



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

DISEÑO WEB ADAPTATIVO PARA LA PLATAFORMA EDUCATIVA ZERA.

Tesis presentada en opción al título de Ingeniero en Ciencias
Informáticas

Autor: **Adrian Arjona Clavelo**

Tutor: **Ing. Yerandy Manso Guerra,
Dra. C. Roxana Cañizares
González**

Ciudad de la Habana, 2014

Agradecimientos

Todo lo que soy y seré no se creó por arte de magia. Mi vida desde que nací, se vió afectada por muchas personas a las cuales quiero mucho y quiero agradecer.

Primero: A todos aquellos que me han apoyado de una forma u otra en esta última etapa de estudios, a mi *hermandad* de la universidad, a Peña, Nuria, Dairon, Anadelis, Dariel, Armandito y Andy que no por ser el último en entrar en el cuarto deja de ser importante; a todos ustedes muchas gracias.

Segundo: A mis tutores; a mis exigentes tutores, que paciencia y comprensión tuvieron hacia mí; a ustedes dos, les deseo lo mejor y les doy las gracias también.

Tercero: A aquellos que no son familia carnal, pero que sin importar los lazos sanguíneos también estuvieron apoyándome todo el tiempo; a mi tía Mirtica, a Nancy, Delvin, Mailen; a ustedes, muchas gracias.

Cuarto: A mi novia, la persona que siempre soñé tener a mi lado y que la vida me ha regalado, a ella que siempre entiende todos mis enredos y me demuestra su amor día tras día; a ti Cindy, muchas gracias.

Quinto: A mi familia, que han estado para mí, en todas las faces de mi vida, que me han apoyado en cada decisión que he tomado, que me han aconsejado y que han soportado todas mis *malas crianzas* y errores. A mi mamá Maura, a mi papá Nestor, a mi tía-abuela Umbelina, a mi tío-abuelo Berto; a todos ustedes, muchas gracias.

Pero sobre todo, a aquellos dos que me dieron la vida, a quienes tengo que agradecer eternamente; a mi papá y a mi mamá; con todo el amor del mundo les dedico esta tesis.

Resumen

La comunicación, es la habilidad que brinda al ser humano la posibilidad de intercambiar ideas, pensamientos, etc. Pero en algunas ocasiones, factores como la distancia o la falta de presencia en el contenido mostrado pueden interrumpir o incluso destruir el flujo de información. Durante el proceso de enseñanza-aprendizaje, problemas como estos pueden hacer fracasar al educando. Por cuestiones de esta índole las herramientas educativas han creado estrategias para corregir este tipo de adversidad y lograr apoyar a la educación de la sociedad. El diseño de cualquier sistema informático es de vital importancia para el usuario ya que es la parte con la que este interactúa. Distintos tipos de diseños son aplicados a diversas aplicaciones en aras de mejorar la aceptación del usuario. El diseño adaptativo forma parte esencial en la construcción de los sitios web de la actualidad, pues con este ya es posible navegar a través de ellos sin importar el dispositivo de navegación. Por tanto se propone la inclusión del paradigma adaptativo en la plataforma educativa **ZERA** para que esta pueda ser visualizada en diferentes anchos de pantalla. Este trabajo muestra una serie de actividades para satisfacer las necesidades que la plataforma educativa **ZERA** posee en estos momentos. La investigación realizada para dar respuesta a la necesidad que presenta la plataforma de poder adaptarse a diferentes tipos de pantallas de dispositivos móviles aborda temas acerca del uso de buenas prácticas para la optimización de sitios web. También se realizan estudios de similares, y finalmente se proponen diseños de interfaces basados en principios de estética, interactividad y comodidad del usuario futuro.

Palabras claves: experiencia de usuario, dispositivo móvil, diseño adaptativo, estrategias de contenido.

Índice general

Introducción	1
1. Fundamentación teórica	7
1.1. Evolución del diseño web	7
1.2. Diseño web adaptativo	8
1.2.1. Características del diseño web adaptativo	9
1.2.2. Tecnología vs educación	13
1.3. Herramientas o tecnologías para lograr la adaptación de contenidos	14
1.3.1. Estrategias de contenido	14
1.3.2. Imágenes fluidas	16
1.4. Elementos para lograr un diseño web adaptativo	18
1.4.1. HTML5	18
1.4.2. Viewport	21
1.5. Cascading Style Sheet para un diseño adaptativo	22
1.5.1. Posicionamiento del código	23
1.5.2. Frameworks	23
1.6. Media queries en el diseño web adaptativo	25
1.6.1. Media queries	25
1.6.2. Puntos de quiebre o break points	26
1.7. Diseño web adaptativo en plataformas educativas	27
1.7.1. Tipos de LMS	27
1.8. Conclusiones Parciales	30
2. Propuesta Solución	32
2.1. Estándares de codificación	33
2.2. Puntos de quiebre	33
2.3. Imágenes y <i>sprite</i>	35
2.4. Modificaciones en el código	35

2.4.1. Lado del cliente	35
2.4.2. Lado del servidor	42
2.5. Diseño de estrategias para los <i>breakpoints</i>	50
2.6. Eventos y animaciones	52
2.7. Uso de jQuery Lazy	52
2.7.1. Integrando jQuery Lazy	53
2.8. Conclusiones Parciales	53
3. Validación	54
3.1. Métodos de prueba	54
3.2. Procedimientos de validación	55
3.2.1. Test de “usabilidad”	55
3.2.2. Encuesta	55
3.2.3. Pruebas de rendimiento	58
3.3. Conclusiones Parciales	60
Conclusiones	61
Recomendaciones	62
Referencias bibliográficas	63
Acrónimos	66

Índice de figuras

1.1. Evolución Web según Google.	8
1.2. Estadísticas comScore	10
1.3. Concepto de Responsive Web Design	11
1.4. Comparación del uso del diseño web adaptativo y el diseño web para móvil en las ultimas décadas.	12
1.5. Desarrollo de aplicaciones para dispositivos.	14
1.6. Visualización de la etiqueta <i>div</i> en diferentes entornos; de izquierda a de- recha: PC y móvil.	21
1.7. El cuadrante rojo delimita el área asociada al viewport.	22
2.1. Resultados obtenidos en el menú de la plataforma.	48
2.2. Prototipo de diseño para ZERA.	50
2.3. Prototipo de diseño para ZERA.	51
3.1. Resultados de la encuesta ZERA.	56
3.2. Resultados de la encuesta ZERA.	57
3.3. Resultados de la encuesta ZERA.	57
3.4. Pruebas de rendimiento.	59
3.5. Pruebas de rendimiento.	59
3.6. Pruebas de rendimiento.	59

Índice de tablas

1.1. Puntos de quiebres para el RWD	27
2.1. Modelación de la propuesta de solución para lograr que la plataforma educativa <i>ZERA</i> incorpore las buenas prácticas existentes para el diseño adaptativo a desarrollar.	33
2.2. Puntos de quiebres para el desarrollo en la plataforma <i>ZERA</i>	34
2.3. Cambios realizados en la plataforma.	49
3.1. Atributos tomados en cuenta para las pruebas de rendimiento.	58

Introducción

El diseño de páginas web es un término que implica una gran gama de actividades tales como la planificación, creación, modelado y diseño; codificación de páginas, portales, la programación y la organización profesional de estas, que en conjunto, componen un sitio web [Peterson, 2014].

La construcción de sitios web es un arte que tuvo su origen hace más de 20 años cuando surgió por primera vez lo que se conoce hoy por Hyper Text Markup Language (**HTML**), el cuál tuvo una rápida evolución de un simple texto a poder incluir imágenes y Cascading Style Sheets (**CSS**). Contando con estos lenguajes, más la introducción del estándar Document Object Model (**DOM**) la web poco a poco fue siendo aceptada en la red de redes [Matthew, 2010].

Al principio cuando toda web era visualizada a través de máquinas de escritorio, el diseño de estas páginas no varió mucho; todo empezó a cambiar a partir del 2007 cuando Apple presentó por primera vez al **iPhone**. Por el momento la solución fue crear sitios web a la medida de esas nuevas pantallas, es decir, sitios separados del *regular* [Peterson, 2014].

La invención del **iPhone** le dio una perspectiva diferente a todos los desarrolladores de sitios web, pues aunque por el momento, la solución consistía en mantener dos sitios web, el gasto para estas mantenciones era elevado. Mientras tanto, las tecnologías de la comunicación se fueron desarrollando y llegó un momento en que no solo era **Apple** quien diseñaba dispositivos capaces de visualizar la web, sino que para ese entonces la telefonía móvil, de manera general, había podido alcanzar a la industria **Apple**; la meta para los diseñadores desde ese instante fue la creación de un único sitio web que pudiera visualizarse en cualquier tipo de dispositivo y tamaño de pantalla.

El deseo de formar una web que se adaptara a cualquier ancho de pantalla requería de complementos que fueran capaces de ajustarse al espacio disponible, en otras palabras; que se *auto-redimensionaran*, esta característica es conocida como *flexibilidad* y, además era necesario saber en que momento se estaba en presencia de una resolución u otra, cosa que se logra con sentencias (*media queries*) **CSS** que preguntan al navegador cual es el espacio disponible [Marcotte, 2011].

Tras algunos intentos de la comunidad de desarrollo web para dar con una respuesta acerca del tema; el diseñador web Ethan Marcotte logra *unir* los conceptos de **flexibilidad** y **media queries** en un artículo denominado *A List Apart* [Peterson, 2014] introduciendo así, por primera vez, el término Responsive Web Design (**RWD**) y sienta las bases para el problema de *los diversos tamaños de pantalla*.

Debido al acelerado avance en las comunicaciones, la búsqueda de un **RWD** ha sido el *santo grial* de todo diseñador web. Muchas soluciones han surgido; los diseñadores han desarrollado incontables *frameworks CSS*¹ como *Bootstrap*, *Blueprint* o *Metro* con el objetivo de facilitar el trabajo de diseño y sentar las bases para un **RWD**.

La creación de un **RWD** es de gran importancia para el diseño de un sitio web hoy en día, pues este te permite [W3C, 2014b]:

- Con una sola versión en **HTML** y **CSS** cubrir todas las resoluciones de pantalla, es decir, el sitio web creado estará optimizado para todo tipo de dispositivos: PCs, tabletas, teléfonos móviles, etc.
- Mejorar la experiencia de usuario a diferencia de lo que ocurre, por ejemplo, con sitios web de ancho fijo cuando se acceden desde dispositivos móviles.
- Reducir los costes de creación y mantenimiento, pues se evita tener que desarrollar aplicaciones para versiones móviles, por ejemplo, una aplicación específica para **iPhone**, otra para móviles **Android**, etc.
- Desde el punto de vista de la optimización de motores de búsqueda, sólo aparecería una URL en los resultados de búsqueda, con lo cual se ahorran re-direcciones y los fallos que se derivan de éstas.
- También se evitarían errores al acceder al sitio web en concreto desde los llamados "social links", es decir, desde enlaces que los usuarios comparten en medios sociales tales como Facebook, Twitter, etc y que pueden acabar en error dependiendo de qué enlace se copió (desde qué dispositivo se copió) y desde qué dispositivo se accede.

Los frameworks **CSS**, aunque son una solución factible para ahorrar tiempo en la construcción de un sitio web y además brindan una gama de estilos que mejoran considerablemente la experiencia del usuario, teóricamente no logran alcanzar un **RWD** pleno. Los frameworks en sí, no integran buenas prácticas para mejorar la accesibilidad web, pues según [Manso, 2014] para lograr **RWD** adecuado, limpio y accesible es necesario tener en cuenta algunos aspectos fundamentales de **HTML**, tal es el caso de la semántica,

¹Hojas de estilo previamente elaboradas, listas para ser usadas en un sitio web.

esta se refiere al uso de etiquetas **HTML** para reforzar la semántica o el significado de la información en los sitios web, la especificación HTML5 incorpora etiquetas con significado semántico como son: **header**, **nav**, **section**, **article**, **aside**, **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 el objetivo de mejorar los diseños web y a su vez la experiencia de los usuarios que navegan la Internet, principalmente aquellos que navegan a través de sus dispositivos móviles, dígase **Smartphones**; la World Wide Web Consortium (**W3C**) ha estandarizado una serie de elementos a tener en cuenta para la creación de sitios web. Estos elementos han sido escritos de manera general para que sean usados a lo largo de la historia sin importar los cambios sufridos en el lenguaje de maquetado.

Estas "buenas prácticas" también tienen en consideración el ancho de banda y el costo, esto es así ya que las redes de los dispositivos móviles pueden ser en algunas ocasiones un poco lenta en comparación con las conexiones de PC de escritorio, pues estos dispositivos están limitados a visualizar algunos tipos de contenidos. Por otro lado, este tipo de dispositivo por lo regular posee un método de entrada de datos bastante complejo, lo cual dificulta este proceso principalmente a la hora de introducir URIs de sitios web. A pesar de contradicciones como estas, esta tecnología se ha vuelto bastante popular ya que es [Guidelines, 2014]: personal, personalizable, portable, siempre está conectado e incrementadamente multi-funcional; más allá de su propósito original; las comunicaciones.

Hoy en día, los dispositivos móviles constan con una serie de funcionalidades muy sencillas y útiles al ser humano, entre ellas [Guidelines, 2014]: reconocimiento de ubicación, funcionamiento operable con una mano, conectado las **24 hrs.** del día la Internet.

Es por ello que, el desarrollo general de una aplicación web con intención para móvil debe explotar al máximo nivel este tipo de funcionalidades, para así desarrollar de manera correcta un **RWD**, sin perder de vista aspectos como [Guidelines, 2014]:

- comportamiento general.
- links y navegación.
- capa de página y contenido.
- definición de página.
- métodos de entrada del usuario.

Hoy en día la sociedad actual tiene acceso a tecnologías de bolsillo de alto procesamiento y conectadas al igual que un computador convencional. Además de proveer acceso a Internet se han convertido en el principal modo de comunicación electrónica [Guy et al., 2010]: correo electrónico y mensajería instantánea.

Las 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* por sus siglas en inglés) se utilizó por primera vez en Estados Unidos a finales de los 90 e instaurándose en Europa a inicios del siglo XXI [Pisanty et al., 2010].

[Manso, 2014] plantea que en Cuba, aunque todavía insuficiente, el desarrollo de las telecomunicaciones ha mostrado un notable ritmo de crecimiento en los últimos 5 años. Juventud Rebelde además anunció que apenas se cuenta con un 25,8% de densidad telefónica, lo que se traduce en un total de 2 900 mil líneas. Cuba brinda más de 30 mil servicios de transmisión de datos y acceso a Internet, que son utilizados por 1,7 millones de usuarios, entre los que se han priorizado fundamentalmente los sectores vinculados a las actividades educacionales, de salud, científicas y otras de carácter productivo en general.

El aprovechamiento de este auge, principalmente en la rama de la educación sería de gran ayuda al proceso educativo estudiantil que hoy tiene Cuba. Es por ello que la Universidad de las Ciencias Informáticas (UCI) tiene, entre otras tareas, la de enriquecer con productos informáticos de alta calidad el sistema educacional del país.

El centro Centro de Tecnologías para la Formación (FORTES) de la UCI en Cuba, cuenta con varios productos en desarrollo entre ellos la Plataforma Educativa ZERA, esta, después de varios ciclos de liberaciones aún no sigue el paradigma del diseño *adaptativo*, pues aunque usa el framework **CSS Bootstrap**, la plataforma no incorpora ninguna estrategia para lograr que las imágenes se adapten a tamaños variados de pantallas y ancho de banda, lo que influye negativamente en el rendimiento y su visualización en diferentes dispositivos, tampoco cuenta con una serie de *breakpoints*² personalizados a las necesidades de la plataforma. Además con la evolución de **HTML4** a **HTML5** no se ha hecho uso de las etiquetas semánticas que este último presenta.

Por lo expuesto se plantea el siguiente **problema científico**: ¿Cómo permitir la visualización de la plataforma educativa **ZERA** en diferentes tamaños de pantalla?

Objeto de estudio: Tecnologías para la visualización de plataformas web en

²breakpoints: Puntos de quiebre; definen los anchos de pantalla que podrá experimentar una aplicación.

diferentes ancho de pantalla.

Objetivo general: Desarrollar un diseño web adaptativo en la plataforma educativa **ZERA** que cumpla con las buenas prácticas existentes para aumentar la visualización de esta en diferentes tamaños de pantalla.

Hipótesis: Si se desarrolla un diseño web adaptativo en la plataforma educativa **ZERA** que incorpore las buenas prácticas existentes entonces se aumentará su visualización en diferentes tamaños de pantalla.

Campo de acción: El diseño web adaptativo en la plataforma educativa **ZERA**.

A continuación se mencionan los métodos científicos utilizados para el desarrollo de la investigación.

Métodos Teóricos:

Histórico – lógico: es utilizado para realizar revisiones críticas sobre la bibliografía existente y conformar un criterio sólido sobre el estado de las investigaciones en el área de la gestión de temas de interfaz de usuario.

Analítico – sintético: es utilizado en el estudio de los conceptos relacionados con la gestión de temas de interfaz de usuario y los sistemas similares existentes, para luego sintetizar toda la información analizada y conformar la propuesta de solución.

Modelación: es usado para crear propuestas de soluciones (prototipos) a través de la modelación de ideas y referentes teóricos vistos en las fuentes bibliográficas consultadas.

Métodos Empíricos

La Observación Científica: sirvió para valorar las tecnologías y productos existentes, dando a conocer de manera detallada lo que se desea, hace falta hacer y cómo hacerlo.

Encuesta: sirvió para comparar el trabajo desarrollado en la plataforma contra la versión anterior de la misma haciendo hincapié en algunos aspectos importantes de la vista.

Tareas a resolver:

- Análisis del **RWD** como estrategia para lograr una mejor experiencia de usuario en

distintos tipos de ancho de pantalla.

- Análisis de las tecnologías usadas en el desarrollo del **RWD**.
- Análisis y selección de estrategias para lograr la flexibilidad en las imágenes y plantilla de la plataforma.
- Modelación de la propuesta de solución para lograr que la plataforma educativa ZERA incorpore las buenas prácticas existentes para el **RWD** a desarrollar.
- Validación de la solución implementada a partir de las pruebas ingenieriles existentes.

Para la elaboración del presunto documento se estableció la siguiente estructura:

Capítulo1: "Fundamentación Teórica". En este, se brinda una visión general de los aspectos relacionados con los portales web y los conceptos fundamentales asociados a ellos para el estudio de los mismos. Se analizan las herramientas y tecnologías a utilizar en el diseño que estos presentan y se realiza un análisis a través de Internet con el objetivo de encontrar en ellos aspectos comunes que tributen a mejorar la accesibilidad de la plataforma.

Capítulo2: "Propuesta de Solución". En esta sección se detalla todo el proceso de desarrollo ejecutado para dar cumplimiento a las tareas listadas así como la estructura final que tendrán las diferentes interfaces que adoptará la plataforma.

Capítulo3: "Validación". En él, se harán las pruebas pertinentes para comprobar si el cumplimiento de las tareas propuestas finalmente satisfacen al requisito de calidad, accesibilidad web.

Capítulo 1

Fundamentación teórica

Introducción

En este capítulo se especifica la fundamentación teórica que soporta la investigación sobre la gestión de interfaz de usuario adaptables, la cual incluye el estudio del estado del arte, donde se realiza el análisis de soluciones existentes. Además se lleva a cabo el estudio de las tendencias en las tecnologías, metodologías y herramientas actuales, en las que se apoya la solución de la problemática a resolver.

1.1. Evolución del diseño web

Hace ya más de una década, cuando la web fue ganado fama y atención, se decía que para poder visualizar una página web solo hacía falta una PC de escritorio con algún navegador web instalado ya fuera **Microsoft Internet Explorer** o **Netscape**. Este pensamiento se mantuvo vigente por más de 10 años.

El cambio de perspectiva fue cuando surgen las redes **WIFI**. Para ese entonces ya no era necesario una PC de escritorio, ni siquiera una laptop para conectarse a Internet. Para realizar una conexión a la red de redes los usuarios solo tenían que disponer de algún dispositivo capaz de hacer esa conexión.

El primer paso para resolver este problema lo dio **Apple** con el lanzamiento del **iPhone** y del **iPod Touch**. Estos dispositivos fueron capaces de visualizar una página web completa al igual que una PC de escritorio, todo gracias a su navegador web **Safari** capaz de soportar tecnologías web que para ese entonces Internet Explorer no podía. Ya en 2007 el **iPhone** era capaz de soportar **HTML5**, cosa que **Microsoft** demoró un poco en conseguir [Peterson, 2014].

En el 2010, gracias a Ethan Marcotte, la experiencia de los usuarios en la web mejoró considerablemente con la inclusión del término **RWD**. Gracias a esto; no importa donde te encuentres, ni que dispositivo uses, no tendrás que esforzarte mucho para poder leer de manera agradable un sitio web.

El desarrollo web ha evolucionado de una página web meramente estática hasta aquella que es capaz de adaptarse al medio que lo visualiza. Algunos autores como [Gustafson, 2011] o [Marcotte, 2011] comparan esta habilidad con la evolución de las especies o con las proporciones de las que dispone un pintor cuando va a pintar. Lo cierto es que ya se cuenta con una web que puede visualizarse en cualquier pantalla.

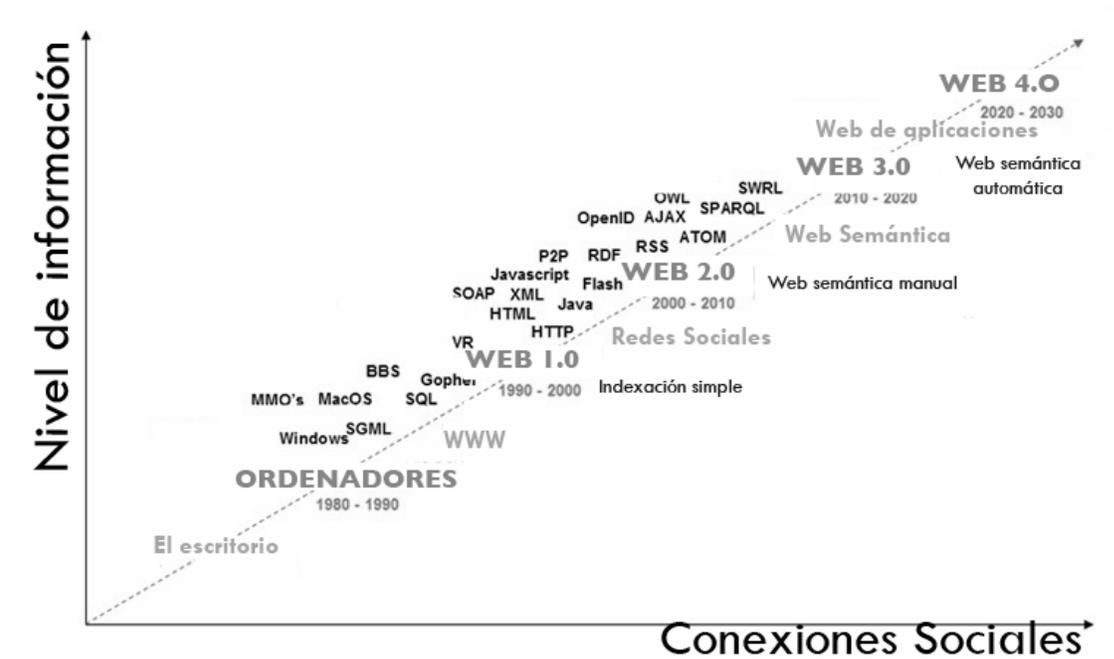


Figura 1.1: Evolución Web según [GoogleTrends, 2015].

1.2. Diseño web adaptativo

El diseño *adaptativo* es usado mayormente con el objetivo de crear páginas web que puedan ser fácilmente visualizadas desde cualquier dispositivo móvil (*laptops, lectores de pantalla, smartphones, tabletas, etc*) o PC de escritorio. Para comprender mejor este concepto es necesario imaginar dos páginas; una, con un ancho de pantalla fijo y otra con

un ancho de pantalla variable y observar su comportamiento y en diferentes ambientes de presentación.

El sitio diseñado con un ancho fijo; al ser visualizado en ambientes que no *encajan* en las dimensiones estipuladas para este, deberá ser desplazado manualmente sobre la horizontal para poder visualizar todo el contenido presentado, aunque es posible que existan otros sitios prácticamente idénticos a este, pero personalizados para resolver el problema con estos cambios de ambiente[Matthew, 2010].

De otra manera [Peterson, 2014] afirma que con un diseño adaptativo es posible tener una sola página que pueda adaptarse a cualquier ambiente de presentación, es decir, el diseño puede re-dimensionarse de ser necesario.

[Marcotte, 2011] plantea que el **RWD** es un método para diseñar sitios web flexibles, que no se vasa 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.

Otros autores también dan opiniones acerca del tema, 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.

Prácticamente se habla de un diseño en el que las plantillas web son flexibles y a medida que cambien los dispositivos el diseño debe irse adaptando al espacio disponible. En este caso [Snell, 2013] concluye que el **RWD** provee una experiencia positiva e intuitiva independientemente del dispositivo, sin requerir sub-dominios separados para los dispositivos móviles, to do esto haciendo uso de las reglas de **CSS** media queries logrando que todos los elementos se adapten al ancho y orientación de los dispositivos.

1.2.1. Características del diseño web adaptativo

El diseño *adaptativo* parte de dos componentes principales: **flexibilidad**, que significa que las dimensiones horizontales de los objetos deben ser tratadas en unidades porcentuales para que puedan responder de manera correcta a los anchos de pantalla, y los **media queries**, los cuales permiten usar reglas **CSS** para cambiar el diseño del sitio web dependiendo del ancho de pantalla[Peterson, 2014].

El diseño *adaptativo* te permite crear interfaces en el sitio capaces de adaptarse a cualquier tamaño de pantalla haciendo uso de un solo documento de código. Cuando se implementa un diseño *adaptativo* quiere decir que tu sitio también esta optimizado para la búsqueda. Es la manera de hacer sitios web para que sean fácilmente visualizados y usados

en cualquier tipo de dispositivo y tamaño de pantalla, desde los móviles más pequeños hasta los monitores más grandes de computadoras de escritorio.

Con el aumento de la utilización de los **Smartphone** por parte de los usuarios para realizar cualquier tipo de consulta en Internet, el **RWD** se ha convertido en imprescindible para cualquier sitio web que quiera mantenerse a la altura de sus competidores.

El **RWD** es una serie de técnicas que permiten que el diseño de una web se adapte automáticamente al tamaño y características de la pantalla del usuario que la está viendo. De esta manera se evitan tener versiones móviles adaptadas para cada dispositivo.

El principal concepto en el que se apoya el Diseño Web Adaptativo es en abandonar los anchos fijos de nuestra web. Estos deberán ser fluidos. De esta manera cuando la web sea mostrada a través de la pequeña pantalla de un móvil todos los elementos de la web se harán más pequeños guardando la proporción entre ellos.

Entre las ventajas que proporciona el **RWD** se puede destacar las siguientes [[comscoreanalytics, 2014](#)]:

- Hoy en día los usuarios de Internet no realizan la navegación desde su ordenador personal el 100 % de las veces y muchos de los usuarios quieren que el mismo contenido mostrado en la web sea el mostrado en la versión para móvil. Con el **RWD** solo se necesita una versión del **HTML** y **CSS** para cubrir todas las resoluciones de pantalla, es decir, el sitio web creado estará optimizado para todo tipo de dispositivos: PC, tabletas, teléfonos móviles, etc. Esta característica es conocida como **”Paridad de Contenido”**.

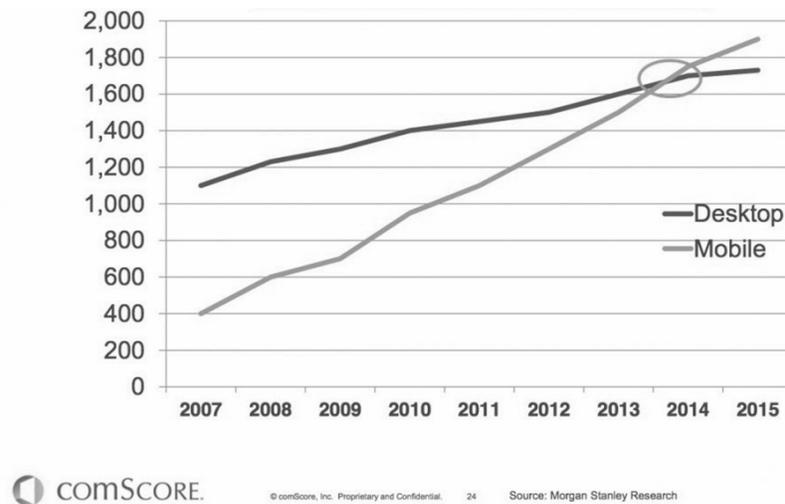


Figura 1.2: Navegación desde PC y móviles según comScore [[comscoreanalytics, 2014](#)].

- Mejora la experiencia de usuario a diferencia de lo que ocurre, por ejemplo, con sitios web de ancho fijo cuando se acceden desde dispositivos móviles. La usabilidad es un factor muy importante para fidelizar con los usuarios.
- Se evitan errores al acceder al Sitio Web en concreto desde los llamados “Social Links”, es decir, desde enlaces que los usuarios comparten en medios sociales tales como Facebook, Twitter, Google+, entre otros, y que pueden acabar en error dependiendo de qué enlace se copió y desde qué dispositivo se accede.
- Se evita tener que desarrollar aplicaciones nativas para versiones móviles, como por ejemplo, una aplicación específica para iPhone, otra para móviles Android o para móviles Blackberry, entre otros.
- Desde el punto de vista de la optimización de motores de búsqueda, sólo aparecería una URL en los resultados de búsqueda, con lo cual se ahorran redirecciones y los fallos que se derivan de éstas.



Figura 1.3: Concepto de Responsive Web Design[Marcotte, 2011].

La *optimización en motores de búsqueda* por sus siglas en inglés Search Engine Optimization (**SEO**) es una característica que deben poseer los sitios web para que estos puedan ser una opción válida cuando se realizan búsquedas en la web, y el **RWD** brinda grandes ventajas en cuanto al tema. Por ejemplo **Google** recomienda el uso de un **RWD**, para mejorar la visualización de un mismo sitio en varios dispositivos.

Con un **RWD** se evitarían problemas con el duplicado de datos, si fuese el caso se utilizaría una sola **URL** para los accesos a la información que se muestran dentro del sitio,

puesto que la misma vista estaría disponible para escritorio y móvil. El diseño adaptativo, nos permite que la misma **URL** sirve para ser compartida en diferentes medios, como redes sociales (Facebook, Twitter, Gplus), por lo cual se podría popularizar nuestro sitio mediante dichas redes bajo una misma URL, sin tener que desviar tráfico a vistas para entornos móviles o de escritorios, en los cuales podría encontrarse el usuario que lee la información o publicidad referente a nuestro sitio.

Por ende se puede decir que el **RWD** no es más que un técnica que basa su comportamiento en el uso de componentes flexibles y que pueden reajustarse de acuerdo al dispositivo en el que se renderiza el sitio web. Este evita problemas con la paridad de contenido y las URL, perfeccionado así el sitio web.

Es de entender que el desarrollo de sitios web esté volcado hacia este tipo de diseño, ya que la principal problemática con los dispositivos móviles es la incapacidad de visualizar correctamente las aplicaciones web, al contrario del **RWD**. Como solución aparecieron las *web app*¹, las cuales resolvían el problema pero aún representan una carga de más a la hora del mantenimiento, por lo que no son consideradas una opción factible.

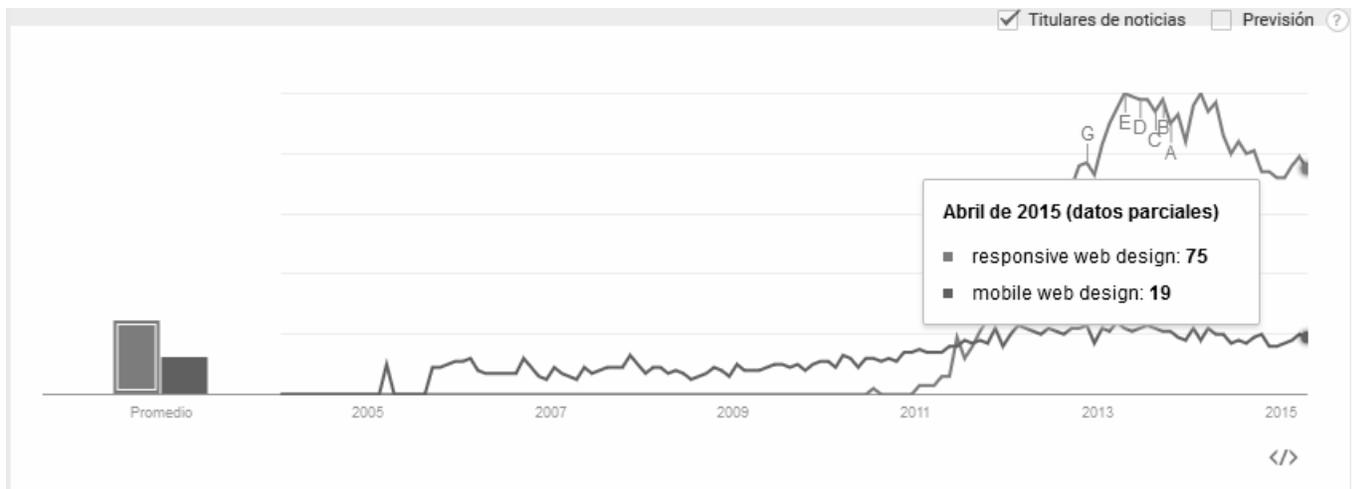


Figura 1.4: Comparación del uso del diseño web adaptativo y el diseño web para móvil en las ultimas décadas. [GoogleTrends, 2015].

¹Sitios web desarrollados para poder ser visualizados en móviles, pero que carecen de interacción y presencia en el diseño.

1.2.2. Tecnología vs educación

El concepto de que la web solo sirve para propaganda y recreación ya es un mito. Actualmente la tecnología, en especial la tecnología móvil ha sido aceptada en la rama de la educación por el potencial que esta representa en la vida de la sociedad. Según estudios realizados[Google, 2012] está demostrado que el 90 % del tiempo es basado en el uso de pantallas, digase *laptops*, *móviles*, *televisores* y *tablets* y solo el 10 % del tiempo se dedica a escuchar la radio o leer revistas o periódicos.

El ser humano normalmente se adapta a lo más fácil de hacer o modifica el entorno para que sea mas cómodo. Es por ello que en términos de portabilidad, las laptops ya no representan una comodidad con respecto a los *smartphones* o *tablets*[Google, 2012]. Por tales razones los centros educacionales en lugar de hacerle la guerra a este tipo de tipo de dispositivo han aceptado su presencia y a la ves han aprendido a hacer uso de estos. Estos artefactos; son regidos por normas, que se deben cumplir por los educandos para evitar el uso incorrecto de este instrumento, y así aprovechar al máximo las opciones que estos brindan. La relación *estudiante-dispositivo-profesor* representa un eslabón importante en la educación pues ella permite que ambas partes puedan sacar provecho del tiempo de estudio disponible, independientemente de su ubicación, además permite enriquecer todas las materias que se imparten en el centro.

De aspectos como estos surge así la política Bring Your Own Device (**BYOD**) o Bring Your Own Technology (**BYOT**) con la cual el sistema educacional puede resolver problemas como la falta de recursos para la educación, ya que la gran mayoría del alumnado, posee un terminal móvil con conexión o tableta y se puede contemplar la opción de que lo traigan al centro como **parte de su material escolar**[Ministerio de Educación, 2015].

Dicho esto, es de entender que el desarrollo de aplicaciones sea en gran medida hacia estos dispositivos, sobre todo el desarrollo de *aplicaciones de soporte a la educación* (ver 1.7).

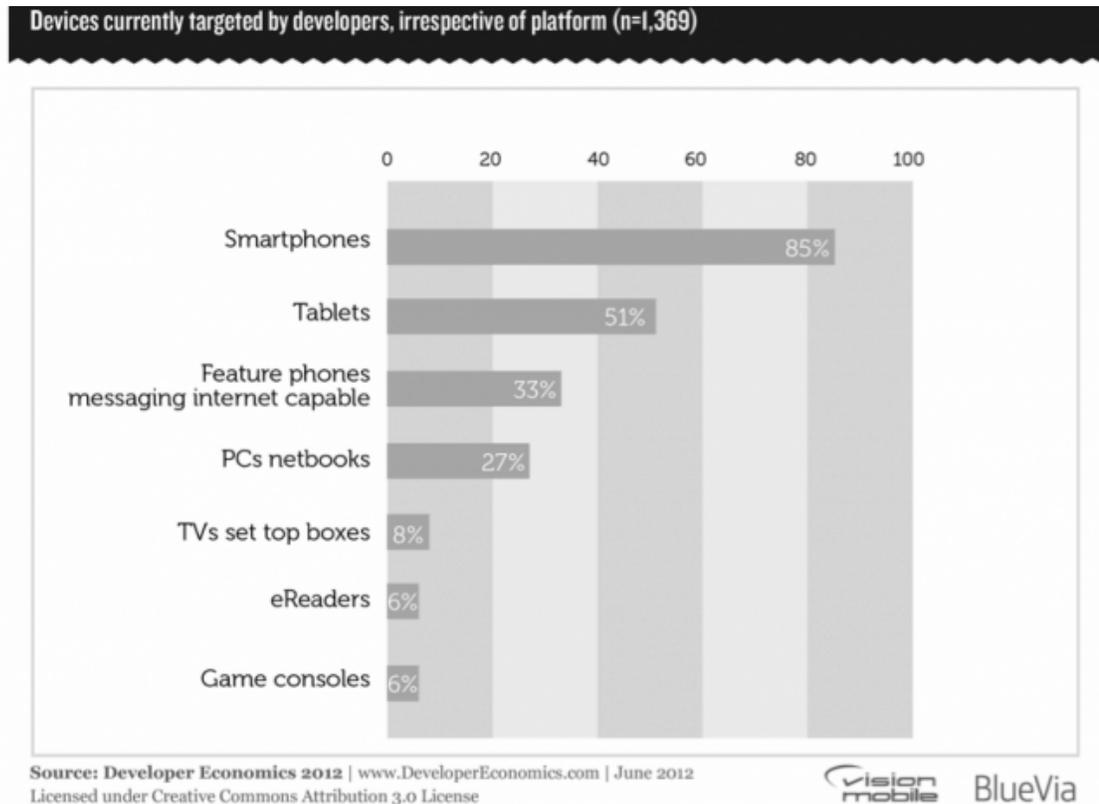


Figura 1.5: Desarrollo de aplicaciones para dispositivos. [DeveloperEconomics.com, 2012].

1.3. Herramientas o tecnologías para lograr la adaptación de contenidos

Uno de los problemas más grandes existentes en la red, es que el contenido en la web no logra ajustarse a los anchos de pantalla y en muchos casos este se distorsiona, pues es obligado a adoptar dimensiones que hacen que la tipografía usada en el sitio se vea con problema. Uno de los principales problemas que esto acarrea es que muchas veces el usuario opta por abandonar el sitio ya que le es muy engorrosa la lectura.

1.3.1. Estrategias de contenido

Luego de la aparición del libro *Estrategias de Contenido para la Web*, escrito por Kristina Halvorson en 2009, este tópico ha sido debatido bastante y su atención ha subido considerablemente.

Según [Peterson, 2014] cuando se habla de estrategias de contenido se esta hablando

además del uso de palabras; también al uso de imágenes, sonidos y vídeos, y a la forma en que se administran. Este aspecto no es sencillo de desarrollar, pero el resultado final, cuando todo el contenido de la web halla sido organizado de manera correcta, la experiencia del usuario durante su navegación será la más grata posible.

De manera general esto es lo que se conoce como *Estrategia de Contenido* pero, el uso de la tecnología móvil ha llevado este concepto un poco más allá de lo conocido, pues el hecho de que la web debe adaptarse a las resoluciones de pantalla ofrecidas por estos dispositivos, ha cambiado la forma de administrar este contenido y es lo que se conoce como "*mobile content strategy*" o estrategia de contenido para móviles, la misma no difiere tanto de lo regular en la web, pero se debe tener en cuenta el espacio visual de los **smartphones**.

Este término hace referencia a la manera en que se controla el flujo de contenido que se debe presentar en un sitio web; qué publicar y qué no publicar en la web. Para una mejor comprensión del tema, [Peterson, 2014] plantea para la elaboración de una estrategia de contenido los siguientes puntos:

- **Olvidarse de la idea de que todo el contenido va en la web.** A veces el usuario se cansa cuando lo que busca es muy difícil de encontrar aún cuando el sitio lo ofrece.
- **Como acortar el monto de información.** No es necesario redundar, la información debe ser mostrada en una sección no en múltiples.
- **Auditoría de contenido.** Siempre es bueno asignar responsables de revisión de contenidos para evitar contenidos no deseados.

De igual manera se enuncian cuatro estrategias para la adaptación de contenidos, que pueden ser aplicadas en cualquier contexto para brindarle a los usuarios una mejor experiencia. A continuación se explican las estrategias a partir de un análisis realizado por [MobiForge, 2009].

- **One Size Fits All**(Un tamaño sirve para todos): El sistema es diseñado para todos los usuarios y todos los dispositivos, se utilizan las mismas interfaces y los mismos contenidos sin tener en cuentas las características de los dispositivos desde donde se accede.
- **Minor Adaptation**(Adaptación mínima): A diferencia de la estrategia anterior esta es capaz de detectar las características de los dispositivos y realizar pequeñas

adaptaciones a los contenidos como cambio en el tamaño de las imágenes, ocultamiento de contenidos, entre otros.

- **Redirection**(Re-direccionamiento): El sistema crea sitios optimizados para móviles y redirecciona a los usuarios a estos sitios, requiere el doble de esfuerzo y es considerado una solución temporal.
- **Unified**(Unificado): Este enfoque necesita mas planificación y experiencia que otros, puede recodificar el contenido original en el contenido adaptado de acuerdo al contexto y las capacidades de los dispositivos.

En relación con la ubicación donde se realiza la adaptación, [WAI-ARIA, 2014] clasifica tres tipos de adaptación de contenido: adaptación del lado del cliente, adaptación a través de un proxy y adaptación del lado del servidor.

En el presente trabajo, se abordará la adaptación de los elementos del lado del cliente, centrándose en técnicas de diseño, **CSS** y buenas prácticas, y del lado del servidor, las técnicas de tratamiento de imágenes, ya que el **RWD** depende en gran medida de aspectos como estos. Como se puede apreciar, los puntos no son nada específicos; por tanto en el desarrollo de la solución se puede *jugar* con el tratamiento de la interfaz, siempre y cuando cumpla con los puntos anteriormente expuestos. De igual manera será necesario alterar algunos archivos de construcción en el lado del servidor para facilitar el trabajo en la visualización de componentes creados dinámicamente.

1.3.2. Imágenes fluidas

El uso de imágenes es algo muy común cuando se está desarrollando un sitio web, sin embargo es muy común también cometer errores en el uso de este tipo de recurso. Es importante en una web, pensar en el peso que las imágenes tienen, pues tienden a sobrecargar el servidor cuando se hace una petición. Cuando la petición es hecha, el navegador, comienza a cargar toda la interfaz de usuario, y si las imágenes usadas en el sitio son de gran tamaño(en el orden de los *megabytes*) es muy probable que la velocidad de navegación en el sitio se dificulte y no sea la más adecuada.

[Frain, 2012], [Marcotte, 2011], [Peterson, 2014], entre otros concuerdan que para escalar las imágenes en la web y que estas se adapten sin problemas a cualquier tamaño de pantalla, es necesario hacer uso de la propiedad *max-width* que propone **CSS3** siempre y cuando los valores que adquiera esta propiedad sean porcentuales.

Código 1.1: Uso de la propiedad *max-width* en CSS.

```
img, object, video, embed{  
  max-width: 100%;  
}
```

La versatilidad de esta propiedad le permite ser aplicada a varios tipos de *recursos media* siempre y cuando estos sean especificados a través de los selectores **CSS**. Pero aún cuando la imagen logra adaptarse a las diversas pantallas, es necesario lograr que estas bajen su peso (*megabytes*) para así resolver el problema de la sobrecarga en las peticiones al servidor. Para esto [Frain, 2012] recomienda el uso de la librería JavaScript(<http://adaptive-images.com>), pues con esta es posible además de redimensionar la imagen, también se le puede dar solución al tema de las peticiones Hypertext Transfer Protocol (**HTTP**). Otra librería Java Script que ayuda a soportar la carga de imágenes en el servidor es *jQuery Lazy*, la cual retrasa la carga de las imágenes según sean visualizadas en el viewport 1.7. El uso de **sprites CSS** también es una técnica muy reconocida para el uso de imágenes dentro del desarrollo web y al igual que las librerías JavaScript también aligeran la carga en el servidor.

Teniendo en cuenta que los contenidos presentados en la plataforma tienen la posibilidad de hacer uso imágenes y que además pueden existir muchos contenidos en dicha plataforma, se considera como mejor opción la de retrasar la carga de las imágenes a través de la librería *jQuery Lazy* ya que esta se integra de manera más fácil a cualquier aplicación web.

Finalmente se puede comprender que contenido adaptable no es solo lo que se logra con un **RWD**, pero básicamente es lo que al final se quiere lograr en todas las vías de información. Lo importante aquí es como se ven las cosas; todo (*imágenes, tipografía, vídeos y elementos web*) deben *encajar* a la perfección y mostrarse adecuadamente en el espacio disponible. Algunas vías de información como (Really Simple Syndication (**RSS**) o Instapaper) también usan contenido adaptable pero muchas veces con la ausencia de imágenes, ellas se centran más en el contenido publicado. El uso de estas alternativas está en el propósito que se les da.

1.4. Elementos para lograr un diseño web adaptativo

1.4.1. HTML5

Sin duda, para crear un sitio web es necesario hacer uso de **HTML**, este representa el esqueleto de lo que se va a construir. La evolución de este lenguaje en los últimos años esta dada principalmente por la necesidad de conseguir códigos mucho más limpios y entendibles, y que apoyen ciertamente a la creación de sitios web adaptativos.

Para mejorar al **HTML**, el nuevo **HTML5** se ha preocupado por lo que es necesario para un perfecto desarrollo de sitios web durante la próxima década y más. **HTML5** consta con los siguientes elementos[[Matthew, 2010](#)]:

- Estructura de página.
- Presentaciones visuales.
- Herramientas de gráfico.
- Enriquecimiento para recursos media.
- Mejoramiento en el Java Script.

Nuevos elementos en HTML5

[[Matthew, 2010](#)] afirma que **HTML5** incorpora nuevos elementos que permiten el control del código fuente. Estos nuevos elementos están organizados en el siguiente orden:

- Estructura de *bloques* en la página.
- Administración de recursos media.
- Estructura de formularios.

Anteriormente el uso de esta estructura de bloques en **HTML** era desarrollada a través de complejas tablas y de la *no tan popular* etiqueta DIV. Ahora **HTML5** introduce algunos nuevos elementos que permiten insertar de manera fácil contenidos en bloques en una página. Convenientemente estos nuevos elementos tienen nombres que identifican el tipo de bloque de contenido que representan en la página: HEADER, SECTION, ARTILCE, ASIDE, FOOTER, NAV.

El rol de esta nueva capa de página de elementos es el de describir de manera más fácil las partes específicas del documento. El comportamiento de estas etiquetas es muy

similar a como se escribe en un documento Word. Este posee secciones de contenido que pueden ser separadas en párrafos, barras laterales, encabezados y pies de página. Estos nuevos elementos organizan el código en secciones lógicas, o bloques y dan a la página una estructura visual que según la ubicación dada por el diseñador aportará más, o menos experiencia de navegación al usuario.

Los formularios han recibido su mayor mejora desde **HTML2**, en 1994. Ahora es posible:

- Incluir solo números telefónicos.
- Incluir campos que validan direcciones web.
- Incluir campos que validan direcciones de correo.
- Activar campos que permiten escoger fechas en un calendario.
- Forzar campos a que sean el primer campo por defecto del formulario.
- Resaltar campos(cuando se escribe o en caso de errores) si es necesario.

Con la inclusión de estos formularios 2.0 ya no se necesita **Ajax** o **Flash** o cualquier otra tecnología para dar animaciones a estas funcionalidades.

El atributo data-*

En aras de mejorar la integración del lenguaje **HTML**, **HTML5** cuenta con nuevos atributos para sus etiquetas(*elementos*) de trabajo, este es el caso del atributo *data-** usado principalmente para almacenar cierta clase de datos que serán usados según estime el desarrollador. Estos valores pueden ser usados desde las páginas *Java Scripts* para mejorar la experiencia de usuario a través de la interactividad con la aplicación sin necesidad de funciones *Ajax* o llamadas del lado del servidor[[W3C, 2014a](#)].

El atributo consta de dos partes fundamentales[[W3C, 2014a](#)]:

- Su nombre no puede contener letras mayúsculas, y debe existir al menos una palabra después del prefijo "data-".
- El valor del atributo puede ser cualquier cadena de texto(*string*).

HTML semántico

Para comprender mejor este término es mejor abstraerse del hecho de que el **HTML** es un lenguaje de programación. Este lenguaje en su última versión posee etiquetas las cuales sus nombres hacen alusión a la funcionalidad que cumplen, por tanto semánticamente poseen un significado. El uso de estas etiquetas facilita el trabajo para los diseñadores ya que estas poseen comportamientos predeterminados en su visualización a través de dispositivos móviles (se auto-redimensionan) Figura 1.6 no obstante muchas etiquetas conocidas de la versión anterior ya incluyen esta habilidad.

Código 1.2: Ejemplo del uso de la etiqueta div

```
<!DOCTYPE HTML>
<html lang="en-GB">
<head></head>
<body>
<div style="background: grey;">
  <h1>Esto solo es una etiqueta div, sin modificaciones en los
    valores de ancho</h1>
</div>
</body>
</html>
```

Para mejorar la semántica del lenguaje existen otras vías; la W3C's Accessible Rich Internet Applications (**WAI-ARIA**) propone además de el uso correcto de las etiquetas de **HTML5**, el uso de el atributo *role* disponible para cualquier etiqueta **HTML**.

WAI-ARIA describe una serie de puntos para regiones de navegación y estructuras web como contenido principal, menús, información del banner del sitio, etc. De esta forma los programadores pueden facilitar no solo el trabajo de otros programadores, sino que también pueden añadir características peculiares a determinadas regiones de la aplicación con el objetivo de facilitar la navegación del usuario.

Dentro de las especificaciones que propone este grupo de desarrollo se puede encontrar:

- Roles que ayudan a describir el tipo de componente que se presenta (menús, drop-down, treeitem, sliders, etc).
- Roles que describen la estructura de los sitios web, tales como headings, regiones y tablas (grids).
- Propiedades que describen los estados de los componentes, tales como checked o haspopup.

- Propiedades para los eventos *drag* o *drop* que describen los orígenes y los destinos de los ficheros.

Por tanto en este caso, el uso de la terminología *código semántico* es solo para evidenciar que el código usado en la programación del sitio web en cuestión es **HTML5** y que además cada elemento de este posee un significado con respecto a la función que ejecuta.

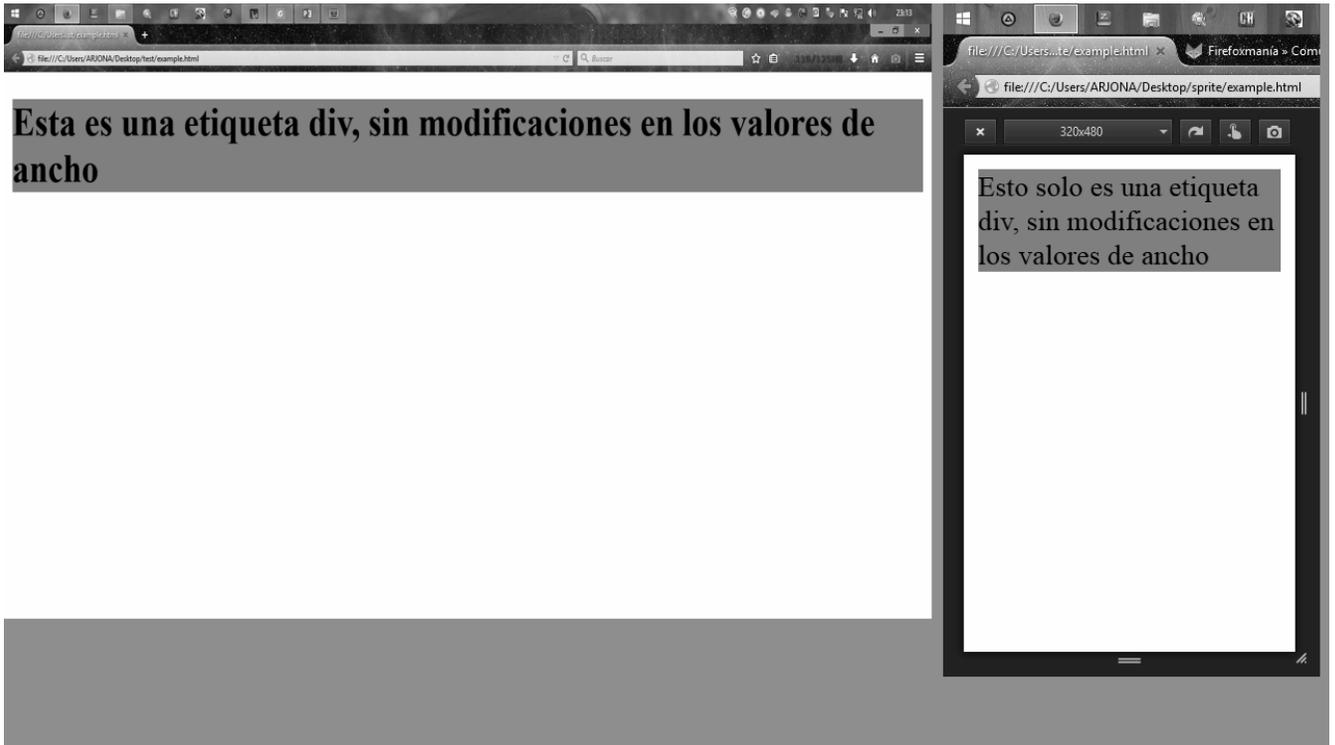


Figura 1.6: Visualización de la etiqueta *div* en diferentes entornos; de izquierda a derecha: PC y móvil.

1.4.2. Viewport

Diversos autores definen al **Viewport** de muchas maneras [Marcotte, 2011] lo define como el lienzo de un pintor; pero en un final todos concuerdan en que el **Viewport** es el área del navegador en donde se visualiza la página web. El tamaño de esta área difiere según el tamaño del dispositivo, sabiendo esto es normal saber que el **Viewport** en el navegador de un dispositivo móvil (**smartphones**) será más grande que el de una máquina de escritorio. Este era el principal problema que afrontaba el diseño web hace algunas décadas; la carencia de espacio en los navegadores web de los dispositivos móviles.

Nota: El tamaño del *viewport* no tiene nada que ver con la resolución que puede admitir el dispositivo, sino con el tamaño de pantalla físico con el que cuenta el mismo.



Figura 1.7: El cuadrante rojo delimita el área asociada al viewport.

1.5. Cascading Style Sheet para un diseño adaptativo

Que es Cascading Style Sheet

Una parte de lo que es la creación de sitios web y no menos importante es el **CSS**. Este lenguaje es el encargado de describir como el **HTML** va a ser presentado o estilizado. **CSS3** es la última actualización de este lenguaje y solo es un término usado para hacer referencia a las nuevas incorporaciones que este presenta. **CSS3** tiene incluido **CSS2.1**, pero también adicionó una serie de características para ayudar a los desarrolladores con problemas como el maquetado no semántico o imágenes extras. Entre las nuevas inclusiones en el **CSS3** se puede observar el uso de selectores nuevos, animaciones, transparencias, etc. [Goldstein et al., 2011].

1.5.1. Posicionamiento del código

Cuando se desarrolla un sitio web es posible que a pesar de que se logre el objetivo de manera visual; internamente el código utilizado no esté correctamente organizado, lo cual al principio no importa, pero puede tener graves consecuencias.

Según [Alvarez, 2014] lo óptimo sería colocar el *vínculo* al archivo **CSS** al principio pues aunque de esta forma se hace otra petición al servidor, de este modo tarda menos en cargar la página y la petición al servidor apenas se nota porque tarda mucho menos que en cargar **CSS** mezclado con el código **HTML**.

Es decir; el código **CSS** en el documento **HTML**, es recomendable escribirlo al principio del documento para que, al cargar la web, lo primero que se cargue sean el **HTML** y el **CSS** y el usuario tenga la página con estilos lo antes posible y no pueda ver la web sin formato ni estilos.

Código 1.3: Ejemplo de como optimizar el código CSS.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<body>
  ....
</body>
</html>
```

1.5.2. Frameworks

Frameworks CSS

Se toma como opción, el framework **Bootstrap**, teniendo en cuenta que es el framework con que fue desarrollado desde sus inicios la plataforma educativa ZERA [Herrera Santos, 2013], no obstante el estudio de otros frameworks **CSS** no queda descartado, pero solo se hará revisión de las nuevas propuestas de diseño e interactividad que estos traen.

Frameworks JS

De igual manera se continuará trabajando con el framework Java Script (**JS**) **jQuery** en su versión 1.9, el mismo que permite simplificar la interacción con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Dentro de sus ventajas se encuentra el soporte a extensiones o plugins los cuales amplían las facilidades de uso y mejoran los tiempos de producción al poder reutilizar componentes desarrollados por terceras personas[Herrera Santos, 2013].

Librerías de terceros

jQuery Lazy

Lazy es una librería ligera y rápida para el tratamiento de imágenes soportado mediante el plugin **jQuery**. Esta diseñado específicamente para agilizar la carga de imágenes en la web y así decrementar el trafico de datos. Con **Lazy** es posible demorar la carga de imágenes al desplazar el *scroll* hacia abajo o hacia arriba y de igual manera poner una imagen por *default* mientras la fuente es cargada. Es compatible con la mayoría de los navegadores usados.

Picturefill

Picturefill es una librería igualmente usada para tratar las imágenes en la web, la misma detecta las resoluciones de pantalla automáticamente e intercambia las imágenes en el **HTML**. La librería hace uso del atributo *srcset* y en ella añade varias de direcciones para dependiendo de la resolución captada se cargará una imagen o *thumbnail*.

HISRC

HISRC al igual que las demás librerías mostradas esta no difiere con la anteriormente mencionada su estrategia es la misma, hacer cambios de una imagen por otra dependiendo de la resolución, solo que en este caso lo hace usando el atributo *data-**(ver 1.4.1).

Adaptative images

Adaptative images es una librería que permite escalar las imágenes presentadas en la web dependiendo del tamaño del viewport y al mismo tiempo reducir el peso de las mismas.

Finalmente se mantiene el uso del framework **CSS Bootstrap** y se incluye la librería *jQuery Lazy* y **Adaptive images** por ser las que mejor se ajustan a las condiciones que tiene la plataforma, pues no hacen uso de imágenes extras y solo trabaja con la imagen originalmente guardada por el usuario.

1.6. Media queries en el diseño web adaptativo

1.6.1. Media queries

Desde **CSS2.1**, las hojas de estilo han disfrutado de un cierto grado de sensibilización a través de diferentes tipos de dispositivos. Cualquiera que haya escrito una hoja de estilos de impresión, ya está familiarizado con el concepto:

Código 1.4: Ejemplo de uso de código CSS

```
<link rel="stylesheet" type="text/css" href="core.css" media="screen" />
<link rel="stylesheet" type="text/css" href="print.css" media="print" />
```

La inclusión de *media queries* como parte del **CSS3** es gracias a la **W3C**. Los *media queries* nos permiten inspeccionar las características físicas del dispositivo que en ese momento está *renderizando* el sitio web. Por ejemplo para que funcionen los *media queries* se debe incorporar una *consulta* dentro de la etiqueta *link*, el atributo *media*:

Código 1.5: Ejemplo de media query dentro de la etiqueta link

```
<link rel="stylesheet" type="text/css"
media="screen and (max-device-width: 480px)"
href="shetland.css" />
```

Como se puede observar la *consulta* solo se compone de 2 partes:

- Un *media type*(screen), y
- la verdadera consulta encerrada entre paréntesis, la cual pregunta por una característica del dispositivo(max-device-width), seguido del valor(480px).

En otras palabras, se está preguntando si la resolución horizontal que posee el dispositivo en cuestión es de 480px o menos(max-device-width: 480px). Si esa consulta es valor verdadero entonces el dispositivo debe cargar el archivo *shetland.css*, de suceder lo

contrario, este enlace es ignorado. Aun así esta no es obligado incluir estas *consultas* en la etiqueta "link", también es posible hacerlo desde el propio "*fichero.css*" como una regla `@media`[[Marcotte, 2010](#)]:

Código 1.6: Ejemplo de media querie dentro de un archivo CSS a través de la regla `@media`

```
@media screen and (max-device-width: 480px) {  
  .column {  
    float: none;  
  }  
}
```

O como parte de una regla `@import`:

Código 1.7: Ejemplo de media querie dentro de un archivo CSS a través de la regla `@import`

```
@import url("shetland.css") screen and (max-device-width: 480px);
```

1.6.2. Puntos de quiebre o break points

Como se explica anteriormente, los *media queries* son los encargados de inspeccionar el dispositivo que realiza la visualización del sitio web.

La consulta que el *media querie* realiza desconoce por completo el *ancho de pantalla* que el dispositivo admite, para mejorar este problema cada dispositivo fue creado con un *ancho de pantalla* específico y conocido por muchas personas ya sean portadores o no del objeto. Esta gama de *anchos de pantalla* son conocidos como **breakpoints**, que no son mas que los puntos por los cuales el *media querie* va a cambiar el diseño del sitio. Este diseño puede tener una o mas variantes, dependiendo de como este situado(orientación) el dispositivo[[Peterson, 2014](#)].

Tabla 1.1: Puntos de quiebres enunciados en [Marcotte, 2011] para lograr **RWD**.

Pixeles	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, como <u>Amazon Kindle</u> (600x800) y <u>Barnes & Noble Nook</u> (600x1024), en modo <i>portrait</i> .
768	Tables de 10 pulgadas como los <u>iPad</u> (768x1024), en modo <i>portrait</i> .
1024	Tables como el iPad (1024x768), en modo <i>landscape</i> , también algunas <u>laptop</u> , <u>netbook</u> y monitores.
1200	Para pantallas <i>widescreen</i> .

1.7. Diseño web adaptativo en plataformas educativas

Plataforma educativa (LMS)

Según [Boneu, 2007], un Learning Management System (**LMS**) parte de su antecesor el Content Manager System (**CMS**) y esta orientado al aprendizaje y la educación, proporcionando herramientas para la gestión de contenidos académicos, permitiendo mejorar las competencias de los usuarios de los cursos y su intercomunicación, en un entorno donde es posible adaptar la formación a los requisitos de la empresa y al propio desarrollo profesional.

Este tipo de herramienta también posee la característica de disponer de herramientas que permiten la distribución de cursos, recursos y contenidos relacionados con cierto tema.

1.7.1. Tipos de LMS

Moodle

Moodle es un **LMS** libre orientado para que los educadores puedan crear sus propios cursos de manera dinámica haciendo extensible el conocimiento hacia cualquier parte en cualquier momento. Esta plataforma cumple con los estándares de **WAI-ARIA** por lo que es apta para personas discapacitadas, presenta soporte para *screen-readers* además de poseer una interfaz adaptable a partir de la versión 2.7.[Moodle, 2014].

Blackboard

Blackboard se ha comprometido a garantizar que la plataforma sea usable y accesible. El código y la interfaz de usuario son auditadas continuamente para asegurarse de que la aplicación se puede utilizar por todo el mundo, en la mayor medida posible, sin importar la edad, habilidad o situación. Los cambios técnicos a la navegación principal en **Blackboard Learn** hacen la navegación más fácil para los usuarios con discapacidad visual. Los marcos se han eliminado por completo de la aplicación y reemplazado con **HTML5 DIVs** compatibles e *iFrames*. Este cambio mejora la impresión de páginas, la accesibilidad, y la página de navegación [Blackboard, 2014].

Este cambio se traduce en una mejora de la accesibilidad para los usuarios de lectores de pantalla, donde son necesarias la identificación de títulos ubicación en esta página.

Sakai

Sakai es un **LMS** desarrollado para cumplir o exceder todos los principios de diseño de la accesibilidad en las normas internacionales reconocidas. Esta aplicación es un sistema con todas las funcionalidades que ayudan a mejorar la enseñanza, el aprendizaje, la investigación y la colaboración para la educación. **Sakai** hace de la educación, una empresa colaborativa. **Sakai** puede soportar prácticamente cualquier proyecto o actividad que requiera la comunicación, la colaboración y el intercambio de conocimientos para lograr los objetivos del proyecto con eficacia. Su objetivo también es cumplir con las Pautas de Accesibilidad establecidas por la **W3C** a nivel A y AA. También el uso de los estándares emergentes y las mejores técnicas de diseño de la práctica (por ejemplo, la suite de **WAI-ARIA**) que apoyan las tecnologías de adaptación existentes y emergentes [Sakai, 2014].

Chamilo

Chamilo es de Código Abierto, bajo la licencia GNU/GPLv3, y está protegido por una asociación sin fines de lucro, lo que significa que es literalmente imposible para cualquier empresa tomar la propiedad del proyecto y cerrarlo. Pero también significa que eres libre de desarrollar tus propias extensiones o mejoras, y compartirlo (o no) con la comunidad [Chamilo, 2014]. Pero a pesar de estas ventajas este software no consta con una integración de **HTML5** y tampoco hace uso de *media queries* para su visualización en dispositivos móviles, aunque existe entre sus metas el desarrollo de una versión para móvil.

Dokeos

Dokeos es una plataforma libre de código abierto que propicia un intercambio dinámico entre estudiantes y profesores a través de actividades. Esta plataforma presenta un diseño adaptativo al igual que las mencionadas anteriormente por lo que se puede decir que, el tema de accesibilidad es una barrera que en estos momentos ya esta superada.

ZERA

Es un **LMS** desarrollado en la **UCI**, el cual tiene como características principales [Manso, 2014]:

- Aplicación basada en hiper-entornos de aprendizajes.
- Permite la creación de cursos con una estructura capitular donde el contenido se muestra como un libro.
- Permite la creación de 30 tipos de recursos educativos.
- Permite la creación de 11 tipologías de ejercicios.
- Soporte para las especificaciones IMS Question & Test Interoperability Specification (**IMS-QTI**) y Sharable Content Object Reference Model (**SCORM**)
- Sugerencias de uso y registro de avance.
- Evaluación por rúbricas y por competencias.
- Ideal para el trabajo semi-presencial *b-learning* y a distancia *e-learning*.
- Atención diferenciada: recorridos dirigidos, softareas, orientaciones de trabajos.
- Sistema distribuido, ideal para instituciones con problemas de conectividad.
- Gestión académica.
- Fórum, chat, herramientas de la web 2.0, sistema con varios roles, etc.

Aunque se reconoce que el mayor impacto de los dispositivos móviles es en la formación informal. La tendencia de que a los estudiantes les gusta llevar dispositivos móviles a las escuelas ha sido demostrada y apoyada por la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (**UNESCO**) siempre y cuando los estudiantes las

usen con vías educativas[UNESCO, 2012]. De la misma manera [Horizon, 2015] afirma que en cortos plazos de tiempo la educación, principalmente la educación superior hará uso de tecnologías para el enriquecimiento de las clases a impartir así como de los servicios que se prestarán. Se está hablando de una educación totalmente digital y basada en el análisis de los progresos de cada estudiante por separado. Con esto se asegura un mejor aprovechamiento de las clases, pues estas son destinadas a realizar actividades colaborativas, donde ya el profesor no transmite contenidos en el aula, sino que se apoya en videolecciones, podcasts, textos y foros en línea a los que el estudiante accede antes y después de las clases, es lo que se conoce como *clase invertida* o (**flipped classroom**). Por tanto la inclusión de esta técnica en la plataforma educativa **ZERA** será de gran importancia ya que permitirá apoyar la enseñanza con mayor fuerza y asegurar un paso mas hacia esta tendencia.

1.8. Conclusiones Parciales

Después de la investigación realizada a través de los diferentes temas abordados en este capítulo se arriban a las siguientes conclusiones:

- Diseñar una estrategia de contenido que logre adaptarse a los anchos de pantalla sin romper con la estética que actualmente posee la plataforma y que mejore grandemente la experiencia de usuario.
- Con el objetivo de apoyar la inclusión del paradigma adaptativo y crear una estructura semántica en el código **HTML** que posee la aplicación, se debe cambiar el código existente por elementos **HTML5** según sea necesario.
- Crear un **sprite** para agrupar todas las imágenes usadas en la navegación de la plataforma.
- Incluir el uso de la librería **jQuery Lazy** para retrasar la carga de las imágenes en la interfaz de la plataforma.
- Incluir el uso de la librería **Adaptive images** para reducir el peso de las imágenes de la plataforma.
- Definir los **breakpoints** necesarios para que la plataforma pueda visualizarse en diferentes anchos de pantalla.

- Incluir el paradigma adaptativo a la plataforma educativa **ZERA**, para que la misma logre visualizarse en todos los dispositivos móviles existentes, así como PC de escritorio, para de esta manera poder apoyar el proceso de enseñanza y aprendizaje y esté a la altura de las demás plataformas educativas existentes.

Capítulo 2

Propuesta Solución

Introducción

En el presente capítulo serán detalladas las actividades propuestas para satisfacer las necesidades que en estos momentos presenta la plataforma educativa **ZERA**. Se describen los cambios significativos en la visualización de los componentes de la misma así como en algunos archivos presentes en el modelo de negocio, con vistas a mejorar la experiencia del usuario. Para el desarrollo de la plataforma se continúa con las herramientas utilizadas como parte de la arquitectura del sistema, las cuales serán usadas a través de las siguientes propuestas.

El nivel de prioridad estará dado por los números del 1 al 5, dotando de mayor prioridad al número 1 y de menor prioridad al número 5. De esta manera se explica el orden de cada tarea a cumplir. La prioridad de estas actividades fue otorgada teniendo en cuenta el orden lógico que se usa para realizar un **RWD** común, como por ejemplo es necesario definir los **breakpoints** antes de realizar las *estrategias de contenido*, ya que estas dependen totalmente de los anchos de pantallas establecidos por los **breakpoints**.

El uso de técnicas para lograr un **RWD** en un sitio web se basa principalmente en solo mostrar el contenido necesario, por tanto, la navegación queda descartada y aunque es un elemento fundamental, esta será mostrada de manera sutil al usuario. De esta manera el usuario, de manera intuitiva, se dirigirá a ciertas áreas del navegador con el objetivo de navegar a través de la aplicación. Con esto se evita sobrecargar de contenido la plataforma y si aumentar la experiencia de usuario.

Tabla 2.1: Modelación de la propuesta de solución para lograr que la plataforma educativa *ZERA* incorpore las buenas prácticas existentes para el diseño adaptativo a desarrollar.

Tarea	Prioridad
Definir los <i>breakpoint</i> para las diferentes vistas.	1
Crear un <i>sprite</i> para las imágenes usadas en la navegación de la plataforma.	3
Cambiar el código HTML existente por etiquetas HTML5 haciendo uso del atributo <i>role</i> .	4
Diseñar estrategias de contenido para las vistas definidas por los <i>breakpoints</i>	2
Integrar la librería Adaptive Image a la plataforma para reducir el peso en las imágenes presentadas en los contenidos.	3
Integrar la librería jQuery Lazy a la plataforma para optimizar la carga de las imágenes presentadas en los contenidos.	3

2.1. Estándares de codificación

Para el desarrollo de las tareas es necesario establecer un estándar de codificación para asegurar un código plenamente entendible. Cuando la aplicación incorpore código fuente previo, o cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación permitirá operar con la base de código existente.

Al ser **ZERA** una aplicación previamente elaborada, se continúa con el estándar definido previamente. Con esto se evitarán complicaciones si en un futuro la aplicación vuelve a sufrir cambios en su código[Herrera Santos, 2013].

2.2. Puntos de quiebre

Teniendo en cuenta la tendencia que existe en la actualidad de navegar a través de los **smartphones** y **tablets** se han propuesto los siguientes puntos de quiebre, sobre los cuales se ha de elaborar una estrategia de contenido que permita adaptarse a dichos puntos. Estos han sido propuestos basándose en la visualización que en estos momentos posee la plataforma y del ancho disponible que poseen los dispositivos anteriormente

mencionados. Como se observa los puntos de quiebre van desde 320 pixeles de ancho a 768, pues los elementos de la plataforma logran adaptarse a anchos de pantalla superiores a 768 pixeles.

Tabla 2.2: Puntos de quiebres para el desarrollo en la plataforma ZERA.

Pixeles	Descripción
320	Para dispositivos de pantallas pequeñas, como los celulares en modo <i>portrait</i>
360	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, como Amazon Kindle (600x800) y Barnes & Noble Nook (600x1024), en modo <i>portrait</i> .
640	Para dispositivos de pantallas pequeñas, como los celulares en modo <i>landscape</i> .
768	Tables de 10 pulgadas como los iPad (768x1024), en modo <i>portrait</i> .

2.3. Imágenes y *sprite*

Ésta técnica se basa en agrupar varias imágenes en un sólo archivo para después posicionarlas con **CSS** según sea necesario. Dicha técnica no interfiere con el uso de otras clases en el posicionamiento de los elementos web, ya que solo altera las dimensiones de este para poder mostrar en ese espacio la imagen deseada.

Código 2.1: Ejemplo de sprite CSS

```
.sprite { background-image: url('sprite.png'); background-color:
    transparent; background-repeat: no-repeat; }

.animation { width: 100px; height: 100px; background-position:
    -196px -497px; }
```

Finalmente el **HTML** se vería como el siguiente código:

Código 2.2: Ejemplo de uso de sprite CSS

```
<div class = "sprite animation"></div>
```

Como resultado, se ha reducido considerablemente el número de solicitudes **HTTP** y evitado cualquier retraso potencial de otros recursos en la página, ya que no es necesario hacer uso de direcciones hacia la imagen, por cada una de las imágenes que se presentan. Solo se visualizará una imagen, aquella que contendrá a todas las demás que se desean presentar, y con el **CSS** se muestra solo la que se desea presentar a través de un sistema de coordenadas que será entendido por la propiedad **CSS** *background-position*. Cuando se hizo uso del *sprite*, se evitó dejar espacios en blanco entre imágenes. Esto no afectará al tamaño del archivo, pero tendrá efecto en el consumo de memoria.

El uso de esta técnica es de gran ayuda a la aplicación ya que la misma tiende a presentar un gran cúmulo de imágenes. La técnica en sí se debe aplicar a las imágenes que se usan en la navegación del sitio, pues estas no varían y por ende pueden ser incluidas en el *sprite*. El resto de las imágenes serán tratadas a través de técnicas diferentes.

2.4. Modificaciones en el código

2.4.1. Lado del cliente

El uso de frameworks **CSS**, aunque puede resultar muy cómodo, a veces representa una mala práctica en el diseño web, pues este tipo de herramientas puede obviar el uso de las nuevas etiquetas **HTML** existentes en la última versión de este lenguaje.

Una de las pautas que se persigue en el diseño web actual es el uso de estos componentes. Los frameworks **CSS** trabajan sobre clases (estilos) que son adheridos a los componentes usados en el sitio web. Por lo general este tipo de herramientas propone el uso de etiquetas `div`, que de igual manera pueden ser sustituidas por las siguientes etiquetas: *section*, *aside*, *header*, *nav*, *footer*, *image*, *video*, etc. Ya que la mayoría de estas, funcionan como elementos contenedores, además de que todos poseen el atributo **class**, atributo mediante el cual trabajan todos los frameworks CSS.

De esta manera se está proponiendo cambiar la estructura que tiene en estos momentos la plataforma **ZERA** en cuanto a su **HTML**. De esta manera dicha plataforma será más compatible con los dispositivos móviles actuales. Este cambio es casi transparente al usuario, pues los estilos visuales no se alterarán, solo se cambiarán las etiquetas pertinentes que se consideren claves en el cuerpo de la plataforma. Junto a esta propuesta también está la inclusión de librerías *Java Script* para mejorar la interacción con el usuario.

Código 2.3: Plantilla layout.php existente

```
<html lang="en">
<head></head>
<body onload="setTimeServer(<?php echo '\'' . __(date('l')), null,
    'global') . ', ' . date('j ') . __(date('F')), null, 'global') .
    ', ' . date('Y') . '\', ' . '\'' . date('h') . '\', ' . '\'' .
    date('i') . '\', ' . '\'' . date('s') . '\', ' . '\'' . date('a')
    . '\'; ?>);" class="<?php echo $body_class ?>" data-time="<?
    php echo time() ?>">
<div class="container-fluid">
<div>
<!-- Banner -->
<div class="row-fluid">
<?php
$userId = sfUserTools::getConnectedUser()->getId();
$nowTime = date('Y-m-d H:i:s');
?>
<?php
include_partial('eView/layoutTopBar', array(
    'events' => array(
    'count' => zfCalendar::getCountEventsInTime($userId, $nowTime),
    'elements' => zfCalendar::getEventsInTime($userId, $nowTime, 5)
    ),
    'messages' => array(
    'count' => zfMessage::getCountMessageNew($userId),
    'elements' => zfMessage::getMessageNew($userId),
    ),
    ),
```

```

'notifications' => array(
'count' => zfNotification::getCountNotificationNew($userId),
'elements' => zfNotification::getNotificationNew($userId),
),
))
?>

</div>
<!-- Fin del banner -->

<!-- Menu primario -->
<div class="row-fluid">
<?php
/**
 * TODO cambiar sf_cache_key en dependencia del roll
 */
include_partial('eView/layoutNavbar', array(
'sf_cache_key' => 'layoutNavbar_' . zfeView::getModuleNav(
    $sf_request) . '_' . $sf_user->getCulture(),
));
?>
</div>
</div>
<!-- Fin del Menu primario -->

<!-- Contenido -->
<div id="main" class="row-fluid">
<?php
include_partial('eView/layoutContent', array(
'layout_left' => $layout_left,
'layout_right' => $layout_right,
'layout_breadcrumb_course' => $layout_breadcrumb_course,
'layout_breadcrumb_content' => $layout_breadcrumb_content,
'sf_content' => $sf_content)
)
?>
</div>
<!-- Fin del Contenido -->

<!-- Pie de pagina -->
<?php if (!$bach_principal): ?>
<div class="row-fluid">
<div class="footer-container span25">

```

```

<div class="user_name span15">
<?php if (sfAOCache::hasSession('bachelor_user_full_name')): ?>
<?php $full_name = sfAOCache::getSession('bachelor_user_full_name
    '); ?>
<?php else: ?>
<?php $full_name = sfUserTools::getConnectedUser()->getFullName();
    ?>
<?php sfAOCache::setSession('bachelor_user_full_name', $full_name)
    ?>
<?php endif; ?>
<p><?php echo __('Welcome', null, 'global') ?>...&nbsp;&nbsp; <span
    class="big_text_msg"><?php echo $full_name ?></span></p>
</div>

<div class="bach_data span9 right">
<span id="bach_date" class="bach_date"></span>
<span id="bach_time"></span>
</div>
</div>



</div>
<?php endif ?>
<!-- Fin del Pie de pagina -->
</div>
</body>

<?php jq_add_my_ui() ?>
<?php jq_add_plugins_by_name(array("zalerts", "fancybox", "
    ajaxfileupload", "ezmark", "tree", "parsley")) ?>

<?php
/**
 * TODO cambiar sf_cache_key en dependencia del i18n
 */
include_partial('jsMessages/bachelor_js_messages', array(
'sf_cache_key' => 'bachelorI18N_' . $sf_user->getCulture()
));
include_partial('jsMessages/global_js_messages');
?>

<?php //use_javascript('bootstrap/bootstrap-collapse') ?>

```

```

</script>
*/ ?>
</html>

```

Luego de realizar los cambios pertinentes, los elementos que representan las secciones principales de la plataforma adquirieron el atributo *role* y fueron sustituidos por sus similares en **HTML5**, quedando de la siguiente manera.

Código 2.4: Plantilla layout.php después de los cambios

```

<html lang="en">
<head></head>
<body onload="setTimeServer(<?php echo '\'' . __(date('l')), null,
    'global') . ', ' . date('j ') . __(date('F'), null, 'global') .
    ', ' . date('Y') . '\', ' . '\'' . date('h') . '\', ' . '\'' .
    date('i') . '\', ' . '\'' . date('s') . '\', ' . '\'' . date('a')
    . '\'; ?>);" class="<?php echo $body_class ?>" data-time="<?
    php echo time() ?>">
<div class="container-fluid">
<header role="banner">
<!-- Banner -->
<div class="row-fluid">
</div>
<!-- Fin del banner -->

<!-- Menu primario -->
<nav role="navigation" class="row-fluid">
</nav>
</header>
<!-- Fin del Menu primario -->

<!-- Contenido -->
<section id="main" role="main" class="row-fluid">
</section>
<!-- Fin del Contenido -->

<!-- Pie de pagina -->
<?php if (!$bach_principal): ?>
<footer role="contentinfo" class="row-fluid">
</footer>
<?php endif ?>
<!-- Fin del Pie de pagina -->

```

Código 2.5: Cambios realizados en la plantilla layout.php.

```

<!-- Develop by Arjona -->
<!-- Menu de Navegacion para moviles chicos -->
<a class="middle-button hide" data-target="menu">men&uacute;</a>
<?php if ($layout_left): ?>
  <a id="btn_left" class="btn_left sprite sprite-content-menu"
    data-target="menu-contenidos" href="#"></a>
  <?php endif; ?>
<nav id="menu" role="navigation" class="mm-menu mm-theme-black
  mm-hasheader">
<div class="title-modal nav-header"><h5>Plataforma Educativa ZERA<
  /h5>
<a id="menu-close-btn" class="sprite red-close-btn" data-target="
  menu"></a>
</div>
<?php
include_partial('eView/layoutNavBarBottom', array(
'sf_cache_key' => 'layoutNavBarBottom_' . zfeView::getModuleNav(
  $sf_request) . '_' . $sf_user->getCulture(),
));
?>
</nav>
<!-- Fin del menu de navegacion para moviles chicos -->
</div>
</body>

<?php use_javascript('course/principal/jquery.lazy/jquery.lazy.js
  ') ?>
<?php use_javascript('course/principal/jquery.lazy/
  jquery.lazy.min.js') ?>
<?php // use_javascript('/sfA0TemplatesPlugin/js/menu/jquery.js')
  ?>
<?php // use_javascript('/sfA0TemplatesPlugin/js/menu/
  second_jquery.js') ?>

</html>

```

Igualmente muchos *estilos CSS* fueron alterados a la hora del desarrollo de la estrategia de contenido para móviles pequeños (ver 2.5). El uso de la propiedad *display* que posee **CSS3** permitio ocultar de manera permanente aquellos componentes que no podían ser visualizados de manera correcta. La idea no fue prescindir de ellos sino que pudieran ser visualizados de manera correcta y cuando el usuario estimara necesario; para ello se hizo

uso de funciones **jQuery** (ver 2.6).

Código 2.6: Desplazamiento del menú lateral izquierdo para mejorar la vista en móviles

```
@media (max-width: 320px) {

    aside[class="span5 layout-left sidebar-offcanvas
        sidebar-offcanvas-left"]{
        left: -110%;
        z-index: 10006;
        top: 0px;
        width: 45% !important;
        height: 480px;
        position: fixed !important;
        overflow: scroll;
    }

}
```

Código 2.7: Animación del menú lateral izquierdo para mostrarlo u ocultarlo en móviles

```
/*
 * Funcion que anima la aparicion del menu izquierdo
 */
$(document).ready(function (){
    $('#btn_left').click(function (){
        var objetivo = $(this).data('target');
        $('#'+ objetivo).animate({left: "-1%"});
        $('.middle-button').toggle();
        $('.layer').toggle();
        $(this).toggle();
    });
});
/*
 * Funcion que anima la retirada del menu izquierdo
 */
$(document).ready(function (){
    $('#close_btn_menu_left').click(function (){
        var objetivo = $(this).data('target');
        $('#'+ objetivo).animate({left: "-110%"});
        $('.middle-button').toggle();
        $('.layer').toggle();
        $('#btn_left').toggle();
    });
});
```

```

});

/*
 * Funcion que anima la retirada del menu izquierdo
 */
$(document).ready(function (){
$('.layer').click(function (){
$('#menu-contenidos').animate({left: "-110%"});
$('.middle-button').toggle();
$('.layer').toggle();
$('#btn_left').toggle();
});
});

```

2.4.2. Lado del servidor

Aún cuando todo el trabajo se basa en crear interfaces agradables, también hay que tener en cuenta las funcionalidades que la plataforma posee y que alteran dramáticamente a la vista. Este es el caso del menú de navegación que posee la aplicación para hacer uso de los módulos que propone la plataforma. Dicha funcionalidad será tratada por separado para no alterar en ningún momento los flujos de información que atraviesan estas rutas. Por esto, se crearán copias de los archivos que intervienen en este flujo para luego ser tratados y modificados según sea necesario. Durante este proceso se verán afectado los archivos pertenecientes a los módulos que la plataforma tiene y así como el archivo que los visualiza (*layoutNavBar.php*).

En las resoluciones más pequeñas la visualización del archivo **layoutNavBar.php** será alterada y en su lugar se visualizará el archivo **layoutNavBarBottom.php** archivo que contendrá el menú que será capaz de adaptarse a los *breakpoints* definidos. Este archivo posibilitará un mejor tratamiento por separado de los módulos que el administrador de la plataforma activará o desactivará, facilitando el diseño de estrategias de contenido que se propondrá en la plataforma. El archivo **layoutNavBarBottom.php** es el encargado de visualizar los elementos construidos por el archivo **layoutNavbarItemResponsiveA-part.php** y que serán usados más tarde en el menú de navegación. Por tanto, para evitar los efectos que propone la plataforma en su menú, se definieron otra serie de *estilos* para colocarlos en el nuevo menú.

Código 2.8: Construcción del menú según **layoutNavbarItem**

```
<?php if (isset($items)) : ?>
```

```

<li class="dropdown <?php if($principal['active']) : ?>active<?php
    endif ?>">
<a href="<?php echo isset($principal['action']) ? url_for(
    $principal['action']) : '#'; ?>" class="dropdown-toggle"
    data-hover="dropdown" data-delay="100"><?php echo __($principal
    ['name'], null, 'global') ?> <b class="caret"></b></a>

<ul class="dropdown-menu">
<?php $total_items = count($items); $limit = 3; $open_group = 0;
    $open_single = false; $single_item = false; ?>
<?php foreach ($items as $key => $item) : ?>
<?php if(isset($item['name'])): ?>
<?php if(isset($item['sub_items'])) : ?>
<?php $sub_items = $item['sub_items'] ?>

<?php if ($single_item) : ?>
<?php $single_item = false ?>
<li class="divider"></li>
<?php endif ?>

<?php if ($open_single) : ?>
<?php $open_single = false; ?>
</ul>
<?php endif ?>

<?php if ($open_group == 0 || $open_group == $limit) :?>
<?php if ($open_group == $limit) { $open_group = 0; } ?>
<li class="dropdown-group">
<?php endif ?>

<?php $open_group++ ?>
<ul class="unstyled">
<li class="nav-header"><?php echo __($item['name'], null, 'global
    ') ?></li>
<?php foreach ($sub_items as $sub_key => $value): ?>
<li><a href="<?php echo url_for($value['action']) ?>"><?php echo
    __($value['name'], null, 'global') ?></a></li>
<?php endforeach ?>
</ul>

<?php if ($open_group == $limit || $total_items == ($key - 1)) :?>
</li>
<?php $open_group = 0; ?>

```

```

<?php endif ?>

<?php else : /* ! isset($item['sub_items']) */?>

<?php if (isset($navbar_single_group) && $navbar_single_group) : ?
    >
<?php $tmp_new_op = false ?>
<?php if (!$open_single && ($open_group == $limit || $open_group
    == 0)) :?>
<?php if ($open_group == $limit) { $open_group = 0; } ?>
<li class="dropdown-group single-item">
<?php $open_single = true; $tmp_new_op = true; ?>
<?php endif ?>

<?php if ($tmp_new_op || ($open_group != $limit && !$open_single))
    : ?>
<ul class="unstyled dropdown-single">
<?php $open_single = true; $open_group++ ?>
<?php endif ?>

<li><a href="<?php echo url_for($item['action']) ?>"><?php echo __
    ($item['name'], null, 'global') ?></a></li>

<?php if ($open_group == $limit || $total_items == ($key + 1)) :?>
</ul>
</li>
<?php $open_single = false; ?>
<?php endif ?>

<?php else : ?>

<?php if ($open_group != 0) : ?>
</li>
<?php $open_group = 0 ?>
<?php endif ?>
<?php $single_item = true ?>
<li><a href="<?php echo url_for($item['action']) ?>"><?php echo __
    ($item['name'], null, 'global') ?></a></li>

<?php endif ?>

<?php endif ?>
<?php endif ?>

```

```

<?php endforeach ?>
</ul>
</li>
<?php else : ?>
<li <?php if($principal['active']) : ?>class="active"<?php endif ?
    >><a href="<?php echo url_for($principal['action']) ?>"><?php
        echo __($principal['name'], null, 'global') ?></a></li>
<?php endif ?>

```

Código 2.9: Construcción del menú según `layoutNavbarItemResponsiveApart`

```

<?php if (isset($items)) : ?>
<li id="modal_<?php echo strtolower($principal['name'])?>" class="
    menu-apart">
    <a class="sprite sprite-<?php echo strtolower($principal['name'])
        ?>" style="margin-top: 1%;"></a>
    <a class="mm-next mm-fullsubopen navigation_btn" data-modulo="
        modulo_<?php echo strtolower($principal['name']) ?>"
        data-target="modal_<?php echo strtolower($principal['name']) ?
        >" href="#" ></a><span id="module_name"><?php echo __ (
            $principal['name'], null, 'global')?></span>

    <ul class="mm-listview inner" style="display: none;">
    <?php $total_items = count($items); $limit = 3; $open_group = 0;
        $open_single = false; $single_item = false; ?>
    <?php foreach ($items as $key => $item) : ?>
    <?php if(isset($item['name'])): ?>
<?php if(isset($item['sub_items'])) : ?>
<?php $sub_items = $item['sub_items'] ?>

    <?php if ($single_item) : ?>
    <?php $single_item = false ?>
    <!--<li class="divider"></li>-->
    <?php endif ?>

    <?php if ($open_single) : ?>
    <?php $open_single = false; ?>
    <?php endif ?>

    <?php if ($open_group == 0 || $open_group == $limit) :?>
    <?php if ($open_group == $limit) { $open_group = 0; } ?>
    <!--<li class="dropdown-group">-->
    <?php endif ?>

```

```

        <?php $open_group++ ?>
            <ul class="mm-listview" style="display: none;">
        <li class="nav-header"><?php echo __($item['name'], null, '
            global') ?></li>
            <?php foreach ($sub_items as $sub_key => $value): ?>
        <li><a href="<?php echo url_for($value['action']) ?>"><?php echo
            __($value['name'], null, 'global') ?></a></li>
            <?php endforeach ?>
            </ul>
    <?php if ($open_group == $limit || $total_items == ($key - 1)) :?>
        <!--</li>-->
    <?php $open_group = 0; ?>
    <?php endif ?>

    <?php else : /* ! isset($item['sub_items']) */?>
    <?php if (isset($navbar_single_group) && $navbar_single_group) : ?
        >
        <?php $tmp_new_op = false ?>
    <?php if (!$open_single && ($open_group == $limit || $open_group
        == 0)) :?>
        <?php if ($open_group == $limit) { $open_group = 0; } ?>
        <li class="dropdown-group single-item">
        <?php $open_single = true; $tmp_new_op = true; ?>
        <?php endif ?>

    <?php if ($tmp_new_op || ($open_group != $limit && !$open_single))
        : ?>
        <ul class="unstyled dropdown-single">
        <?php $open_single = true; $open_group++ ?>
        <?php endif ?>

    <li><a href="<?php echo url_for($item['action']) ?>"><?php echo __
        ($item['name'], null, 'global') ?></a></li>

    <?php if ($open_group == $limit || $total_items == ($key + 1)) :?>
        </ul>
    </li>
        <?php $open_single = false; ?>
        <?php endif ?>

    <?php else : ?>

```

```

    <?php if ($open_group != 0) : ?>
        </li>
        <?php $open_group = 0 ?>
    <?php endif ?>
    <?php $single_item = true ?>
<li><a href="<?php echo url_for($item['action']) ?>"><?php echo __
    ($item['name'], null, 'global') ?></a></li>

    <?php endif ?>

    <?php endif ?>
        <?php endif ?>
    <?php endforeach ?>
    </ul>
</li>
<?php else : ?>
    <li id="menu_component">
    <a class="sprite sprite-<?php echo strtolower($principal['name
    ']) ?>" style="margin-top: 1%;"></a>
    <a class="mm-next mm-fullsubopen" href="<?php echo url_for(
    $principal['action']) ?>"></a><span id="module_name"><?php
    echo __($principal['name'], null, 'global') ?></span>
    </li>
<?php endif ?>

```

Finalmente la reconstrucción del menú de navegación general logró mejorar el entorno destinado a dicha tarea, la interactividad de este componente pasó de ser engorrosamente difícil, a ser fácil y sencilla. Con la nueva propuesta de menú, el usuario no tendrá dificultad en viajar a través de los módulos activos, pues los efectos *dropdown* del anterior menú, fueron sustituidos y tratados por separado para una mejor comprensión del contenido(ver 2.1).

De la misma manera será necesario alterar algunos atributos en los elementos de tipo imagen que aunque forman parte de la interfaz(*lado del cliente*), estos son construidos de manera dinámica a través de código Hypertext Preprocessor (**PHP**). Esto es necesario ya que más tarde serán usados por las funciones **jQuery** encargadas de esta tarea (ver 2.7). Además se añadió el archivo **adaptive-images.php** a la carpeta raíz del sitio y se colocó el siguiente *script* al inicio de la cabecera(*head*)

Código 2.10: Sentencia que identifica la resolución del dispositivo.

```

<script>document.cookie= 'resolution= '+Math.max(screen.width ,
screen.height)+'; path=/'; </script>

```



Figura 2.1: Resultados obtenidos en el menú de la plataforma.

También se modificó el archivo **.htaccess** para añadir la librería.

Código 2.11: Añadiendo librería.

```
# Adaptive-Images -----
# Add any directories you wish to omit from the Adaptive-Images
  process on a new line, as follows:
# RewriteCond %{REQUEST_URI} !some-directory
# RewriteCond %{REQUEST_URI} !another-directory

RewriteCond %{REQUEST_URI} !assets

# Send any GIF, JPG, or PNG request that IS NOT stored inside one
  of the above directories
# to adaptive-images.php so we can select appropriately sized
  versions
```

```
RewriteRule \.(?:jpe?g|gif|png)$ adaptive-images.php

# END Adaptive-Images
-----
```

De igual manera la librería se modificó para que tuviese en cuenta los **breakpoints** especificados anteriormente.

Código 2.12: Aspectos tratados.

```
/* CONFIG ----- */
$resolutions = array(768, 640, 480, 360, 320); // the resolution
break-points to use (screen widths, in pixels)
$cache_path = "ai-cache"; // where to store the generated
re-sized images. Specify from your document root!
$jjpg_quality = 75; // the quality of any generated JPGs on a
scale of 0 to 100
$sharpen = TRUE; // Shrinking images can blur details,
perform a sharpen on re-scaled images?
$watch_cache = TRUE; // check that the adapted image isn't stale
(ensures updated source images are re-cached)
$browser_cache = 60*60*24*7; // How long the BROWSER cache should
last (seconds, minutes, hours, days. 7days by default)
```

Tabla 2.3: Cambios realizados en la plataforma.

Lado del cliente

- 1- Definir los **breakpoints** a usar para que la plataforma identifique los tamaños de pantalla y la misma pueda ajustarse.
 - 2- Integrar un **sprite** con las imágenes usadas en la navegación.
 - 3- Trazar una estrategia de contenido haciendo uso de los lenguajes **JS**, **HTML** y **CSS**.
 - 4- Cambiar las etiquetas **HTML** existentes por sus similares de **HTML5** en las secciones más importantes de la plataforma.
 - 5- Integrar la librería **jQuery Lazy** para retrasar la carga de imágenes.
-

Lado del servidor

- 1- Re-definir los estilos de los elementos **HTML** creados dinámicamente en los archivos **layoutNavbarItem.php** y **layoutNavbar.php**
- 2- Integrar la librería **Adaptive images** para reducir los pesos de las imágenes y aumentar la velocidad de navegación.

2.5. Diseño de estrategias para los *breakpoints*

Cuando se navega a través de dispositivos móviles, se puede notar la falta de espacio para visualizar los componentes usados en una web. Es por ello que durante el proceso de desarrollo de la tarea en cuestión se hayan tomado en cuenta una serie de aspectos para poder ocultar componentes de la plataforma que no pudieran ser visualizados de manera correcta.

Componentes como los menús laterales y el menú de navegación de la aplicación, ocupaban mucho espacio o en algunos casos poseían un efecto de animación a través del cual era difícil navegar. Para resolver obstáculos como este se decidió agruparlos en lugar de mostrarlos por defecto. La figura 2.2 muestra cual será el comportamiento por defecto de la plataforma, de esta manera la aplicación solo mostrará al usuario el contenido educativo que en ella se presenta. En caso de que el usuario necesite interactuar en la aplicación, deberá disponer de los botones que se presentan en la figura.

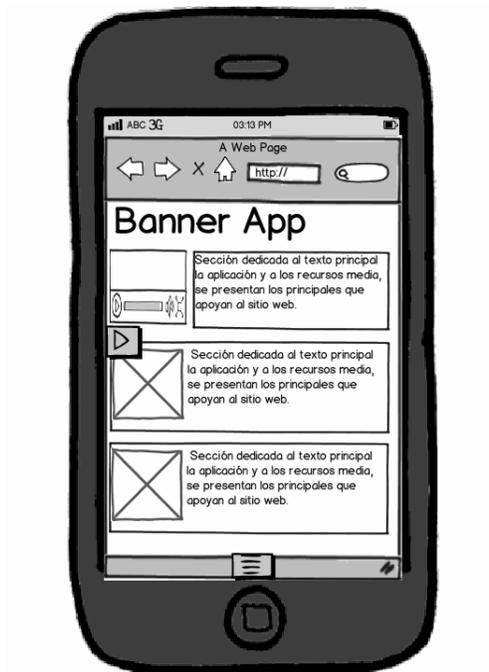


Figura 2.2: Prototipo de diseño para ZERA.

Como antes se explicó, la aplicación en algunos tamaños de pantalla no dispone de suficiente espacio. Por lo que depende totalmente de los botones situados en el lateral derecho y en la parte inferior central de la pantalla; con estos es posible navegar o interactuar

directamente con el contenido. Partiendo de ahí es posible visualizar componentes como el menú de navegación de contenidos y el menú de navegación general de la aplicación(ver 2.3).

Para la elaboración de estos conceptos fue necesario el uso de lenguajes como **CSS3** acompañado de librerías *Java Script* como *jQuery* y como base se usaron componentes **HTML**.

Los componentes **HTML** se agruparon de esta manera, para que el usuario tuviera al alcance de un clic la posibilidad de navegar a través de los contenidos de manera más fácil. El menú mostrado luego de la interacción con el botón inferior de la pantalla es mostrado en primer plano sobre una capa semitransparente, la cual dará la posibilidad de abandonar la navegación en caso que se desee, con solo dar un clic sobre el botón de salir.

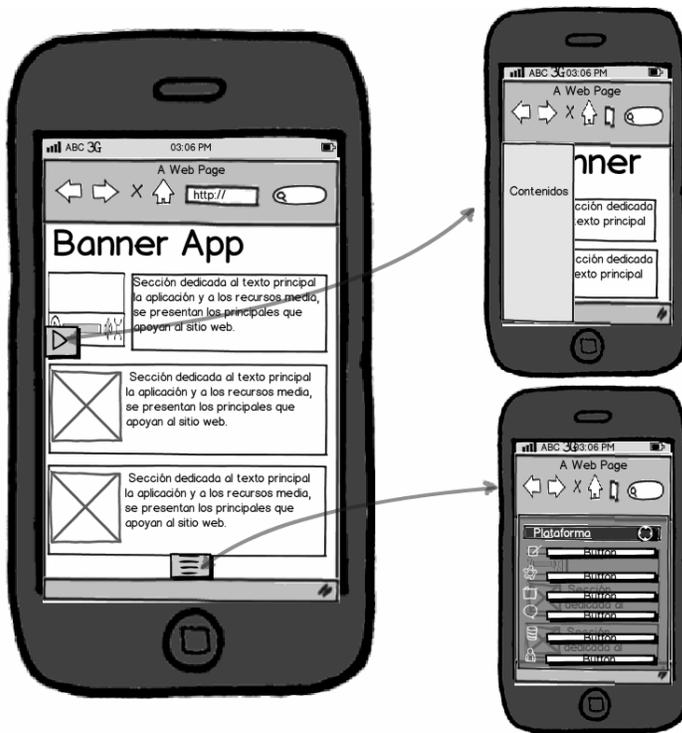


Figura 2.3: Prototipo de diseño para ZERA.

2.6. Eventos y animaciones

Como parte del proceso de diseño de estrategias de contenido para apoyar las interfaces que la plataforma tomará en los diferentes anchos de pantallas se hace uso de **JQuery**. Con este framework *Java Script* se podrá dar animación y capturar eventos en los diferentes componentes **HTML** que la aplicación posee, de tal manera también se logrará brindar una mayor compatibilidad del código con los navegadores existentes.

Actualmente la mayoría de los navegadores poseen ciertas dificultades para mostrar animaciones. Frameworks *Java Scripts* como **JQuery** o diversas librerías *Java Scripts* para generar animaciones, tienen en cuenta este tipo de dificultades.

No obstante, para evitar futuros problemas, siempre es bueno estar actualizado con las últimas versiones del navegador a usar. Pues, el avance de las tecnologías garantiza que las nuevas liberaciones de este tipo de productos ya tengan corregido errores de esta índole. La librería que se usó en el desarrollo de este trabajo (*jQuery Lazy*) posee total compatibilidad con navegadores como: **Internet Explorer**, **Firefox / Firefox Mobile**, **Chrome / Chrome Mobile**, **Safari / Safari Mobile** y **Android Browser**.

De igual manera, en el desarrollo de estos eventos se hace uso de la función **animate()** que posee **JQuery**, la cual es el equivalente de crear una función de animación a través del **CSS**, de esta forma también se está ahorrando tiempo de desarrollo en la aplicación. De la misma manera también se hace uso de la función **toggle()**, que aunque no es explícitamente para hacer animaciones, también posee dentro de sus parámetros, la posibilidad de crear tiempos en la aparición y desaparición de componentes **HTML**.

2.7. Uso de jQuery Lazy

Finalmente para el tratamiento de las restantes imágenes presentes en la aplicación se hará uso del *plugin*¹ **jQuery Lazy**. La propuesta de esta librería va acorde con las necesidades de la plataforma en este ámbito, ya que al ser la plataforma una aplicación en la cual las imágenes son añadidas por un usuario, las mismas podrían tener varios pesos(dígase *MegaBytes*).

Visto desde este punto de vista esto es un problema cuando se está navegando en la web a través de un smartphone. La librería en sí ayuda a retrasar la carga de las imágenes en la web según sean visualizadas en el *viewport*, evitando así carga extra en el servidor.

La librería consta de una serie de características que hacen de esta una solución factible entre las cuales es de interés: la alta compatibilidad que tiene con la mayoría de los

¹Librería Java Script que permite el manejo del comportamiento de las imágenes en la web.

navegadores usados hasta el momento. La misma consta con una serie de funciones para armonizar el uso de las imágenes durante el proceso de carga.

2.7.1. Integrando jQuery Lazy

La librería sera añadida en la cabecera(*header*) de la aplicación de manera que esta pueda estar activa durante todo el uso de la plataforma a través de los diferentes módulos de la misma. Una vez arribado a este punto, una función **jQuery** hará aparecer las imágenes señaladas como objetivo de esta funcionalidad.

Código 2.13: Uso de jQuery Lazy

```
jQuery(document).ready(function() {  
  jQuery("img.lazy").lazy({  
    delay: 2000 });  
});
```

Para el apropiado funcionamiento de la librería se hicieron algunos cambios pertinentes en los atributos de la imagen, pues como se puede observar la función anterior es ejecutada a todos los elementos **HTML** de tipo *imagen* con la clase *lazy*.

2.8. Conclusiones Parciales

Luego de puestas en vigor las propuestas anteriormente comentadas, se estableció una estrategia de contenido en la plataforma, para que la misma pudiera renderizar todos su componentes de forma correcta y a la vez adaptarse a los anchos de pantalla establecidos. También se incluyeron nuevas técnicas y librerías para el tratamiento de las imágenes presentadas en la plataforma y contribuir a mejorar los tiempos de respuesta de la misma. Además se tomaron una serie de elementos **HTML** existentes en la plataforma y se sustituyeron por sus semejantes de **HTML5** para de esta forma aprovechar su nivel semántico y establecer en la plataforma código limpio y fácil de entender.

Capítulo 3

Validación

Introducción

En ocasiones durante la etapa de diseño, los desarrolladores necesitan saber si sus ideas reflejan las necesidades de los usuarios. En consecuencia, el proceso evaluador de la interfaz debe entrelazarse con el de diseño, proporcionándole así una comunicación fluida con el usuario potencial. El objetivo de este tipo de metodologías de evaluación es mejorar la interfaz lo máximo posible antes de su distribución comercial, en el momento en el que los cambios son medianamente sencillos y no demasiado caros de acometer[Vega, 2014].

3.1. Métodos de prueba

Para el desarrollo del proceso de validación se tuvo en cuenta las siguientes formas de validación[Vega, 2014]:

- Pruebas de rendimiento.

Las **pruebas de rendimiento** son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos.

- Test de “usabilidad”

Tests “destructivos”: Se propone a los usuarios el vencer a la máquina y encontrar errores de programación.

- Encuestas.

Una **encuesta** es un procedimiento dentro de los diseños de investigación descriptivos (no experimentales), en el que el investigador busca recopilar datos por medio de un cuestionario previamente diseñado o una entrevista a alguien, sin modificar el entorno ni el fenómeno donde se recoge la información.

3.2. Procedimientos de validación

3.2.1. Test de “usabilidad”

Tests “destructivos”

El diseño del siguiente test fue concebido bajo la idea de que la plataforma esta destinada a cualquier tipo de enseñanza, por tanto, cualquier persona, sin importar origen o educación previa, debe ser capaz de explotar todo su potencial. Es por ello que se escogieron varios sujetos para que probaran la plataforma y trataran de encontrar errores en la programación de las vistas.

Resultados : Con el uso de este método se lograron encontrar una serie de no conformidades en las estrategias de contenido de la plataforma ver(??) que fueron solucionadas en una segunda iteración inspeccionando cada propiedad **CSS** de los elementos para determinar cuales distorsionaban su renderizado.

3.2.2. Encuesta

Para poner en práctica esta forma de validación se procedió a realizar una encuesta a diferentes sujetos. Para la selección de los mismos se tuvo en cuenta varios aspectos, como son: experiencia profesional que poseen, su reconocido prestigio profesional o estudiantil, los años vinculados a la **UCI** tanto de trabajador como de estudiante y disposición a participar en la encuesta, para así obtener criterios diversos y poder realizar un análisis más detallado del problema de estudio; al tener tantos puntos de vista con respecto al tema. Este método de evaluación se emplea para comprobar la calidad y efectividad de los resultados de las investigaciones.

Elegir los sujetos de investigación atendiendo a las particularidades mencionadas anteriormente, ayuda a obtener resultados con calidad, junto a otras cualidades propias de éstos como pueden ser: la seriedad, la honestidad, la sinceridad, la responsabilidad y otras en este sentido, que hacen que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

Para realizar la validación se seleccionaron 37 sujetos siguiendo los criterios expuestos inicialmente, quienes respondieron a la encuesta. La misma se aplicó de forma anónima, con el objetivo que la opinión de un sujeto no repercuta en otro, de esta forma cada sujeto arrojará opiniones diferentes. Para la validación se tuvo en cuenta los siguientes parámetros(??).

Resultados: Con la realización de la encuesta se demostró que el uso de buenas prácticas en los lenguajes **CSS**, **HTML** y **JS** pueden hacer de una aplicación web común, una web adaptativa, así como mejorar la experiencia de usuario para motivar al usuario futuro a mantenerse conectado. De manera general la encuesta demostró que existe un antes y un después con respecto a la versión anterior de dicha plataforma. Las estadísticas mostradas (ver 3.2.2) corroboran que la *versión 1.2.1* de **ZERA** tuvo gran aceptación entre los usuarios encuestados, ya que en una escala del 3 al -3, donde 3 es la calificación más alta y -3 la mas baja, más de la mitad de ellos dieron buenas opiniones acerca de las nuevas inclusiones asociadas. Esta encuesta también arrojó una serie de no conformidades que repercutían en las vistas diseñadas, las cuales fueron corregidas en una segunda iteración.

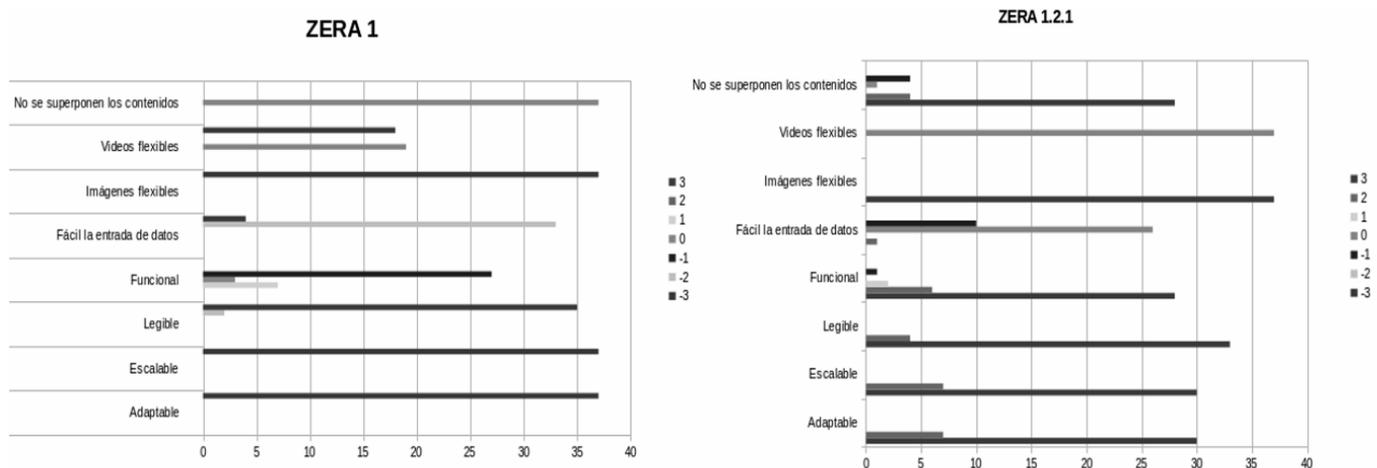


Figura 3.1: Resultados de la encuesta ZERA.

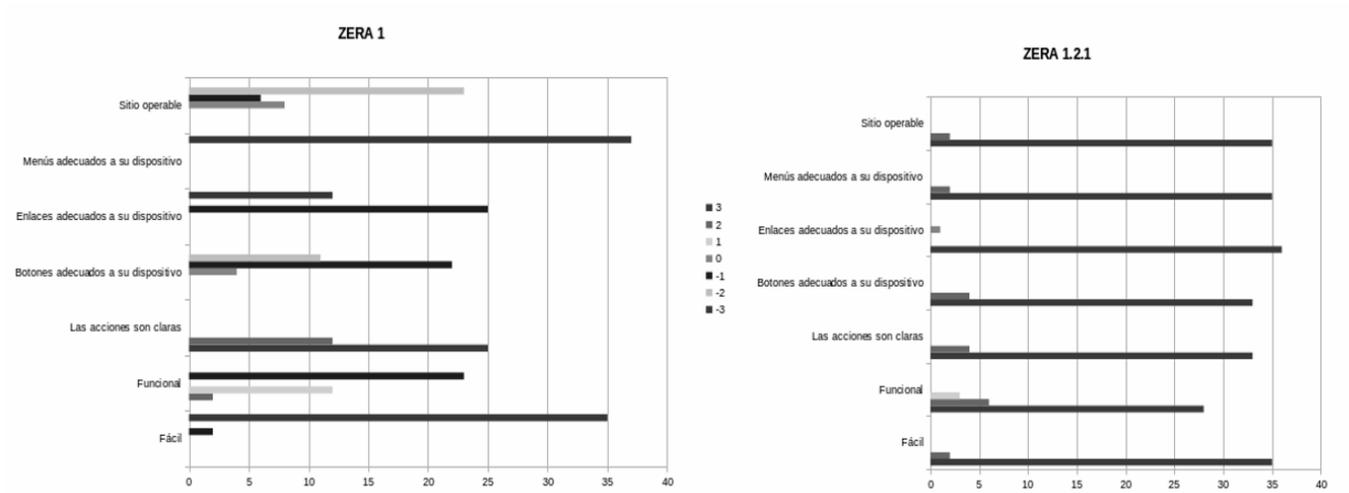


Figura 3.2: Resultados de la encuesta ZERA.

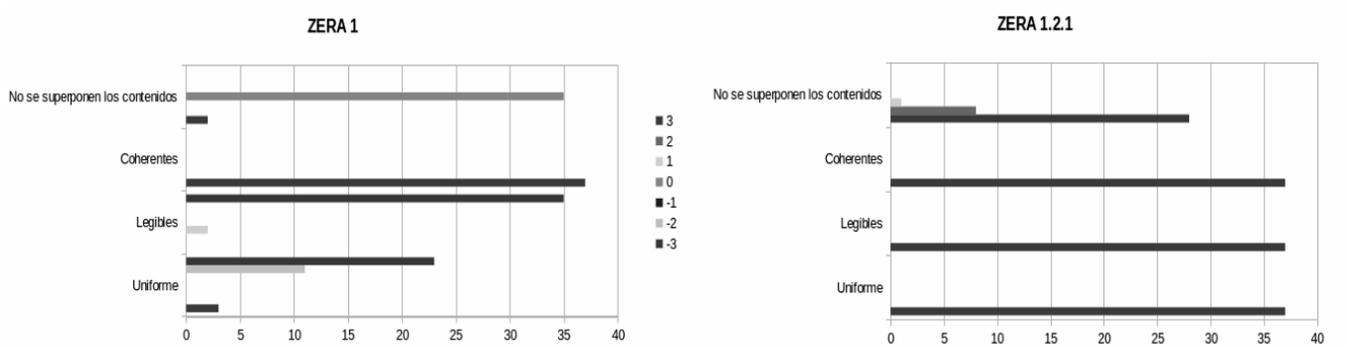


Figura 3.3: Resultados de la encuesta ZERA.

3.2.3. Pruebas de rendimiento

Metodología

Para la elaboración de esta etapa de prueba se siguen ciertos pasos para ejecutar de manera exitosa dicha prueba, entre los cuales se encuentran los siguientes:

- Evaluación e Identificación de escenarios de Pruebas.
- Planificación de actividades.
- Análisis, diseño, construcción de los scripts de prueba (Scripting).
- Ejecución de las pruebas.
- Análisis de resultados.
- Informes de resultado que evalúan el comportamiento del aplicativo, su **performance** y **escalabilidad**.

Independientemente de que estas pruebas pueden pre-elaborarse de forma manual siguiendo algún patrón, existen muchas aplicaciones que integran estas técnicas. Para la ejecución de esta fase se utilizó el navegador web *Firefox* en su versión 30, el cual propone una herramienta para dicho proceso.

Tabla 3.1: Atributos tomados en cuenta para las pruebas de rendimiento.

Tiempos de respuesta en el uso de los recursos de la aplicación.

- 1- Uso de imágenes.
- 2- Uso de **JS**.
- 3- Uso de **CSS**.
- 4- Uso de **HTML**.

Resultados: Tras esta prueba se observó que el uso de los recursos **JS**, **CSS** y de las imágenes disminuyó considerablemente(ver 3.4). El uso del recurso alegórico al color azul(*imágenes*) comenzó a desaparecer gradualmente gracias a la aplicación de buenas prácticas en el uso de los mismos, de la misma manera los recursos *naranja* y *amarillo* correspondientes al uso del **JS** y **CSS** se mantuvieron al margen, dato que debemos resaltar pues ZERA 1.2.1 consta de un mayor tamaño con respecto a su homólogo anterior. No obstante algunas vistas mostraron no conformidades en cuanto al uso de estos recursos, estos errores fueron corregidos en una segunda iteración haciendo uso de compresores **JS** y eliminando estilos **CSS** que no eran usados.

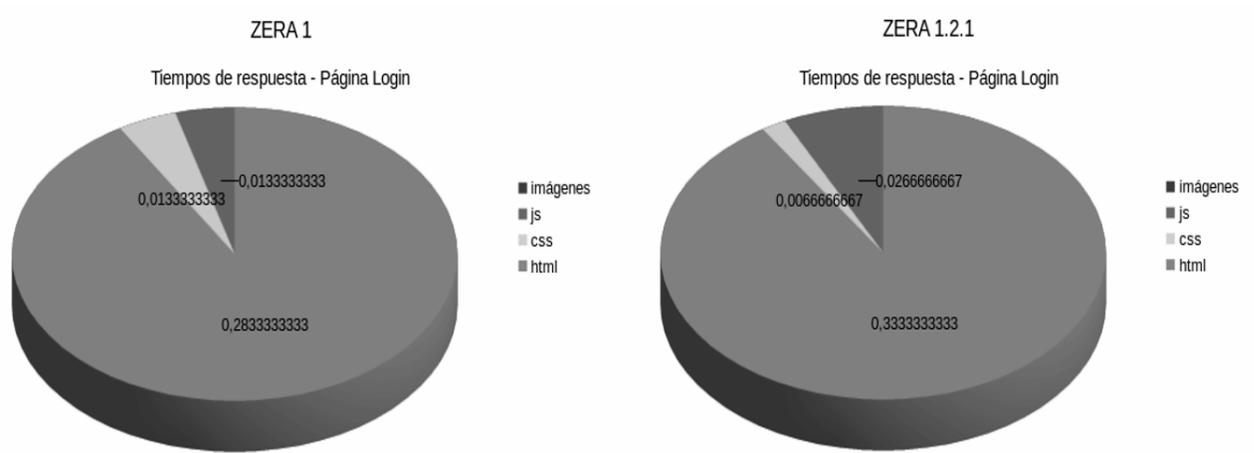


Figura 3.4: Pruebas de rendimiento.

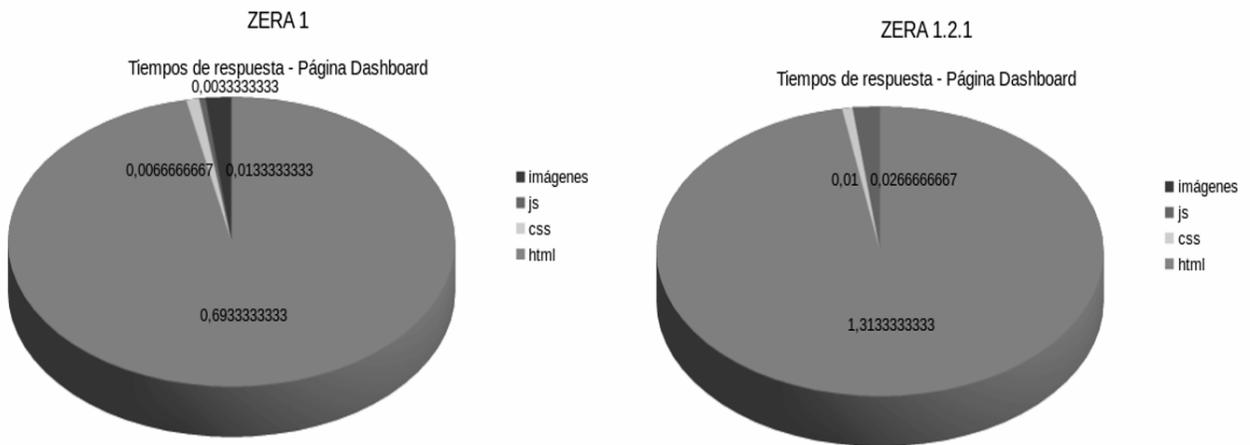


Figura 3.5: Pruebas de rendimiento.

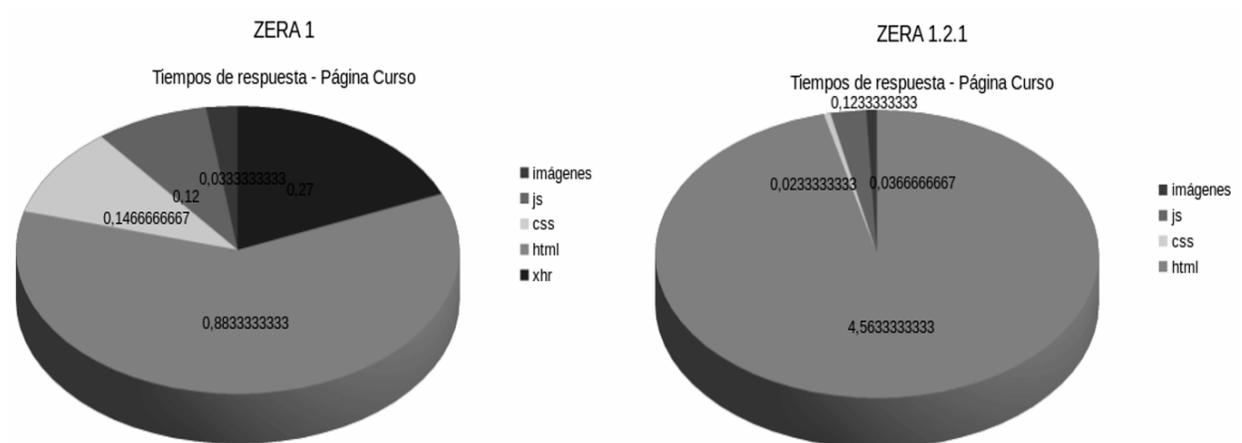


Figura 3.6: Pruebas de rendimiento.

3.3. Conclusiones Parciales

El uso de estas técnicas de validación permitió corroborar el cumplimiento de las tareas previstas en la investigación. También fue posible establecer un nivel de calidad alto para el futuro despliegue de esta aplicación. Las validaciones se realizaron con una gran variedad de sujetos y herramientas, para así lograr abarcar la mayor cantidad de puntos de vistas posibles y poder eliminar cualquier error o indicio de error, obteniéndose así una mejor opinión acerca de la calidad de la aplicación.

Conclusiones

Con la elaboración de este trabajo se da cumplimiento a las tareas propuestas, obteniéndose como principales resultados las funcionalidades que permiten la total interacción del usuario de forma eficiente con los componentes de la aplicación. Se identificaron las tendencias actuales que tratan el uso de dispositivos móviles para el mejoramiento del proceso de enseñanza y aprendizaje, así como la inclusión del paradigma adaptativo en las plataformas educativas. Se generó la documentación necesaria para la implementación de las tareas listadas y además se generaron los prototipos que dieron cumplimiento a cada uno de los flujos de interacción que se propusieron. Como resultado de toda la investigación, se logró mejorar la adaptación de todos los contenidos existentes en la plataforma y al mismo tiempo se optimizó la navegación del sitio. Finalmente se validó la propuesta de solución a través de encuestas y pruebas realizadas a las interfaces, demostrando así que; si se desarrolla una interfaz que incorpore buenas prácticas de diseño adaptativo en la plataforma educativa **ZERA** entonces todos los componentes mostrados en la aplicación se podrán visualizar en cualquier tipo de dispositivo.

Recomendaciones

- Ampliar la estrategia de adaptación de contenidos a Unificado **Unified** logrando una adaptación al contexto del usuario.
- Extender el uso de la librería *jQuery Lazy* a las próximas versiones de la plataforma y al resto de esta.

Referencias bibliográficas

- [Alvarez, 2014] Alvarez, S. (2014). Optimización de tu web mediante ordenación del código.
- [Blackboard, 2014] Blackboard (2014).
- [Boneu, 2007] Boneu, J. M. (2007). Plataformas abiertas de. 4.
- [Chamilo, 2014] Chamilo (2014). Chamilo - Chamilo Classic Campus.
- [comscoreanalytics, 2014] comscoreanalytics (2014). Number of Global Users.
- [DeveloperEconomics.com, 2012] DeveloperEconomics.com (2012). Devices Currently targeted by developers.
- [Frain, 2012] Frain, B. (2012). *No Title*.
- [Goldstein et al., 2011] Goldstein, A., Lazaris, L., and Weyl, E. (2011). *HTML5 & CSS3 for the real world*.
- [Google, 2012] Google (2012). The new Multi-screen world: Understanding Cross-platform Consumer Behavior.
- [GoogleTrends, 2015] GoogleTrends (2015). Evolucion web.
- [Guidelines, 2014] Guidelines, W. (2014). Mobile Web Best Practices 1.0. (July 2008):1–30.
- [Gustafson, 2011] Gustafson, A. (2011). *Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement*.
- [Guy et al., 2010] Guy, R., Chaka, C., Hoorebeek, M. V., Cohen, E. B., Masrom, M., Ismail, Z., and Sagrario, M. (2010). *Mobile Learning Pilot Projects and Initiatives*. Informing Science Press, Santa Rosa, California.

- [Herrera Santos, 2013] Herrera Santos, Adia y Viamontes Herrera, L. (2013). Módulo de gestión de temas de interfaz de usuario para el subsistema Bachiller de la Plataforma Educativa Zera.
- [Horizon, 2015] Horizon, T. N. (2015). The NMC Horizon Report: 2015 Higher Education Edition.
- [Knight, 2011] Knight, K. (2011). *Responsive Web Design: What It Is and How to Use It*.
- [Manso, 2014] Manso, Y. (2014). SOLUCION INFORMATICA INTEGRAL PARA INCLUIR A LA PLATAFORMA EDUCATIVA ZERA EN EL PARADIGMA M-LEARNING.
- [Marcotte, 2010] Marcotte, E. (2010). RESPONSIVE WEB DESIGN.
- [Marcotte, 2011] Marcotte, E. (2011). *Responsive Web Design*.
- [Matthew, 2010] Matthew, D. (2010). *HTML5 Designing Rich INTERNET Applications*.
- [Ministerio de Educación, 2015] Ministerio de Educación, C. y. D. E. (2015). BYOD – Regulación Interna en un Centro Educativo.
- [MobiForge, 2009] MobiForge (2009). Adaptation Strategies.
- [Moodle, 2014] Moodle (2014). Features - MoodleDocs.
- [Peterson, 2014] Peterson, C. (2014). *Learning Responsive Web Design*.
- [Pisanty et al., 2010] Pisanty, A., Enríquez, L., Chaos-Cador, L., and 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.
- [Sakai, 2014] Sakai (2014). Sakai.
- [Snell, 2013] Snell, J. (2013). *Flexible Everything: Getting Responsive With Web Design*.
- [UNESCO, 2012] UNESCO (2012). Activando el aprendizaje móvil en américa latina.
- [Vega, 2014] Vega, M. A. C. (2014). *METODOLOGÍAS DE EVALUACIÓN DE INTER-FACES GRÁFICAS DE USUARIO*.
- [W3C, 2014a] W3C (2014a). HTML data-* Attributes.

[W3C, 2014b] W3C (2014b). Mobile Web Best Practices, Basic Guidelines.

[WAI-ARIA, 2014] WAI-ARIA (2014). WAI-ARIA Overview.

Acrónimos

LMS Learning Management Systems

UCI Universidad de las Ciencias Informáticas

FORTES Centro de Tecnologías para la Formación

HTML Hyper Text Markup Language

CSS Cascading Style Sheets

RWD Responsive Web Design

DOM Document Object Model

W3C World Wide Web Consortium

SEO Search Engine Optimization

RSS Really Simple Syndication

LMS Learning Management System

CMS Content Manager System

WAI-ARIA W3C's Accessible Rich Internet Applications

HTTP Hypertext Transfer Protocol

UNESCO Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura

IMS-QTI IMS Question & Test Interoperability Specification

SCORM Sharable Content Object Reference Model

PHP Hypertext Preprocessor

BYOD Bring Your Own Device

BYOT Bring Your Own Technology

JS Java Script