

Tipo de artículo: Artículo original  
Temática: Desarrollo de aplicaciones informáticas  
Recibido: 05/03/2015 | Aceptado: 23/02/2016

## Diseño web adaptativo para la plataforma educativa ZERA

### *Responsive web design for educative platform ZERA*

Yerandy Manso Guerra <sup>1\*</sup>, Roxana Cañizares González <sup>1</sup>, Juan Pedro Febles<sup>1</sup>

<sup>1</sup>Universidad de las Ciencias Informáticas, {[ymguerra](mailto:ymguerra@uci.cu), [rcanizares](mailto:rcanizares@uci.cu), [febles](mailto:febles@uci.cu)}@uci.cu

\* Autor para correspondencia: [ymguerra@uci.cu](mailto:ymguerra@uci.cu)

---

#### Resumen

La sociedad actual tiene acceso a tecnologías de bolsillo de alto procesamiento y conectadas al igual que un computador convencional. Estas tecnologías móviles ofrecen nuevas oportunidades para el aprendizaje. Algunos Sistemas de Gestión del Aprendizaje han adoptado variantes para incorporar o integrar el *m-learning*, ya sea realizando diseños web adaptativos, aplicaciones web, aplicaciones nativas o aplicando alguna combinación de estas; sin embargo, son insuficientes las soluciones desarrolladas. En la Universidad de las Ciencias Informáticas se desarrolló una plataforma educativa denominada ZERA, a pesar de sus potencialidades en su concepción no se tuvo en cuenta su utilización en dispositivos móviles lo que afecta la disponibilidad de sus contenidos y su correcto funcionamiento. En el presente artículo se destacan los referentes teóricos relacionados con el diseño web adaptativo y se describe su implementación en ZERA; haciendo uso de HTML5 y sus etiquetas semánticas, el *framework* de CSS Bootstrap, puntos de quiebres personalizados, un alto grado de flexibilidad en el diseño, así como el uso del *polyfills Adaptive Image* para la adaptación de las imágenes. Las pruebas realizadas después de implementado el diseño web adaptativo en la plataforma educativa ZERA permitieron comprobar su adaptabilidad a distintos tamaños de pantallas y dispositivos móviles.

**Palabras clave:** diseño web adaptativo, aprendizaje móvil, plataforma educativa ZERA

#### Abstract

*Today's society has access to pocket technology connected like a conventional computer. These mobile technologies offer new opportunities for learning. Some Learning Management Systems have adopted alternatives to incorporate or integrate m-learning, whether performing responsive web design, web applications, native applications or using some combination of these; however are insufficient the solutions developed. At the University of Information Science an educational platform was developed called ZERA, even though its potential in its design was not taken into*

*account its use on mobile devices which affects the availability of its contents and its proper functioning. This article highlights the theoretical references related to the responsive web design and its application in ZERA; using HTML5 and semantic tags, the Bootstrap CSS framework, custom breakpoints, a high degree of flexibility in the design and use of polyfills Adaptive Image for adapting images. The tests performed after the responsive web design was implemented in the educational platform ZERA allowed to check their adaptability to different screen sizes and mobile devices.*

**Keywords:** responsive web design, m-learning, educative platform ZERA

---

## Introducción

La sociedad actual tiene acceso a tecnologías de bolsillo de alto procesamiento y conectadas al igual que un computador convencional. Estas tecnologías móviles ofrecen nuevas oportunidades para el aprendizaje, extendiéndolo más allá de la tradicional aula *de clases*. Los avances en los dispositivos móviles y las tecnologías inalámbricas han impactado en el proceso enseñanza-aprendizaje. Se plantea que el término aprendizaje móvil (*m-learning*) se utilizó por primera vez en Estados Unidos a finales de los 90 e instaurándose en Europa a inicios del siglo XXI (Pisanty, Enríquez, Chaos-Cador, & García Burgos, 2010).

En relación al término *m-learning* existe un amplio debate entre los investigadores (Campanella, 2012; Caudill, 2007; Georgiev, Evgenia, Angel, Georgieva, & Smrikarov, 2004; Pinkwart, Hoppe, Milrad, & Perez, 2003; Traxler, 2005). Los debates se centran principalmente en si el *m-learning* es el siguiente paso del aprendizaje electrónico [*e-learning* (Baelo Álvarez, 2009)] o si es una herramienta avanzada que se integra con el *e-learning*; en cualquiera de los casos es considerado un nuevo componente que ha surgido para apoyar la educación a distancia.

Los Sistemas de Gestión del Aprendizaje (LMS), uno de los sistemas más utilizados en la educación a distancia, se han incorporado con éxito al *m-learning*, sin embargo, son insuficientes las soluciones desarrolladas. Dentro de las punteras se encuentra el LMS privativo Blackboard (Blackboard, 2014), con su aplicación Blackboard Mobile Learn (Mobile, 2014) en sus versiones para los sistemas operativos Android<sup>1</sup>, IOS<sup>2</sup> y Blackberry OS<sup>3</sup>. Moodle (Moodle, 2014) después de varios intentos fallidos a partir de su versión 2.0, cuenta con funcionalidades que permiten la

---

<sup>1</sup> Sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles, perteneciente a la empresa Google.

<sup>2</sup> *iPhone Operating System*, sistema operativo móvil de la empresa Apple Inc.

<sup>3</sup> Sistema operativo móvil desarrollado por RIM para los dispositivos BlackBerry.

creación de diseños adaptativos y la integración con Moodbile (Moodbile, 2014) (aplicación nativa para IOS y Android).

Los LMS han adoptado variantes para incorporar o integrar el *m-learning*, ya sea realizando diseños web adaptativos (*responsive web design*), aplicaciones web (*web app*), aplicaciones nativas (*native app*) o aplicando alguna combinación de estas soluciones.

En Cuba se constató, en un diagnóstico realizado a especialistas de tele-formación de universidades del país (Cañizares González, Estrada Sentí, & Febles Rodríguez, 2012), que existe un uso generalizado del LMS Moodle y que ninguna plataforma nacional ha adoptado en el paradigma de aprendizaje *m-learning*, a pesar de que la tendencia a nivel internacional reflejada en informes como Horizon 2012 (Johnson, Adams Becker, & Cummins, 2012) plantean el uso generalizado de las aplicaciones móviles con fines educativos en los próximos años.

En la medida en que se amplíe el estudio a distancia en Cuba y el acceso a las tecnologías de bolsillo, se necesitarán aplicaciones educativas desarrolladas para móviles. Estas deberán seguir las políticas de migración hacia aplicaciones y plataformas de código abierto, en concordancia con el desarrollo del proceso de informatización de la sociedad como parte de la ejecución por el país de una política orientada a alcanzar la seguridad, invulnerabilidad e independencia tecnológica, estipulado así por el Acuerdo No. 84 del 2004 del Comité Ejecutivo del Consejo de Ministros de la República de Cuba.

En la Universidad de las Ciencias Informáticas se desarrolló un LMS denominado Plataforma Educativa ZERA que cumple con lo estipulado en el Acuerdo No. 84 del 2004 del Comité Ejecutivo del Consejo de Ministros de la República de Cuba y debido a sus características puede ser utilizada en Cuba y en otros países. Sin embargo, en su concepción no se tuvo en cuenta la utilización de esta en dispositivos móviles. Esto ha traído consigo las siguientes afectaciones: no se muestran de forma correcta los distintos componentes visuales y sus contenidos, afectando la usabilidad y la aceptación por parte de los usuarios, no se brinda soporte para la gran variedad de dispositivos móviles existentes, no se tuvo en cuenta ninguna optimización referente a las bajas prestaciones de los dispositivos móviles, dificultades en su comercialización por no poseer las características necesarias para su inclusión en el paradigma *m-learning*.

Estas afectaciones constituyeron las bases para desarrollar una investigación que tuvo como **problema científico**: las limitaciones en el diseño y arquitectura de la plataforma educativa ZERA impiden su incorporación en el paradigma *m-learning*, afectándose la disponibilidad de sus contenidos y su correcto funcionamiento en dispositivos móviles. El presente artículo tiene como objetivo mostrar uno de los principales resultados alcanzados en la investigación

relacionada con el diseño web adaptativo desarrollado en ZERA, destacando los principales referentes teóricos que sustentan la solución.

## **Materiales y métodos o Metodología computacional**

La introducción del *iPhone* en el año 2007, fue un punto de cambio para el diseño web. Anteriormente los sitios web solo tenían que trabajar en monitores, que, aunque variaban sus tamaños la diferencia no era significativa. Así surgió la interrogante: ¿cómo hacer que los sitios web funcionaran en estas pantallas tan pequeñas?

Por un tiempo se realizaron sitios web optimizados para las pantallas de los *iPhone*, estos eran sitios separados (web app). Mantener dos sitios no es tan complejo, pero en los últimos años han surgido nuevos dispositivos con tamaños variables y eventualmente se ha hecho imposible desarrollar sitios separados para cada dispositivo.

Ante esta problemática se necesitaba una solución que abarcara diferentes tamaños de pantalla, una manera de diseñar sitios web que se adaptaran al tamaño de las pantallas desde donde se accedía (Peterson, 2014). No fue hasta el año 2010 que Ethan Marcotte en un artículo publicado en la web *A List Apart* (Marcotte, 2010) acuñó por primera vez el término *responsive web design*. (Marcotte, 2010) plantea que el *Responsive Web Design* (RWD) es un método para diseñar sitios web flexibles, que no se basa únicamente en anchos de pantallas fijos sino que es capaz de detectar el ancho de pantalla y ajustar el diseño para proveer una experiencia adecuada de visualización en cada dispositivo.

Existen otros investigadores que han dado su propia definición de RWD, tal es el caso de (Knight, 2011) que lo define como: el enfoque que sugiere que el diseño y desarrollo de un sitio web debe responder al comportamiento del usuario y al medio ambiente de este, basado en el tamaño de pantalla, la plataforma y la orientación de los dispositivos que este use para acceder al sitio. También (Snell, 2013) concluye que el RWD provee una experiencia positiva e intuitiva independientemente del dispositivo, sin requerir sub-dominios separados para los dispositivos móviles, todo esto haciendo uso de las reglas de *Cascading Style Sheets* (CSS) *media queries* logrando que todos los elementos se adapten al ancho y orientación de los dispositivos.

El RWD consiste en una combinación de diseño, plantillas flexibles y un uso inteligente de reglas de CSS. A medida que el usuario cambie de dispositivo, el sitio debe cambiar automáticamente para adaptarse a la resolución de pantalla. En otras palabras, el sitio web debe tener la tecnología para responder automáticamente al contexto del usuario. Una razón convincente para el uso de RWD es que se crea un sitio web que no sólo se ve bien y funciona

correctamente en los dispositivos que están en el mercado, sino que probablemente se verá bien y funcionará en los nuevos dispositivos que estarán disponibles en el futuro.

Una ventaja obvia de usar RWD es que requiere menos esfuerzo y posibilita el mantenimiento futuro (Peterson, 2014), solo es necesario crear un sitio web que se visualizará de distintos dispositivos, usando el mismo diseño, código y un mismo contenido, además este último se verá correctamente no importando el tamaño de pantalla donde se visualice. Otra ventaja es la ubicación del sitio web en los motores de búsquedas. Un sitio para móviles con un conjunto separado de URL puede traer problemas con el posicionamiento en los resultados de los buscadores. Google desde el 2012 recomienda el uso de RWD, no solo por la mejor experiencia que les brindan a los usuarios sino también porque le permite a Google obtener el contenido de los sitios más eficientemente, significando que los cambios de los sitios son actualizados en los motores de búsqueda rápidamente.

## HTML y CSS

El uso correcto de *HyperText Markup Language* (HTML) y de *Cascading Style Sheets* (CSS) son fundamentales para lograr un RWD, de ambos existen varias versiones siendo HTML 5 (W3C-HTML5, 2014) y CSS 3 (W3C-CSS, 2011) las versiones más actuales.

Para lograr un RWD adecuado, limpio y accesible es necesario tener en cuenta algunos aspectos fundamentales de HTML, tal es el caso de la semántica, esta se refiere al uso de etiquetas HTML para reforzar la semántica o el significado de la información en los sitios web. La especificación HTML 5 incorpora etiquetas con significado semántico como son: <header>, <nav>, <section>, <article>, <aside> y <footer>. Estas etiquetas son entendidas por los lectores de pantalla y cualquier otra herramienta que lea los sitios web, su uso permite aumentar la accesibilidad y mejora el posicionamiento en los buscadores.

Con HTML se crea la estructura de los contenidos de los sitios web y con CSS se le dice al navegador como darle estilo, o sea, como mostrarlo. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. La aplicación de los estilos a un sitio web se puede realizar de varias formas, pero la más recomendada (por los autores) es en ficheros CSS.

## Media queries

Las media *queries* son reglas de CSS “@media” (W3C-Media Queries, 2012) que permiten mostrar diferentes estilos de CSS basándose en las características de los dispositivos, fue incluida como parte de la especificación de CSS 2,

pero solo soportaba consultas tales como *screen* y *print*. Comúnmente se usaba para crear una versión para imprimir de los sitios web pero su utilidad terminaba ahí.

La versión CSS 3 incluye consultas más específicas basadas en las características de los dispositivos, como el ancho, alto y capacidad de colores. Las media *queries* no afectan al HTML solo a los estilos aplicados a las páginas haciendo uso de CSS. Los navegadores empezaron a darle soporte en el 2009. Esto permitió que se le pudieran añadir a los sitios web diferentes plantillas (*layouts*) para diferentes tamaños de pantallas sin tener que crear nuevos sitios web. Usando media *queries* se puede cambiar cualquier aspecto relacionado con el estilo de su sitio web, no solamente el número de columnas sino también pueden mover objetos, cambiar la fuente del texto, mostrar u ocultar contenido, ajustar márgenes y cualquier otro estilo que se pueda aplicar con CSS.

En la práctica su funcionamiento es sencillo, todo comienza con una pregunta, por ejemplo: ¿es el ancho de pantalla (la resolución horizontal) igual o menor que 480px? Si cumple la condición entonces se aplica el estilo. Existen varias formas de utilizar las medias *queries*, los autores recomiendan dos; la primera es su uso en una etiqueta *link* y la segunda su uso dentro de un fichero CSS, ambos casos funcionan de forma similar.

## Puntos de quiebre

Otro elemento fundamental al usar las media *queries* es definir los puntos de quiebres o cambios (*breakpoints*), estos son los puntos donde el diseño cambiará de forma, o sea, se adaptará a los distintos anchos de pantallas y resoluciones. Ethan Marcotte en su libro *Responsive Web Design* (Marcotte, 2011) enuncia una serie de puntos de quiebres (ver Tabla 1), estos son solo sugerencias ya que lo recomendable es probar el diseño en diferentes anchos de pantallas.

Tabla 1. Puntos de quiebres enunciados en (Marcotte, 2011) para lograr RWD

Píxeles	Descripción
320	Para dispositivos de pantallas pequeñas, como los celulares en modo <i>portrait</i> .
480	Para dispositivos de pantallas pequeñas, como los celulares en modo <i>landscape</i> .
600	Tables pequeñas, 600x800 y 600x1024, en modo <i>portrait</i> .
768	Tables de 10 pulgadas como los iPad (768x1024), en modo <i>portrait</i> .
1024	Tables, iPad (1024x768), en modo <i>landscape</i> , algunas laptop, netbook y monitores.
1200	Para pantallas <i>widescreen</i> .

Las medias *queries* son un elemento fundamental en los RWD, sin estas no se podría hablar de diseño adaptativo, estas pueden organizar las plantillas, pero el RWD no funciona sin una base indispensable de flexibilidad.

## Flexibilidad

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 RWD, esto se logra mediante el abandono de los *pixeles*<sup>4</sup>, en favor de porcentajes o unidades relativas llamadas *ems*<sup>5</sup>.

Las imágenes también deben ser capaces de expandirse y contraerse según sea necesario. Respecto a esto existen estrategias enunciadas por varios autores, algunos se centran en el uso de reglas CSS para lograrlo y otros proponen bibliotecas/librerías sobre todo de *JavaScript* para lograr una adaptación óptima. Al utilizar imágenes en el diseño lo primero que se debe tener en cuenta es usar las que no sean posibles reemplazarlas por reglas CSS, de esta forma se reduce el peso de la página y se logra un diseño más adaptativo.

En (Snell, 2013) se plantea que existe dos formas de lograr la adaptación de las imágenes, la primera es establecer el ancho de la imagen al 100% y la segunda ponerle un ancho predefinido ocultando el resto de la imagen con el atributo *overflow*. Los autores concuerdan con la primera opción, pero consideran que esta no resuelve el problema de la eficiencia, las imágenes no siempre se mostrarán en el mismo tamaño, porque son flexibles, así que ¿cómo asegurar no perder el ancho de banda mediante el envío de una imagen grande a un dispositivo con una pantalla pequeña?

## Imágenes en el contenido

En las imágenes que se usan en el contenido hay que tener en cuenta su extensión, estas deben ser JPEG, GIF, o PNG, existen otros formatos de imágenes, pero no son soportadas por todos los navegadores web. Las diferencias entre las extensiones de las imágenes usadas son importantes para el RWD porque el formato afecta el tamaño de la imagen y lo deseable es usar el formato que dé como resultado la menor imagen posible. En (Peterson, 2014) se hace un resumen de cada uno de estos formatos.

Para lograr entregar imágenes de acuerdo a las capacidades de los dispositivos con los que se accede sin desperdiciar ancho de banda se creó *W3C's Responsive Images Community Group* (W3C-RICG, 2014). Este grupo propone dos soluciones potenciales para el problema de la adaptación de las imágenes, el elemento `<picture>` y el atributo `srcset` pero ninguna de las dos soluciones ha sido implementada por los principales navegadores, ni aún son parte de HTML.

---

<sup>4</sup> Pixel: acrónimo del inglés *picture element* es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

<sup>5</sup> La unidad *em* es una unidad escalable y está en relación al tamaño de la fuente.

Sin embargo, ya que aún no es posible usar de forma nativa ninguna de las dos soluciones anteriores han surgido algunos *polyfills*<sup>6</sup>:

**Picturefill** es un *polyfills* (Jehl, Marquis, & Jansepar, 2014) que está conformado por un fichero llamado `picturefill.js` que permite utilizar la etiqueta `<img>` con los atributos *sizes* y *srcset*. El atributo *sizes* es usado para definir el tamaño de la imagen a través de una serie de puntos de quiebre y *srcset* define una serie de imágenes y sus tamaños, de esta forma solo se cargará la imagen que corresponda con los puntos de quiebres definidos.

**Adaptive Images** es otro *polyfills* para lograr la adaptación de las imágenes, esta es una variante del lado del servidor (Wilcox, 2014) que no requiere cambios en el HTML, es necesario tener como servidor web Apache2/Nginx, Hypertext Preprocessor (PHP) 5.x y la biblioteca/librería GD.

Para poner en funcionamiento esta técnica es necesario añadir código *JavaScript* en el `<head>` del sitio web, este crea una *cookie*<sup>7</sup> con el ancho de pantalla del visitante en pixeles, entonces el navegador al encontrar una etiqueta `<img>` realiza la petición de la imagen enviando la *cookie* ya creada con anterioridad. De esta forma el servidor se encarga de devolver la imagen en el tamaño correcto teniendo en cuenta la configuración que se haya establecido en el fichero `adaptive-images.php`, la explicación detallada del funcionamiento y configuración de esta estrategia se puede encontrar en <http://adaptive-images.com/details.htm>. Esta variante es una de las mejores siempre y cuando el servidor web cumpla con los requisitos antes expuestos.

**HiSRC** (Schmitt, 2013) es un plugin de JQuery<sup>8</sup> que permite reemplazar las imágenes teniendo en cuenta la velocidad de la red y la resolución de pantalla. Su funcionamiento es bastante sencillo, primero carga la imagen con menor resolución y luego utiliza *JavaScript* para detectar la velocidad de conexión, si el ancho de banda es bueno descarga una imagen con mayor resolución. Más detalles de esta variante y de su configuración se pueden encontrar en <https://github.com/teleject/hisrc>. El mayor inconveniente de esta opción es que es necesario tener la misma imagen en varias resoluciones.

---

<sup>6</sup> Un *polyfill* es un pedazo de código que replica los elementos nuevos de HTML/ CSS en los navegadores más antiguos, o sea le da soporte a estos elementos.

<sup>7</sup> *Cookie*: es una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario. (Kristol, 2001)

<sup>8</sup> Framework de JavaScript (Benedetti & Cranley, 2011)

Otra forma de adaptar las imágenes es usar **servicios de terceros**, algunas compañías ofrecen servicios de adaptación que automáticamente adaptan las imágenes, tal es el caso de Sencha.io SRC (Sencha, 2014), ReSRC.it (Fee & Thurgood, 2014), Thumbr.io (Empirical Evidence SL, 2013) y Responsive.io (14 Islands AB, 2014).

Estos servicios escalan las imágenes al ancho del dispositivo con que se acceda y guardan esta en cache logrando que las siguientes peticiones del mismo dispositivo sean más rápidas. El problema de usar servicios de terceros está en la posibilidad que tienen de dejar de existir, que este se encuentre temporalmente fuera de servicio o que esté lento, cualquiera de estas posibilidades afectaría el desempeño del sitio web donde se usen.

## Frameworks

Para lograr un RWD de calidad es necesario tener en cuenta una serie considerable de detalles. Por esta razón desarrolladores y compañías internacionales que apuestan por esta variante se han dado a la tarea de realizar *frameworks* y herramientas del lado del cliente que permiten crear RWD en poco tiempo y sin mucha experiencia, estos son utilizados como la base para desarrollar productos web. Existe una amplia variedad de estos *frameworks* con características similares, la selección de alguno estará dada por la documentación, uso y soporte que este tenga, a continuación se presentan una pequeña selección: Twitter Bootstrap (Otto & Jacob, 2014), Foundation (ZURB, 2014), Skeleton (Skeleton, 2014), HTML KickStart (Gatcke, 2014), Less Framework (Korpi, 2014).

En la presente investigación se usará el *framework Twitter Bootstrap*, porque es adaptable a las resoluciones de pantalla, es ágil en la construcción de interfaces y el equipo de trabajo tiene experiencia en su uso, esto implica un gran cambio ya que la plataforma en la versión actual utiliza el *framework Blueprint* el cual ha quedado obsoleto por su imposibilidad de ser adaptativo.

## Resultados y discusión

Para realizar un RWD en ZERA es necesario aplicar un conjunto de técnicas y estrategias de las explicadas, a continuación, se expondrá cuáles de esta se usaron y cómo.

### Puntos de quiebre

Al realizarse un RWD uno de los primeros elementos a definir son los *puntos de quiebres*. En la presente investigación se usan los siguientes puntos de quiebre: 1200px para pantallas grandes, de 1199px a 978px para monitores de tamaño medio, de 977px a 768px para tables en modo *landscape* y algunos monitores y de 767px a 481px para teléfonos en modo *landscape* o tables en modo *portrait*. Quedando de la siguiente forma:

Tabla 2. Puntos de quiebres usados

Pantallas	Media queries
Grandes	@media (min-width: 1200px) { }
Medias	@media (min-width: 978px) and (max-width: 1199px) { }
Tables portrait y landscape a desktop	@media (min-width: 768px) and (max-width: 979px) { }
Teléfonos landscape a portrait tablet	@media (max-width: 767px) { }
Pequeñas	@media (max-width: 480px) { }

## HTML semántico y cambios en el diseño según los puntos de quiebre

La estructura del HTML es de vital importancia para lograr algún nivel de accesibilidad, una estructura que cumpla con las tendencias internacionales es de vital importancia para organizar correctamente el contenido, además le permite a la página web ser accesible desde cualquier lector de pantalla.

En este aspecto se decidió cambiar la estructura del HTML, para añadir las etiquetas semánticas que introduce HTML5: `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`, además se cambió y organizó el maquetado de todas las páginas estructurándolo de tal forma que siempre lo más importante para el usuario este de primero, este principio es descrito en (Peterson, 2014) y hace referencia al seguimiento de un diseño web de forma lineal, debido a que se efectuaron gran cantidad de cambios que no son posibles reflejarlos en el presente artículo.

También en (Peterson, 2014) se concluye que la forma óptima de realizar un RWD, es comenzar de pantallas pequeñas a grandes, sin ningún *JavaScript* y sin reglas CSS, en ZERA fue imposible seguir este principio ya que lo realizado fue un rediseño de la plataforma. El cambio de *framework* de CSS llevó a la modificación del código fuente del 100% de las vistas, aprovechándose estos para adoptar el diseño de la línea de productos de la UCI llamada XAUCE, de igual forma debido a la cantidad de cambios en el presente documento no es posible mostrarlos todos.

## Adaptación de contenidos

La adaptación seleccionada para ser aplicada en la presente investigación está relacionada con la capa de presentación (Pushpa, 2012) y con el lugar donde se realiza la adaptación: en el servidor (W3C-MWBP, 2008). Se utilizó la estrategia *Minor Adaptation* (MobiForge, 2009) centrada sobre todo en la adaptación de las imágenes haciendo uso del *polyfills Adaptive Images*, debido a que ZERA cumple con los requisitos tecnológicos necesarios del lado del servidor para aplicar esta técnica de adaptación.

## Componentes

En los componentes visuales y en el diseño de la plataforma educativa ZERA, como se expresó, no se tuvo en cuenta ninguna optimización referente a los dispositivos móviles, siendo esto una de las principales causas por las que se originó la presente investigación. Como parte de la solución se realizó cambios en varios componentes, a continuación, se exponen los componentes más importantes:

### Componente recurso

Componente perteneciente al subsistema *Recursos*, de este depende toda la gestión y visualización de recursos educativos y de los contenidos en cualquier lugar de la plataforma por esto su gran importancia. El componente recurso es el encargado de realizar la creación, edición y visualización de 28 tipos de recursos educativos agrupados en cuatro categorías: multimedia, interactivos, estructurales y materiales. En este componente se hicieron cambios tanto visuales como arquitectónicos. Los cambios arquitectónicos se enfocaron en mejorar la mantenibilidad y reutilización de este eliminando el anti-patrón espaguetis<sup>9</sup> (Spaghetti Code).

En este sentido se re-escribió todo el componente haciendo uso de patrones de diseños y buenas prácticas, estos fueron el resultado de la investigación de (Araujo Quintana, Duran López, Manso Guerra, & Santos Salgado, 2013). Dentro de los principales cambios está la reorganización del código, además de reutilizar clases y objetos, con el objetivo de que el mismo sea flexible y escalable. Estos cambios permiten la inclusión de nuevos tipos de recursos en la aplicación de forma rápida. Se le añadió la posibilidad de habilitar o deshabilitar los recursos que desee visualizar u ocultar el usuario. Además, se reestructuró las tablas de la base de datos y la información de los recursos, guardándose en su tabla correspondiente.

Para lograr la mantenibilidad y reutilización del código (características deseadas) se hizo uso de los patrones *Factory Method*, *Dependency Injection* y *Strategy* ya que se encontró en estos las características necesarias para darle solución a los problemas existentes.

### Componente de subida de archivos

Este componente es usado por todos los subsistemas de la plataforma que permitan la subida de archivos, en los formularios como un campo más o de forma solitaria, a este se le añadió la posibilidad de ser internacionalizado

---

<sup>9</sup> Este anti-patrón fue mencionado por primera vez por Dennis V. Tassel se refiere al código que está complicado de leer y mal organizado que cambiar alguna parte de él es probable que afecte una funcionalidad distinta de la que se pretende cambiar.

(solucionando de esta forma una no conformidad señalada en todas las revisiones realizadas por calidad) y se cambió su interfaz visual permitiendo que sea adaptativo.

Para lograr estos comportamientos se hizo uso de las buenas prácticas propuestas por el *framework Symfony*, se modificó el *widget*<sup>10</sup> llamado *sfAOWidgetFormInputFile* usado en los formularios heredando de la clase *sfWidgetFormInputFile* que provee *Symfony*, se cambió la forma de visualizarse los *input* de tipo *file* siendo estos sustituidos por un *plugin* de JQuery llamado *FileUpload*, que además de ser adaptativo permite que el usuario pueda arrastrar los ficheros a subir. De igual forma se implementó un componente<sup>11</sup> heredando de la clase *sfComponents* del *framework* llamado *sfFileUploadComponents*, este permite el uso del componente de forma aislada.

### Subsistema Curso

El subsistema *Curso* de la plataforma ZERA es donde ocurre la mayor parte del proceso de aprendizaje, es accedido por profesores, estudiantes, directores y tutores. Debido a esta característica es el subsistema al cual se le dio mayor importancia en la implementación del RWD. En este sentido se cambió el diseño añadiendo una barra de alertas con tres categorías: mensajería, notificaciones y eventos. Para tamaño de pantallas menores 767px se ocultaron los menús laterales a través de reglas CSS. Se crearon componentes que estandarizaron las tablas, menús y los paneles laterales. Se les añadió a los formularios la posibilidad de ser vistos en *tabs* y se estandarizaron todos los mensajes mostrados al usuario.

Con el objetivo de verificar la solución, se realizaron pruebas con dispositivos de tres hasta diez pulgadas de diámetro, abarcando desde *smartphones* hasta *tablets*. Las pruebas consistieron en tomar un grupo de usuarios con estos dispositivos y comparar la versión 1.0 de ZERA (versión sin el diseño web adaptativo) con la versión 1.2.1 que es la que contiene todos los cambios expuestos en el presente artículo. Para medir los resultados se usó el diferencial semántico o prueba del diferencial semántico de *Osgood* (Charles Egerton & James G, 1969) que permitió comprobar que los cambios realizados superan la versión anterior en cuanto a disponibilidad de los contenidos y correcto funcionamiento en dispositivos móviles.

---

<sup>10</sup> En programación, un widget (o control) es un elemento de una interfaz que muestra información con la cual el usuario puede interactuar. Por ejemplo: ventanas, cajas de texto, checkboxes, listbox, entre otros.

<sup>11</sup> En este caso la palabra componente se refiere a los componentes internos del framework *Symfony* y no a la componente subida de archivos.

## Conclusiones

El RWD es una variante a tener en cuenta por cualquier software educativo que desee incluirse en el paradigma de aprendizaje *m-learning*. RWD consiste en dos componentes principales: *flexibilidad*, significando que cada medida horizontal tiene que ser en unidades relativas (porcentuales) y las *media queries* que permiten usar reglas CSS para cambiar el diseño según el ancho de pantalla del dispositivo con el cual de acceda.

La adaptación de las imágenes de forma eficiente es un área del RWD que aún está siendo investigada. Hay algunas soluciones que son de uso común, las cuales se trataron en la presente investigación, pero cada una tiene algunos inconvenientes y ninguna de ellas parece ser la solución perfecta. De esta amplia gama de opciones, la variante a escoger estará dada por las necesidades de los clientes y profundidad del proyecto o producto que se elabore.

La implantación del RWD en ZERA y por tanto su incorporación al paradigma de aprendizaje *m-learning* le da la posibilidad de incursionar en nuevos escenarios de aprendizaje, elevar su calidad visual y le permite competir con otras plataformas reconocidas.

## Referencias

14 ISLANDS AB. (2014). Responsive.io. Retrieved from <https://responsive.io/>

ARAUJO QUINTANA, A., DURAN LÓPEZ, R., MANSO GUERRA, Y., & SANTOS SALGADO, C. (2013). *Rediseño del módulo iResources de la Plataforma Educativa ZERA*. Universidad de las Ciencias Informáticas. Retrieved from [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/8173](http://repositorio_institucional.uci.cu/jspui/handle/ident/8173)

BAELO ÁLVAREZ, R. (2009). El e-learning, una respuesta educativa a las demandas de las sociedades del siglo XXI. *Pixel-Bit. Revista de Medios y Educación*, 35, 87–96.

BENEDETTI, R., & CRANLEY, R. (2011). *Head First jQuery*. (C. Nash, Ed.) (p. 538). USA: O'Reilly Media, Inc. Retrieved from <http://my.safaribooksonline.com>

BLACKBOARD. (2014). Blackboard Oficial Web Page. Retrieved from <http://www.blackboard.com/>

CAMPANELLA, P. (2012). Mobile Learning: New forms of education. *ICETA 2012 - 10th IEEE International Conference on Emerging eLearning Technologies and Applications* (pp. 51–56). Stará Lesná, The High Tatras, Slovakia: IEEE.

- CAÑIZARES GONZÁLEZ, R., ESTRADA SENTÍ, V., & FEBLES RODRÍGUEZ, J. P. (2012). *Repositorio de recursos educativos para las instituciones de educación superior*. Universidad de las Ciencias Informáticas, La Habana. Retrieved from [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/7913](http://repositorio_institucional.uci.cu/jspui/handle/ident/7913)
- CAUDILL, J. G. (2007). Mobile Computing : Parallel developments. *International Review of Research in Open and Distance Learning*, 8(2), 1–13. Retrieved from <http://www.irrodl.org/index.php/irrodl/article/view/348/873>
- CHARLES EGERTON, O., & JAMES G, S. (1969). *Semantic differential technique: A sourcebook* (p. 681). Aldine. Retrieved from <http://library.wur.nl/WebQuery/clc/413744>
- EMPIRICAL EVIDENCE SL. (2013). Thumbr.io. Retrieved from <http://www.thumbr.io/>
- FEE, D., & THURGOOD, E. (2014). ReSRC.it. Retrieved from <http://www.resrc.it/>
- GATCKE, J. (2014). HTML KickStart. Retrieved September 5, 2014, from <http://www.99lime.com/elements/>
- GEORGIEV, T., EVGENIA, G., ANGEL, S., GEORGIEVA, E., & SMRIKAROV, A. (2004). M-Learning - a New Stage of E-Learning Tsvetozar. *International Conference on Computer Systems and Technologies - CompSysTech'2004* (Vol. IV–28, pp. 1– 5).
- JEHL, S., MARQUIS, M., & JANSEPAR, S. (2014). Picturefill. Retrieved from <http://scottjehl.github.io/picturefill/>
- JOHNSON, L., ADAMS BECKER, S., & CUMMINS, M. (2012). *NMC Horizon Report: 2012 Higher Education Edition* (p. 40). Austin, Texas.
- KNIGHT, K. (2011). Responsive Web Design: What It Is and How to Use It. *Smashing eBook | Best of Smashing Magazine*, 234–269. Retrieved from <http://krackerjackdesigns.com/SM-ResponsiveWebDesign.pdf>
- KORPI, J. (2014). Less Framework 4. Retrieved September 5, 2014, from <http://lessframework.com/>
- KRISTOL, D. M. (2001). HTTP Cookies : Standards , Privacy , and Politics. *ACM Transactions on Internet Technology (TOIT)*, 1(2), 151–198. doi:10.1145/502152.502153
- MARCOTTE, E. (2010). Responsive Web Design. A List Apart. Retrieved from <http://alistapart.com/article/responsive-web-design>
- MARCOTTE, E. (2011). *Responsive Web Design*. (J. Zeldman & M. Brown, Eds.) *A List Apart* (p. 157). New York, New York: A Book Apart. Retrieved from [http://www.reposol.be/sites/reposol.beta.the-aim.be/files/responsive-webdesign\(ethan-marcotte\).pdf](http://www.reposol.be/sites/reposol.beta.the-aim.be/files/responsive-webdesign(ethan-marcotte).pdf)

- MOBIFORGE. (2009). Adaptation Strategies. MobiForge. Retrieved July 27, 2014, from <http://mobiforge.com/book/adaptation-strategies>
- MOBILE, B. (2014). Blackboard Mobile. Retrieved from <https://www.blackboard.com/platforms/mobile/products/mobile-learn.aspx>
- MOOBBILE. (2014). Moodbile Oficial Web Page. In. Retrieved from <http://www.moodbile.org>
- MOODLE. (2014). Moodle Oficial Web Page. In., Retrieved from <https://moodle.org>
- OTTO, M., & JACOB. (2014). Bootstrap. Retrieved September 5, 2014, from <http://getbootstrap.com/2.3.2/>
- PETERSON, C. (2014). *Learning Responsive Web Design*. (M. Treseler, K. Ebrahim, & J. Kwityn, Eds.) (First Edit., p. 412). Canadad: O'Reilly Media, Inc. Retrieved from <http://www.oreilly.com/catalog/errata.csp?isbn=0636920029199>
- PINKWART, N., HOPPE, H. U., MILRAD, M., & PEREZ, J. (2003). Educational scenarios for cooperative use of Personal Digital Assistants. *Journal of Computer Assisted Learning*, 19(3), 383–391. Retrieved from <http://dx.doi.org/10.1046/j.0266-4909.2003.00039.x>
- PISANTY, A., ENRÍQUEZ, L., CHAOS-CADOR, L., & GARCÍA BURGOS, M. (2010). “M-learning en ciencia” - Introducción de aprendizaje móvil en física. *RIED. Revista Iberoamericana de Educación a Distancia*, 13(1), 129–155. doi:<http://dx.doi.org/10.5944/ried.1.13.892>
- PUSHPA, M. (2012). ACO in e-Learning : Towards an adaptive learning path. *International Journal on Computer Science and Engineering (IJCSE)*, 4(03), 458–462.
- SCHMITT, C. (2013). HiSRC. Retrieved from <https://github.com/teleject/hisrc>
- SENCHA. (2014). HTML5 App Development for Desktop and Mobile. JavaScript Frameworks and Dev Tools from Sencha. Retrieved from <http://www.sencha.com/learn/how-to-use-srcsencha-io/>
- SKELETON. (2014). Skeleton. Retrieved September 5, 2014, from <http://www.getskeleton.com/>
- Snell, J. (2013, April). Flexible Everything: Getting Responsive With Web Design. *Computers in Libraries*, (April), 6. Retrieved from [www.iRfotoday.coni](http://www.iRfotoday.coni)
- TRAXLER, J. (2005). Defining mobile learning. IADIS International Conference Mobile Learning 2005.
- W3C-CSS. (2011). W3C. W3C. Retrieved July 24, 2001, from <http://www.w3.org/TR/CSS/#css3>

W3C-HTML5. (2014). HTML5. W3C. Retrieved July 24, 2001, from <http://www.w3.org/TR/html5/>

W3C-Media Queries. (2012). Media Queries. W3C. Retrieved July 24, 2001, from <http://www.w3.org/TR/css3-mediaqueries/>

W3C-MWBP. (2008). Mobile Web Best Practices 1.0. W3C. Retrieved July 27, 2014, from <http://www.w3.org/TR/mobile-bp/#d0e128>

W3C-RICG. (2014). Responsive Images Community Group. Retrieved from <http://www.w3.org/community/respimg/>

WILCOX, M. (2014). Adaptive Images. Retrieved from <http://adaptive-images.com/>

ZURB. (2014). Foundation. Retrieved September 5, 2014, from <http://foundation.zurb.com/>