligatorios requisitos de Accesibilidad la posibilidad de acceso multidispositivo.

Este prototipo es tan sólo el primer paso de una andadura muy larga, y ha permitido que el Ayuntamiento de Zaragoza haya dado un primer paso hacia la publicación de contenidos para cualquier dispositivo con capacidades web.

Cabe también destacar que, si bien el enfoque del prototipo se ha centrado en la adaptación de contenidos, existen aproximaciones como MyMobileWeb [11] que permiten el desarrollo de verdaderos servicios multidispositivo, basándose en el concepto de diseño único de una aplicación o un servicio para múltiples dispositivos. La arquitectura SOA de este proyecto se presenta como el enfoque ideal para lograr eAdministración con requisitos de multicanalidad. Por su filosofía de software abierto y basado en estándares, Fundación CTIC, dentro de sus actividades de I+D, se ha comprometido con este proyecto y espera poder ayudar a que las distintas AAPP amplíen y mejoren los contenidos y servicios que actualmente ofrece a los ciudadanos sobre cualquier mecanismo de acceso.

Referencias

- [1] UAPProf Specification, 2001. <http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf>
- [2] WURFL (Wireless Universal Resource File). <http://wurfl.sourceforge.net>
- [3] XHTML 1.0 (Second Edition), <http://www.w3.org/TR/xhtml1/>
- [4] XHTML Basic 1.0, 2000. <http://www.w3.org/TR/2000/REC-xhtml-basic-20001219/>
- [5] JSR 154: Java Servlet 2.4 Specification, 2003, <http://jcp.org/en/jsr/detail?id=154>
- [6] Mobile Web Best Practices 1.0 (Basic Guidelines), 2006. <http://www.w3.org/TR/mobile-bp/>
- [7] The Essentials of Filters, 2004. <http://java.sun.com/products/servlet/Filters.html>
- [8] JSR-188: CC/PP Processing, 2003. <http://jcp.org/en/jsr/detail?id=188>
- [9] XHTML-MP 1.1, http://www.openmobilealliance.org/release_program/docs/Browsing/V2_2-20061020-A/OMA-WAP-XHTMLMP-V1_1-20061020-A.pdf
- [10] Comparison of XHTML-MP and XHTML Basic, 2007. <http://rabin.mobi/docs/MP-Basic.xhtml>
- [11] Sitio del proyecto MyMobileWeb: <http://mobileweb.morfeo-project.org>