

RECIBIDO: 02-05-2017

ACEPTADO: 30-11-2017

EL MITO DEL DISEÑO WEB ADAPTATIVO

THE RESPONSIVE WEB DESIGN'S MYTH

LUCIANO MALDONADO

UNIVERSIDAD DE LOS ANDES INSTITUTO DE ESTADÍSTICA APLICADA Y COMPUTACIÓN,
IEAC, FACES
MALDONAJ@ULA.VE

YESENIA NÚÑEZ

UNIVERSIDAD DE LOS ANDES INSTITUTO DE ESTADÍSTICA APLICADA Y COMPUTACIÓN,
IEAC, FACES
YESENIAN@ULA.VE

MÉRIDA - VENEZUELA

RESUMEN: En este artículo se presenta una descripción del diseño web adaptativo, a partir de la experiencia de trabajar con este tipo de tecnología en el ámbito académico universitario de postgrado, donde se ha podido apreciar que el enfoque de diseño y construcción de aplicaciones web constituye una evolución tecnológica que, en la actualidad, juega un rol muy importante en programas educativos donde se imparten clases en modalidades a distancia o semipresenciales, por lo tanto adoptarla para llevar a cabo cualquier proyecto, incluso fuera del campo educativo, conlleva grandes beneficios en el uso de recursos, en la visualización, en la comunicación y en la diversidad de dispositivos a través de los cuales se pueden acceder las aplicaciones. Sin embargo, también, implica grandes desventajas, por lo que más que presentarla como la solución definitiva hay que considerarla como una tecnología que tiene defectos que constituyen retos por superar.

PALABRAS CLAVE: Diseño adaptativo, aplicaciones web, resoluciones de pantalla, visualización, tecnología.

ABSTRACT: This article presents a responsive web design description, based on the experience of working with this type of technology in the academic postgraduate field, where it has been appreciated that the design and construction approach of web applications constitutes a technological evolution that, at present, plays a very important role in educational programs where classes are taught in distance or blended modalities, therefore adopting it to carry out any project, even outside the educational field, entails great benefits in the use of resources, in the visualization, in the communication and in the diversity of devices through which the applications can be accessed. However, it also implies great disadvantages so that rather than presenting it as the definitive solution, it must be considered as a technology that has defects that constitute challenges to be overcome.

KEYWORDS: Adaptive design, web applications, Screen resolutions, visualization, technology.

I. INTRODUCCIÓN

En la actualidad millones de usuarios se conectan a internet desde diferentes dispositivos electrónicos, y cada vez más lo hacen desde los llamados teléfonos inteligentes y *tablets*. En este sentido, el tamaño de la pantalla y la resolución de la misma es un factor a tomar en cuenta a la hora de desarrollar una aplicación web para ser visualizada en un *smart tv*, un *smartphone*, una *tablet*, una laptop o un computador de mesa; puesto que la navegación y la visualización utilizando un tipo u otro de tales dispositivos puede cambiar, incluso, de forma radical. Por esta razón, el diseño y construcción de una aplicación web tiene que adaptarse a todas las resoluciones.

El diseño adaptativo ha tomado auge desde hace varios años, ya que hasta hace poco los diseñadores orientaban sus aplicaciones a un solo tipo de resolución, por lo que tenían el control absoluto de lo que se presentaba sobre esa resolución, se orientaba el diseño y la programación no sólo a una resolución, sino, incluso, al navegador y a los dispositivos de entrada de datos de los usuarios como el teclado o las pantallas *touch*. Es decir, hasta hace algunos años el diseño y construcción de las aplicaciones web implicaba orientar ese desarrollo a resoluciones específicas y al uso de recursos específicos sobre los que se ejecutarían dichas aplicaciones. Este hecho conllevaba replicar todo el desarrollo cuando se requería esa aplicación para otra resolución, navegador o dispositivo de entrada de datos, lo que, sin duda, en el presente, se consideraría una pérdida de recursos, de tiempo y de falta de flexibilidad en el desarrollo.

En este trabajo se explica, en base a la experiencia de los autores, en qué consiste y cómo se logra el diseño y la construcción adaptativa de las aplicaciones web, en la actualidad, que, sin embargo, desde nuestro punto de vista, es sólo un mito más que una realidad total.

II. EL RESPONSIVE WEB DESIGN O DISEÑO WEB ADAPTATIVO

El término *responsive web design*, creado por Ethan Marcotte (Marcotte, 2017), hace referencia a que el diseño y construcción de una aplicación web debe tener como característica la capacidad para poder visualizarse siempre de la misma manera, independientemente del navegador y de la resolución de la pantalla del dispositivo con el que se acceda a la web. De esta manera, se superaría la redundancia antes mencionada, con lo que se reduciría el tiempo de desarrollo de tales aplicaciones, el conjunto de programas necesarios para cada dispositivo y se realizaría un trabajo más eficiente por parte de los diseñadores y desarrolladores web.

El *responsive web design* estandarizaría tanto el diseño como la programación para cumplir lo establecido por el W3C (*world wide web consortium*) que se guía por los principios de accesibilidad, internacionalización e independencia para que el acceso a la web sea posible desde cualquier lugar, en cualquier momento y utilizando cualquier dispositivo, cualquier idioma y cualquier cultura (Vega, 2013). Por comodidad, en adelante, se utilizará el término Diseño Web Adaptativo o simplemente Diseño Adaptativo, que con frecuencia aparecen, como la versión en español, en artículos y trabajos relacionados con aplicaciones web.

De acuerdo a la experiencia de los autores, los aspectos más relevantes del diseño adaptativo están relacionados con la definición del ancho de los contenedores en *css*, en cuyo caso, se requiere trabajar en porcentajes; el tamaño de las imágenes tiene que ser flexible de tal manera que en

los monitores de los computadores de escritorio se visualicen en tamaño normal y se reduzcan de tamaño en dispositivos de pantallas pequeñas. También, están los casos de los *smartphones* (a los que nos referiremos, en adelante, como celulares) y *tablets*, donde la función y el contenido, a veces, necesitan ser diferentes al sitio visto en un computador de escritorio, lo que es resuelto con el diseño adaptativo al quitar o reducir la visibilidad de ciertos componentes.

III. INFLUENCIA DEL DISEÑO ADAPTATIVO EN LA EDUCACIÓN

El diseño instruccional trata la planificación del contenido que se imparte en una asignatura dirigida de manera presencial o virtual (Belloch, 2017). Esta planificación involucra recursos que pueden estar directamente relacionados con plataformas digitales. Para el caso particular de la educación virtual, las plataformas de aprendizaje digital necesariamente deben crear la percepción de estar basadas en tecnologías con características adaptativas, por lo tanto el trabajo que se describe, en este artículo, cobra importancia para quienes se involucran en investigaciones relacionadas con el desarrollo de contenidos en este tipo de plataformas, más ahora que han surgido las tecnologías del tipo *m-learning* (Mahande y Surhono, 2017). De igual manera, los desarrolladores de las aplicaciones web orientadas a la educación virtual, en la actualidad, tienen que ofrecer sistemas de software que satisfagan a los usuarios, independientemente del dispositivo sobre el cual se visualicen los contenidos e independientemente del ámbito de esos contenidos.

IV. OPINIÓN DE LOS AUTORES RESPECTO A LOS DESARROLLOS WEB ADAPTATIVOS

Esta opinión se basa en la experiencia adquirida en el desarrollo de aplicaciones web, en el Instituto de Estadística Aplicada y Computación, IEAC, adscrito a la Universidad de Los Andes, Venezuela. Institución en la cual los autores trabajan en la automatización de la información administrativa y académica que se genera en dicha dependencia. Hace aproximadamente dos años, se dio inicio a esta automatización a partir del desarrollo de un sistema web que sustituiría a una antigua página web estática, la cual solo contenía datos académicos de carácter informativo para los estudiantes de Maestría y Doctorado en Estadística de esta Universidad. El nuevo sistema web inicialmente estaba basado, solamente, en *php*, *mysql*, *css* y *html*.

Por inexperiencia de los desarrolladores del nuevo sistema, éste fue desarrollado, inicialmente, para el tipo de resolución de los monitores de los computadores de escritorio de sus oficinas. Posteriormente, al realizar las pruebas en pantallas de laptops y de computadores de escritorio de otras oficinas, se determinó que, al tener, esos dispositivos, resoluciones distintas a las de los computadores en las que se estaba construyendo la aplicación, la visualización era diferente, llegando, incluso, a ser poco estética. Por lo tanto, para resolver ese problema se revisó el estado del arte relacionado con el desarrollo de aplicaciones web.

El estado del arte para el desarrollo de aplicaciones web mostraba varias alternativas para resolver el problema surgido: Una alternativa era desarrollar una aplicación por cada resolución; otra alternativa consistía en construir hojas de estilos con diferentes puntos de ruptura por cada resolución haciendo uso de *media query* y *viewport*; una tercera alternativa era utilizar algún *framework* de diseño web y una cuarta alternativa implicaba recurrir a software manejador de contenidos (*CMS*, *Content Management System*) (Centro de Apoyo Tecnológico a Emprendedores, Fundación Parque Científico y Tecnológico de Albacete, 2012).

De las alternativas mencionadas, se descartó la primera debido a que conlleva gran redundancia del software al desarrollar un sistema particular para cada resolución, con el consecuente gasto, innecesario, de recursos y de tiempo. La tercera y cuarta alternativas, también, fueron descartadas porque requerían del estudio del uso de herramientas tecnológicas adicionales para el desarrollo de aplicaciones web, distintas a las que ya conocían los autores; por lo que abordarlas produciría una interrupción en la planificación original para el desarrollo de la aplicación planteada. Por lo tanto, se decidió trabajar en base a la alternativa dos.

La alternativa dos implica disponer de un archivo de estilos *css* que contenga todas, o la mayoría de, las resoluciones de pantalla disponibles para los dispositivos actuales a través de los cuales se accederá a cualquier aplicación web en desarrollo. Es decir, el desarrollador tiene que construir, para cada aplicación particular, un archivo *css* en el cual se ajusta de manera explícita cada contenedor por resolución. Se trata de una tarea muy laboriosa, puesto que se tienen que realizar pruebas por dispositivo y por resolución, hasta lograr que se visualice la web y sus contenidos en la forma esperada, y volver a corregir cuando eso no se logra, por lo tanto, hay una recurrencia de pruebas.

Adicionalmente, a la construcción del archivo *css*, esta alternativa implica agregar en cada archivo *php* y *html*, de la aplicación, tres enlaces asociadas al *media query*, al *viewport* y al mencionado archivo *css*. En este sentido, esta alternativa dos, elimina la redundancia que es el gran defecto de la alternativa uno. Sin embargo, tiene como desventaja el abundante esquema de pruebas que hay que realizar para adaptar la aplicación a las diferentes resoluciones. Para llevar a cabo estas pruebas, también, se tienen dos opciones: la primera consiste en realizar el cambio en cada resolución, en el archivo *css*, y probar físicamente la aplicación sobre el dispositivo de cada resolución, y realizar las correcciones de ser necesarias; la segunda consiste en recurrir al uso de algún simulador de resoluciones de pantalla, los cuales permiten visualizar, de manera virtual, el diseño de una página web en distintas resoluciones. Como referencia para los lectores, los autores tienen experiencia en el uso de *ScreenFly* una aplicación online que puede ser utilizada para observar cualquier página web, bajo muchas resoluciones y tipos de celulares, pero teniendo presente que los simuladores son sólo aproximaciones a las pantallas reales, por lo tanto no siempre aportan la visualización esperada.

Es importante tener en cuenta que la alternativa dos, tiene sus dificultades, básicamente, en el hecho de que hay que ser muy riguroso en la definición de los puntos de ruptura de las resoluciones para abarcar todas las pantallas de manera precisa. Pero se cuenta con la gran ventaja que el desarrollador tiene conocimiento y control completo de la programación de la aplicación.

La alternativa dos aportó muy buena visualización a la aplicación, prácticamente, en todas las resoluciones de pantallas de celulares, *tablets*, mini laptops, laptops y computadores de escritorio, cuando la visualización no requería imágenes y menús de muchas opciones. Sin embargo, al incorporar imágenes y grandes menús a algunos archivos *php* y *html* de la aplicación, la visualización perdió eficacia en las pantallas de los celulares y *tablets*. Por lo que quedaban dos opciones: La primera, consistía en modificar la programación de esos archivos y ajustar los *css*, para que sobre esas resoluciones se pudiera observar en forma correcta las imágenes y los menús, con la consiguiente recarga del esquema de pruebas. La segunda opción, consistía en abordar el uso de algún *framework* de diseño web adaptativo.

Llegando a este punto, se decidió por abordar el estudio y uso de un *framework* de diseño web adaptativo, como una manera, casi obligada, de abordar las nuevas tecnologías asociadas al desa-

rollo web. En particular, se decidió trabajar con *bootstrap*, por ser una de las tecnologías más populares a nivel mundial (*Bootstrap 3*, el manual oficial, 2016). En este sentido, se observó que una de las principales características de *bootstrap* es que abstrae de las compatibilidades entre navegadores, poniendo, a la vez, a disposición del desarrollador un conjunto de herramientas y elementos como formularios, botones, tablas, menús, alertas y otros componentes que facilitan el desarrollo.

El uso de *bootstrap* implica descargar e instalar el paquete de herramientas ya sea en una versión compilada, en su código fuente o por componentes en la medida que se necesiten para personalizar y optimizar una aplicación web. En general, las herramientas *bootstrap* están clasificadas en tres carpetas o directorios que contienen archivos de programas reutilizables por la aplicación a desarrollar: La carpeta que contiene los *CSS* de *bootstrap*, concretamente *bootstrap.css* y *bootstrap-responsive.css*. La carpeta JS que contiene el código *javascript* necesario para el correcto funcionamiento de los *widjets* de *bootstrap*, concretamente el archivo *bootstrap.js* y su versión minimizada. La carpeta *IMG* que incluye los iconos de *bootstrap* (*Bootstrap 3*, el manual oficial, 2016).

Es importante tener en cuenta que *bootstrap*, simplemente, permite crear la estructura básica de la aplicación web, el resto del *front-end* se completa con la programación en, por ejemplo, *php*, *html* y *css* propios de la aplicación. Para la construcción de esa estructura básica, se dispone de un sistema de *grids* o rejillas; de una gran variedad de estilos *css* que se aplican por defecto a las etiquetas básicas empleadas en *html*; de una gran variedad de iconos; herramientas de interfaz como menús desplegables, botones y ejemplos de código *Javascript* para el uso y funcionamiento de *widjets*, entre otras, fácilmente integrables en la aplicación con pocas líneas de código.

Para utilizar *bootstrap* en una aplicación web, básicamente hay que incluir los archivos *css* y *js*, descargados, en cada archivo *html* y *php*, colocando los enlaces correspondientes dentro de las etiquetas *<head>*. El uso de *bootstrap*, en la actualidad, ha llegado a ser tan popular que existen diseñadores web que se han dedicado a producir millones de plantillas, dirigidas a cualquier cantidad de aplicaciones, que colocan a disposición de los desarrolladores web de todo el mundo, como una manera de agilizar su trabajo, y por supuesto como una manera de producir ganancia económica (*Bootstrap 3*, el manual oficial, 2016).

De la experiencia con *bootstrap*, por parte de los autores, se puede afirmar que estos tipos de *frameworks* tienen ventajas y desventajas. Entre las grandes ventajas, está el hecho de lograr compatibilidad de la visualización entre los navegadores y entre los dispositivos. En la capacidad para modificar la apariencia de la aplicaciones en los celulares y *tablets*; y en la capacidad para mostrar u ocultar elementos de la aplicación dependiendo del dispositivo que se esté utilizando para el acceso a la web. Mientras que entre las grandes desventajas está el hecho que no se puede reutilizar el *software* de una aplicación desarrollada donde no se haya recurrido al uso de este tipo de *framework*, como en el caso de las aplicaciones descritas por los autores. En cuyo caso, resulta más conveniente realizar una aplicación completamente nueva que adaptar la que ya se había implementado, lo que implica pérdida de recursos en el desarrollo previo. También, el hecho de tener una gran cantidad de plantillas, disponibles para los desarrolladores, podría considerarse un problema porque la búsqueda de la plantilla que mejor se adapte a una aplicación es una tarea complicada que, muchas veces, podría llegar ser tan extensa como desarrollar la aplicación completa. Por otro lado, el uso de este tipo de tecnologías implica consumo de tiempo y aprendizaje que bien pudiera emplearse en el desarrollo de la aplicación de la manera tradicional. También, está el hecho de que cada desarrollador puede construir su propia plantilla con el consiguiente gasto de tiempo en aprender la tecnología hasta lograr un producto terminado.

Del estudio y aplicación de *bootstrap*, se tomó la decisión de sólo recurrir a este tipo de *frameworks* en la construcción de los módulos futuros de la aplicación en desarrollo, y ajustar, en la medida de lo posible y en la forma tradicional, los módulos existentes, puesto que no resulta conveniente, en estos momentos, rehacer toda la programación de la aplicación sólo por el hecho de adoptar las herramientas contemporáneas del diseño adaptativo.

Siguiendo en la línea de abordar tecnologías recientes, y como una manera de proporcionarle a la aplicación web, en desarrollo, otras funciones y disponibilidad de información, se realizó el estudio y uso de herramientas de administración de contenidos digitales en web, *cms*, como una manera para que la aplicación permita crear, editar, modificar y publicar contenidos web, además de las funciones propias de los módulos ya probados. En este sentido, se decidió trabajar con *wordpress*, porque su amplitud de uso, a nivel mundial, es considerable y porque todos los *cms* actuales reúnen parámetros de calidad suficientemente altos (Casabona, 2014). Y algo muy importante, que contribuyó a esta elección, fue el hecho que *wordpress* utiliza *php* como lenguaje de programación y *mysql* como manejador de base de datos, que son dos de las herramientas con las que han venido trabajando los autores. Es de destacar que este *cms* permite cambiar por completo la apariencia de una aplicación, ya sea instalando plantillas o *themes* de diseño existentes o creando plantillas propias desde cero. Igual como sucede con los *frameworks* de diseño, existen, de manera permanente, millones de desarrolladores, independientes, creando nuevas funciones para *wordpress*, por lo que es posible encontrar todo lo que se necesite para realizar cualquier proyecto web (González, 2011).

Como *bootstrap*, *wordpress* se instala en el servidor donde se almacena la aplicación web. Una característica de este tipo de *cms* es que cada aplicación web tiene dos áreas: El *front-end*, que es el área pública donde los visitantes pueden ver el contenido e interactuar con la aplicación, y el *back-end* que es el área de administración o desarrollo propiamente dicho de la aplicación, desde donde se puede publicar artículos, crear páginas y moderar comentarios, así como agregar o eliminar plantillas, *widgets*, *plugins* y utilizar todas las funciones y opciones de *wordpress*. Otra de las características relevantes de este *cms* es que está orientado al diseño adaptativo.

A manera de información general, recordemos que los *plugins* son programas que agregan funciones a una aplicación web. Estos programas, que pueden ser gratuitos o pagos, son creados por desarrolladores independientes, y se añaden como componentes externos al desarrollo propio de la aplicación web. En general, los *plugins* constituyen una forma de añadir a *wordpress* funciones con las que no cuenta originalmente. Por otro lado, los *widgets* son componentes que se utilizan sólo para personalizar la apariencia de aplicación web que, a diferencia de los *plugins*, son visibles a los usuarios, y pueden ser colocados en cualquier parte de la plantilla de diseño.

En relación con *wordpress*, una de sus grandes ventajas es que tiene gran flexibilidad a la hora de decidir cómo se desea mostrar el contenido. Una muestra de ello son las plantillas o *themes* disponibles, por lo que cuando se crea una aplicación web una de las actividades principales es definir la plantilla con la que desea se visualice.

La experiencia de trabajar con *wordpress* lleva, a los autores, a afirmar que los manejadores de contenidos, constituyen, en la actualidad, herramientas muy útiles para el desarrollo de aplicaciones web, sin embargo, tiene desventajas evidentes. Entre las ventajas, por supuesto, resalta el hecho que permiten construir aplicaciones con diseño adaptativo, fáciles de construir y de apariencia fácilmente modificable, una vez que se tienen las plantillas determinadas de manera precisa,

mientras que entre las desventajas está el hecho que no se pueden reutilizar, de manera directa, las aplicaciones desarrolladas con tecnologías tradicionales, lo que implica que, prácticamente, hay que desechar todos los desarrollos previos y comenzar el proyecto desde cero.

V. OPINIÓN DE LOS DESARROLLADORES EXPERTOS

A continuación se resume la opinión general de desarrolladores de nivel mundial, como una forma de contrastar o encontrar similitudes con la opinión de los autores del presente artículo.

Las ventajas del diseño web adaptativo se encuentran en el hecho que los usuarios ven la aplicación de la mejor manera posible en el dispositivo desde el que acceden, provoca gran impacto en el usuario ya que se suele asociar con mucha creatividad e innovación tecnológica en el diseño, se reducen los costos de creación puesto que no es necesario desarrollar aplicaciones para distintas resoluciones, cualquier actualización o modificación en el diseño se ve reflejado en todas las plataformas lo que reduce tiempos y márgenes de errores, al hacer una búsqueda aparecerá una *url* única en los resultados, con lo que se ahorra redirecciones a distintos dominios o subdominios por lo que se optimiza su posicionamiento en los rankings de los buscadores. El acceso a la web desde un celular permite al desarrollador sacar beneficio de las tecnologías móviles, explotando funciones como la geo localización y los eventos táctiles (Vega, 2013, Martínez y Salgado, 2013).

Como todas las tecnologías, el diseño adaptativo, no implica que sea la panacea del desarrollo web. Una de las desventajas la constituyen los tiempos de carga, del lado del cliente, ya que el usuario debe descargar todo el código fuente, aún sin ser necesario, muchas veces, e incluso si el diseño no está optimizado con imágenes adaptadas a distintas resoluciones debe descargar imágenes de gran resolución. Se emplea más tiempo en optimizar el diseño para las distintas resoluciones respecto de un desarrollo web no adaptativo. Es muy difícil convertir un sitio web en adaptativo si no lo es desde el principio (Vega, 2013, Martínez y Salgado, 2013).

VI. CONCLUSIONES

El diseño web adaptativo es realmente un mito dentro del mundo del desarrollo de aplicaciones web, ya que a pesar de constituir una marcada evolución de las tecnologías, tiene evidentes desventajas. Entre las grandes ventajas se destaca el uso de plantillas de diseño construidas por desarrolladores independientes, de gran vistosidad para los usuarios, pero que sacrifica control por parte de quienes implementan las aplicaciones, recargándolas de un conjunto de componentes que no siempre se requieren en todas las aplicaciones. Por otro lado, se pierde, en muy alto grado, la reusabilidad del software de las aplicaciones implementadas sin este tipo de tecnologías por lo que su migración es, prácticamente, imposible; para ello se tiene que comenzar de cero abandonando los antiguos desarrollos, sin tomar en cuenta los recursos que se hayan invertido en aquellos, por lo que, con seguridad, un altísimo porcentaje de aplicaciones tradicionales seguirán utilizándose por varios años más. En realidad, las tecnologías de los componentes basados en diseño adaptativo siguen siendo los mismos desarrollos por resoluciones, sólo que abstraen de este tipo de desarrollos, por lo que no es pecado mortal crear aplicaciones con la tecnología tradicional, siempre y cuando se logre eficacia en cuanto a la velocidad de respuesta y al uso de recursos computacionales.

REFERENCIAS

- Belloch C. (2017). *Diseño Instruccional*. Disponible en línea: <http://www.uv.es/bellochc/pedagogia/EVA4.pdf>. Consultada en enero de 2017.
- Bootstrap 3. El manual oficial*. Disponible en línea: http://librosweb.es/libro/bootstrap_3/. Consultada en diciembre de 2016.
- Casabona J. (2014). *Responsive Design with WordPress*. Disponible en línea: <http://ptgmedia.pearsoncmg.com/images/9780321957412/samplepages/0321957415.pdf>. Consultada en enero de 2017.
- Centro de Apoyo Tecnológico a Emprendedores, Fundación Parque Científico y Tecnológico de Albacete. (2012). *Estudio de los sistemas de gestión de contenidos web*. Disponible en línea: https://www.bilib.es/fileadmin/estudio_sistemas_gestion_contenidos_web_cms.pdf. Consultada en marzo de 2017.
- González A. (2011). *Tutorial de wordpress*. Disponible en línea: <http://antoniogonzalezm.es/wp-content/uploads/2011/06/Tutorial-de-Wordpress.pdf>. Consultada en abril de 2017.
- Mahande R. y Surhono H., (2016). *Mobile Learning: Trends and Challenges for Vocational Education in Indonesia*. Disponible en línea: <http://www.ridwanmahande.net/wp-content/uploads/2016/09/Paper-ICVET2016-Ridwan-Herman.pdf>. Consultada en febrero de 2017.
- Marcotte E. (2017). *Responsive Design*. Disponible en línea: [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). Consultada en febrero de 2017.
- Martínez E. y Salgado C. (2013). *Diseño web adaptativo o responsivo*. Disponible en línea: <http://www.revista.unam.mx/vol.14/num1/art07/art07.pdf>. Consultada en diciembre de 2016.
- Vega A. (2013). *Responsive Web Design: Interfaces Web Adaptables al dispositivo empleando HTML5 y CSS3*. Trabajo de fin de carrera para optar al grado de Ingeniero en informática. Universidad de Alcalá, España.

Como citar este artículo:

Núñez, Y. y Maldonado, L. (2017). El mito del diseño web adaptativo. *Aprendizaje Digital*, Vol 2 N° 2, pp. 9 -16.