

Guía metodológica para el desarrollo de sitios web móviles adaptativos con el enfoque Mobile First

Vera, Pablo M.^{1,2}; Rodríguez, Rocío A.^{1,2}; Martínez, M. Roxana²

¹ *Departamento de Ingeniería e Investigaciones Tecnológicas*

Universidad Nacional de La Matanza

Florencio Varela 1903, San Justo, Buenos Aires

gidfis@ing.unlam.edu.ar

² *Centro de Altos Estudios en Tecnología Informática*

Universidad Abierta Interamericana

Montes de Oca 745, Barracas, Ciudad Autónoma de Buenos Aires

Resumen

Si bien tener un sitio web móvil permite establecer los contenidos que se van a priorizar, disminuir la cantidad de información visual que carece de importancia, ofreciendo una solución ágil para un usuario en movimiento; son pocas las instituciones que cuentan con una solución web de escritorio y otra móvil; siendo el diseño adaptativo (responsive design) una primera iniciativa significativa para mejorar la visualización de los sitios en pantallas reducidas. El diseño adaptativo es una buena solución para poder ajustar los controles de una página web a distintos tamaños de pantalla. En este artículo se analizan las ventajas y desventajas del diseño adaptativo, cuando el mismo se aplica para mejorar la visualización de sitios web en dispositivos móviles. Con la finalidad de generar diseños adaptativos usables en este artículo se presenta una metodología formada por pautas que permitirán solucionar los problemas más frecuentes a los que se enfrentan los usuarios de la web móvil.

1. Introducción

Actualmente existe una clara tendencia en el crecimiento del uso de los dispositivos móviles para conectarse a internet. Cada vez menos personas esperan llegar a su casa y sentarse frente a una computadora para consultar los emails, leer alguna noticia o consultar alguna página particular. La disponibilidad y practicidad de los dispositivos móviles hacen que sean la primera fuente de referencia para estos casos. Sin embargo, al querer ingresar a muchos sitios web desde los dispositivos móviles surgen diversos problemas dado que la mayoría de los sitios no fueron diseñados y estructurados pensando en dispositivos móviles sino para ser visualizados desde una computadora. Por ello se

propone el enfoque mobile first (pensar primero en el usuario móvil), “es pensar en el móvil como punto de partida, esto obliga a concentrarse en lo esencial de un producto y a hacer foco solo en lo que tiene sentido para este dispositivo” [1].

En este artículo se plantean primeramente los problemas a los que se enfrentan los usuarios cuando acceden desde sus dispositivos móviles a la web (sección 2), algunos de estos problemas no son ajenos a los sitios web móviles (sección 3). Dado que son muy escasos los desarrollos de sitios web móviles, el diseño adaptativo es una importante iniciativa para ofrecer una vista adaptada al dispositivo de acceso (esto se aborda en la sección 4). En la sección 5 se ofrece un relevamiento de frameworks, que permiten construir soluciones adaptativas, comparándolos y estableciendo las posibilidades que estos ofrecen para evitar los problemas detallados en las secciones anteriores. Finalmente, en la sección 6 se propone un conjunto de pautas para implementar el diseño adaptativo asegurando la calidad del resultado; en la sección 7 se presentan las conclusiones y en la sección 8 las referencias.

2. Los usuarios móviles y la web

Las dificultades de navegar en un dispositivo móvil, un sitio web no preparado para tal fin, son muy diversas. Basta con mencionar dos ellas para poder comprender la dificultad de los usuarios móviles: (1) La cantidad de elementos que se muestran en la pantalla sobre los cuales el usuario debe hacer zoom para poder visualizarlos, (2) El scroll que es necesario hacer para moverse por la pantalla.

A modo ilustrativo se muestra en la figura 1, dos sitios web correspondientes a instituciones universitarias argentinas: Universidad Nacional de la Patagonia San Juan Bosco y Universidad Nacional del Centro de la

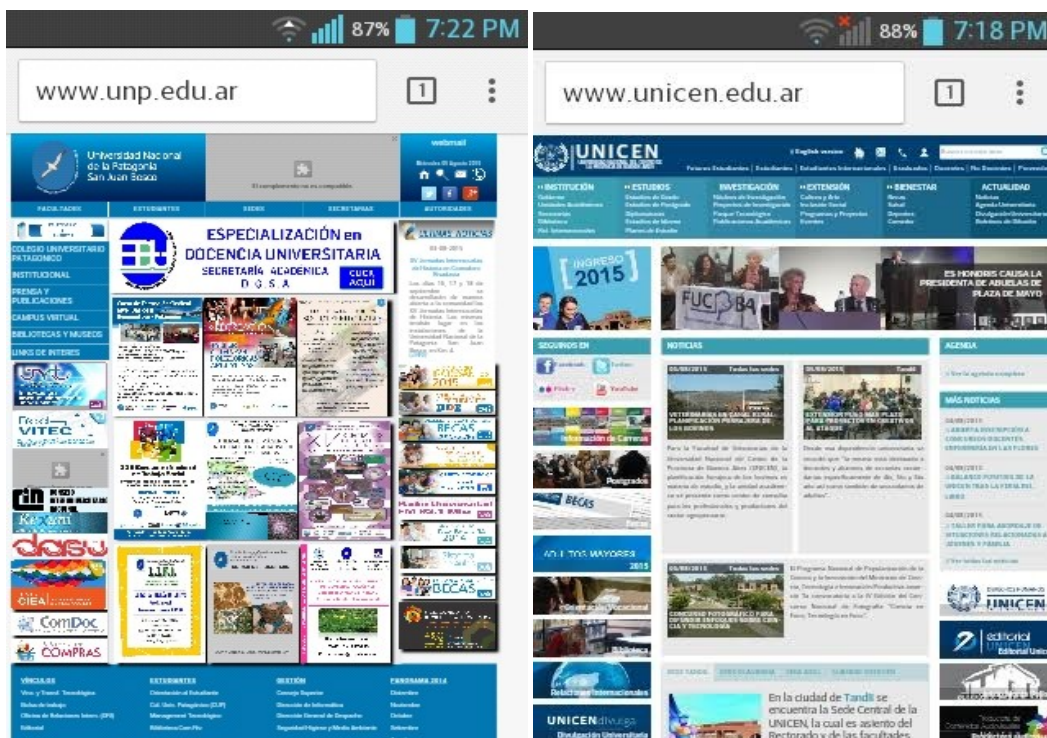


Figura 1. Visualización desde un dispositivo móvil de sitios web vinculados con instituciones universitarias

Provincia de Buenos Aires, los cuales han sido accedidos desde un dispositivo móvil (teléfono celular avanzado - smartphone).

Evidentemente cuantos más elementos haya en pantalla más engorroso será para el usuario final poder ubicar lo que necesita [2]. La sobrecarga de información de los sitios web fijos visualizada en los dispositivos móviles va en contra de la usabilidad. “Después de todo la usabilidad sólo significa que algo funcione bien: que una persona con capacidad y experiencia media (o incluso por debajo de la media) pueda ser capaz de usar algo... con el objetivo deseado sin sentirse completamente frustrado” [3].

Por otra parte, hay sitios web que utilizan tecnologías que requieren plugins tal como Flash y no son visibles desde algunos dispositivos móviles, entonces el usuario no puede visualizar parte del contenido del sitio. A modo de ejemplo se muestra en la figura 2 la captura efectuada desde un celular del sitio web de la Universidad Católica de Santa Fe (Argentina). Puede notarse que el banner y otros elementos del sitio no se visualizan.

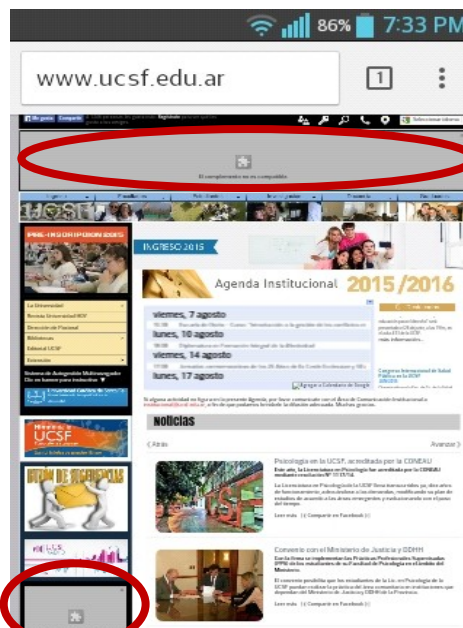


Figura 2. Elementos en Flash no visibles desde dispositivos móviles

3. La web móvil

La web móvil tiene diversos problemas que parecen ser heredados de las páginas web tradicionales esto se debe a que no se ha puesto el foco en los usuarios móviles. Entre esos inconvenientes es posible mencionar:

1. Peso de la página
2. Uso y abuso de imágenes
3. Imágenes no preparadas para dispositivos móviles
4. Contenido no priorizado
5. Páginas demasiado grandes
6. Visualización pobre, teniendo que hacer zoom y scroll en el sitio

Para el desarrollo de sitios web móviles el W3C (Consortio Web a Nivel Internacional) brinda una guía de buena de buenas prácticas a tener en cuenta así como también establece el conjunto de características mínimas que un dispositivo móvil debe cumplir para navegar por internet. Siguiendo esas pautas es posible desarrollar sitios web móviles accesibles desde cualquier dispositivo [4]. En cambio si solo se busca llegar a una audiencia con teléfonos de gama media-alta, se puede optar por desarrollar un sitio móvil en HTML5, existiendo gran cantidad de frameworks que combinando hojas de estilo CSS y javascript logran crear interfaces adecuadas para dispositivos con pantalla táctil de tamaño reducido incluso simulando la interfaz nativa de los distintos sistemas operativos móviles. Como por ejemplo JQueryMobile [5], IUI [6]. Aún con estas posibilidades son muy pocos los sitios web que se desarrollan específicamente para dispositivos móviles habiendo visto incluso algunos esfuerzos que rápidamente quedan desactualizados por el hecho de tener que mantener dos sitios web diferentes.

En un relevamiento realizado previamente pudo observarse que de los 135 municipios que conforman la provincia de Buenos Aires, a lo que se suma la Ciudad Autónoma de Buenos Aires (dando por resultado 136 municipios), tan sólo 5 tienen sitio web móvil, es decir tan sólo un 3,7% de ellos [7]. Sin embargo 40 de ellos tienen diseño adaptativo un 29, 4%.

4. Diseño Adaptativo

Para lograr evitar desarrollar distintas versiones de un sitio web y hacer que el mismo se visualice correctamente tanto en un dispositivo móvil como en una computadora de escritorio nace el diseño adaptativo.

Es importante comprender que el diseño adaptativo no es una versión diferente del sitio web, sino que es el mismo sitio que se va adaptando y acomodando sus elementos para una mejor visualización en el dispositivo

de acceso. Con lo cual no cuenta con todas las ventajas que tendría la construcción de un sitio móvil.

El diseño adaptativo se basa en la utilización de hojas de estilo de nivel 3 con CSS Media Queries [8] lo que permite aplicar distintos estilos según sea el tamaño de la pantalla en la cual se está visualizando el sitio web. De esta forma una misma página se podrá visualizar “correctamente” indistintamente del tamaño o resolución de pantalla del dispositivo de acceso.

“El diseño web adaptativo utiliza hojas de estilo en cascada complejas, para modificar de forma fluida la apariencia de un sitio web dependiendo del ancho de la ventana de visualización del dispositivo en uso” [9].

La figura 3 muestra como varía la visualización de un mismo sitio web al cambiar el tamaño de la pantalla. En la misma pueden diferenciarse tres medidas: (A) tamaño grande, que es adecuado por ejemplo para la visualización en una computadora de escritorio; (B) tamaño mediano, que se adapta por ejemplo para ser visualizada en una tablet; (C) tamaño pequeño adecuado para un dispositivo con pantalla reducida como por ejemplo un teléfono celular.

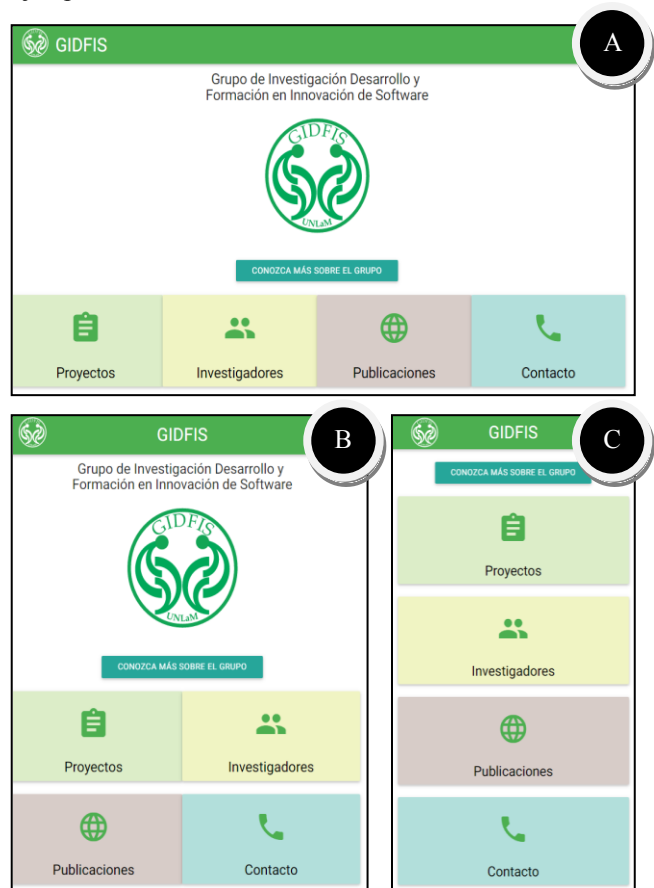


Figura 3. Visualización de un sitio adaptativo en tres tamaños de pantalla

Sin embargo, la mayoría de los sitios implementados utilizando esta tecnología no lo hacen pensando en el usuario móvil y presuponen que por el solo hecho de emplearla ya están creando un sitio amigable para un usuario móvil.

Un mal uso del diseño adaptativo incorporará prácticamente todos los errores antes mencionados:

- (1) Peso de la página
- (2) Uso y abuso de imágenes
- (3) Imágenes no preparadas para dispositivos móviles
- (4) Contenido no priorizado
- (5) Páginas demasiado grandes

Solamente mejora el punto 6 haciendo que el usuario no deba hacer zoom y scroll en ambos sentidos para visualizar la página, sino que limita el scroll al menos en una sola dirección y no necesita hacer zoom ya que los controles se adaptan al ancho de la pantalla.

5. Framework Adaptativos

Tal como se comentó anteriormente, el desarrollo adaptativo se basa en la utilización de CSS Media Queries aplicando distintos estilos según el tamaño de visualización de la página. No obstante existen diversos frameworks con estilos pre-diseñados que se pueden aplicar rápidamente para la construcción de sitios web adaptativos. Estos estilos generalmente están complementados con archivos JavaScript que agregan funcionalidad a ciertos controles, haciéndolos dinámicos e incluso permitiendo además animar parte de la interfaz gráfica.

Sin embargo, al utilizarlos hemos notado que la mayoría de estos frameworks no están correctamente diseñados ya que se basan en unidades absolutas (píxeles) para determinar los distintos tamaños de pantalla cuando lo correcto sería utilizar unidades relativas (em o porcentajes) para asegurar una correcta visualización en pantallas pequeñas más allá de la resolución que posean.

A continuación se explica cómo se logra en diversos frameworks tener una visualización en base al dispositivo de acceso:

- Materializecss [10]: Este framework está basado en la corriente de diseño Material Design [11] creada por Google. Divide la pantalla en 12 columnas las cuales pueden utilizarse para organizar el contenido pudiendo acomodar los controles en dicha grilla y cambiar el tamaño y/o ubicación de los mismos considerando tres tamaños de pantalla posible: grande, mediano ó chico.
Por ejemplo si se desea crear una página web que tenga tres columnas para la versión grande, dos columnas para la versión mediana y una para la

versión móvil chica, se puede utilizar el siguiente código:

```
<div class="col s12 m6 l4">
```

Como puede observarse en la línea de código, los tamaños small, medium, large, son representados con la inicial de cada palabra, escribiéndose a continuación la cantidad de columnas que ocupará el contenido del div. La cantidad de columnas visibles (CCV) resultará de considerar la cantidad de columnas definidas (CCD) en el código dividido el total de columnas que es 12, siendo siempre un valor fijo. Lo cual se expresa en la fórmula 1.

$$CCV = CCD / 12 \quad (1)$$

De este modo el código presentado hará que el contenido del div se visualice utilizando 4 de las 12 columnas disponibles cuando la pantalla es grande (l4), por lo tanto al mismo tiempo pueden visualizarse 3 de estos divs. En cambio si la pantalla es de tamaño mediano (m6) el div ocupara la mitad de la pantalla y para el caso de la pantalla chica (s12) cada div ocupara todo el ancho de pantalla.

Además dispone de modificadores para poder organizar el contenido permitiendo cambiar la visualización según el tamaño de pantalla en cuanto a contenido visible, centrado, espaciado, etc.

- Bootstrap [12]: La utilización del espacio disponible en la pantalla y de los modificadores de contenido es similar a Materializecss dividiéndola en 12 columnas pero este framework agrega un tamaño adicional, el extra pequeño, quedando conformado de la siguiente manera:
 - Grande: para computadoras de escritorios con pantallas grandes
 - Mediano: para computadoras de escritorio, notebooks o netbooks con pantallas de menor tamaño que las primeras
 - Pequeño: para tablets
 - Extra Pequeño: para teléfonos celulares
- Foundation [13]: Es un framework adaptativo diseñado para seguir la corriente Mobile First. Es uno de los pocos frameworks que trabaja con tamaños de pantalla definidos por unidades relativas. Permite trabajar con grillas con cantidad de columnas parametrizables y además tiene una herramienta para generar las hojas de estilo solo con los componentes utilizados reduciendo notablemente el tamaño del archivo vinculado en las páginas web.
- Skeleton [14]: Es un conjunto de estilos para poder diseñar sitios adaptativos. No es un framework tan

completo como los otros pero se caracteriza por ser liviano y simple. Incluye el manejo de pantalla también mediante una grilla de 12 columnas y define 5 tamaños de pantalla predefinidos: Escritorio de alta resolución, Escritorio, Tablet, Celular con pantalla Grande, Celular.

6. Pautas para Diseños Adaptativos Usables

Para realizar un correcto diseño de un sitio web utilizando diseño adaptativo es preciso considerar algunos principios que resultan fundamentales. Es por ello que en esta sección se proponen un conjunto de pautas, las cuales se muestran en la figura 4.

Estas pautas son necesarias para poder asegurar un sitio web adaptativo de calidad. A continuación, se detallan brevemente cada una de ellas:

1. Establecer prioridades de contenidos: No se deben volcar todos los contenidos del sitio web en un sitio web móvil. Deben priorizarse los contenidos más relevantes del sitio poniendo más al alcance de la mano lo primordial para un usuario móvil: direcciones, forma de contacto, horarios y ubicación, son los principales ítems que un usuario en movimiento puede necesitar de cualquier institución que esté consultando, por lo que debería ser una de las primeras cosas que usuario visualice sin necesidad de realizar scroll. “Cuando esté escribiendo para usuarios móviles, enfoque su atención en el contenido esencial” [15].

2. Limitar el contenido de cada página: el W3C [4] establece que tiene que existir un balance entre la profundidad de un sitio web y el contenido de cada página por que lo no deben sobrecargarse las páginas sino dividir las en unidades lógicas correctamente estructuradas, evitando además el excesivo scroll vertical por parte del usuario.
3. Controlar el scroll y limitarlo a una sola dirección: [4]. Esto evitará que el usuario tenga que desplazarse tanto horizontal como verticalmente para alcanzar el contenido.
4. Construir páginas livianas: Si un sitio tiene mucho contenido e imágenes la descarga será lenta en especial en un dispositivo móvil. Además para el caso de que no se disponga de una conexión WIFI sino que se esté conectado a una red de datos, la descarga será aún más lenta incluso en caso del uso de redes de alta velocidad como por ejemplo 4G siempre puede existir pérdida o disminución en la señal por lo que es conveniente reducir el tamaño final de cada página a ser visualizada. Una estrategia sencilla que permite reducir el tamaño es la reducción de espacios en blanco [16]. Quién desarrolla la solución podrá utilizar espacios, tabulaciones y demás elementos que permitan visualizar mejor el código. Luego existen herramientas que se encargan de eliminar esos espacios en blanco, comentarios, tabulaciones e incluso optimizan los códigos javascript renombrando los nombres de las variables y funciones, con identificadores más cortos.

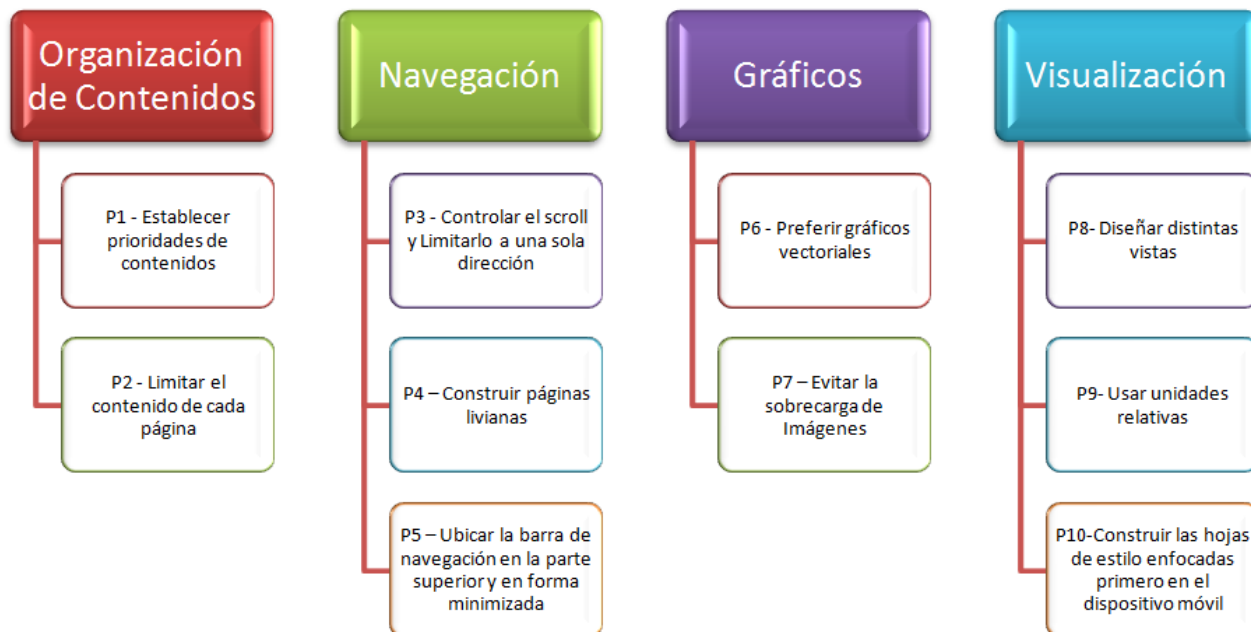


Figura 4. Pautas para el Diseño Adaptativo con el Enfoque Mobile First

La minimización puede realizarse de dos formas:

- a) Hosteando directamente en el servidor las páginas minimizadas. Esto puede realizarse de forma manual con alguna herramienta de minimización como por ejemplo HTML Minifier [17]. Siendo recomendable tener algún proceso automático de minimización que se realice cada vez que se haga una implementación de una nueva versión en el servidor productivo. En la figura 5 se muestra a la izquierda el lenguaje de etiquetado de un sitio web adaptativo copiado tal como lo descarga el usuario al visualizar la página principal, a la derecha en la parte superior se ha pegado el resultado luego de realizar la minimización. En la parte inferior derecha la herramienta web utilizada para generarla HTML Minifier [17].
- b) Hosteando las páginas sin minimizar, pero haciendo el proceso de minimización cuando un usuario accede a la página. Esta última opción solo es posible para sitios dinámicos donde mediante plugins o herramientas de los distintos entornos se dispare el proceso de minimización cada vez que se recupera una página web.

De esta forma se logra disminuir el peso de las páginas (KB) a descargar por los usuarios en sus dispositivos móviles. Es bueno que al momento de la construcción de una página se realice de una forma cómoda utilizando espacios, comentarios y otros recursos que puedan mejorar la legibilidad del lenguaje de etiquetado. Pero al subirlas a los servidores este tipo de herramientas son muy útiles para que las páginas a subir sean más livianas. De este modo el desarrollador puede trabajar cómodamente.

Un caso opuesto a lo planteado (en donde se hacía uso de los comentarios y espacios para mejorar la legibilidad de lo realizado), es cuando se abusa de los comentarios para agregar elementos que no aportan valor alguno. Un ejemplo de ello es el sitio web adaptativo del Municipio de Almirante Brown (Provincia de Buenos Aires), en donde al finalizar el lenguaje de etiquetado pueden verse gráficos hechos con caracteres (ver figura 6). Actualmente no son muy visibles los ASCIIART en las páginas web pero lamentablemente si aparecen embebidos en el lenguaje de etiquetado aumentado el peso de la página que debe descargar el usuario final.

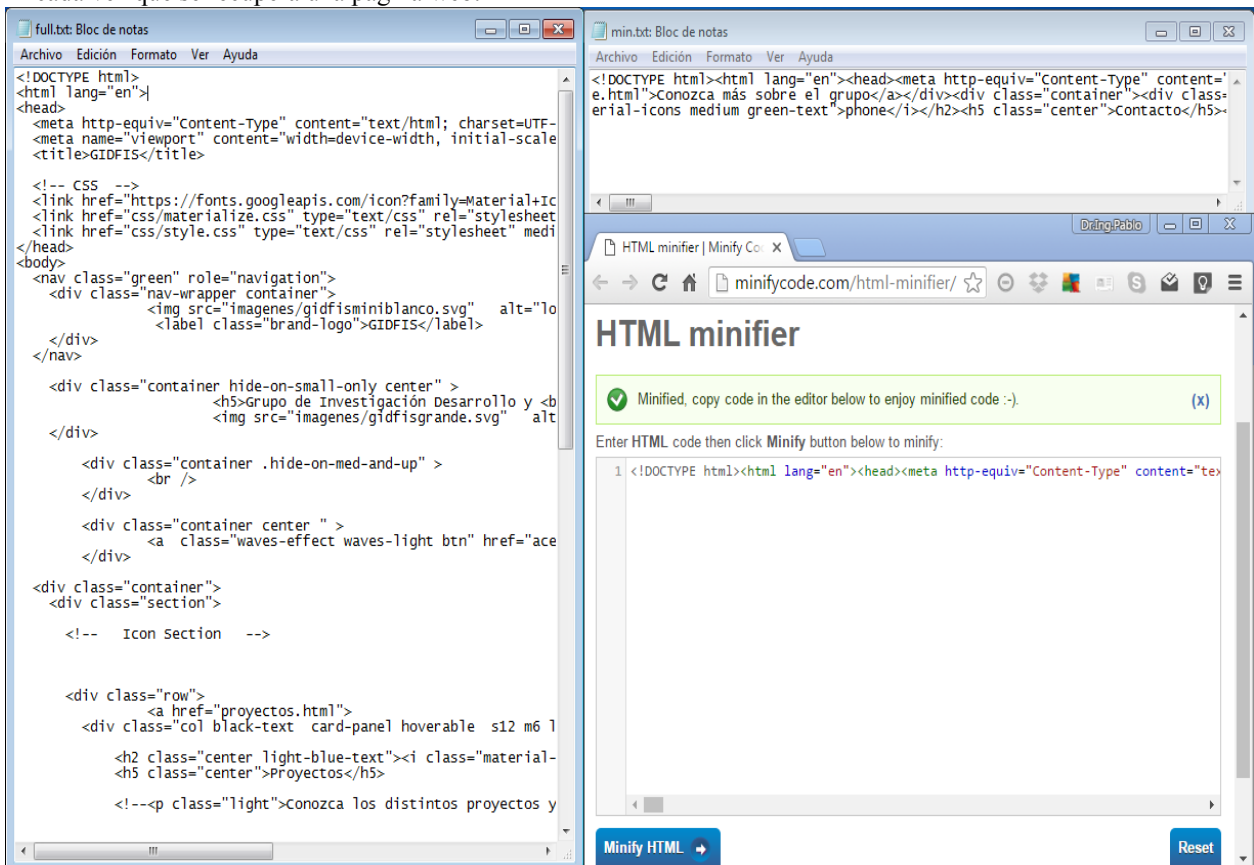


Figura 5. Herramienta online que permite disminuir el peso del código

imágenes las mismas pueden perder calidad al mostrarse en un tamaño demasiado grande, o si se opta por incorporar una imagen de buena calidad para que se vea correctamente la misma será ineludiblemente de mayor peso. Es por eso que se recomienda siempre que sea posible el reemplazo de las imágenes tradicionales por el uso de imágenes vectoriales SVG [16]. Las imágenes SVG son dibujadas por el navegador y están especificadas mediante XML por lo que pueden ser redimensionadas sin perder calidad alguna [18]. En la figura 8 se muestran capturas de pantalla al visualizar un archivo SVG en un sitio web desde un celular. A la izquierda se visualiza la imagen en tamaño completo, luego hacia la derecha se muestran distintos niveles de zoom realizados donde puede apreciarse que no se pierde calidad al utilizar este tipo de imágenes.

7. Evitar la sobrecarga de imágenes: Dado que en un dispositivo móvil la pantalla es reducida la atención del usuario en un contexto dinámico se dispersará aún más. Cuanto más sobrecargada de elementos esté la pantalla menos usable será la misma. En la figura 9 se muestra a modo de ejemplo el caso del Municipio de Marcos Paz en donde todo el contenido de la página principal se muestra por medio de imágenes.

“Las aplicaciones móviles presentan... mayores desafíos ya que los usuarios interactúan con ellas en contextos dinámicos, en muchos casos realizando otras tareas al mismo tiempo, con su atención no siempre centrada en el dispositivo. Lo cual indica



Figura 8. Imagen en SVG con distintos niveles de zoom

que la forma de uso debe ser lo más simple y clara para que el usuario no se frustre al intentar utilizar el software en cuestión” [19].



Figura 9. Ejemplo de sobrecarga de imágenes

8. Diseñar distintas vistas: Pensar en al menos 3 tamaños de pantalla, pequeño, mediano y grande, (esto viene soportado por diversos framework tal como se explicó en la sección anterior). No todos los diseños adaptativos toman en cuenta verdaderamente las posibilidades de los distintos

dispositivos, tomando el ejemplo presentado en la figura 10, puede verse que la parte superior es muy grande y desperdicia pantalla. En la figura 10 se presenta la captura de pantalla en un teléfono celular que cuenta con una pantalla de 5,5 pulgadas y una resolución de 1440 x 2560, pero al verse horizontalmente no se aprovecha la pantalla quedando mucho espacio en blanco y obligando al usuario a realizar scroll para acceder al contenido.



Figura 10. Ejemplo de sobrecarga de imágenes

9. Usar unidades relativas: Usar siempre unidades relativas [16], lo cual va a permitir que el contenido se ajuste al tamaño de pantalla, considerando la resolución de la misma. Esto va a implicar no utilizar una cierta cantidad fija de pixeles para indicar un tamaño (px) sino incorporar unidades relativas por ejemplo al tamaño de letra por defecto (em).
 “Para que un sitio sea usable cuando se ve en una variedad de tamaños de pantalla, un cierto grado de flexibilidad debe ser incorporada en el diseño. Div, fuentes, botones, y todos los demás elementos deben ser capaces de crecer y contraerse sobre la base del tamaño de la pantalla. En el diseño web adaptativo, esto se logra mediante el abandono de los pixeles, en favor de porcentajes o unidades relativas llamadas ems” [20].
10. Construir las hojas de estilo enfocadas primero en el dispositivo móvil: Los estilos se deben escribir para lograr una correcta visualización en los dispositivos móviles y luego utilizando CSS Media Queries adaptar la visualización a pantallas de mayor tamaño. Al utilizar CSS Media Queries se definen lo que se conoce como breakpoints para aplicar diferentes estilos según el área de visualización sea mayor o menor a determinado tamaño. Por ejemplo el siguiente CSS cambia el color de fondo según el área de visualización:

```
body
{
  background-color: lightgreen;
```

```
}
@media only screen and (max-width: 50em)
{
  body
  {
    background-color: red;
  }
}
```

```
@media only screen and (max-width: 25em)
{
  body
  {
    background-color: yellow;
  }
}
```

Este CSS no es Mobile First ya que se define un color de fondo verde que luego se modifica a rojo cuando la pantalla tiene 50em o menos y luego se modifica a amarillo en una pantalla pequeña cuya área de visualización es de 25em o menos. Es decir se diseñó pensando primero en cómo se visualiza el contenido en una pantalla grande y se modificó luego para pantallas más pequeñas. Esto se lo denomina diseño Mobile First. Para adaptar esta hoja de estilo siguiendo la corriente Mobile First se debe cambiar el enfoque haciendo que por defecto el fondo sea el adecuado para una pantalla pequeña (amarillo) y luego cambie el color si la pantalla es más grande. El CSS Mobile First resultante es:

```
body
{
  background-color: yellow;
}
```

```
@media only screen and (min-width: 25em)
{
  body
  {
    background-color: red;
  }
}
```

```
@media only screen and (min-width: 50em)
{
  body
  {
    background-color: lightgreen;
  }
}
```

Esto es solo un ejemplo simplificado ya que en cada breakpoint en un diseño real, las hojas de estilo

modifican como se ubican y visualizan los controles en la pantalla para lograr una mejor experiencia de uso.

Estas 10 pautas deben ser consideradas al momento de diseñar un sitio web adaptativo. Las mismas surgen de una revisión exhaustiva de la bibliografía y de relevamientos realizados en soluciones existentes donde se ponen de manifiesto errores que podrían ser solucionados de considerarse las mismas. En resumen, se debe diseñar primero pensando en el usuario móvil, diseñando y probando el sitio para que se vea correctamente primero en una pantalla reducida y luego para las pantallas más grandes, esto muchas veces hace que se tengan que cambiar los controles elegidos y dividir la información en unidades más pequeñas.

7. Conclusiones

Los sitios web adaptativos son una buena solución para permitir visualizar correctamente los sitios web en los dispositivos móviles así como también para reducir el esfuerzo necesario en desarrollo al evitar tener que realizar varias versiones de un mismo sitio. Sin embargo, en la mayoría de los casos la falencia está en el diseño elegido para los mismos, ya que no se toma en cuenta al usuario móvil como punto de partida. Las pautas propuestas en este artículo contribuyen a construir sitios adaptativos con el enfoque mobile first, es decir tomando como punto de partida al usuario móvil, teniendo en cuenta sus necesidades y posibilidades primero para luego sí, mejorar la visualización en pantallas de mayor tamaño. Los frameworks para el desarrollo de sitios web adaptativos actuales en su mayoría se encuentran mal diseñados ya que trabajan con unidades absolutas en lugar de utilizar unidades relativas para determinar el tamaño de la pantalla visualizada ya que hoy en día son cada vez más comunes los dispositivos de pequeño tamaño con pantallas de muy alta resolución por lo tanto deben adaptarse para obtener mejores resultados.

8. Referencias

- [1] Cuello J.; Vittone J. "Diseñando Apps móviles". Primera edición: junio de 2013. ISBN: 978-84-616- 5070-5
- [2] Nielsen, J; Loranger, H. "Usabilidad. Prioridad en el diseño web". 2006.
- [3] Krug, S. "No me hagas pensar". Prentice-Hall, 2006.
- [4] W3C. Mobile Web Best Practices 1.0. 2008
<https://www.w3.org/TR/mobile-bp/>
- [5] jQuery. "A Touch-Optimized Web Framework", 2010
<https://jquerymobile.com/>
- [6] Hewitt J. "iUI web framework for smartphones & high-end devices", 2007
<http://www.iui-js.org/>
- [7] Rodríguez, R., Vera, P., Marko, I., Alderete C., Conca A. "El Gobierno Electrónico y la Implementación de las TIC para Brindar Nuevos Canales de Comunicación. *Revista Latinoamericana de Ingeniería de Software*, 3(5), 2015, pp. 187-196. ISSN 2314-2642
<http://sistemas.unla.edu.ar/sistemas/redisla/ReLAIS/relais-v3-n5-187-196.pdf>
- [8] W3C Media Queries. 2012
<https://www.w3.org/TR/css3-mediaqueries/>
- [9] Walsh, T. A., McMinn, P., & Kapfhammer, G. M. "Automatic Detection of Potential Layout Faults Following Changes to Responsive Web Pages". In *Automated Software Engineering (ASE)*, 2015 30th IEEE/ACM International Conference on (pp. 709-714). IEEE.
- [10] Wang A, Chang A, Mark A, Louie K, "Carnegie Mellon University"
<http://materializecss.com/>
- [11] Google. "Material Design". 2014
<https://material.google.com/>
- [12] Otto M., Thornton J. "Bootstrap". 2010
<http://getbootstrap.com/>
- [13] Zurb. "Foundation - The most advanced responsive front-end framework in the world". 2011
<http://foundation.zurb.com/>
- [14] Gamache D. "Skeleton". 2014
<http://getskeleton.com/>
- [15] Nielsen, J., & Budiu, R. "Mobile usability". MITP-Verlags GmbH & Co. KG. 2013.
- [16] W3C, Mobile Web Application Best Practices. 2010
<https://www.w3.org/TR/mwabp/>
- [17] Minify Code. "HTML Minifier". 2013
<http://minifycode.com/html-minifier/>
- [18] W3C. "Scalable Vector Graphics (SVG) 1.1 (Second Edition)". 2011.
<https://www.w3.org/TR/SVG11/>
- [19] Rodríguez, R. A., Merchán Rodríguez, V. R., Marko, I., Vera, P. M., Valles, G. Y., Alderete, C., ... & Conca, A.. Evaluación de la usabilidad en las soluciones móviles ofrecidas por las entidades públicas. In XVII Workshop de Investigadores en Ciencias de la Computación, 2015
<http://hdl.handle.net/10915/45841>
- [20] Guerra, Y. M., González, R. C., Febles, J. P. "Diseño web adaptativo para la plataforma educativa ZERA". *Revista Cubana de Ciencias Informáticas*, 10(2), pp. 100-115. 2016.