

PROYECTO DE GRADUACION
Trabajo Final de Grado

Multiplicidad digital
Hacia la expansión de interfaces web para dispositivos móviles

Reynoso, María Amantani
Cuerpo B del PG
24 de Julio de 2014
Diseño de Imagen y Sonido
Ensayo
Nuevas Tecnologías

Agradecimientos

Quisiera reconocer a las personas que me acompañaron en este proceso de aprendizaje y de crecimiento.

Principalmente agradezco el constante apoyo brindado por mi familia, por dejarme siempre seguir mis sueños y permitirme desarrollar mis capacidades sin presiones. A mis papás Darío y Cecilia por solventar mi formación, y confiar en mí. También a mis hermanos, Mailén, Guadalupe y Manuel, por darme una mano cuando fue necesario.

A su vez, quiero agradecer a Sebastián, por transitar este largo recorrido conmigo y ser un compañero fundamental en todo el proceso, y más allá también. Por alentarme a nunca bajar los brazos, y ser un sostén primordial en mi vida.

Agradezco también a todos los compañeros y profesores con los que cruzamos camino, y me hicieron creer y crecer tanto personal como profesionalmente. Y en el contexto de este trabajo, a las profesoras Marisa Cuervo y Mónica Incorvaia por guiarme en la realización de la culminación de mi carrera.

A todos ellos, ¡gracias!. No creo haber podido llegar a esta instancia en mi vida sin su apoyo y acompañamiento.

Índice

| | |
|---|-----------|
| Introducción | 6 |
| Capítulo 1: El fascinante mundo del diseño web | 12 |
| 1.1. ¿Qué hay dentro de un sitio web? | 13 |
| 1.1.1. El almacén: HTML..... | 14 |
| 1.1.2. El maquillaje: CSS..... | 16 |
| 1.1.3. La magia: <i>Javascript</i> | 19 |
| 1.1.4. La lógica: lenguajes de programación del lado del servidor | 21 |
| 1.2. <i>Surfeando</i> la Web: los navegadores | 22 |
| 1.3. Navegantes satisfechos: la experiencia de usuario..... | 24 |
| 1.3.1. Facilitando caminos mediante la usabilidad | 25 |
| 1.3.2. Accesibilidad para todos y todas..... | 27 |
| Capítulo 2: La Web en todos lados | 28 |
| 2.1. La red al alcance de la mano: dispositivos móviles..... | 29 |
| 2.1.1. Teléfonos pensantes: <i>smartphones</i> | 30 |
| 2.1.2. Pequeñas computadoras: <i>tablets</i> | 32 |
| 2.1.3. ¿Aún más?: otras clases de dispositivos | 33 |
| 2.2. El creciente avance de las plataformas móviles..... | 34 |
| 2.3. Un <i>bicho</i> raro: el usuario móvil..... | 35 |
| 2.4. Hay una <i>app</i> para todo | 37 |
| 2.4.1. Aplicaciones nativas | 37 |
| 2.4.2. Aplicaciones web | 38 |
| 2.4.3. Aplicaciones híbridas..... | 40 |
| 2.5. Cuando el entorno móvil marca fronteras | 41 |
| 2.5.1. <i>Small, large</i> y <i>XL</i> : resoluciones..... | 42 |
| 2.5.2. ¿Vertical o apaisado?: orientaciones de pantalla..... | 43 |
| 2.5.3. Cada navegador con sus reglas | 46 |

| | |
|--|------------|
| Capítulo 3: La Web líquida | 47 |
| 3.1. Adaptar o morir..... | 48 |
| 3.2. Responder acorde: <i>responsive web design</i> | 50 |
| 3.3. ¿Uno, dos o cinco? Diseñar responsivamente en vez de por separado..... | 52 |
| 3.4. Avancen los móviles: la filosofía <i>mobile first</i> | 55 |
| 3.5. ¿Mejora progresiva o degradación gradual?..... | 56 |
| Capítulo 4: ¡Manos a la obra! <i>Responsive web design</i> en camino..... | 58 |
| 4.1. Condicionar las reglas: <i>media queries</i> | 59 |
| 4.2. Medir relativamente con unidades flexibles | 61 |
| 4.3. Verdaderos tipos móviles: fuentes adaptables..... | 63 |
| 4.4. Imágenes elásticas..... | 65 |
| 4.5. Cada cosa en su lugar: sistemas de grillas..... | 70 |
| 4.6. Cerrando grietas con <i>polyfills</i> | 72 |
| 4.7. La era <i>touch</i> : gestos táctiles | 74 |
| 4.8. Marche un <i>combo</i> para llevar: <i>frameworks</i> | 76 |
| Capítulo 5: De la teoría a la práctica: diseño web responsivo en acción | 78 |
| 5.1. ¿Quién lo usa?: análisis de aplicación..... | 78 |
| 5.1.1. <i>Responsive web design</i> en las grandes ligas: Infobae..... | 84 |
| 5.1.2. Pensando un portfolio: rediseño del sitio web personal de la autora..... | 86 |
| 5.2. Ganan todos: beneficios del uso de esta práctica..... | 89 |
| Conclusiones | 91 |
| Imágenes seleccionadas..... | 94 |
| Lista de referencias bibliográficas | 98 |
| Bibliografía..... | 102 |

Índice de figuras y tablas

| | |
|---|----|
| Figura 1: Anatomía de una página web | 94 |
| Figura 2: Declaración de estilo en CSS | 94 |
| Figura 3: Formulario de contacto en sitio de juegos creado por la autora dentro del contexto universitario | 94 |
| Tabla 1: Unidades absolutas en CSS | 95 |
| Tabla 2: Unidades relativas en CSS | 95 |
| Figura 4: Aplicaciones nativas vs. web | 96 |
| Figura 5: Orientaciones de pantalla | 96 |
| Figura 6: Declaraciones de <i>media queries</i> | 97 |

Introducción

En las últimas dos décadas, Internet ha crecido hasta ser una de las herramientas más valiosas con las que cuenta la humanidad. Es la fuente de información más dinámica y extensa, en la cual quien lo desee puede tomar parte. Se accede a ella mediante un dispositivo que lo permita: una computadora de escritorio, una *laptop* o *notebook*, una consola de juegos, una *tablet*, un teléfono celular, entre otros. Inicialmente, el primero mencionado era el medio de elección de la mayor parte de los usuarios.

La manera de interactuar con los contenidos que ofrece la Web, casi en su totalidad, es mediante una interfaz gráfica de un sitio, una aplicación, o un software. Ésta sirve como un mediador que permite a los usuarios visualizar los contenidos. Cada una de ellas es diferente, ya que responde a intereses comunicacionales distintos, estilos, culturas, épocas, e interminables variables más que varían entre sí.

El uso y desarrollo de los dispositivos móviles a través de los años recientes ha cambiado el paradigma de lo que se conoce como la Web. La masificación de los mismos en la sociedad ha generado un cambio de estrategia para quienes tienen como tarea desarrollar las interfaces gráficas para ella.

Anteriormente se concebía la creación de un sitio basándose en resoluciones estándares de monitores; se optimizaban para que se adapten a las pantallas de computadoras de escritorio, y a lo sumo a las de computadoras personales, las cuales suelen integrar resoluciones similares. Sin embargo, con el desembarco de la masificación de tecnología móvil y su accesibilidad, ha aparecido una nueva dimensión en la cual Internet es prácticamente omnipresente. El diseño web debe evolucionar y adaptarse a ello.

Multiplicidad digital es un Ensayo cuya temática general es el diseño de interfaces gráficas para la web, enfocando puntualmente su recorte en el diseño para dispositivos móviles. Éste se encuadra dentro de la línea temática Nuevas Tecnologías y pertenece al área de la carrera de Diseño de Imagen y Sonido.

El Proyecto de Graduación (PG) pretende ofrecer al lector una introducción general a este campo, para abocarse de lleno al paradigma móvil: qué es, cómo se desarrolla y domina, cuál es su importancia en el área y por qué debería ser un estándar en cualquier proyecto de desarrollo de un sitio web.

La carrera de Diseño de Imagen y Sonido incluye actualmente en su plan de estudios, en el primer año y la primera mitad del segundo año, materias que enfocan completamente la temática, lo que denota la importancia de este uso del conocimiento. No obstante, en ninguna de las currículas se toma el tema específico del recorte.

La elección de la temática se debe a la novedad y potencial que esta práctica ofrece al desarrollo de sitios web. La autora sostiene la maquetación como enfoque profesional, y en su experiencia de trabajo encuentra que es escasa la puesta en práctica del diseño web para dispositivos móviles.

La pregunta disparadora de este trabajo es ¿cómo puede el diseño web para dispositivos móviles potenciar cualquier sitio?, la cual intentará responderse durante el progreso del mismo. Se intenta explicar qué es este tipo de diseño, sus técnicas, sus utilidades y campos de aplicación y desarrollo. En consecuencia, pretende exponer argumentos que demuestren que la utilización de esta modalidad maximiza la calidad y *usabilidad* del mismo, así también justificar las razones por la cual esta práctica debería ser un estándar en cualquier tipo de proyecto web.

El desglose temático que se desarrollará en cada capítulo consiste en lo siguiente:

El primer capítulo, *El fascinante mundo del diseño web*, tiene como finalidad introducir al lector en este campo y a las prácticas estándares del mismo. Es un acercamiento a la temática principal, y en él se establecen los puntos de partida que luego serán desarrollados a lo largo del proyecto. Aquí se exponen técnicas básicas para el desarrollo de un sitio web, dentro de los parámetros de visualización normales en computadoras.

El segundo capítulo, titulado *La Web en todos lados*, introduce al lector en el campo de los dispositivos portables y las razones por las cuales el paradigma de diseño y desarrollo

antes expuesto debe modificarse en estos casos. En este capítulo, se demuestra el creciente avance de la tecnología móvil y se hace un análisis pronosticando el futuro de la misma.

La Web líquida, el tercer capítulo del trabajo, avanza en la exposición de una de las técnicas que aborda el diseño web para móviles: el diseño web *responsivo*. Este procedimiento es tomado como caso de estudio para este Proyecto por ser accesible y conveniente. Se introduce al lector en esta práctica, su desarrollo y algunos conceptos que complementan la idea general.

En el cuarto capítulo, llamado *¡Manos a la obra! Responsive web design en camino*, se busca ahondar en las prácticas específicas que requiere la aplicación de esta práctica de diseño adaptativo. Se sugieren y exponen diferentes herramientas que son complementarias entre sí para alcanzar un desarrollo compilante con esta técnica.

En el último capítulo, *De la teoría a la práctica: diseño web responsivo en acción*, se pretende exponer las razones por las cuales quien encare un proyecto de sitio web, debería incluir en la planificación del mismo el desarrollo para dispositivos móviles, y también las razones por la cual es provechoso generar un diseño web para móviles en un sitio ya desarrollado. En este capítulo se presenta una investigación sobre el estado de la Web móvil en la escena nacional. Posteriormente se toman dos casos de estudio para ser analizados con el marco teórico desarrollado a lo largo del Proyecto: el portal de noticias Infobae y un sitio personal creado por la autora.

Dentro del ámbito de las publicaciones en la Universidad de Palermo se ha recopilado una serie de trabajos que se encuentran ligados – en diferentes niveles – con este PG.

En orden de relevancia, comenzando por los más generales, son:

Pousadela, L. I. (2013). *Red Mundial Universitaria. Desarrollo Web con presencia en la Argentina para que los usuarios se relacionen con universidades*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. *Red Mundial Universitaria* desarrolla un proyecto profesional en el cual se plantea un sitio

web con índole social de intercambio para alumnos universitarios. Si bien ni la temática ni el recorte coinciden con *Multiplicidad digital*, el primer capítulo del PG – titulado *Internet, la red mundial* – formula una introducción interesante a la vasta red de información aquí también tratada, la Internet.

Tomassino, P. I. (2013). *Una página web para el casco histórico de Buenos Aires. Buscando su espacio en Internet*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. Este PG recorre el desarrollo de una propuesta de diseño web para el casco histórico de Buenos Aires. En él, su autora realiza un proyecto web íntegro y sostiene teóricamente sus decisiones de diseño. Este trabajo tiene como punto en común con *Multiplicidad digital* la temática desarrollada, pese a que el planteo de diseño de interfaces web para dispositivos portables no es considerado.

Espinoza Lozada, L. C. (2013). *Una nueva forma de comunicar. El nacimiento de la publicidad digital y el diseñador web*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. Este proyecto no comparte específicamente la temática con el presente PG, ya que se plantea desde la perspectiva del diseño gráfico y la publicidad. No obstante, se toma como antecedente ya que utiliza como foco de estudio la evolución del rol del diseñador/desarrollador web, análisis que es relevante a *Multiplicidad digital* ya que se intenta proponer aquí una evolución en este rol, al incluir nuevas técnicas y metodologías al paradigma de trabajo de este sujeto.

Paredes Uribe, J. C. (2012). *La marca móvil. Los aportes de los dispositivos móviles en la comunidad y sus efectos en la comunicación de marcas*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. *La marca móvil* toma como punto de partida la utilización de dispositivos móviles en el contexto de la web, como también lo hace este PG. Sin embargo, dicho proyecto se enmarca dentro del campo de la carrera de Publicidad, por lo cual su enfoque no pasa por el desarrollo de interfaces para dispositivos, sino el aporte de los mismos a la comunicación. Se toma de este proyecto como aporte su tercer capítulo, *Marketing digital*, el cual trata

específicamente sobre un tema desarrollado en *Multiplicidad digital*, las aplicaciones móviles.

Pardo Vargas, J. F. (2011). *Accesibilidad web y discapacidad. Una web sin barreras*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. Este trabajo desarrolla la necesidad de aplicación de herramientas de accesibilidad a proyectos web, con enfoque en usuarios discapacitados. La relación encontrada con el presente PG es la temática global, y el enmarcado en la accesibilidad, ya que *Multiplicidad digital* pretende demostrar esta misma necesidad, pero desde el enfoque de dispositivos, y no específicamente de un grupo de usuarios.

Botero, L. (2013). *Nuevos sistemas de comunicación. Tecnologías aplicadas al diseño de plataformas interactivas*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. Este proyecto comparte un acercamiento similar al tema de diseño de interfaces gráficas. Si bien *Nuevos sistemas de comunicación* es un Proyecto Profesional y apunta a la aplicación del marco teórico – parcialmente compartido con este PG – en una aplicación multiplataforma, la temática no es la misma ya que no trata específicamente del diseño para la web.

Lasso Guerrero, J. G. (2013). *Ergonomía en el diseño web. Usabilidad de sitios web dedicados al comercio electrónico en Buenos Aires*. Tesis de Maestría. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. Este trabajo, al ser presentado como Tesis de Maestría, a diferencia de este PG, se enmarca en otras normas de presentación y desarrollo. La temática, no obstante, es la misma. *Ergonomía en el diseño web* analiza específicamente cuestiones de usabilidad en un campo determinado – sitios web dedicados al comercio electrónico – en un área determinada, Buenos Aires, mientras que el presente PG intenta ser comprensivo a todo tipo de proyectos web.

Salgueiro, G. J. (2013). *Diseño de interfaces web efectivas y usables. El rol encubierto del diseñador web en el mercado local*. Proyecto de Graduación. Buenos Aires: Facultad

de Diseño y Comunicación. Universidad de Palermo. Este proyecto sirve de sustento teórico e inspiración, ya que los temas en él son tratados de manera novedosa, tal como apunta este ensayo. Si bien la temática es la misma, el recorte cubre un espectro más general – es decir, no se limita a modalidades de visualización – y hace enfoque en la usabilidad.

De Mingo, R. A. (2011). *Diseño de interfaces gráficas para la web móvil. Objeto de estudio: Facebook*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. Este trabajo comparte tanto el tema como el recorte con *Multiplicidad digital*. Sin embargo, el autor de *Diseño de interfaces gráficas para la web móvil* utiliza como eje el estudio de las redes sociales y el uso del diseño para móviles enfocado en un caso específico. Otra diferencia con el presente PG es la novedad, ya que en el lapso entre la publicación de uno y otro ha habido un desarrollo extenso en el campo de aplicación tratado.

Luzardo Alliey, A. M. (2011). *Diseño de la interfaz gráfica web en función de los dispositivos móviles. Caso de estudio: diarios digitales*. Tesis de Maestría. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo. Este trabajo fue publicado en el marco de tesis para la Maestría de Diseño, por lo tanto cumple con otras normas al presente PG. Sin embargo, la temática tratada, y su recorte, son compartidos. El enfoque que le da su autora al análisis de la teoría está basado en los diarios digitales. Es el trabajo que más se asemeja al objetivo que persigue este PG.

Multiplicidad digital busca convertirse en un modelo de consulta que pueda facilitar la comprensión de las temáticas tratadas, y su aplicación específica, para estudiantes en las áreas curriculares y la actualización para profesionales que se desempeñen en diseño web o afines. El aporte que este PG desea dejar es un antecedente justificativo sobre la importancia de integrar esta problemática en la actualidad digital.

Capítulo 1: El fascinante mundo del diseño web

El diseño de páginas y sitios deriva sus fundamentos del diseño gráfico y editorial. Comparten principios morfológicos y conceptuales, pero con el avance tecnológico que experimentó la *World Wide Web* en los últimos años, es de esperarse que el diseño web haya tomado un camino diferencial para establecerse por sí mismo. “Diseñar para la web es diferente a diseñar para cualquier otro medio. [...] Sin embargo, el medio atrae a diseñadores de todos los ámbitos de la práctica del diseño: desde la ingeniería y la arquitectura, hasta el diseño gráfico y de productos.” (Boulton, 2009, p. 1).

Diseñar para este medio significa un proceso de planificación, creación y construcción (Kyrnin, 2013). Este mismo, para ser realizado efectivamente, de manera que el producto resultante comunique correctamente sus objetivos, no debe ser tomado a la ligera sino que quien lo encare debe invertir tiempo y conocimientos. El diseño web requiere incorporar interactividad con el usuario – ésta tiene a su vez diferentes niveles – mediante imágenes, audio y video, además de otras nociones como usabilidad, arquitectura de la información, y programación, dependiendo la complejidad que requiera el sitio. “El diseño es una parte clave del diseño web. Esto incluye tanto a los principios de diseño: equilibrio, contraste, énfasis, ritmo, y la unidad y los elementos de diseño: líneas, formas, textura, color, y la dirección.” (Kyrnin, 2013). La combinación de todos estos factores, tanto de diseño y estética, como de construcción y de programación, generan los pilares básicos del diseño web, y son necesarios para desarrollar cualquier página o sitio.

Es pertinente señalar la diferencia entre página y sitio web. En el colectivo imaginario cultural, se entiende por página web lo que es realmente un sitio web. Una página es, en términos generales y de ilustración, la unidad mínima dentro de un sitio. Éste último es el conjunto de páginas web, distribuidas dentro de diferentes directorios. Sin embargo, en la jerga cotidiana cuando se habla de página web, a menos que sea un caso específico, no se está refiriendo a un archivo o fichero único, sino a la globalidad del sitio. Ejemplificando, se toma el sitio educativo de la W3C (*World Wide Web Consortium*,

consorcio que desarrolla los estándares web), el *W3C Schools*: <http://www.w3schools.com/>. Éste es el sitio web, ya que cumple con una ubicación en la WWW apuntada por el nombre de dominio a un servidor específico. Allí se alojan una cantidad variable de páginas web, que en su conjunto e interacción forman la globalidad del sitio. Una página web dentro del sitio de el *W3C Schools* es http://www.w3schools.com/html/html_examples.asp. No necesariamente una página tiene que tener extensión .html, sino que, dependiendo de las características del servidor en el cual esté alojado, puede variar. Sin embargo, estos archivos, en la mayoría de los casos, son codificadas como cualquier archivo HTML, ya que es la manera que tiene el navegador de interpretar una página web. Dentro de un sitio se pueden encontrar otras clases de recursos, como imágenes, videos, entre otros, que son accesibles de la misma manera que lo es una página web.

1.1. ¿Qué hay dentro de un sitio web?

A la hora de llevar a cabo un desarrollo para la Web, es imprescindible conocer y manejar, como mínimo, las tecnologías básicas y fundamentales que permiten generar una interfaz visual en un navegador. Partiendo desde la simplicidad de una única página web, sin ningún tipo de complejidad interactiva, para su creación se requiere al menos el uso de del lenguaje HTML.

A continuación se desarrollarán las tecnologías elementales en el diseño web, desde el punto de vista del desarrollo. Se asume que el lector se encuentra familiarizado con herramientas de diseño gráfico que le permitirán confeccionar imágenes y gráficos utilizables dentro de una página web.

Para poder utilizar estas tecnologías únicamente es necesario, en una primera instancia, un editor de texto como lo es el Bloc de Notas de *Windows* – o cualquier editor o procesador que maneje texto plano y permita guardar archivos en diferentes formatos – y un navegador web. No es específicamente necesario utilizar un *software* de edición de

páginas web visual, como *Adobe Dreamweaver*, aunque éstos si son recomendados para quien no tenga interés en aprender código y desee trabajar únicamente desde el punto de vista de la interfaz. Esta clase de programas utilizan el sistema WYSIWYG (abreviación de *What you see is what you get*, es decir, lo que ves es lo que obtienes), los cuales al trabajar desde el *cómo quedaría*, no producen páginas codificadas de manera correcta, sino que generan lo que se conoce como *código basura*. Por lo tanto, éste no es el objetivo del desarrollo de este PG, y tampoco debería serlo el de quien persiga el diseño web como campo de aplicación profesional.

1.1.1. El almacén: HTML

HyperText Markup Language, o su abreviación, HTML, es el lenguaje que estructura el contenido de cualquier página web y le da formato semántico y en menor medida, visual, al contenido. "HTML es el lenguaje de publicación de la World Wide Web." (*World Wide Web Consortium*, 2013b).

La última versión estandarizada por la W3C es HTML 4.01, la cual fue lanzada en diciembre de 2004. Aunque aún su especificación no ha sido completada, HTML5 es la versión en la que los desarrolladores trabajan actualmente, ya que todos los navegadores modernos soportan la mayoría de sus módulos.

HTML es un lenguaje que se conoce como de marcado, ya que envuelve el contenido en marcas, o etiquetas, las cuales indican al navegador la manera en que éste se debe *renderear* en pantalla.

Se reconocen las etiquetas en un documento HTML, ya que son delimitadas por el signo menor que y mayor que (<>), y son predefinidas, es decir, son establecidas según la estandarización vigente. Las etiquetas encierran un contenido, por lo tanto se cierran al final del mismo. Una etiqueta con su contenido adentro genera lo que se conoce como elemento. La diferencia entre la etiqueta de apertura y la de cierre, es que esta última agrega una barra diagonal luego del signo menor que.

El siguiente ejemplo ilustra un texto con formato de negrita: `Texto a resaltar`. Como se puede observar, la frase es el contenido y se encuentra encapsulada entre una etiqueta de apertura y otra de clausura, las cuales se evidencian por los signos que las separan y el nombre de la misma que es auto-referencial, por su nombre en inglés. Hay, sin embargo, etiquetas que no guardan contenido y se cierran en sí mismas, como la etiqueta de imagen o de salto de línea: `
`. Se conocen como elementos vacíos.

Las etiquetas pueden aceptar atributos, los cuales también son predefinidos por la estandarización vigente. Éstos son siempre declarados dentro de la etiqueta de apertura. Los atributos son declarados en pares de manera que al nombre de atributo le corresponde un valor, el cual puede ser arbitrario o pre-establecido.

El siguiente es un ejemplo de como se establecería un hipervínculo: `Google`. En el ejemplo, dentro de la etiqueta `<a>` se pueden distinguir dos atributos diferentes: el atributo *href* establece el destino del hipervínculo – valor arbitrario – y el atributo *target*, el cual indica de qué manera debe abrirse ese enlace: dentro del mismo marco, en la misma ventana o en una nueva. Éste ejemplo tiene como valor `_blank`, que abre el vínculo en una ventana independiente del navegador.

Las etiquetas pueden ser anidadas, es decir, una etiqueta puede contener en su contenido otras etiquetas adentro suyo y así sucesivamente. En términos generales, una página web clásica contiene por lo menos tres etiquetas diferentes, que son los cimientos de la misma, y sólo pueden ser utilizadas una vez por documento. La etiqueta `<html>` envuelve a todas las demás y no puede estar dentro de ninguna otra etiqueta, por eso suele ser la primera y última etiqueta en un documento. Se considera que suele ser la primera ya que en muchos de los casos antes de la apertura de `<html>` se coloca una declaración que define el tipo de documento, el *doctype*.

En el mismo nivel, dentro de <html> hay dos etiquetas que separan al documento en dos bloques: <head> y <body>. <head> contiene una serie de etiquetas específicas que definen información global del documento. Dentro de este bloque se coloca la etiqueta de título, la de llamada a hojas de estilos CSS, la de llamada a *scripts* de *Javascript*, y etiquetas que serán utilizadas por los motores de búsquedas: palabras relacionadas, autores, contenido, entre otras. La etiqueta <body> contiene lo que el navegador interpretará como visible en pantalla. Comienza al cierre de la etiqueta <head> y cierra justo antes del cierre de <html>. Aquí pueden utilizarse todas las etiquetas, salvo las que pertenecen dentro del <head>, y por supuesto <html>, <head> y <body>, que no corresponden y no pueden volver a ser utilizadas. Entonces, una estructura clásica, utilizando las etiquetas que ya han sido presentadas y algunas nuevas para ilustrar el ejemplo puede visualizarse en la Figura 1: *Anatomía de una página web*, pág. 94, anexo de imágenes seleccionadas.

1.1.2. El maquillaje: CSS

CSS es la abreviatura de *Cascading StyleSheets*, cuya traducción es hojas de estilo en cascada. Éste es un lenguaje que permite declarar instrucciones de estilo visual para el navegador, estando relacionado a un documento HTML, o a varios. La última versión estandarizada por la W3C es la 2.1, aunque es posible utilizar módulos de la versión 3.0 (CSS3) si se dispone de un navegador moderno que la soporte.

Estas declaraciones en lenguaje CSS pueden estar definidas en un documento separado de la página web, con extensión .css, o pueden estar dentro del HTML envueltas en la etiqueta <style>. Si se utiliza un archivo externo, es necesario vincularlo a la página HTML en cuestión mediante una llamada con la etiqueta <link>, la cual se debe colocar en el <head> del mismo. Es recomendable utilizar un archivo CSS aparte, por varios motivos. De esta manera se separa la estructura con su texto y contenido de las declaraciones visuales de estilo: “Mantener estrictamente separados la estructura

(marcado), presentación (estilos), y comportamiento (scripts), y tratar de mantener la interacción entre los tres a un mínimo absoluto” (Google, 2013). Por otro lado permite un nivel de escalabilidad mayor a un proyecto web, ya que un mismo archivo de hoja de estilos puede ser utilizado por la cantidad de páginas web que se desee, facilitando así la edición del mismo; si las declaraciones fuesen repetidas dentro de cada página web, mantenerlas actualizadas en cada uno de los documentos sería una tarea trabajosa.

Las declaraciones CSS son instrucciones que apuntan a dar estilo y/o modificar la manera en que el navegador *rendera* el contenido HTML nativamente. Si bien es posible generar cambios en la interfaz visual desde el HTML mismo, mediante diferentes atributos en los elementos, esta práctica está completamente desacreditada por Google en su *Guía de Estilos para HTML/CSS* (Google, 2013).

Una instrucción en CSS comienza indicando el destino de ese comando. El destino al cual se le aplicarán los cambios, llamados selectores, pueden ser, desde lo más general a lo más particular: elementos, pseudo-elementos, clases o IDs.

Como ya se ha dicho, un elemento es lo que se forma al encerrar un contenido con una etiqueta, aunque también se ha mencionado que pueden existir etiquetas que se cierran en sí mismas. Tomando como ejemplo la etiqueta `<p>` de párrafo, el elemento se construiría de esta manera: `<p>Esto es un párrafo.</p>`. Si con CSS se quisiera apuntar al elemento `<p>`, únicamente se debe escribir el nombre de ese elemento y apertura de llaves: `p { }`. Esta instrucción es válida para todas las apariciones de elementos `<p>` en todas las páginas web que compartan la misma hoja de estilo. Es recomendable, por este motivo, apuntar directamente a elementos para hacer cambios genéricos y no modificaciones particulares.

A su vez, se puede encontrar un tipo de selectores más específico que sirve para apuntar a elementos que, dentro del marcado, son en realidad inexistentes. Se los denomina pseudo-elementos. Sirven para seleccionar elementos virtuales, como *before* y *after*, que crean contenido antes y/o después de una etiqueta, como también para hacer

selecciones delimitadas dentro de un elemento, con *first-line* o *first-child*. Su uso más común es, probablemente, la selección de diferentes estados, como *hover* que apunta a cuando se coloca el *mouse* por encima de un elemento en pantalla. En un documento CSS se hace referencia a los pseudo-elementos con el símbolo de dos puntos, antecedido por el selector al que se intenta aplicar. Por ejemplo, para darle estilos a un hipervínculo únicamente cuando se le pasa el *mouse* encima, se utiliza `a:hover { }`, siendo `a` el selector de todos los elementos de enlace, y *hover*, el pseudo-elemento de estado.

Así también otro selector es una clase, o *class*, la cual es un atributo que puede dársele a un elemento HTML con un valor arbitrario. Esta clase puede ser utilizada en cualquier tipo de etiqueta que permita el atributo, a la cantidad de elementos que se considere necesario. El atributo *class* también soporta múltiples valores, es decir, un mismo elemento puede tener una o más clases diferentes. Dentro del HTML se define un elemento con una clase de la siguiente manera: `<p class="copete">Esto es un párrafo con clase copete</p>`. Para apuntar desde CSS a cualquier clase se utiliza un punto antes del nombre de la misma: `.copete { }`. Éstas sirven para elementos que comparten estilos, por más que sean elementos de distintos tipos.

Un ID también es un atributo en un elemento HTML. La diferencia esencial con las clases es que un ID puede ser utilizado una sola vez por página web, y tampoco es posible que este atributo contenga más de un valor. Es, de todos los selectores, el más específico, y debería ser utilizado en ocasiones que realmente lo requieran. Un ejemplo de ID dentro del HTML es: `<p id="intro">Este es un párrafo con ID intro</p>`. En el CSS se declara antecediendo el símbolo numeral, de la siguiente manera: `#intro { }`.

Adentro de cada selector, entre las llaves, se colocan las declaraciones de estilo, indicando las propiedades y el valor que se les desea atribuir. Cada una de ellas se define como propiedad, dos puntos, valor, y debe finalizar con un punto y coma, salvo la última del bloque, para la cual no es necesario. Muchas propiedades poseen una sintaxis propia para definir estos valores, y debe ser respetada para su correcta interpretación por

el navegador. Un ejemplo concreto de cómo realizar una declaración CSS dentro de un bloque de selección es: `width: 200px;`. En la Figura 2: *Declaración de estilo en CSS* se presenta un ejemplo ampliado de cómo puede formatearse un párrafo para que cumpla con las normas de presentación de este PG.

Como se puede apreciar en la declaración de estilos, CSS acepta diferentes tipos de unidades de medidas, tanto absolutas como relativas. En la Tabla 1: *Unidades de medidas absolutas en CSS*, y en la Tabla 2: *Unidades de medidas relativas en CSS* se explica cada tipo de medida.

Ver Figura 2 en pág. 94, y Tabla 1 y Tabla 2 en pág. 95 del anexo de imágenes seleccionadas.

A su vez, los colores se pueden declarar de tres maneras: mediante una palabra clave, ya que algunos valores de colores están relacionados con sus nombres en inglés, llamados colores web; usando el sistema RGB mediante un valor hexadecimal (#123456), o mediante el uso de las notaciones funcionales como `rgb()` o `rgba()`; o utilizando el sistema HSL a través del uso de las notaciones funcionales `hsl()` o `hsla()` (Mozilla, 2014).

Los selectores pueden ser combinados para generar capas más complejas de selección y estilo. Estas declaraciones de propiedades pueden, a su vez, ser superpuestas y autoeliminadas. En caso de repetición de una propiedad para un mismo elemento específico, el navegador tomará como válida la última declarada, o aquella a la que se le anexe el comando *!important* antes de su cierre.

1.1.3. La magia: *Javascript*

Ya se ha hablado de dos de los tres pilares que sostienen el armado de una página web: la estructura y la presentación. El comportamiento, el último de ellos, puede ser pensado como una capa de interactividad. Para poder romper con las bases estáticas de la página, es necesario introducir un lenguaje de programación llamado *Javascript*, también conocido como *ECMAScript*, o su abreviación JS.

Javascript es un lenguaje de programación que funciona del lado del cliente. La diferencia entre que un lenguaje corra del lado del cliente o del lado del servidor, es que en el último caso la petición realizada por el usuario es enviada al servidor para su análisis, y una respuesta es devuelta. Esto generalmente requiere una recarga en la página. Por el contrario, un lenguaje que corre del lado del cliente trabaja desde la máquina y navegador del usuario, llamado el cliente, sin la necesidad de pasar por un servidor, tomando los recursos de ésta para procesar la información. Para profundizar, en el próximo apartado se tratan lenguajes de programación que interactúan con el servidor.

Por lo tanto, “*Javascript* depende de las capacidades y configuraciones del navegador Web” (Niederst Robbins, 2012, p. 459). Es un lenguaje que no necesita más que el *browser* para ser interpretado. Como depende del navegador, quien diseñe una página web debe tener en cuenta que el usuario puede desactivar *Javascript*, y por ende, perder las funcionalidades que mediante este lenguaje se inyecten en ella.

De la misma manera que con las hojas de estilo, el diseñador web puede incluir código de *Javascript* de dos maneras diferentes y simultáneas: haciendo una llamada a un archivo externo, que debe tener extensión de archivo .js mediante la etiqueta <link>, o dentro del documento, dentro de la etiqueta <script>.

Al ser un agregado de interacción, generalmente estas funcionalidades se ejecutan mediante eventos y a demanda. Éstos son llamados que se activan en el momento que el usuario realiza una acción con los elementos de la página, como puede ser por ejemplo un botón. Con *Javascript*, es posible asignarle a diferentes elementos una función específica que corra cuando un determinado evento ocurra.

El lector puede reconocer el uso de *Javascript* en diferentes sitios web dada la interactividad que esta herramienta le proporciona. Un ejemplo frecuente de su uso es la validación de formularios. En la Figura 3 se ilustra el caso cuando un usuario completa uno, e ingresa una dirección de correo electrónico que no es correcta en su sintáxis – por

lo tanto no es válida como tal – e inmediatamente la página devuelve un mensaje de error. Ver pág. 94, anexo de imágenes seleccionadas.

Existe también el caso de un paso más avanzado en validación de formularios, como puede suceder en grandes sitios comerciales que requieran registración por parte del usuario. Es usual que al introducir un nombre de usuario que ya ha sido registrado la página devuelva inmediatamente, sin ningún tipo de participación del sujeto, o sin recarga de la misma, un mensaje que indique el error. Para este caso se utiliza una tecnología llamada *Ajax* que corre en *Javascript* y hace pedidos asincrónicos a un servidor. *Ajax* es una tecnología intermediaria entre lenguajes del lado del cliente con los del lado del servidor.

Dentro del amplio espectro de soluciones que brinda *Javascript*, es posible beneficiarse del uso de librerías. Las librerías son paquetes de funcionalidades modulares enfocadas en que al programador o desarrollador web se le facilite su trabajo, ya que provee de funciones predefinidas listas para utilizar. La mayormente utilizada en la industria es *jQuery*, ya que posee una sintáxis de sencillo aprendizaje y puede ser extendida con *plug-ins*. *jQuery* “hace que cosas como el recorrido de un HTML y su manipulación, control de eventos, animación y Ajax sean mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores” (*jQuery*, 2013). Esta librería también cuenta con una versión para desarrollar sitios web para plataformas móviles, *jQuery Mobile*, la cual es retomada en el cuarto capítulo.

1.1.4. La lógica: lenguajes de programación del lado del servidor

Ya ha sido establecida la diferencia entre lenguajes y programas que trabajan del lado del servidor y los que viven del lado del cliente. *Javascript* pertenece al último caso. No obstante, la mayoría de los lenguajes de programación utilizados en la Web trabajan en interacción con un servidor.

Del lado del servidor se refiere a las operaciones que se realizan por el servidor en una relación de cliente-servidor en redes de computadoras. Un [...] servidor es

un programa de software que se ejecuta en un servidor remoto, accesible desde el ordenador o estación de trabajo local del usuario. Las operaciones se realizan del lado del servidor, ya que requieren acceso a la información o funcionalidad que no está disponible en el cliente, o requieren comportamiento típico que no es fiable cuando se realiza del lado del cliente. Las operaciones del lado del servidor incluyen el procesamiento y almacenamiento de datos de un cliente al servidor, que puede ser visto por un grupo de clientes. (Armada de los Estados Unidos, 2013).

Los lenguajes más utilizados dentro del contexto de la programación para la web son PHP, *Ruby*, *Python*, ASP.NET, entre otros. Una característica que iguala estos lenguajes es que son de tipo interpretado, es decir, no requieren una compilación para ser ejecutados. A su vez, suelen interactuar con bases de datos, en donde se almacena información de manera permanente.

Es pertinente introducir al lector en el conocimiento básico de estas tecnologías ya que es posible, mediante por ejemplo PHP, reconocer desde dónde accede el usuario a la página web en cuestión, entre otros datos. Esta información puede ser de suma importancia a la hora de planificar la puesta en marcha del sitio web en dispositivos móviles, dándole al desarrollador valiosas herramientas para conocer al usuario del sitio y poder proveerle de la mejor experiencia de navegación posible.

1.2. Surfeando la Web: los navegadores

“Un navegador es un programa en tu computador que te permite visitar sitios web” (Toff, 2009). Esta herramienta es la puerta de entrada a la web, siempre y cuando el dispositivo desde el cual se acceda posea conexión a Internet. El navegador es el *software* que resuelve y traduce en pantalla las interfaces gráficas de las páginas web. Al momento de adquirir un dispositivo o máquina de escritorio, el sistema operativo con el que corre, suele venir empaquetado con un navegador predeterminado. Adquirir un nuevo navegador es gratuito y sencillo, ya que el proceso de instalación es intuitivo, considerando que es uno de los *softwares* principales que un nuevo usuario tecnológico es propenso a adquirir.

Actualmente se encuentra disponible un espectro de diferentes soluciones en materia de navegadores web, cada una de ellas adaptada a diferentes tipos de usuarios. Tomando como base las estadísticas recolectadas por la W3C sobre los visitantes a su sitio web, se puede realizar una comparación sobre el uso de navegadores en el transcurso de un año. Los primeros datos fueron recolectados en mayo de 2013, y los más recientes, en junio de 2014. El navegador más utilizado a mayo de 2013 es *Google Chrome*, el cual acapara más de la mitad de los usuarios (52.9%). Es el navegador con mayor crecimiento, considerando que apareció en el mercado a mediados del año 2008. Detrás de éste, y en caída, viene *Mozilla Firefox*, con el 27.7% (World Wide Web Consortium, 2013a). Un año más tarde, los primeros puestos se mantienen, pero con nuevos valores que soportan la idea de crecimiento de uno y decadencia del otro: *Chrome* acapara el 59.3% de los usuarios, mientras que *Mozilla Firefox* bajó al 25.1% (World Wide Web Consortium, 2014).

Otros navegadores populares son *Internet Explorer*, exclusivo para plataformas que corren con el sistema operativo *Windows*; *Safari*, navegador predeterminado en sistemas operativos de *Apple*, tanto en máquinas de escritorio como en sus dispositivos móviles y *Opera*.

La elección del navegador queda a cargo del usuario, aunque existe la probabilidad que un porcentaje elevado de las personas utilicen el navegador instalado por defecto en su sistema operativo. Si se habla de *Windows*, el navegador de la compañía es *Internet Explorer*, y en *Mac OS X* e *iOS* – sistemas operativos de terminales comercializadas por *Apple* – el navegador predeterminado es *Safari*.

Quien emprenda tareas de diseño web debería tener en cuenta que tanto *Mozilla Firefox* como *Google Chrome* ofrecen herramientas nativas que facilitan la creación y desarrollo web. Mediante el uso de estos instrumentos es posible modificar en tiempo real cualquier página web desde su código fuente de manera local, lo cual significa que estos cambios son visibles únicamente para quien los haga, y los mismos no son publicados, además de

varias funcionalidades que permiten mejorar el flujo de trabajo de manera eficiente. Por ejemplo, al poder modificar reglas CSS en tiempo real, se ahorra el trabajo de modificar el archivo, guardar los cambios y recargar la página para ver el resultado; con este tipo de herramientas se puede modificar en vivo, ya que el navegador interpreta los cambios al momento de hacerlos.

A su vez, es fundamental hacer *tests* del diseño desarrollado en la mayor cantidad de navegadores como sea posible. Considerando que no todos los usuarios ingresan a la Web de la misma manera, y que cada navegador tiene sus propias reglas sobre cómo interpretar en pantalla cada elemento, es absolutamente necesario revisar que el sitio web funcione correctamente por lo menos en las versiones estables de los programas mencionados anteriormente.

1.3. Navegantes satisfechos: la experiencia de usuario

El concepto de experiencia de usuario, comúnmente abreviado UX por *user experience*, no es únicamente aplicado en el campo del diseño web, o al campo del diseño en términos globales. La experiencia de usuario se destina a cualquier área en la que se requiera de la interacción de un usuario con un producto determinado. La experiencia de usuario es la sensación que se genera en un sujeto determinado al interactuar con un producto bajo circunstancias particulares (Arhippainen y Tähti, 2003). Esta experiencia puede derivar en diferentes tipos de respuestas, siendo la más positiva posible la que, quien genera un producto, busca de sus usuarios. Esta respuesta es lo que marcará al usuario hacia futuro respecto a la decisión de volver a consumir el producto ofrecido.

“Los factores clave que contribuyen a la calidad de la experiencia de los usuarios del producto son la facilidad de aprendizaje, la usabilidad, utilidad y el atractivo estético.” (Gabriel-Petit, 2013). Más adelante se explicará también el concepto de accesibilidad, que a pesar de que no fue mencionado por Gabriel-Petit, es de suma importancia en relación a lo que este proyecto trata de demostrar, siempre hablando de la experiencia de

usuario aplicada al campo del diseño web. La facilidad de aprendizaje refiere al nivel de intuición que el producto debe tener para ahorrar al usuario un proceso de entendimiento que, si bien es necesario, si no se encuentra optimizado para obtener una comprensión efectiva en el menor tiempo posible, puede generar un sentimiento negativo hacia el producto. Un concepto similar es el de la usabilidad, que se desprende del aprendizaje. Si el producto es excesivamente complicado de utilizar puede disminuir la frecuencia de uso por parte del usuario. Este concepto se retoma más adelante. El atractivo estético se liga en cierta medida con la decisión de diseño en el producto; cuanta mayor partida estética sea aplicada en el proceso de elaboración al producto, en términos generales, mayor será la atracción por parte del usuario hacia el mismo.

La globalidad, o suma de las experiencias particulares que generen estos factores, crearán en el usuario una valoración específica referente al producto. Es válido aclarar que no siempre es positiva, aunque se debe procurar que así lo sea. Sino, se estaría fallando como diseñador y a su vez comunicador, si se enfoca desde el punto de vista que compete a este trabajo.

Para poder desarrollar una buena experiencia de usuario desde el punto de vista del diseño, es imprescindible pensar en términos de diseño centrado en el usuario. El diseño centrado en el usuario, abreviado DCU, o UCD en inglés por *user-centered design*, se basa en realmente conocer al usuario, para así poder diseñar concretamente para él y proveerle de una experiencia satisfactoria. “El objetivo final del DCU es, por tanto, lograr la satisfacción de las necesidades de todos sus usuarios potenciales, adaptar la tecnología utilizada a sus expectativas y crear interfaces que faciliten la consecución de sus objetivos” (Hassan Montero y Ortega Santamaría, 2009).

1.3.1. Facilitando caminos mediante la usabilidad

Usabilidad es un concepto que se desprende inherentemente de la globalidad de la experiencia de usuario. Éste refiere entonces a “la medida en que un producto puede ser

usado por determinados usuarios para lograr los objetivos especificados con efectividad, eficiencia y satisfacción en un contexto de uso específico” (*International Organization for Standardization*, 1998).

Para poder llevar a cabo un correcto diseño centrado en el usuario y que responda a las buenas prácticas de la usabilidad, es necesario realmente conocer al usuario. No es el mismo usuario el que, generalmente hablando, visite un sitio de vestidos de novia desde una *laptop* a quien navegue el sitio oficial de la NBA desde un teléfono celular. Cambia el tipo de usuario, sus costumbres, su locación, su edad, su bagaje cultural, entre muchos otros factores, que también incluyen al contexto del usuario, ya sea el dispositivo utilizado y/o su navegador. Es necesario conducir una investigación previa mediante herramientas analíticas que permitan descubrir el *target* del producto en profundidad.

Una vez finalizado todo el proceso de creación del mismo, es necesario someterlo a diferentes evaluaciones. Es muy factible pedirle a diferentes tipos de potenciales usuarios que naveguen el sitio en presencia de un especialista en UX y que éste recolecte información sobre cómo es realmente usado el sitio web: si tal objeto funciona en su contexto, si el menú de navegación es confuso de comprender, entre otras cuestiones. No obstante, la prueba real del análisis de usabilidad se puede medir una vez lanzado el producto, con las herramientas analíticas anteriormente mencionadas; siendo la más popular de ellas *Google Analytics*.

Después de todo, la usabilidad en realidad sólo significa asegurarse de que algo funciona bien: que una persona con la capacidad y la experiencia media (o incluso por debajo del promedio) puedan usar el objeto – sea este un sitio Web, un avión de combate, o una puerta giratoria – para su objetivo previsto sin que se sienta irremediabilmente frustrada. (Krug, 2000, p. 5)

Cuando se trate específicamente el caso del diseño web para dispositivos móviles en los próximos capítulos, se hablará de la aplicación de los conceptos de usabilidad aquí descritos para fomentar la mejor experiencia de usuario posible en un proyecto web.

1.3.2. Accesibilidad para todos y todas

En términos generales se denomina así al factor que facilita que todas las personas puedan acceder a determinadas situaciones, contextos, contenidos, entre otros, sin diferenciación de condiciones sociales y/o físicas, derribando así barreras culturales, y a su vez generando inclusión y democratización de las oportunidades para todos los miembros de una sociedad, por más que se encuentran disminuidos por diferentes razones.

Dentro del ámbito del diseño web, se puede definir de la siguiente manera:

La accesibilidad a la Web y a Internet en general (medio electrónico), se refiere al conjunto de elementos que facilitan el acceso a la información web de todas las personas en igualdad de condiciones, y ello independientemente de la tecnología que utilicen (ordenador, PDA, teléfono y otros) y de la discapacidad del usuario (física, psíquica, sensorial y otras). (Lara Navarra y Martínez Usero, 2006, p.10).

Es importante mencionar que existe en cualquier circunstancia un grupo de usuarios que se ven impedidos a obtener una experiencia de usuario máxima dentro del mismo contexto que otro tipo de sujetos. Como mencionan Lara Navarra y Martínez Usero, los factores que separan a este grupo de usuarios del resto se desprenden o de la discapacidad, en cualquiera de sus formas, del usuario en sí, o de la tecnología con la que acceden a la Web.

Este último caso es el que interesa al desarrollo de este Proyecto. El avance de la tecnología y las diferentes puertas de acceso a la Web han limitado a muchos usuarios a desarrollar una experiencia óptima en relación a la que obtendrían en un contexto de acceso como lo es una computadora de escritorio. Aquellos que naveguen frecuentemente la Web en dispositivos con características fuera de los estándares de escritorio probablemente hayan visto restringidas sus experiencias como usuarios en alguna ocasión, y se hayan sentido disminuidos frente al resto. La experiencia negativa por dificultades de accesibilidad puede ser revertida si se aplican serias medidas de consideración hacia todo tipo de usuarios, o al menos abarcando la mayoría de las posibles situaciones en la que éstos estén inmersos.

Capítulo 2: La Web en todos lados

En términos generales, cualquier sitio web se construye en base a las tecnologías introducidas en el primer capítulo. Entendiendo y aplicando los conceptos presentados, el lector puede generar una página básica. El espectro del diseño web es mucho más amplio que lo presentado aquí, y la concepción de un producto de calidad depende de la creatividad y la correcta utilización de las tecnologías dentro de las buenas prácticas por parte cada autor. Se entiende por tales al seguimiento de los estándares de la W3C y otras convenciones, como la *Guía de Estilos de HTML/CSS* de Google.

La primera página web fue lanzada en diciembre de 1990, por Tim Berners-Lee, el creador de la *World Wide Web* (CERN, 2014). Desde aquel momento, Internet no ha cesado de crecer y la Web ha ido evolucionando permanentemente.

El diseño de interfaces gráficas para la Web se basa en el diseño gráfico editorial, ya que es parte de la naturaleza humana reproducir lo que está comprobado que funciona. Los seres humanos “adoptamos modelos mentales familiares. Optamos por lo seguro y familiar sobre lo nuevo y riesgoso” (Kadlec, 2012, p. 3). Repetir aquel paradigma de comunicación visual puede ser suficiente, pero no aprovecha el hecho de que las herramientas brindadas por este medio digital son inmensamente superiores a las anteriormente conocidas. “El problema es que basar el futuro en experiencias pasadas limita la evolución de las nuevas ideas y medios de comunicación” (2012, p. 3).

El diseño web *de escritorio*, por su término en inglés *desktop*, refiere a una interfaz gráfica optimizada para ser vista en monitores estándares de computadoras de escritorio y a lo sumo en computadoras portátiles. Ambas modalidades de visualización suelen portar resoluciones de pantallas similares, siendo las más clásicas y comúnmente usadas 1024x768 píxeles (monitor de 15 pulgadas), 1366x768px (monitores de 18,5 pulgadas y portátiles de 14 y 15,6 pulgadas), 1280x800 (portátil de 14 pulgadas), entre otras (StatCounter, 2014). Este modelo de visualización sigue siendo el estándar, pero con la introducción de dispositivos móviles con capacidad de navegar la Web, se abre un

paradigma distinto que cuenta con nuevas reglas y lineamientos. “En los últimos años, el uso de Internet móvil ha aumentado considerablemente al punto que los desarrolladores y diseñadores web ya no pueden darse el lujo de ignorarlo” (Raasch, 2012).

2.1. La red al alcance de la mano: dispositivos móviles

La diversidad de aparatos que pueden ser clasificados como dispositivos portátiles para el contexto de este ensayo es amplia. No sólo se consideran las plataformas dentro del ámbito de la telefonía móvil – celulares – sino que con el correr de la última década y media han surgido nuevos instrumentos con funcionalidades equivalentes pero que, en algunos casos, prescinden del factor telefónico.

El ecosistema móvil, como lo define Maximiliano Firtman, comprende un sistema de dispositivos que comparten una serie de características tales como: que son portables, son personales, son acompañantes (es decir, se llevan encima), son fáciles y rápidos de utilizar y poseen capacidad de conexión, ya sea permanente o limitada (2012). Es pertinente agregar que también son inalámbricos, a pesar que Firtman no lo considere en su clasificación.

En la actualidad, todos los dispositivos móviles modernos, que se adapten a la clasificación de Firtman, poseen características propias de un ordenador de escritorio, con gran poder de procesamiento y funciones interactivas. Pero la funcionalidad o característica que hermana a todas estas plataformas, la más importante a lo que este Ensayo respecta, es la posibilidad de acceso a la Web a través de un navegador. Como ya se ha desarrollado en el primer capítulo, éste es la puerta de entrada a la Web y la pieza de *software* encargada de procesar y mostrar en pantalla las interfaces gráficas de usuario, término comunmente abreviado GUI por su traducción al inglés – *graphic user interface*.

En este PG se hace una distinción entre teléfonos inteligentes, comunmente llamados *smartphones*, y *tablets*. El resto de los dispositivos que recaen dentro del ecosistema móvil también son definidos al final de este subcapítulo.

2.1.1. Teléfonos pensantes: *smartphones*

Para poder definir qué es un *smartphone*, primero es necesario recorrer brevemente la evolución de los teléfonos móviles, y los diferentes tipos de celulares que hoy pueden encontrarse en el mercado. Para hacer estas distinciones se toma como base la categorización planteada por Firtman en su libro *Programming the mobile web* (2013).

Los teléfonos celulares básicos que todavía hoy pueden ser encontrados en algunos, aunque ya pocos mercados, especialmente los emergentes, de bajo costo y con limitadas prestaciones, son el punto de partida evolutivo hacia los teléfonos de alta tecnología actuales. Estos dispositivos poseen únicamente capacidades fundamentales, como la posibilidad de realizar y recibir llamados, y soporte para SMS (*short message service*). No incluyen la capacidad de conexión a Internet, ni reproductor de música, ni cámara fotográfica incorporada, las cuales son funcionalidades estándar en todos los teléfonos celulares actuales. Dado que no poseen conectividad, son excluidos del ecosistema móvil que se desarrolla en este trabajo. Si bien siguen siendo fabricados por algunas compañías, el futuro de estos dispositivos es finito. “Esto se debe a que el ecosistema móvil [...] quiere ofrecer servicios a los usuarios, y un navegador es el punto de entrada” (Firtman, 2013, p. 4).

La siguiente generación de teléfonos celulares son dispositivos conocidos como de gama baja. Estos aparatos cuentan con funcionalidades comunicativas básicas e incluyen cámara de fotos incorporada, reproductor de música, conexión a Internet, y por consecuente, un navegador. Todas estas prestaciones son aún, a los estándares actuales, básicas. No suelen a su vez tener un sistema operativo masivo, como por ejemplo *Android*, sino que poseen un sistema operativo perteneciente al fabricante, como

por ejemplo *Symbian* de *Nokia*. Si bien estos dispositivos entran en la categorización del ecosistema móvil, el usuario característico de celulares de gama baja no suele ser un ávido consumidor de contenidos web.

Más avanzados que los celulares de gama baja son los de gama media y gama alta. Firtman los define como teléfonos sociales (2013). Tal como los celulares de gama baja, no tienen un sistema operativo masivo y multiplataforma, característica que los distancia de los *smartphones*. Estos teléfonos cuentan con las funcionalidades de los de gama baja pero con prestaciones más avanzadas. En algunos de los casos, especialmente en los de gama alta, también cuentan con pantallas táctiles. En cuanto a conectividad, tienen soporte de tecnología 3G, o datos móviles, lo cual provee al usuario de una experiencia de consumo de contenidos web más rica. Son a su vez teléfonos sociales ya que, en muchos de los modelos, aplicaciones sociales como *Facebook*, *Twitter* o *Whatsapp* vienen pre-instaladas de fábrica, y/o el usuario cuenta con la posibilidad de instalarlas manualmente (2013).

El siguiente escalón evolutivo es la última generación en telefonía celular: los teléfonos inteligentes, mejor conocidos como *smartphones*. Dentro de esta clasificación recaen muchos modelos de celulares, y el término está en constante actualización, ya que lo que era inteligente hace un par de años, hoy ya no lo es.

Un teléfono inteligente, como se define hoy en día, cuenta con un sistema operativo multitarea, un entorno de escritorio completo, un navegador, LAN inalámbrica ([...] WiFi) y soporte de conexión 3G/4G, reproductor de música, y varias de las siguientes características: GPS, acelerómetro, [...] cámara con soporte de video, [...] Bluetooth, [...]. (Firtman, 2013, p. 6)

Los *smartphones* también se separan en terminales de variadas gamas: baja, media y alta. Como ya se ha señalado, lo que unifica a esta clase de dispositivos es que cuentan con un sistema operativo (SO) masivo y multiplataforma. Los más expandidos son *iOS*, de *Apple*, que corre únicamente en terminales de su propia fabricación – es decir, el *iPhone* y el *iPad*, en todos sus modelos –; *Android*, que corre en una gran cantidad de dispositivos de diferentes fabricantes, y que acapara más de un tercio del mercado (*Net*

Market Share, 2014); *Windows Phone*, distribuido por *Microsoft*, principalmente incluido en nuevas terminales *Nokia*; *Blackberry* – aunque sólo sus más recientes modelos pueden ser considerados *smartphones* dada la categorización anteriormente descrita –; entre otros SO no tan extendidos.

Todos estos sistemas operativos ofrecen a los usuarios un navegador web de calidad, y/o la posibilidad de descargar e instalar otro a elección. Los navegadores presentados en el primer capítulo tienen una versión especial y adaptada para terminales portátiles.

Dentro de los dispositivos móviles funcionales a lo que este Ensayo respecta, los *smartphones* son los medios más importantes ya que son el elemento y herramienta primera y fundamental del usuario móvil, más adelante descrito.

2.1.2. Pequeñas computadoras: *tablets*

A pesar que suelen ser concebidas en el imaginario colectivo como un dispositivo nuevo y/o moderno, las *tablets* remontan su invención e inicio evolutivo a la década de los '60 (McLellan, 2014). En su origen se denominaban *pen computers* (computadoras manejadas por lapiceras), y más adelante también recibieron nombres como *touch computers* (computadoras táctiles). Se caracterizan por ser manejadas mediante gestos táctiles, ya sea con la mano o con un lápiz o lapicera que funcione por presión.

La diferencia entre una *tablet* y un *smartphone* moderno se encuentra principalmente en el tamaño del dispositivo. Las primeras se comercializan en tamaños estándar que varían entre las 7 y 11 pulgadas (Firtman, 2013), mientras que los *smartphones* son mucho más pequeños. Hay teléfonos como la gama *Samsung Galaxy Note* que tienen un tamaño excepcional para un teléfono – 5,3 pulgadas –, pero que siguen sin ser clasificados como *tablets* por otra característica esencial entre los dos tipos de dispositivos: las *tablets*, salvo casos específicos, no poseen la función fundamental de los celulares: la capacidad de telefonía.

La revolución de las *tablets* en el mercado actual fue iniciada con la introducción del *iPad* en la escena tecnológica, a principios del año 2010 de la mano de *Apple*. A partir de allí, los fabricantes de celulares e incluso otras compañías de *hardware* incursionaron en el desarrollo de sus propias *tablets* para competir con éste. Aquí nuevamente aparece *Android*, sistema operativo desarrollado por *Google*, el cual es núcleo de muchos de los modelos de *tablets* hoy comercializados. *Windows* también tiene una versión dedicada a estos dispositivos; como también *Blackberry* es el sistema operativo de la *PlayBook*.

2.1.3 ¿Aún más?: otras clases de dispositivos

Existen otros tipos de dispositivos móviles que no recaen directamente en alguna de las dos categorías anteriormente exploradas. Estos aparatos, como las *tablets*, no poseen la funcionalidad de telefonía, por eso Firtman (2013) los llama dispositivos móviles no telefónicos.

Un ejemplo claro es el *iPod Touch*, de *Apple*, un reproductor multimedia que evolucionó desde un simple reproductor de música como lo era el primer *iPod*. Este dispositivo es en casi todos los sentidos lo mismo que un *smartphone iPhone*, salvando la principal diferencia, explícita en el nombre del producto: el teléfono. Si bien no son teléfonos, cumplen con las características planteadas para el ecosistema móvil (Firtman, 2013, p. 7), por lo tanto entran en el espectro de este Ensayo.

Otro clase de dispositivos que puede ser considerada pero que no se ubica cómodamente en las otras dos categorías son los *e-readers*, conocidos también como libros electrónicos. No todos ellos cumplen con las características del ecosistema móvil, pero algunos modelos de las últimas generaciones de estos aparatos poseen un navegador avanzado y posibilidad de conexión a Internet, tanto por *WiFi* como por 3G. Los ejemplos más claros son el *Kindle Fire*, fabricado por *Amazon*, y el *Nook HD*, distribuido por *Barnes & Noble*. Si bien comparten características físicas con las *tablets*, difieren en su función principal.

2.2. El creciente avance de las plataformas móviles

El crecimiento de la tecnología móvil es innegable. No solamente es palpable la evolución tecnológica, sino también el uso de los dispositivos, que crece exponencialmente año tras año.

Cisco Systems, compañía multinacional dedicada a la manufactura y distribución de equipos de telecomunicaciones, elabora periódicamente un informe sobre el tráfico de datos móviles (2014). La información provista en este pronóstico permite realizar un análisis sobre el crecimiento de uso y formar especulaciones sobre el futuro del mercado móvil.

Según este informe, el promedio de uso de smartphones creció un 50% en 2013 (*Cisco Systems*, 2014). Si bien la medición se realiza a través del análisis de datos móviles – conexiones 3G y/o 4G –, sirve para darse una idea global del impacto que tienen los *smartphones* en el mercado actual, y esta reseña apunta tanto a demostrar la migración de usuarios de teléfonos no inteligentes hacia este tipo de tecnología, como el incremento de uso en los usuarios ya efectivos. Es decir, que los individuos ya poseedores de estos dispositivos realizaron un uso más intenso de los mismos. Esto puede darse por una cuestión cultural, la masificación del uso móvil y la necesidad de adaptación a este modelo. El dato concreto indica que “la cantidad media de tráfico por smartphone en el año 2013 fue de 529 MB al mes, frente a los 353 MB al mes en 2012.” (2014).

Esta estadística también aporta información valiosa sobre el uso específico de *tablets*: “En 2013, el número de tablets móviles conectadas aumentó 2,2 veces hasta las 92 millones, y cada tablet genera 2,6 veces más tráfico que el smartphone promedio” (2014). El hecho que solamente en el período de doce meses el número de terminales se haya duplicado, claramente indica que el mercado apunta a este tipo de tecnología, y que aparentemente no hay vuelta atrás hacia el pasado. Los usuarios son cada vez más, y hacen un uso más intenso de estos dispositivos, ya que “[...] el tráfico de datos móviles

por tablet fue de 1.374MB al mes, frente a los 529MB por mes por cada smartphone” (2014).

Quizá la información más clara e ilustrativa de este evidente crecimiento es que “el número de dispositivos móviles conectados superará la población mundial para el año 2014” (2014). Es evidente que este dato no significa que todas las personas del mundo poseerán un dispositivo móvil; estos aparatos siguen estando reservados para ciertas clases sociales de la población, las cuales cuentan con un determinado poder adquisitivo. Pero sí demuestra que disponer de un dispositivo no es exclusivo, sino que se poseen en cantidad y generalmente en variedad; es decir, no es inusual por ejemplo, un individuo dueño de un *smartphone*, una o más *tablets*, un *iPod* y un *Kindle Fire*, todo a la vez.

Estos datos son contundentes y pronostican únicamente un crecimiento continuo, probablemente de manera más acelerada año tras año. Son un indicador que no puede seguir siendo ignorado por diseñadores y desarrolladores web. Estadísticamente está demostrado que es necesario atender las necesidades de este grupo emergente de usuarios. Si ya se está de acuerdo en que es necesario proveer a todos los usuarios de una experiencia de navegación óptima, ¿por qué se sigue relegando e ignorando a la primera minoría, pronta a ser la mayoría en un futuro no muy lejano?

2.3. Un bicho raro: el usuario móvil

Antes de intentar definir al usuario móvil, es necesario entender a qué se denomina usuario en términos generales. Se trata de un individuo que hace uso de un objeto, ya sea un producto, una herramienta, o en el caso que compete a este Proyecto, un sitio web, para obtener de él un resultado o servicio luego de una interacción con el mismo.

Desde el punto de vista tecnológico, éste es quien interviene en el sistema enviándole peticiones para recibir una respuesta esperada. Como ya se ha explicado, la capa entre el usuario y el sistema es la interfaz gráfica en dispositivo por el cual acceda.

Partiendo de esta generalización técnica, es posible entonces diferenciar distintos tipos de usuarios, cuya separación depende de variados factores. Cada individuo puede llegar a tener distintas expectativas en relación al sistema y sus respuestas, condicionadas por una combinación de cuestiones sociodemográficas, como pueden ser género, edad, nacionalidad, profesión, entre otras. Pero estos suelen actuar de manera similar al ser agrupados por razones tecnológicas.

Es decir, no todos los usuarios son iguales, cada uno de ellos tiene características que lo definen y diferencian, pero sus acciones pueden ser predecibles si se toma en cuenta la manera de acceso que hace a la Web. Diferentes individuos pueden buscar o esperar encontrar distintos contenidos, pero su comportamiento colectivo es similar en cuanto a la navegación.

Es aquí donde entran las distinciones de acceso. Una persona que ingresa a la Web desde una computadora de escritorio cuenta con una diferencia fundamental sobre quien accede desde un dispositivo móvil: una computadora es, en la mayoría de los casos, inamovible y el usuario permanece frente a ella enfocando gran parte de su atención a esa interacción. Sin embargo, el usuario móvil muta este paradigma de uso dada la principal característica de los dispositivos con los que accede a la Web: la movilidad (Ballard, 2007). Si bien puede darse el caso en el que el usuario móvil, ingresando desde un aparato con estas características, haga uso estático del mismo tal como lo haría con una *desktop* o *laptop*, es más común y probable que el factor de movimiento se incluya en su experiencia.

Hay otro factor que esta clase de usuario valora: la conveniencia de uso. Al ser portables, y generalmente rápidos, además de poder ser manejados mediante gestos táctiles, en el caso de *smartphones* y *tablets*, es usual que sean la primera opción al momento de acceder a la Web.

El usuario no siempre ha interpretado el entorno de la Web de la misma manera. Con la introducción de diferentes tecnologías a su cotidianidad, entre otras causas, éste ha

sabido moldearse a estos fenómenos e ir mutando a diferentes tipos de usuarios preparados para afrontar distintos tipos de situaciones. Desde el comienzo de la Web, el usuario es quien evolucionó para adaptarse a ella, es momento de que la Web sea quien se adapte al usuario (Luzardo Alliey, 2011).

2.4. Hay una *app* para todo

Actualmente este término es utilizado constantemente para referirse a una de las maneras de interactuar digitalmente con un dispositivo, ya sea móvil o no. Son comunmente llamadas *apps*, por la abreviación del término en inglés, *application*. Pero, ¿qué son realmente?.

Una aplicación es un programa, un *software*, que puede ser muy simple en su desarrollo y/o diseño, como compleja y extensa. Generalmente se piensa en aplicaciones como algo exclusivo de las plataformas móviles, pero éstas se encuentran en todo tipo de dispositivos, mismo en las computadoras, en las cuales son llamadas simplemente programas. Éstas son un sistema en sí mismas, y absolutamente todas presentan una interfaz gráfica, que es con la cual el usuario interactúa.

Para lo que compete a este Ensayo se analizan únicamente las aplicaciones móviles. Dentro de esta clasificación según Raluca Budiu, investigadora del Nielsen Norman Group, se reconocen tres clases de *apps*: nativas, web e híbridas (2013).

2.4.1. Aplicaciones nativas

Este tipo de *app* es lo más similar que se puede encontrar a un clásico programa de computación. Incluso muchos de ellos son portados a aplicaciones para móviles. Una aplicación nativa se instala y reside en el aparato pudiendo ser accedida desde el menú o desde la pantalla principal.

Se caracterizan por estar disponibles para su descarga en los mercados (o *stores*) oficiales de cada sistema operativo, como en el *Google Play* o en el *App Store*, aunque también pueden ser descargadas de sitios no oficiales.

Este tipo de aplicaciones están desarrolladas específicamente para una plataforma, es decir, un tipo de dispositivo con determinado sistema operativo; por ejemplo, no se desarrolla de la misma manera una *app* para un *smarthphone* que corre con *Android* que otra para un *iPad* con *iOS*.

A su vez, las aplicaciones nativas aprovechan funcionalidades propias del aparato, como la cámara, el sistema de posicionamiento (GPS), el teléfono, entre otras, y hacen uso del sistema de notificaciones del mismo. Muchas de ellas, también, incorporan la funcionalidad de ser manejadas mediante gestos táctiles, tanto los estándares del sistema operativo como nuevos (Budiú, 2013). Una ventaja primordial de las nativas es que son accesibles sin la necesidad de estar conectada a la Red, aunque algunas puedan requerir conexión para hacer envíos a un servidor, como puede ser el caso de, por ejemplo, un juego que almacene y conecte los puntajes obtenidos con una cuenta de *Facebook*.

Es este tipo de aplicaciones las que, en el imaginario colectivo, se cree que son las únicas que existen. Sin embargo, el espectro es aún mayor.

2.4.2. Aplicaciones web

Esta clase de *apps* son esencialmente sitios web, ya que son accesibles desde un navegador en un dispositivo móvil, pero con la particularidad de poseer características propias de una aplicación nativa. Para desarrollar este tipo de sitios se utilizan muchas funcionalidades incorporadas en HTML5, que permiten un mayor control sobre algunos complementos del dispositivo desde el que se acceda. “Las aplicaciones web no son aplicaciones reales; en realidad son sitios web que, en muchos sentidos, se ven y se sienten como aplicaciones nativas, pero no se implementan como tal” (Budiú, 2013).

Este tipo de aplicaciones existen ya que son convenientes tanto para los desarrolladores, quienes no se necesitan crear diferentes versiones para cada sistema operativo – dado que las nativas se programan en diferentes lenguajes según la plataforma – sino que con una misma versión en HTML5 pueden servir a un gran rango de dispositivos, como para la gran cantidad de usuarios que por variados motivos deciden no instalar *apps* – por escaso espacio de almacenamiento, desconfianza del origen, evitar el paso de instalación – obteniendo así una experiencia similar con la añadida rapidez. Es pertinente aclarar, que al ser un sitio web, a menos que el mismo esté desarrollado para que almacene datos, muchas aplicaciones de este estilo son únicamente accesibles estando conectado a Internet. Una ventaja fundamental sobre las nativas es que las web pueden ser accedidas desde dispositivos no tan modernos, como los de gama media y gama alta, los cuales prestan conexión a Internet pero no siempre existen *apps* instalables para esa plataforma que sean compatibles.

Las aplicaciones de sitios o redes sociales más importantes generalmente ofrecen tanto aplicaciones nativas como web de su servicio, dado que tratan de abarcar la mayor cantidad de usuarios posibles.

Un ejemplo concreto es *Twitter*. La aplicación nativa de esta red social se distribuye para los mayores sistemas operativos, distinguiendo distintos tipos de dispositivos: *iPhone*, *tablets* con *Android*, *Windows Phone*, *iPad*, entre otros. La aplicación web, sin embargo, es una sola y puede ser accedida desde cualquier *celular*, *tablet*, y hasta computadoras, con sólo ingresar a la siguiente dirección: <https://mobile.twitter.com/>.

El costo de desarrollo para tantas versiones de *apps* diferentes es elevado, que sólo grandes empresas pueden tomar. No obstante, esta clase de aplicación suele ser una adaptación del sistema original, y suele ser más económica de realizar, teniendo en cuenta que sirve a una amplia gama de usuarios. En muchas ocasiones, ambas suelen mantener una línea estética similar, pero, como en el caso de *Twitter*, se demuestra que la versión web suele quedarse atrás en términos de actualizaciones. Esta comparación

de diseño puede visualizarse en la Figura 4: *Aplicaciones nativas vs. web*, en donde además se observa que ambas cumplen las mismas funcionalidades. Ver Figura 4, pág. 96, anexo de imágenes seleccionadas.

A partir del tercer capítulo se analiza de qué manera puede un sitio llegar a transformarse en una aplicación web, utilizando técnicas de adaptabilidad.

2.4.3. Aplicaciones híbridas

Como su nombre lo revela, este tipo de *apps* son una conjunción entre las dos presentadas anteriormente. En esencia son aplicaciones web, ya que son sitios enteramente desarrollados para funcionar en móviles. Pero no son accesibles únicamente desde el navegador, sino que son instalables como las nativas. Si bien no todas, la mayoría pueden conseguirse en los *stores*.

Se les llama híbridas dado que básicamente son aplicaciones web enmarcadas dentro de una nativa. “A menudo, las empresas construyen aplicaciones híbridas como contenedores para una página web existente; de esa manera, esperan obtener presencia en la tienda de aplicaciones, sin tener que gastar un esfuerzo significativo en el desarrollo de una aplicación diferente” (Budi, 2013). Considerando que el desarrollo específico de distintas versiones para cada plataforma es costoso, y que los *stores* son una salida publicitaria masiva, muchas compañías deciden lanzar sus aplicaciones en este formato, ya que obtienen lo mejor de ambos extremos.

Al estar instaladas en el dispositivo, muchas de ellas pueden ser accedidas sin la necesidad de estar conectada a Internet, aunque suelen hacer peticiones asincrónicas a un servidor. A su vez, pueden hacer uso de componentes del mismo, ya que al instalarse generalmente solicitan permisos de uso.

Un caso de este estilo es la *app ¿Cómo llego?* desarrollada por el Gobierno de la Ciudad de Buenos Aires. La misma puede ser descargada desde los *app stores* más populares, y

a su vez se puede acceder desde un navegador ingresando la dirección: <http://comollego.ba.gob.ar/>, para obtener exactamente lo mismo en pantalla.

El marco nativo en una aplicación híbrida sirve de navegador para *renderear* en pantalla el sitio web.

2.5. Cuando el entorno móvil marca fronteras

Este paradigma de visualización de una página web se caracteriza por restringir a los diseñadores a un apretado marco de opciones al momento de plasmar su creatividad en una interfaz funcional y efectiva. Estas limitaciones suelen ser razones suficientes para que muchos de ellos decidan no tomarse la molestia de plantear la necesidad de un abordaje al medio *mobile*. La complejidad en la correcta aplicación de portabilidad en los desarrollos para la web es otra razón que hace huir a algunos diseñadores. “Portabilidad refiere a la capacidad de una aplicación móvil para ser utilizada en varios dispositivos con diferente hardware, software y plataformas” (Firtman, 2013, p. 51).

No obstante, y afortunadamente, hay muchos otros que adoptan estos condicionamientos como desafíos para el desarrollo de la experiencia profesional-personal y el legado de aportes para la extensa comunidad de creadores web. Es de esta manera como muchas de las técnicas y acercamientos teóricos explicados en el tercero y cuarto capítulo fueron concebidas, las cuales son directamente aplicadas más adelante en los casos de estudio proyectados para la última sección de este Ensayo.

El entorno móvil plantea condicionamientos ligados básicamente a sus prestaciones tecnológicas y de tamaño. Estos mismos puntos, en otro plano, también generan ventajas, los cuales han sido desarrollados anteriormente.

A continuación se delinean una serie de cuestiones que deben ser tenidas seriamente en cuenta a la hora de plantear y diseñar una propuesta de interfaz gráfica y la manera en que ésta interactuará con el dispositivo que la muestra.

2.5.1. *Small, large y XL: resoluciones*

Tal como las pantallas de las computadoras de escritorio y las de *laptops* presentan una serie de resoluciones casi estándares, los dispositivos móviles también cuentan con un sistema de medidas de resoluciones, que si bien son muy variadas, pueden agruparse y/o simplificarse en una serie valores comunes.

Cuando se genera un diseño para un sitio *desktop*, es decir, pensado para su visualización en monitores fijos, se toma como base un ancho seguro mínimo de aproximadamente 960 píxeles, ya que, en promedio, las resoluciones más utilizadas van desde 1024x768 a 1280x800 píxeles – estos valores deben leerse como ancho por alto –. A éstos deben descontársele horizontalmente algunos píxeles que son ocupados por elementos de la interfaz gráfica de los navegadores, como por ejemplo la barra de *scroll*, los cuales quitan espacio en el área real de *renderización* de la página web. Es así como se obtiene este número, que es considerado por algunos estudiosos de la materia como una proporción *mágica*, para exhibir correctamente el contenido deseado en pantalla. A medida que la tecnología avanza, los monitores crecen y/o presentan mayor densidad de píxeles por pulgadas (abreviado PPP, o PPI por sus siglas en inglés, *pixels per inch*) como las pantallas retina comercializadas por *Apple* en los nuevos modelos de sus *MacBooks*, y algunos de sus dispositivos móviles. Es por esto que las resoluciones en *desktops* aumentan y un diseño basado en el ancho de 960 píxeles resulta insuficiente. Actualmente, muchos sistemas de grillas y *frameworks*, los cuales son desarrollados en el cuarto capítulo, establecen su ancho base en números mayores, como 1170 píxeles. Como tantas otras variables, al momento de trasladarse al paradigma de visualización en dispositivos móviles, estos cálculos y aproximaciones pierden su validez. Físicamente, una de las características principales de estas terminales es su tamaño. Suelen ser por lo menos 1/5, en el caso de *smartphones*, o 1/2 en el caso de las *tablets*, aproximadamente, del tamaño de los monitores convencionales por lo cual su resolución en pantalla también debe ser menor.

Al haber en existencia en el mercado centenares de modelos, cada uno con sus medidas específicas y resoluciones variadas – ya que el hecho que dos dispositivos que cuenten con la misma medida física no garantiza que ambos porten la misma resolución –, es meramente imposible tratar de cubrir todas las variables y combinaciones posibles para satisfacer a todos los usuarios por igual.

Es por esta razón que han sido trazadas, por convención, una serie de medidas para definir patrones de estandarización al momento de diseñar, y posteriormente desarrollar una interfaz web para móviles. Es del interés de un diseñador, primordialmente, conocer el ancho ya que, en la mayoría de los casos, se construyen sitios verticales con un ancho fijo y un alto variable que depende del contenido de cada página. Algunos de estos valores, tomando como base los mencionados por Firtman (2013) en *Programming the mobile web* son, en orden ascendente, 320px para celulares más antiguos con resoluciones reducidas, 480px también para celulares y algunos *smartphones*, 768px para *smartphones*, y 1024px y 1280px para *tablets*, por mencionar los más utilizados (2013).

Es fundamental que el lector comprenda y memorice estos datos ya que sientan las bases sobre las cuales deberá trabajar al momento de definir la estructura de su diseño para dispositivos móviles.

Los valores presentados son puestos en práctica cuando se desarrolle el tema de *media queries* y su aplicación en el diseño web adaptable, en el cuarto capítulo de este Ensayo.

2.5.2. ¿Vertical o apaisado?: orientaciones de pantalla

Además de tomar en consideración las resoluciones de pantallas, es necesario tener en cuenta que muchos móviles poseen la capacidad de ser rotados a conformidad del usuario. Al girar el dispositivo, el contenido mostrado en pantalla, y su contenedor, muta para adaptarse a la nueva manera de presentación. Esto es posible gracias al acelerómetro incorporado en el mismo.

Aún cuando esta funcionalidad se encuentra disponible en un gran porcentaje de la totalidad de los modelos, es pertinente recordar que el usuario siempre posee la opción de desactivarla.

A menos que el dispositivo sea una *tablet* o a lo mejor un lector de libros electrónicos, los cuales suelen ser utilizados tanto horizontal como verticalmente casi por igual, en el resto de los casos el usuario suele utilizar el aparato en orientación vertical mayormente. Este comportamiento puede deberse tanto a la costumbre de los usuarios, a la comodidad a la hora de hacer llamadas, como al hecho de que generalmente se publicitan comercialmente en esta posición.

Muchas aplicaciones nativas como web aprovechan esta característica brindada para optimizar el diseño de su información. En la Figura 5: *Orientaciones de pantalla*, se toma como ejemplo una aplicación nativa, el juego *Candy Crush Saga*, dado que presenta distintas estructuras dependiendo de la posición en la que se visualice el dispositivo. Ver pág. 96, anexo de imágenes seleccionadas.

Además de llamársele por la posición a la que apunta la orientación, vertical u horizontal, suelen ser denominadas en retrato, por su término en inglés *portrait*, o paisaje, por *landscape*, respectivamente.

El acelerómetro integrado en nuestros dispositivos inteligentes se suele utilizar para alinear la pantalla en función de la orientación del dispositivo, es decir, cuando se cambia entre los modos de retrato y paisaje. Esta capacidad proporciona grandes oportunidades para crear mejores experiencias de usuario, ya que ofrece una disposición adicional con un simple giro de un dispositivo, y sin necesidad de pulsar ningún botón. (Itzkovitch, 2012)

Al generarse una rotación, ya no se cuenta con la misma resolución. Los valores son iguales, pero invertidos. Es decir, la cantidad de píxeles que antes representaban el alto de la pantalla, ahora son funcionales al ancho. Por esta razón al momento de diseñar, si se desea plantear una interfaz adicional considerando orientaciones, es imprescindible tener en cuenta ambos valores, ya que pueden ser intercambiables con un simple giro.

Iitzkovitch (2012) a su vez identifica cuatro patrones de diseño funcionales a la variación en la posición de la pantalla. Estos sirven a la hora de desarrollar una óptima estrategia de experiencia de usuario ya que aportan lineamientos que pueden ser plasmados a cualquier aplicación.

El primero, diseño fluido, es el mayoritario dada la facilidad de empleo. Consiste básicamente y simplemente en que los elementos de la interfaz se reubiquen y adapten según la posición del dispositivo, como es el caso ilustrado en la Figura 5: *Orientaciones de pantalla*, anteriormente mencionada.

El siguiente patrón es el extendido. Este diseño comparte principios con su antecesor, ya que también se adapta y reubica sus componentes, pero con la particularidad que hace uso y desuso de los mismos al esconderlos, expandirlos y/o modificarlos según sea acorde.

Ya en un plano más complejo se define el diseño complementario, el cual establece que al girar el aparato, la pantalla mute y provea una visualización auxiliar de la misma información. Iitzkovitch da como ejemplo una “[...] aplicación financiera móvil que muestre datos en el modo de retrato predeterminado y proporcione un gráfico visual cuando el usuario gira al modo horizontal. Las orientaciones muestran datos similares o complementarios y dependen unos de otros” (2012).

Por último, el patrón de diseño continuo plantea la generación de una pantalla secundaria con una interfaz completamente diferente al girar el dispositivo, que provea de información distinta, pero complementaria a la visualizada en la anterior.

Los últimos dos ejemplos son considerablemente complicados, y no es común encontrarlos en el campo real de aplicación, si bien existen. Sin embargo, los dos primeros sirven y se aplican no solamente en *apps* sino también en simples sitios y páginas web. Es la conjunción de ambos patrones, además de otros factores, los que delinean el concepto de diseño web responsivo, el cual es abarcado en su totalidad en el próximo capítulo de este Ensayo.

2.5.3. Cada navegador con sus reglas

A pesar que en la actualidad la mayoría de los navegadores en su versión *mobile* portan con niveles de soportes entre aceptables a buenos en cuanto a tecnologías de desarrollo para esta clase de dispositivos, es aún pertinente verificar que las funcionalidades que se quieran aplicar estén disponibles, antes de ponerlas en práctica.

Existe la posibilidad que, al momento de diseñar una interfaz, se desee utilizar una funcionalidad propia de HTML5, o CSS3, tecnologías que aún no se encuentran completamente estandarizadas. Es por esta razón que no todos los navegadores interpretan dichas órdenes de la misma manera, o directamente no las soportan.

La palabra 'soporte' se utiliza muy vagamente. No es una propiedad booleana, hay grados. Muchos navegadores son compatibles con la misma característica, pero usan una sintaxis ligeramente diferente. Otros soportan sólo algunas partes de un estándar. Y otros, los peores culpables, se las arreglan para mezclar normas junto con implementaciones propietarias, creando un confuso lío sintáctico. (Kadlec, 2012, p. 7)

Hay varias herramientas en línea que permiten buscar, investigar y comparar casos específicos de uso y qué aceptación tienen éstos en las versiones de los navegadores más utilizados por los usuarios móviles. La más conocida de ellas es *Can I Use*, disponible en <http://caniuse.com>, la cual mediante tablas completas muestra y analiza datos de soporte y uso de funciones específicas.

Si bien las oportunidades en las que el lector pueda encontrarse en la situación de utilizar algunas características con soporte limitado son escasas, ya que, a menos que esté desarrollando una aplicación web que haga uso extensivo de funciones del dispositivo en sí, lo más convencional es utilizar funcionalidades clásicas.

Capítulo 3: La Web líquida

En el primer capítulo se han delineado las variables que deben ser tenidas en cuenta, a grandes rasgos, para desarrollar una página web, aunque básica. Es mediante el permanente estudio e interés por investigar y aprender, además de la experiencia práctica, la manera en que un profesional se forma en esta área. Quien ya se encuentre instruido y posea la habilidad para crear, encontrará en su carrera profesional, inevitablemente, la necesidad de portar sus conocimientos al mundo *mobile*.

En el segundo capítulo se presenta la evolución de este nicho, y se hace una predicción sobre el futuro cercano. Negar la realidad de la Web móvil, como desarrolladores, es cerrarse a un potencial porcentaje, significativo, de usuarios. En *La Web líquida*, asimismo, se exploran algunas de las condiciones de este medio definidas por los dispositivos y los contextos de navegación. De esta manera, el lector puede sumergirse en este nuevo marco y adoptar ciertos patrones para concebirlo.

Afortunadamente, el avance tecnológico y la extensa participación de la comunidad mundial hacia el progreso para una Web mejor, aportan numerosas prácticas y maneras en las que se pueden desarrollar sitios y aplicaciones que se amolden a dispositivos de variados tamaños. Y la Web, en constante perfeccionamiento técnico, permite que la transición de un paradigma a otro sea fluido e impacible. “La Web es universal. Es ubicua. A diferencia de cualquier medio que la ha precedido, la Web puede adaptarse a cualquier pantalla y a cualquier contexto. Es inherentemente flexible y maleable” (Kadlec, 2012, p. 2).

Aprovechando esta particularidad innata de la misma, surgen diferentes maneras de encarar esta transición, o mejor dicho, acomodación al entorno móvil. En este capítulo se presenta y prioriza una de ellas, valorada por la autora de este PG como la más conveniente, para ser desmenuzada en su composición y, más adelante, la viabilidad de su puesta en práctica en el cuarto capítulo.

3.1. Adaptar o morir

El lector puede encontrarse, al leer este Ensayo, con interrogantes muy frecuentes: ¿es realmente necesario amoldar toda clase de sitio web para su visualización en dispositivos móviles?, ¿vale la pena el esfuerzo?. Las mismas no solamente son realizadas por personas que recién inician su camino en este ámbito, ante su desconocimiento, sino que suele ser una traba muy habitual impuesta por algunos profesionales de informática, diseñadores y gerentes de proyectos, entre otros. Empero, quien suele insistir más decididamente con esta cuestión es el cliente, aunque su posición puede llegar a ser comprendida ya que no tiene en su poder, en la mayoría de los casos, el nivel de información sobre el estado de la industria con el que sí cuentan los anteriormente mencionados.

Desde la experiencia profesional, se llega a la conclusión que la respuesta a ambas preguntas es lisa y llanamente sí. Sí, es necesario amoldarse, y sí vale la pena el esfuerzo.

En algunos casos se puede llegar a considerar que, dado el *target* al cual se dirige el producto no es mayoritaria la cantidad de usuarios móviles, el planteo es injustificado. Sin embargo, ya se ha demostrado que el panorama de uso es propenso a mutar constantemente, creciendo de manera exponencial. Aún si continuase siendo minoría este número de individuos, es responsabilidad de los desarrolladores brindárles a todos, mayoría y minoría por igual, la mejor experiencia posible. No es imperativo que se cree una interfaz puntillosa para cada dispositivo móvil, ni que se intente mimetizar aquella planteada originalmente para resoluciones estándares de escritorios, pero se debe obligatoriamente, por lo menos, garantizar una navegación óptima aunque ésta sea limitada. “Aunque no todos los proyectos exijan gran nivel de sofisticación, en casi todos los casos, sigue siendo preferible ofrecer a los usuarios una vista adaptada de nuestro contenido dependiendo del tamaño de su ventana gráfica” (Frain, 2012, p. 10).

Es usual, generalmente en el ámbito laboral y profesional en Argentina, por más que el producto web y sus características de contenido y comerciales lo justifiquen, que se intente poner límites al desarrollo para móviles. Las razones suelen ser los apretados presupuestos y los tiempos con los que se cuenta para la realización del proyecto. Es una realidad que debe ser replanteada y torcerse para dar paso a una Web inclusiva y sinceramente global. La responsabilidad de cambiar ideales recae en las convicciones de cada profesional del diseño y desarrollo web, y en su compromiso con la mejora continua de la misma. Ellos son quienes deben tener la claridad y plasticidad para vender e inculcar a sus superiores vacilantes, y a los clientes, el verdadero valor del *mobile*, para que pase a ser parte inamovible de la cultura en el flujo de trabajo. “A medida que la distinción entre la Web ‘de escritorio’ y para móviles sigue desvaneciéndose, soportar *mobile* es algo que cada vez más hacemos como parte de los diseños que construimos. Debería convertirse en una segunda naturaleza para nosotros” (Reeves, 2013).

Considerar e incluir en los planes de un proyecto web el desarrollo específico para dispositivos móviles puede ser costoso, dependiendo la manera en que se lo encare, pero es necesario recalcar que dada la constante progresión y evolución, cuanto antes en el proceso de creación sea previsto, menos costoso resultará en términos económicos. Ignorarlo en etapas iniciales, e incluso hasta después de lanzado el producto, únicamente generará mayor inversión al querer trasladar un proyecto pre-existente a *mobile*, que si bien puede ser realizado, implica doble o triple cantidad de trabajo.

Éste es uno de los aportes que *Multiplicidad digital* desea asentar en el área del diseño y desarrollo web: hacer foco en la necesidad de formar más profesionales en la industria que sean conscientes de esta problemática y que actúen y se capaciten en pos del futuro hacia la unificación de una mejor experiencia para todos los usuarios, sin importar su contexto.

3.2. Responder acorde: *responsive web design*

Una vez que se toma la decisión de abarcar toda clase de modelos de visualización, y que se comprende que el sitio web en cuestión podrá ser accesible desde una inmesurable cantidad de dispositivos, ¿cuál es la manera más eficaz de presentar los contenidos para que éste sirva al mayor rango de posibilidades y combinaciones probables?

Una forma de responder a este interrogante, es mediante la confección de sitios paralelos. Para desarrollar esta técnica, es primordial establecer parámetros estándares de visualización que pueden llegar a redondear y englobar un gran número de dispositivos. Es decir, se toman valores medios de, por ejemplo *smartphones*, como los comentados en el capítulo anterior, cuando se definen las resoluciones de pantalla. De la misma manera se establecen valores para *tablets*, ambos casos sin enfocarse en un solo dispositivo específico, sino en promedios que abarquen muchos modelos distintos. Se cuenta entonces con bases para realizar un diseño *desktop*, otro para *tablets*, y otro para celulares. Al momento de maquetarlos, en otras palabras, traducir a HTML y CSS las imágenes, se debe entonces triplicar el trabajo ya que es necesario crear un sitio diferente para cada base, cada uno de ellos con sus reglas específicas. Una vez terminados, ya *online*, se desarrolla un detector que, antes de cargar el sitio al usuario, identifica desde dónde accede y le provee la interfaz que mejor se amolde a las características de su plataforma, redireccionándolo a una URL determinada. “Este enfoque ciertamente tiene sus méritos. La creación de un sitio separado para cada tipo de dispositivo hace que sea más fácil de adaptar la experiencia significativamente. [...] Por desgracia, no escalan bien: esos [...] sitios necesitan ser actualizados, probados y mantenidos” (Kadlec, 2012, pp. 9-10).

Si bien en un pasado no muy lejano, éste era el principal abordaje para servir a todos los usuarios, es fácil notar las desventajas que el planteo ocasiona: mantener un número variable de sitios paralelamente, perder la capacidad de confluir todo el tráfico hacia una

misma dirección, ya que, dependiendo desde dónde entre cada usuario, se le asignará una URL diferente, y a su vez la pérdida de la portabilidad. Estas consecuencias son analizadas en el siguiente apartado.

“En lugar de crear diseños desconectados, cada uno adaptado a un dispositivo o navegador en particular, deberíamos en cambio tratarlos como facetas de una misma experiencia” (Marcotte, 2011, p. 8).

Lo que Ethan Marcotte propuso en 2010 a través de un artículo en un *blog* especializado, revolucionó el estado de la industria, ya que rompió con modelos clásicos, proponiendo una impronta, una mirada sencilla e increíblemente adecuada para reorganizar el flujo de trabajo a la hora de diseñar con dispositivos móviles en mente. Él lo llamó *Responsive web design*, que en español se suele convenir en la traducción diseño web responsivo, y en algunas oportunidades se le llama también adaptativo. Basó sus observaciones en conceptos de una reciente disciplina en el campo de la arquitectura, que propone la interactividad del objeto con el usuario, llamada justamente *responsive architecture*. “En lugar de crear espacios que influyen en el comportamiento de las personas que pasan a través de ellos, los diseñadores responsivos están investigando maneras para que una pieza de arquitectura y sus habitantes se influyan e informen mutuamente” (Marcotte, 2011, p. 7).

El concepto que propone, entonces, es generar sitios web que se nutran del usuario para brindarle la mejor experiencia posible. ¿De qué manera se logra entonces esta simbiosis? Sorprendentemente, esta técnica no requiere de procesos complicados ni nuevas tecnologías. Simplemente hace uso de herramientas ya presentes en el entorno pero de un modo distinto para el que fueron planteadas inicialmente. El diseño web responsivo se lleva a cabo únicamente “[...] mediante la incorporación de tecnologías basadas en estándares a nuestro trabajo, y haciendo un pequeño cambio en nuestra filosofía hacia el diseño online” (Marcotte, 2011, p. 8).

La propuesta inicial de diseño responsivo es la unificación de tres técnicas, que serán tratadas individualmente en el cuarto capítulo: un *layout* flexible basado en un sistema de grillas, imágenes y contenido multimedia elásticos, y el uso de *media queries*.

La flexibilidad del diseño, como de sus elementos y componentes – multimedia o no – refiere a la capacidad de estos de escalar y adaptarse a sus respectivos contenedores.

Las *media queries* son una funcionalidad introducida en la especificación de CSS3 que permiten definir puntos de quiebre en una página para variar la estructura dependiendo la *consulta* – de aquí el término *query* – que se realice en la hoja de estilos.

Para tratar de poner la filosofía de diseño web responsivo en pocas palabras, yo diría que es la presentación de los contenidos de la manera más accesible para cualquier ventana gráfica que lo acceda. Por el contrario, un ‘sitio web móvil’ realmente es necesario cuando se requiere una experiencia de contenido y funcionalidades específicas basado en el dispositivo [...]. En estos casos, un sitio web para móviles presenta al usuario una experiencia completamente diferente a su equivalente de escritorio. (Frain, 2012, p. 11)

A raíz de todas las ventajas que pueden extraerse de esta práctica, es que la tendencia de trabajo en lo profesional apunta a ella cuando de diseño para dispositivos móviles se trata. Por esta razón, y por el hecho que no le es requerido al lector aprender nuevas tecnologías para aplicar el diseño web responsivo, es que se toma como eje para el desarrollo de este Proyecto de Grado. En las próximas secciones se explican sus métodos de empleo para culminar con la aplicación y análisis del mismo en el último capítulo.

3.3. ¿Uno, dos o cinco? Diseñar responsivamente en vez de por separado

En términos generales, se perfilaron las maneras en las que se puede servir contenido web para dispositivos *mobile*: generando un sitio diferente para cada conjunto de aparatos, o utilizando diseño adaptativo. A su vez, también se señalaron, aunque superficialmente, ciertas ventajas y desventajas que surgen de la práctica de ambas. Esta comparación entre ellas sigue siendo vigente ya que, en muchos casos, la creación de un sitio apartado suele ser la primera opción para ciertos desarrolladores.

Brad Frost, renombrado conferencista sobre diseño web, propuso en un artículo publicado en *Smashing Magazine* – revista en línea especializada – una comparación de los diseños para móviles de los candidatos presidenciales de EE.UU. de 2012. Allí presenta una pirámide jerárquica diseñada para evaluar ambas opciones. Ésta presenta, desde su base hacia arriba, los siguientes procesos: acceder, interactuar, rendir, mejorar (2012). Es posible, siguiendo estos lineamientos realizar un análisis caso por caso para evaluar qué solución funciona mejor.

Acceder es el proceso básico, y por ello se encuentra en los cimientos de la pirámide. Por más estéticamente agradable o funcional que sea el sitio, si no es accesible desde cualquier lado, nunca logrará cumplir sus objetivos. Es en esta instancia donde se hallan algunas complicaciones en los sitios web apartados, ya que fallan en servir correctamente el acceso a todos sus usuarios. La redirección depende de un detector que no siempre se encuentra bien implementado. A su vez, la multiplicidad de direcciones existentes de acceso hacen que, por ejemplo al compartirlas, se genere confusión entre distintos tipos de personas queriendo entrar desde otras clases de dispositivos; se le estaría sirviendo entonces una URL que no corresponde a su puerta de acceso. Es aquí donde los sitios web diseñados responsivamente persisten ya que es un mismo archivo ofrecido, que se encarga él mismo de adaptarse al usuario, sin importar de donde éste ingrese.

Interactuar refiere a la capacidad de permitir que el individuo, una vez que ya haya accedido, pueda hacer uso y como el término explicita, interactuar con el contenido. Tanto la navegación como la forma en la que se hace *scroll* en el sitio son factores fundamentales, ya que son las maneras en la que él se desplazará por el sitio, y eventualmente encontrará, o no, la información que busca. Este aspecto depende de cómo se optimicen estos recursos en cada uno de los casos.

Rendir es un elemento prácticamente insoslayable. Representa el nivel de desempeño que tiene el sitio, ya sea adaptable o apartado, considerando las limitaciones que

imponen los dispositivos, entre ellas la calidad de conexión, y las impuestas por el propio usuario, así como el tiempo que esté dispuesto a esperar hasta que el mismo cargue todo su contenido.

Mejorar, no menos importante, refiere a la optimización que puede dársele a cada experiencia para que, incluso, supere a la de escritorio. La mayoría de los dispositivos móviles tienen funcionalidades que pueden ser aprovechadas por estos sitios, como una cámara, GPS, o teléfono (Frost, 2012).

La aplicación en la práctica de esta pirámide sirve para analizar qué solución se amolda mejor a cada caso. Como ya se ha señalado antes, en un gran porcentaje de los casos, si se hace correctamente, el diseño web responsivo es más conveniente. Si bien muchos estudiosos de la materia coinciden con este abordaje, como Luke Wroblewski, Josh Clark, Tim Kadlec, entre otros firmantes de la iniciativa *Future Friendly*, hay algunos que sostienen que un sitio web apartado es la mejor opción. Este es el caso de Jakob Nielsen (2012), experto y pionero en el área de usabilidad y experiencia de usuario, quien luego de analizar junto a su equipo una cantidad de ejemplares, llega a la conclusión de que un sitio apartado es el enfoque correcto. En sus conclusiones establece que es necesario eliminar contenido y funcionalidades de lo que él llama *sitio completo – full site* – cuando se escale a un diseño para móviles. “Una buena experiencia de usuario móvil requiere un diseño diferente a lo que se necesita para satisfacer a los usuarios de escritorio. Dos diseños, dos sitios, y entrecruzado para hacer que todo funcione” (Nielsen, 2012).

Si bien es real que los usuarios requieren un diseño de interfaz que se ajuste a sus condiciones, la reducción en la cantidad y/o calidad de la información no es el camino correcto. Éstos necesitan del sitio un contenido, un servicio de él, y ninguno de ellos debería ser discriminado y privado de la información debido al contexto en el que se encuentren.

Yo creo que los sitios web móviles apartados son como poner una bandita sobre el problema; por lo general, no deberíamos tener sitios web para móviles distintos, más de lo que deberíamos tener sitios web para lectores de pantalla distintos. La razón de que muchos ‘sitios web completos’ no se pueden utilizar en los teléfonos

móviles se debe a que muchos sitios web completos son inutilizables en cualquier tipo de dispositivo. (Lawson, 2012, p. 21)

3.4. Avancen los móviles: la filosofía *mobile first*

Luke Wroblewski acuñó este término en 2009 cuando esbozó el concepto, en un *posteo* en su *blog*, el cual establece que el proceso de diseño para móviles debería empezar justamente allí. La idea detrás de *mobile first* es que el diseñador o desarrollador comience su planteo desde los dispositivos, para poder delinear específicamente con qué herramientas se cuentan, así luego poder escalar hacia más pantallas, sin perder lo esencial en el camino, como puede llegar a pasar haciendo el camino inverso.

Wroblewski (2011) sostiene que acomodar los procesos mentales para amoldarse a los parámetros de *mobile* en primera instancia permite tomar ventaja de lo siguiente:

“Aprovechar el enorme crecimiento en el uso de Internet móvil y encontrar nuevas formas para que la gente use nuestros sitios web y aplicaciones” (2011, p. 120). Esta conclusión es compartida, intrínsecamente, con este Ensayo. El avance de la Red en móviles ha superado el punto en el que puede ser ignorado, por lo tanto, hacer uso del mismo hacia la expansión de las interfaces compatibles es elemental.

“Aceptar las limitaciones móviles para enfocar y priorizar los servicios que estamos diseñando y construyendo” (2011, p. 120). Al adoptar una aproximación *mobile first* se hace a su vez un abordaje de *contenido primero*, ya que al optimizar los recursos disponibles, los componentes primordiales prevalecen y se mantienen.

“Utilizar las capacidades del móvil para innovar la completa experiencia del cliente” (2011, p. 120). Hay muchas maneras en las que se puede mejorar el patrón de uso que lleva a cabo el usuario si se consideran las funcionalidades nativas de los dispositivos desde el comienzo del desarrollo.

“Tomar lo que sabemos sobre el diseño para la web y empezar a pensar de manera diferente acerca de la organización en móviles [...]” (2011, p. 120). Básicamente lo que

propone este concepto es cambiar la filosofía y modelos mentales asentados en el profesional para que adopte una técnica que beneficiará su flujo de trabajo.

Si bien la propuesta aporta importantes avances, no todos los desarrolladores comparten esta visión. Principalmente esto ocurre ya que la puesta en práctica de *mobile first* no es sencilla. Requiere de mucho más tiempo de planificación, conocimientos y experiencia en UX para diseñar interfaces que puedan soportar la escala sin fracasar en el trayecto. Además, implica una mutación tanto en modalidades de pensamiento como de trabajo, que simplemente muchos profesionales no están dispuestos a tomar.

3.5. ¿Mejora progresiva o degradación gradual?

Desde los comienzos de la Web, los navegadores fueron diseñados para ignorar instrucciones que no reconocieran. Esta capacidad, inherente a muchos sistemas informáticos, es conocida como tolerancia al fallo, o *fault tolerance* por su traducción al inglés. Esto está dado así para que los lenguajes HTML y CSS pudiesen evolucionar sin ser frenados por el riesgo a conflictuar con ellos. “Entender la tolerancia a fallos es clave para entender la mejora progresiva. [...] es la razón por qué [...] funciona y hace posible asegurarse que todo el contenido servido en la web es accesible y al alcance de todos” (Gustafson, 2011, p. 5).

A medida que la Web crecía, las técnicas utilizadas en el campo del diseño web fueron evolucionando con ella. Lo más habitual consistía en aplicar gran parte de estos cambios en cualquier tipo de proyecto, los cuales generalmente eran únicamente soportados por las últimas versiones de los navegadores más populares. Esta práctica se convirtió en una tendencia, que duró más de una década. “Esta época dio lugar a la filosofía de desarrollo conocida como ‘degradación gradual’” (Gustafson, 2011, p. 6).

Actualmente pueden distinguirse restos de *graceful degradation*, su término original en inglés. Es aún habitual diseñar y maquetar con técnicas disponibles sólo para navegadores modernos, y dejar para el final del proyecto la comprobación de que otros

anteriores las soporten, aunque sea medianamente, y luego, a lo sumo, hacer pequeños cambios para suplir la falta. “El concepto no es especialmente amigable con el futuro. Muestra una falta de respeto a los usuarios de navegadores más antiguos e ignora la creciente realidad de que hay nuevos dispositivos (móviles), que también cuentan con navegadores menos capaces” (Kadlec, 2012, p. 15).

La degradación gradual fue siendo desvalorizada, hasta la aparición de un nuevo concepto denominado mejora progresiva. Esta nueva técnica es esencialmente lo contrario. Su premisa es comenzar por un planteo que abarque la estructura delineada en base a los estándares de la W3C, de manera que sea en primera instancia accesible desde cualquier navegador, para luego ir sumándole capas y niveles de funcionalidades, y características soportadas únicamente en algunos más modernos (Frain, 2012).

Tanto diseñadores novatos como profesionales del área, deben procurar servir una experiencia óptima a la mayor cantidad de usuarios como les sea posible. Por eso mismo, la mejora progresiva es una filosofía que, correctamente aplicada, promete cumplir con tal objetivo.

Capítulo 4: ¡Manos a la obra! *Responsive web design* en camino

El avance de la tecnología móvil ha sabido insertarse en la sociedad de consumo a ritmos cada vez mayores. Esto proporciona nuevas vías de entrada a la *World Wide Web*, la cual debe ser moldeada para ofrecer sus contenidos a cada usuario sin importar el entorno. Su maleabilidad es una característica propia e innata que puede ser trabajada y explotada al máximo para servir al propósito que todos los diseñadores y desarrolladores avocados a la Web deberían perseguir: la masividad y efectividad de conversión de sus productos *online*.

Es en este punto donde surgen dudas y trabas sobre el camino a tomar para cumplir este objetivo. ¿De qué manera se puede incluir a la mayor parte de los navegantes?, ¿cómo se plantea un enfoque a usuarios de dispositivos móviles?, y por sobre todo, ¿cómo se lleva a cabo?. Se presentaron diversas técnicas y filosofías que intentan atacar estas problemáticas y estandarizar una solución. La que más se destaca por sus beneficios es el diseño web responsivo – neologismo que deriva del término original en inglés forjado por su ideólogo, *responsive web design* –, a veces referido simplemente como RWD.

En la teoría, esta técnica es convincente. No obstante, al momento de aplicar esta filosofía a un proyecto concreto, muchos diseñadores novatos o sin experiencia en este tipo de abordaje suelen verse en una encrucijada, ya que no cuentan con las herramientas y conocimientos precisos para hacerle frente. Afortunadamente, numerosos estudiosos y entusiastas de la materia han pensado y publicado soluciones rápidas y efectivas que hacen el trabajo de diseñadores, maquetadores y desarrolladores al menos un poco más sencillo.

En este capítulo se efectúa un recorrido de diferentes nociones que el lector debe tener en cuenta al momento de emprender un proyecto que haga uso de diseño web responsivo. Se delinearán técnicas fundamentales en esta práctica, como las *media queries* y unidades relativas, como también se sugieren sistemas completos ya desarrollados sobre los cuales sentar bases a la hora de diseñar.

4.1. Condicionar las reglas: *media queries*

La característica más destacable del diseño web responsivo es su adaptabilidad a los límites y espacios que el dispositivo de visualización le imponga. Básicamente, al diseñar siguiendo esta pauta, se le puede indicar al navegador que establezca nuevas reglas de estilo dependiendo el tamaño del contenedor, entre otras variables. Es decir, es posible reubicar, reorganizar y hasta volver a diseñar elementos de una página web para cada situación mediante una serie de comandos.

Anteriormente se explicó que en cualquier diseño web deberían haber tres niveles separados de armado que hacen al funcionamiento integral del mismo: estructura, presentación y comportamiento. La estructura es su esqueleto, en donde a su vez se plasma el contenido embebido en etiquetas HTML. Esto le indica al navegador cómo debe mostrar los distintos elementos. Donde realmente se evidencia el diseño es en la capa de presentación. Ésta se caracteriza por dictar una serie de órdenes sobre cómo interpretar cada elemento gráficamente. Es aquí donde se definen los colores, tamaños, ubicaciones y variables estilísticas de todos los componentes visuales de una página. Para estas declaraciones se utiliza el lenguaje CSS, y es en este nivel donde las mismas reglas de estilo pueden ser replanteadas dinámicamente. El comportamiento puede servir como complemento para optimizar la experiencia en variados entornos, pero en términos generales, no debería ser en este nivel donde se marquen pautas de visualización al navegador.

Ya que al diseñar responsivamente se está sirviendo únicamente un mismo archivo HTML, es necesario hacer la separación de *layouts* mediante una indicación al navegador para que levante una u otra hoja de estilo, o si se unifica todo en una sola señalarle qué declaraciones en ella específicamente deben ser interpretadas en cada situación. Para lo primero, se utiliza el atributo *media* en la llamada a cada documento CSS. En el segundo caso se hace mediante un bloque *@media* en la hoja de estilos.

Desde la versión 2 de CSS, *media* puede recibir valores para distintos tipos de casos: *screen* para la visualización en pantalla, *print* para impresión, entre otros no tan utilizados como *tv* o *handheld*. Por defecto se aplica *all*, que unifica todo el espectro. Es decir, si se declara que *media* es igual a *screen*, todas las instrucciones CSS allí definidas serán únicamente aplicables cuando se muestre ese sitio en una pantalla. Caso contrario, si equivale a *print*, esos estilos serán aplicados al momento de impresión de la página. Esto es útil ya que el soporte es diferente, por lo tanto es necesario adaptar la presentación del contenido para que sea óptimo en la mayoría de las situaciones. Exactamente lo mismo ocurre cuando se diseña para dispositivos móviles, ya que surge la necesidad de dar prioridad a la presentación del contenido en las mejores condiciones. Sin embargo, con definir meramente este valor, no es posible separar estilos para diferentes pantallas, ya que *screen* apunta a todas ellas por igual.

Desde la introducción de CSS3, algunos de sus módulos ya han sido estandarizados por la W3C, como es el caso de las *media queries*. “Las media queries permiten definir los estilos que deben aplicarse en determinadas circunstancias dependiendo los valores de características como resolución, profundidad de color, la altura y el ancho” (Kadlec, 2012, p. 54) del área visible del dispositivo. Éstas son condiciones que se adicionan al atributo *media* y permiten limitar el entorno al que se aplica un conjunto de declaraciones de estilo. Estas variables permiten, entre otras cosas, indicar puntos de quiebre en una estructura, conocidos técnicamente como *breakpoints*, sobre los cuales se harán las distinciones. Los valores que toman son generalmente dictados por convención, y responden a números derivados de resoluciones estándares de diversos dispositivos. En el segundo capítulo, al hablar de resoluciones y orientaciones de pantalla, se presentaron algunos números de referencia. Estos *breakpoints* dividen el flujo del contenido y la estructura, presentando así diferentes *layouts* según sus valores. Generalmente, se utilizan únicamente segmentaciones basadas en el ancho de la ventana gráfica con las declaraciones *max-width* y *min-width*, es decir, el ancho máximo y mínimo que ésta debe

tener para que las consiguientes reglas de estilo le sean aplicadas. A su vez, se pueden crear condiciones complejas al juntar varios atributos mediante operadores como *and*, *or*, y *not*, los cuales enriquecen los límites impuestos en la *media query*.

Las dos maneras de incluirlas, tanto desde la etiqueta <link> en el HTML como desde un bloque *@media* en el CSS, son ilustradas en la Figura 6: *Declaraciones de media queries*, pág. 97., anexo de imágenes seleccionadas.

4.2. Medir relativamente con unidades flexibles

El diseño web responsivo sienta sus bases en la flexibilidad de sus componentes. Para poder conseguirla, es necesario despojarse de las nociones que todo diseñador tiene acerca de plantear sus creaciones en tamaños fijos y estáticos. Sin ir más lejos, su principal herramienta es el pixel, una unidad de medida mínima y casi abstracta. Al establecer dimensiones fijas al diseñar para la Web, se limitan las posibilidades que ésta tiene para adaptarse a diferentes resoluciones en variados dispositivos. No todos ellos interpretan un pixel de la misma manera, ya que no es una medida estándar, algo que pueda ser mensurado más allá del mundo digital. Es únicamente un punto.

Esto hace que al utilizar números en píxeles, los diseños sean un poco menos *líquidos* y fluidos, con menor capacidad de adaptarse a su entorno. ¿De qué manera es conveniente entonces definir los tamaños de elementos? Se considera que una de las mejores opciones para hacerle frente a este problema es el uso de porcentajes. Al establecer un valor porcentual, ese elemento ocupará esa proporción relativa sobre su padre. Ese padre puede ser tanto un contenedor, como un <div>, o mismo la ventana gráfica general. En diseños complejos, se establece una estructura jerárquica en la que casi todos los elementos están contenidos por otros, salvo los de mayor nivel. Cuando se define, por ejemplo, el ancho porcentualmente, el navegador interpreta que éste ocupará esa parte del tamaño que a su vez ocupa su padre en la ventana gráfica.

Esto permite que, simplemente con escalar la misma, ya sea manualmente o porque ésta toma diferentes tamaños según el dispositivo, los elementos se acomodarán automáticamente. Si se estuviesen usando valores fijos, como píxeles, esto no pasaría, ya que éstos no son relativos a ningún contenedor. Estos conceptos son fundamentales en el diseño de sistemas de grillas, explicado más adelante.

“Sin embargo, un layout fluido por sí mismo no es suficiente para asegurar que un diseño se vea bien en todo, desde un smartphone a un televisor.” (Kadlec, 2012, p. 25). Al utilizar valores porcentuales se establece un lineamiento hacia un buen diseño responsivo, pero no lo es todo. Es necesario hacer ajustes en el camino, mediante *media queries*, en muchos casos. Es por eso, que a su vez también se utilizan otros tipos de medidas relativas, especialmente para elementos tipográficos, como *em* y *rem*.

Existe una simple fórmula que ayuda a calcular valores porcentuales ideales tomando como referencia números fijos en píxeles: objeto ÷ contexto = resultado (Marcotte, 2011). Esta fórmula se traduce al tomar el valor que se le asigna al objeto en píxeles, dividirlo por el tamaño de su padre o contexto en el que éste se encuentre, para así obtener un resultado relativo que puede trasladarse a porcentaje o a *em*. Un ejemplo concreto sería considerar un escenario en el que se quiere establecer un ancho porcentual de una caja de clase artículo dentro de una caja contenedor. El contenedor tiene un ancho fijo de 960 píxeles, y el artículo de 900. Utilizando dicha fórmula, se puede calcular 900 (el objeto) dividido por 960 (el contexto), para obtener 0,9375. Luego se multiplica el resultado por 100, o simplemente se corren dos decimales, para obtener 93,75. Éste es el ancho porcentual que debería asignársele a la caja artículo para que ocupe la misma proporción espacial que si ambos valores fueran definidos en píxeles (Marcotte, 2011).

La fórmula objeto/contexto se considera extremadamente útil al momento de flexibilizar componentes en una estructura web responsiva, y es mencionada por varios autores como una base fundamental del mismo. Si bien parece ser complicada y tediosa, utilizándola regularmente, la práctica hace a la costumbre. Estableciendo entonces todos

tamaños con unidades relativas, el mantenimiento del diseño resulta más fácil y natural, ya que no es necesario, en el caso de querer hacer alguna modificación general, tener que definir todos los valores nuevamente. Y por sobre todo, a lo que hace al diseño responsivo, se cumple el objetivo que el mismo diseño se adapte ingeniosamente a sus ilimitados contextos, de los que, desde la perspectiva de diseñadores, son prácticamente imposibles de predecir.

4.3. Verdaderos tipos móviles: fuentes adaptables

Al plantear un diseño fluido, en donde los elementos se escalan en relación a sus contextos, es necesario utilizar unidades relativas en, hasta donde sea posible y lo permita el bosquejo gráfico, la mayoría de sus componentes. Habitualmente, en proyectos web se suelen utilizar valores en píxeles para definir cuerpos tipográficos ya que, como cuando se usan anchos fijos, se tiene la sensación de poseer mayor control sobre el flujo estético de la página. Sin embargo, ya se ha demostrado que ésta es una práctica inflexible, y que es contraria a los principios del diseño web responsivo. No es posible, al menos no de manera dinámica, que el contenido fluya líquidamente en sus contenedores si se restringe la adaptabilidad del mismo.

Con este fin se utilizan entonces otras unidades de medidas, como *em* y *rem*. Ver Tabla 2: *Unidades relativas en CSS*, pág. 95, anexo de imágenes seleccionadas. Si bien ambos comparten raíces, no son exactamente lo mismo.

El término *em* es simplemente una manera de expresar la letra 'M' en forma escrita y se pronuncia como tal. Históricamente, se usaba la letra 'M' para establecer el tamaño de una fuente determinada, debido a que [...] es la más grande (más ancha) de las letras. Hoy en día, EM (sic.) como una unidad de medida define la proporción de la anchura y la altura de una letra con respecto al tamaño en puntos de una fuente dada. (Frain, 2012, p. 76)

Al ser una unidad de medida relativa, es complicado imaginar un *em* en el espacio. Gráfica y físicamente no existe como tal, sino que se configura sobre la marcha en relación a un valor pre-establecido, a partir del cual adapta sus propios valores. Un *em*

equivale al tamaño base de la fuente de un elemento, definido en la propiedad *font-size*. Esto quiere decir que todas las declaraciones que se midan en *em* en ese elemento o en sus descendientes que hereden este valor, serán un múltiplo del número establecido como tamaño de fuente. “Dimensionar las fuentes de tus elementos relativamente significa que sólo es necesario ajustar la línea base inicial y el resto del contenido se ajustará automáticamente – proporciones intactas” (Kadlec, 2012, p. 29).

Un caso de aplicación real es, por ejemplo, un elemento de lista desordenada – `` – cuyo *font-size* es de 22px. Al establecer márgenes en sí mismo, o en sus descendientes – etiquetas de lista `` – de manera relativa con *em*, por ejemplo 0.5em, este valor será computado como $22 \times 0.5 = 11$, es decir 11px de márgenes. Si no se define un valor de base en px, sino que se utilizan siempre *ems*, el navegador tomará como cuerpo el que establece por defecto a todas las páginas por igual, 16px. Es importante resaltar que el usuario cuenta con la posibilidad de cambiar ese número manualmente desde las opciones de la aplicación, y se han reportado navegadores en dispositivos móviles que establecen el tamaño de fuente en números mayores, así que es una buena idea o práctica definir inicialmente un valor en píxeles sobre el cual sentar bases, siendo también 16px una cantidad recomendada.

Con la llegada de CSS3, una nueva unidad de medida ha sido introducida. *Rem*, abreviación de *root em* – es decir, *em* raíz – funciona de manera similar a *em*, excepto que la formulación matemática para calcular su valor no se hace sobre la propiedad *font-size* del padre del elemento directamente, sino que absolutamente todos ellos, sin importar en qué nivel jerárquico se encuentren, heredan el mismo tamaño de cuerpo, definido en el elemento raíz de toda página web, `<html>`. Esto quiere decir que únicamente es necesario establecer un número para el *root* y luego establecer valores con unidad *rem* que sean proporcionales a aquél. Esto soluciona los problemas de herencia y agravamiento progresivo que puede ocasionar un uso inconsistente de unidades *em* en elementos anidados.

Si bien es un módulo introducido con la última versión de CSS, todos los navegadores modernos soportan correctamente esta unidad de medida relativa. No obstante, hay versiones anteriores de los mismos que no saben interpretarla, como por ejemplo *Internet Explorer* 8 y previos, los cuales cuentan, aún, con una gran base de usuarios, por lo cual hay que evaluar si la utilización de *rems* afectará, y en qué medida, a la experiencia de algunos navegantes. Una sugerencia habitual es proveer valores fijos en píxeles antes de usar *rems*, de manera que los navegadores que tengan soporte para este módulo, utilicen la última declaración como válida, ya que como se ha mencionado anteriormente, a menos que se indique lo contrario, al superponerse declaraciones en CSS el navegador graficará en pantalla sólo la última definida.

Entonces, ¿qué opción conviene usar para diseños responsivos? Si bien ambas opciones son similares y escalables, es preferible, al menos por el momento, utilizar *em*. Como bien explica Tim Kadlec, *rem* aún es una novedad, por lo que no es recomendable obviar una recurrencia a un valor en píxeles.

Usar *ems* no sólo permite que la tipografía escale, sino que también puede facilitar el mantenimiento. Si decide aumentar el tamaño de fuente en todo su sitio, basta con cambiar el porcentaje que se aplica al cuerpo y ya está todo listo. Con *rems*, ya que tienes que recurrir al uso de píxeles, tendrás que actualizar cualquier elemento con píxeles como tamaño en todo el código. (2012, p. 33)

4.4. Imágenes elásticas

Hasta el momento se explicaron distintos métodos para atacar problemáticas derivadas del uso de medidas fijas para los componentes de una página web. Sin embargo, los casos mencionados refieren a elementos que pueden readaptarse con naturalidad al flujo de la misma. Las imágenes, como así también otros elementos conocidos como *media* – videos, por ejemplo – poseen de manera inherente un tamaño fijo y real, con las más variadas relaciones de aspecto. Por este motivo, la manera en la que deben ser escalados en un diseño responsivo es también distinta.

Hay dos maneras de utilizar imágenes en una página web, y aunque sus usos suelen ser diferentes, en ocasiones pueden ser confundidos, y hasta intercambiados para cumplir con algún objetivo específico.

Desde la estructura, es posible agregarlas utilizando la etiqueta `` en el documento HTML, como es el caso de una fotografía de cabecera en una entrada de *blog*, o una miniatura, llamada también *thumbnail*. En este caso, la imagen pasa a ser un elemento más del *markup* y ocupa su tamaño real en pantalla, a menos que se le indiquen medidas específicas por CSS o mediante atributos de la etiqueta.

La otra posibilidad es definir las desde el CSS como imágenes de fondo de un elemento, utilizando la propiedad *background-image*. Como su nombre lo indica, su funcionalidad es la de servir como complemento decorativo de segundo plano. De esta manera, al no ser un elemento en sí, sino un agregado de otro, no ocupan gráficamente su tamaño original, sino que llenan su contenedor y son mostradas dependiendo únicamente del tamaño de éste. Además del uso clásico decorativo, esta propiedad puede ser utilizada para variados propósitos. Ya que las llamadas a la imagen son manipuladas desde CSS, se pueden conjugar con otras propiedades que permitan mover, cambiar y hasta escalar la misma en el caso de, por ejemplo, cuando el usuario pase el *mouse* por encima del elemento que la contiene. Por esto suelen ser utilizadas para generar el efecto *rollover* en logos, entre otros componentes.

El principal problema de las imágenes insertadas en diseños responsivos es que éstas mantienen su tamaño sin acomodarse automáticamente a su entorno. Se mantienen fijas con el tamaño que les fue dado inicialmente, o en su defecto en su dimensión original. Para solucionar este problema, se podría pensar en la utilización de *media queries*, en donde se le definan nuevos valores de alto y ancho para cada *breakpoint*. Si bien es posible hacer esto, sería muy complejo realizarlo para cada imagen que haya en el sitio, presente o futura, además de trabajoso de mantener.

Existe una propiedad en CSS llamada *max-width* – y a su vez también existe *max-height*, aunque no viene al caso para esta técnica – que establece hasta qué punto debe ese elemento expandirse en ancho. Al fijar ese valor en 100%, es posible asegurarse que las imágenes no sobrepasen el total del ancho de sus contenedores. Es decir, si la fotografía se encuentra embebida dentro de una caja de tamaño menor que la imagen original, ésta se escalará para adaptarse a él. La diferencia de declarar *max-width* a simplemente *width* es que, en el caso que el contenedor en alguna circunstancia sea mayor que la imagen, ésta no crecerá hasta ocupar el 100% del ancho de su padre, corriendo el riesgo que se estire, sino que se mantendrá en su tamaño original. Si éste decreciera, por ejemplo al visualizarlo en un dispositivo pequeño, la imagen sí se adaptaría al mismo hasta un máximo del 100%, o cualquier otro valor que se le defina.

Las imágenes tienen todas una relación de aspecto única a sí misma, lo cual significa que por más que se le establezca un ancho variable, el alto no siempre será el total del de su padre. Es por eso recomendable agregar como regla, además de *max-width*, la propiedad *height* configurada en *auto*, lo cual mantiene esta relación dinámicamente, para cualquier tipo de imagen por igual. Esta técnica de adaptabilidad también puede ser utilizada en otro tipo de elementos que compartan estas características, como videos u objetos *Flash* embebidos.

Este método no aplica, sin embargo, para las imágenes de fondo cargadas desde CSS, ya que no son elementos en sí mismas, como se ha mencionado, sino que dependen del tamaño de su contenedor. Hay dos propiedades que complementan a *background-image* que pueden ser utilizadas en este caso para adaptar el fondo fluidamente: *background-size*, introducida con CSS3, que lo escala a valores arbitrarios y *background-position*, que lo reacomoda en el espacio.

Las opciones comentadas anteriormente son útiles en bastantes casos pero no contemplan un aspecto importante que acarrea el uso de imágenes: el tiempo de carga y el rendimiento de la página. Al servir contenido para dispositivos móviles es fundamental

tener en cuenta el peso de éstas. Por más que hayan sido escaladas, la petición al servidor por ese recurso es la misma, es decir, el tamaño original de la imagen es exactamente igual, incluido su peso. Esto puede demorar el tiempo de carga para usuarios móviles, quienes en muchos casos se encuentran restringidos por su conexión a descargar archivos muy pesados. Por este motivo, sería prudente optimizar las llamadas de imágenes, más que solamente limitarse a mejorar el aspecto visual de las mismas en el diseño.

Hay diferentes maneras de servir imágenes selectivamente según el dispositivo desde donde se acceda. La manera de saber cuál, o qué tipo de plataforma es la que realiza la solicitud, mayoritariamente – a menos que se utilice algún complejo sistema de detección mediante tecnologías del lado del servidor – es obteniendo la resolución de pantalla en la cual se está visualizando el sitio. De esta manera, se puede *adivinar* o realizar una aproximación sobre el entorno del usuario, para entonces ofrecerle una imagen, o mismo un video, acorde a sus posibilidades de descarga.

Si se están cargando imágenes de fondo en CSS, una manera sencilla es definir el *background-image* dentro de cada *media query*, dictando distintas imágenes apropiadas para cada una. De esta manera, el servidor únicamente levanta el archivo correspondiente al ámbito de esa consulta. Por ejemplo, en una *query* que apunta a *smartphones*, se especificará un archivo optimizado para dichos límites. En una de escritorio, en donde se asume que el usuario no sólo tiene más espacio visual, sino también mejor conexión para su descarga, puede ofrecérsele una imagen de mayor calidad y dimensión.

¿Pero cómo se puede solucionar esto cuando las imágenes están insertadas en el HTML? Al estar incrustadas en la estructura, es decir, en el *markup*, van a ser peticionadas al servidor cada vez que se cargue la página, sin importar si luego por CSS se ocultan o se achican. Para atacar esta problemática y optimizar la experiencia de usuarios con recursos limitados de acceso, Kadlec (2012) propone una interesante

técnica, aunque limitada, que la autora de este Proyecto de Graduación toma como base y formula una mejora para servir imágenes condicionalmente. Si bien se debería utilizar programación del lado del cliente y la solución no es extremadamente semántica, cumple con el objetivo planteado. En teoría, requiere que el desarrollador o diseñador cree una serie de archivos optimizados para una cantidad fija de dispositivos por cada imagen. Por ejemplo, una fotografía debería ser almacenada en equis cantidad de versiones (para ilustrar el tema se toman solamente 3), entonces tendría que crear y almacenar `paisaje_smartphones.jpg`, `paisaje_tablets.jpg` y `paisaje_desktop.jpg`, siendo en este caso `paisaje` un nombre de referencia y el segundo valor una reseña al entorno. Cada una de ellas debería responder a valores de calidad y tamaño óptimos para el contexto en el que se vaya a utilizar.

En el *markup* debería agregársele un atributo *data* al elemento contenedor en el cual se ubicaría la imagen y pasarle un valor referencial de la misma, `paisaje` en el caso del ejemplo mencionado.

Luego, programáticamente mediante *Javascript*, se deben recorrer todos los elementos que posean este atributo *data*, y guardar sus valores en variables. A su vez, se obtienen los tamaños de la ventana gráfica, y se hacen cálculos aproximados para identificar desde qué tipo de dispositivo se está accediendo, ya sea *desktop*, *smartphones*, celulares comunes, *tablets*, entre otros. Una vez finalizadas estas dos tareas, se crea un elemento de imagen a continuación de cada etiqueta anteriormente encontrada. Ya teniendo una referencia sobre qué imagen debe cargar, y en qué resolución, sabe exactamente cual archivo levantar.

Otra situación que puede ocurrir es que el dispositivo de acceso sea uno con tecnología *retina*. Ésta es implementada por *Apple* en sus terminales *iPhone* y *iPad* más recientes, y ofrece una densidad de píxeles mucho mayor para obtener pantallas de alta resolución. En estos casos, las imágenes que se sirvan deberían a su vez ser de mayor calidad para

que se vean con buena definición. Mediante condiciones en las *media queries*, es posible realizar esta detección para cargar los archivos adecuados.

Estas técnicas, utilizadas en conjunción con las primeramente mencionadas de escalamiento, aseguran una experiencia optimizada para toda clase de usuarios de manera democrática, y permiten cerciorarse que visualmente el sitio se mantenga consistente.

4.5. Cada cosa en su lugar: sistemas de grillas

Desde los inicios del diseño web, se han implementado diversas formas de estructurar y ordenar el contenido en una página. En un pasado, se consideraba una buena práctica crear contenedores visuales utilizando marcos, más comunmente conocidos como *frames*. Éstos dividían la pantalla en cajas rígidas dentro de las cuales se cargaba un contenido específico, independiente del que se mostraba en otro *frame*. Esta técnica marcó una era en la Web, una tendencia que fue chocando contra sus propios límites, ya que era inflexible y tedioso de mantener.

Con el avance del conocimiento, y con la introducción de nuevas y mejores prácticas, se comenzó a armar los *layouts* y a diseñar en base a sistemas, a veces complejos, de tablas. Muchos sitios solían estar, y aún algunos persisten, configurados en bloques estáticos, que si bien eran un poco más flexibles que cuando se diseñaba con `<frame>`, su mantenimiento seguía siendo arduo. Las tablas, semánticamente, cumplen un objetivo diferente al de servir como soportes de presentación, sino que están pensadas para ser funcionales al manejo de datos y su comparación.

Por esta razón, los diseñadores más puristas fueron migrando a nuevos horizontes en los que se pudiera diseñar con herramientas destinadas para tal fin. Es en este momento donde entró en escena la etiqueta `<div>`. Su función es la de división, de allí su nombre, y su representación visual no es más que un bloque vacío sin estilos. Sus características innatas la convierten en la herramienta perfecta para diseñar estructuras jerárquicas y

maleables, y a su vez permiten dar rienda suelta a la imaginación, ya que no presentan límites estéticos, sino que esos son impuestos por la creatividad del diseñador.

No obstante, aquí no comenzó la Web como se la conoce ahora. Por diversos motivos, los diseños seguían siendo inflexibles y hechos a gusto del diseñador, y no con el objetivo de ser utilitarios a la mayor parte de los usuarios. Los <div>s eran utilizados no como lienzos dúctiles, sino como un medio para posicionar arbitrariamente contenido en la pantalla. Mediante la declaración de la regla de estilo *position: absolute*, se ubicaban los componentes de manera fija, generando un diseño que solamente le era funcional al diseñador y a los usuarios que compartiesen sus mismas características de visualización. El diseño para la Web, como ella misma, ha ido evolucionando y tomando caminos más universales. Si bien el concepto de grilla reticular proveniente del diseño gráfico y editorial siempre ha sido aplicado, o por lo menos se ha intentado, su implementación mediante el uso de <div> ha adquirido popularidad recién en los últimos años.

“Los diseños de grillas pueden agregar ritmo visual para guiar el ojo del usuario, haciendo el diseño limpio, ordenado y proporcionarle consistencia. Permiten estabilidad y estructura en la que se puede agregar fácilmente nuevos elementos y reorganizar los existentes” (Grannell, Sumner y Synodinos, 2012, p. 250).

Básicamente, un sistema de grillas es una cuadrícula compuesta por líneas horizontales y verticales que se intersectan y permiten al diseñador regular la disposición de los componentes en su diseño. Su uso es recomendado ya que, si están bien utilizadas, ayudan a unificar visualmente el mismo. Mediante el uso de columnas de igual tamaño y equidistantes, separadas por *calles* – en inglés denominadas *gutter* – los elementos de una página pueden ser colocados de manera que su organización sea precisa y prolija. Confeccionar una grilla requiere conocer los límites en donde ésta será aplicada. Sin embargo, en la cambiante Web, donde las fronteras son elásticas y prácticamente infinitas, la implementación de un sistema de estas características suele ser una problemática. Existe la posibilidad de ignorar la flexibilidad que éstas deberían tener, y

establecer un sistema que ocupe un tamaño fijo, como lo hace el popular sistema 960, el cual establece un tamaño horizontal de esa cantidad de píxeles, y puede ser dividido en 12 o 16 columnas. Pero esto no sería ideal.

Como ya se ha mencionado anteriormente, mediante la utilización correcta de *media queries*, y unidades relativas de medida, es posible crear sistemas propios que respondan a las necesidades específicas de cada diseño, y que a su vez sean adaptables a la mayor cantidad de pantallas como sea posible. En vez de definir valores fijos para el contenedor principal y sus columnas interiores, pueden utilizarse porcentajes, e incluso *em*, para que éstos se amolden a sus contextos.

De cualquier manera, como se puede imaginar, no es recomendable que el sistema de grillas mantenga la misma configuración en todos los dispositivos, esto sería simplemente la misma estructura pero reducida en tamaño. La premisa del diseño web responsivo es que éste responda a las necesidades de cada caso. Esto quiere decir que la organización debe generarse dinámicamente para reestructurarse acorde a la situación. Si bien suena complicado de realizar, es factible su implementación mediante las técnicas anteriormente mencionadas.

Para un proyecto sin gran escala, quizás una mejor opción sería utilizar un sistema que haya sido pre-armado con este fin, en vez de crear uno de cero. Éstos son llamados *frameworks*, y son tratados al final de este capítulo.

4.6. Cerrando grietas con *polyfills*

Desde que HTML5 y CSS3 fueron introducidos a la escena del desarrollo web, hace ya algunos años, muchos adelantados han ido incluyendo en sus proyectos elementos propios de estas nuevas tecnologías de manera indiscriminada. Si bien las últimas versiones de los navegadores más populares soportan casi todas las novedades – más información al respecto se detalla en el apartado 2.5.3 – aún hay usuarios que no han migrado a ellos, por variadas razones.

La mayor parte de los problemas de compatibilidad son ocasionados por versiones antiguas de *Internet Explorer* que no son capaces de interpretar nuevos elementos en HTML5, como por ejemplo <video>, <audio>, <section>, entre otros. Al no reconocerlos, meramente los ignora, es decir, no permite que se le puedan agregar estilos, y mucho menos funcionalidad. A su vez, navegadores desactualizados en teléfonos móviles de gama baja, o simplemente viejos, sufren la misma problemática.

Si bien los desarrolladores deberían diseñar en función de todos sus usuarios, posean la última tecnología o no, estas nuevas características son provechosas para el buen funcionamiento de los sistemas, y verse limitado en el desarrollo por navegadores desactualizados es desalentador.

Ya Jakob Nielsen había establecido una regla sobre estas cuestiones en su libro *Designing web usability*, en la que recomienda evitar utilizar cualquier tecnología web innovadora hasta que hayan pasado, por lo menos, uno o dos años luego de su lanzamiento oficial y estable (1999), para así evitar que un gran porcentaje de los usuarios se vean impedidos de experimentarla. Si bien es un plazo prudente, es importante resaltar que la especificación de HTML5 aún no es estándar, a pesar de que tiene un buen soporte en los navegadores modernos.

Para suplir la falta en dichos casos, que suelen ser específicos, existe una serie de soluciones que apunta a unificar la experiencia de la mayoría de los usuarios. Los *polyfills* son pequeñas librerías desarrolladas en *Javascript* que hacen creer al navegador que los elementos nuevos en verdad existen, para que los consideren como tal.

El término polyfill fue originado [...] como una alusión a llenar las grietas en los navegadores antiguos con Polyfilla (conocido como masilla de pegar en los EE.UU.). Por lo tanto, un polyfill es un suplemento de JavaScript que replica de manera efectiva nuevas características en los navegadores más antiguos. (Frain, 2012, p. 98)

Estos *scripts* garantizan que la experiencia sea por lo menos un poco más democrática para todos los navegantes, sin importar su entorno. A su vez, han sido desarrollados *polyfills* que se aseguran que los usuarios móviles, y no solamente los de escritorio con

versiones viejas de *Internet Explorer* puedan disfrutar estos avances tecnológicos por igual.

Probablemente el *polyfill* más conocido y utilizado es *Modernizr*, una completa librería que detecta carencias en el navegador de acceso y, además de permitir el uso de elementos de HTML5, habilita la posibilidad de incluir condicionalmente nuevos *polyfills* específicos para casos más puntuales.

Si bien estas soluciones son útiles y efectivas, no dejan de generar una petición al servidor en cada carga. Esto puede conllevar a problemas de rendimiento en dispositivos móviles. “Los polyfills no siempre son una buena idea para los navegadores móviles [...] debido a la sobrecarga de rendimiento que añaden a las (por lo general más antiguas y menos capaces) plataformas móviles” (Firtman, 2013). Por esta razón, sería prudente hacer una evaluación previa sobre el público al que apunta el proyecto web en cuestión para identificar si la utilización de una librería de este tipo se ve justificada, considerando todos los factores.

4.7. La era *touch*: gestos táctiles

Una particularidad que presentan la mayoría de los dispositivos móviles actuales, y que debe ser tomada en cuenta por los diseñadores web para ofrecer una excepcional experiencia a sus usuarios, es la capacidad de ser manipulados mediante gestos táctiles. Como ya se han distinguido anteriormente, gran porcentaje de los *smartphones* y las *tablets* suelen portar pantallas *touch*, siendo ésta, en la mayoría de los casos, la única manera de interactuar con el aparato. Es importante aclarar que actualmente se comercializan, también, diversos modelos de computadoras de escritorio equipadas con la funcionalidad táctil, aunque éstas no están limitadas únicamente a dicha modalidad de manejo y operación.

Esta condición presenta a quienes encaren un proyecto que contemple dispositivos móviles en su espectro de presentación, un desafío especial. Si bien estos aparatos

están optimizados para operar los navegadores con gestos táctiles predefinidos para todas las páginas por igual, como por ejemplo, la simulación de *scroll* al deslizar el dedo sobre la pantalla en línea recta desde la parte inferior a la superior de la misma, y viceversa, existe la posibilidad de automatizar ciertos comandos con gestos definidos manualmente.

Un ejemplo común y utilizado frecuentemente es el de un carrusel de fotografías. En una computadora de escritorio, la manera de desplazarse por las imágenes es utilizando el click del *mouse* en botones estratégicos que comandan el paso de las mismas. En dispositivos móviles, es posible realizar lo mismo, aunque en vez de utilizar un *mouse* – inexistente en este caso – se presiona con el dedo, y en ciertas plataformas con un lápiz óptico, sobre dichos botones en la pantalla.

Pero utilizando gestos táctiles hay muchas opciones intuitivas para el usuario que pueden ser aplicadas para optimizar su experiencia. Una de ellas sería establecer un gesto similar al de *scroll*, pero horizontal, es decir que pase las fotografías *dibujando* una línea recta desde la derecha hacia la izquierda en caso de avanzar, y para el lado contrario en caso de retroceder.

Este tipo de mejoras, aplicadas con creatividad y conciencia, pueden generar una sensación positiva en el usuario con respecto al sitio, y aumentar las posibilidades de conversión del mismo.

El diseño de gestos es cada vez más frecuente a través de las aplicaciones móviles, pero por desgracia, lo mismo no puede decirse para los sitios web. Muchos de ellos se diseñan como aplicaciones y luego pierden grandes oportunidades de utilizar gestos. Usar algunos va a mejorar la usabilidad, ayudar a aumentar visitas desde móviles y quién sabe qué más. Diseñar gestos táctiles debe estar tan alto en la lista de prioridades como el diseño web responsivo (Simpson, 2013).

Para aplicarlas técnicamente es necesario recurrir a la utilización de librerías en *Javascript* especialmente desarrolladas para reconocer gestos táctiles nativos de cada dispositivo, y que les asignan diferentes eventos, los cuales serán disparados cuando el usuario los accione.

4.8. Marche un *combo* para llevar: *frameworks*

En términos generales, desarrollar un sitio web que cumpla con un enfoque responsivo no debería ser muy complicado para profesionales experimentados. Si bien requiere que se ajusten algunos modelos mentales pre-existentes de la concepción del trabajo que puedan llegar a tener, migrar hacia esta técnica implica práctica, creatividad y paciencia.

Empero, los diseñadores que recién se insertan en la industria, o quienes se avocan al diseño web de manera esporádica o recreacional, pueden sentirse abrumados con el caudal de conocimientos que deben ser empleados para poner el diseño responsivo en práctica.

Afortunadamente, hay disponibles para el uso masivo algunas herramientas que intentan derribar estas barreras, y facilitar el trabajo colectivo para una Web más estandarizada e inclusiva. Es muy común, y además es considerado buena práctica en el campo del desarrollo de *software*, la publicación y distribución de instrumentos y funcionalidades de manera *open source*, es decir, de código abierto, para que puedan ser utilizadas, mejoradas y hasta redistribuidas por cualquier usuario. Dentro de este marco, se encuentran distintos tipos de soluciones, de diferentes niveles de complejidad, que sirven de base para desarrollar proyectos propios. Son conocidas como *frameworks*.

Hay muchas clases de éstos en el ámbito del desarrollo web, pero los que competen a la temática de este PG se dividen en las siguientes categorías: *frameworks* de grillas, *frameworks* de interfaz de usuario y *frameworks mobile*.

Los sistemas de grillas pre-armados facilitan al diseñador la creación de las propias, ofreciendo un módulo listo para usar, con documentación extendida para que pueda ser aplicado como base de cualquier proyecto en cuestión de minutos. Al estar basados únicamente en CSS, éstos aportan una hoja de estilos ya codificada, y una serie de nombres de clases que deben ser aplicados a los `<div>` en los que se vaya a armar la estructura, para que el flujo se genere automáticamente. Muchas veces, los que recaen en esta categoría son también llamados *boilerplates*, término que puede traducirse como

plantilla. No todos los *frameworks* CSS son ofrecidos en modalidad responsiva, sino que son grillas de ancho fijo. Para el propósito que aquí se está persiguiendo, aquellos deben ser descartados. Algunas buenas opciones que sí son adaptables son *Less Framework*, *Skeleton*, *Responsive Grid System*, y *Unsemantic*, entre otras.

La siguiente clasificación de *frameworks* es inclusiva de la anterior, pero aún más compleja. Estos módulos tienen como base también un sistema de grillas responsivo, pero además aportan elementos gráficos pre-diseñados que sirven para unificar la interfaz de usuario. En casi todos los casos son mucho más robustos e implican menor intervención del diseñador, ya que muchas decisiones estéticas ya están resueltas. Es recomendable, igualmente, también sentarlos solamente como piso para la construcción de diseños propios, y no utilizarlos tal como se ofrecen, por el hecho de que, al ser masivos, se corre el riesgo de ser visualmente repetitivos. Se consideran como las mejores opciones disponibles a *Bootstrap*, *ZURB Foundation*, *Gumby*, y *Pure*.

Por último, existen *frameworks* mucho más complejos que apuntan más allá del diseño de un sitio o aplicación web, sino que persiguen la facilitación para los desarrolladores de la funcionalidad, además de la estética. Estos módulos son todavía más robustos que los anteriores, y están diseñados exclusivamente para su utilización en aplicaciones *mobile*, aunque sean web o híbridas. Generalmente, requieren un nivel de conocimiento más avanzado, porque aplican técnicas de programación en su núcleo, pero igualmente cuentan con documentación detallada, además de una comunidad de usuarios activa. Las opciones más populares dentro de esta categorización son *jQuery Mobile*, *PhoneGap*, *Sencha Touch* e *Ionic*.

Basándose entonces en una herramienta pre-armada, es más sencillo dar el salto hacia un proyecto más sólido, y preparado para afrontar las muchas situaciones que la gigante y diversa Web actual propone a los desarrolladores y usuarios.

Capítulo 5: De la teoría a la práctica: diseño web responsivo en acción

A lo largo del desarrollo de este Ensayo, se han presentado desde conceptos inherentes al diseño web en sus bases hasta nociones cada vez más específicas que moldean el *workflow* en la actualidad profesional dentro de este campo. Se ha tratado de recorrer junto al lector un camino que se apega a las buenas prácticas, y de ofrecerle a éste herramientas útiles que le permitan mejorar su propio trabajo. Todo esto se ha hecho, sin embargo, desde la teoría y mediante la explicación de situaciones cotidianas y ejemplos prácticos, para que él mismo pueda armar las piezas del conocimiento junto con las que ya posee, de forma que construya su propio flujo de trabajo adaptable a su realidad.

Este capítulo busca demostrar la aplicación efectiva de todo el marco teórico ya presentado en muestras concretas, de manera que el lector sienta confianza en su capacidad para luego implementarlo en sus propios proyectos.

Actualmente, en la Argentina, son numerosos los sitios que ya abarcan una iniciativa hacia su expansión a dispositivos móviles. La mayoría de ellos lo hacen porque, como se ha comprobado, no pueden darse el lujo de ignorarlo debido a la cantidad de tráfico que reciben desde estas vías de entrada. Tantos otros, porque entienden desde su visión de negocios que el futuro se aloja allí. Sea cual fuere el caso, es importante resaltar la intención de cambio. A continuación se realiza una investigación al respecto, a modo de ilustrar los planteos realizados.

5.1. ¿Quién lo usa?: análisis de aplicación

Para delinear un panorama general de la escena local, se realiza un relevamiento de los sitios más visitados por los argentinos a la fecha (junio de 2014), para luego analizarlos, en rasgos generales, en la manera de manejar a sus usuarios móviles. Tomando como base los datos estadísticos proporcionados por Alexa (2014), la empresa de medición de data sobre tráfico en Internet considerada más confiable, se seleccionan los primeros diez portales nacionales que figuran en su lista de sitios más visitados por los argentinos.

Es pertinente aclarar que en ella se listan también variados sitios globales, pero el fin de este análisis es tomar únicamente los que tienen base en el país. Por orden de mayor a menor cantidad de visitas se ubican: Mercado Libre, La Nación, Clarín, Infobae, Taringa!, Olé, TN, Santander Río, Ciudad.com y Minutouno.com.

Este listado puede ser dividido en categorías que permiten comprender, por lo menos de forma general, de qué manera cada uno de ellos elige servir su contenido a todos sus usuarios, especialmente a los que acceden desde plataformas móviles. Dicha categorización es: No cuenta con interfaz optimizada para dispositivos móviles; Ofrece un sitio web apartado en móviles; y Presenta web diseñada responsivamente.

Dado que uno de los objetivos de este PG es promover el uso del diseño web responsivo, los resultados de estas observaciones servirán como crítica al desarrollo web actual en el país, y además buscan utilizarse como validación a la hipótesis planteada. Es pertinente señalar que durante el transcurso del desarrollo de este análisis – aproximadamente un mes – dos de los portales renovaron su interfaz, migrando hacia una propuesta más inclusiva.

Para la comprobación de visualización, se hicieron pruebas en cuatro dispositivos: una notebook *HP Pavilion dv5* corriendo con *Windows 7*, de resolución 1366x768; un *smartphone Google Nexus 4*, de resolución 768x1280 en posición vertical; un *smartphone Samsung Galaxy SIII Mini*, de resolución 480x800 en posición vertical; y una *tablet Samsung Galaxy Tab 10.1 2*, de resolución 1280x800 en posición horizontal y viceversa en vertical. Los últimos tres dispositivos corren con *Android*, en diferentes versiones. Para unificar las pruebas, dado que es un análisis superficial, en todos los casos se utiliza *Google Chrome* como navegador predeterminado.

En el Cuerpo C se desglosan visualmente las interfaces por plataforma, para cada sitio analizado. Las capturas fueron realizadas por la autora en cada uno de los dispositivos mencionados.

Mercado Libre redirecciona el tráfico que ingresa desde dispositivos medianos a pequeños, los *smartphones*, a un sitio diferente con un diseño creado especialmente para estos casos. Éste a su vez invita al navegante a descargar la aplicación nativa que la empresa distribuye, ya que buscan centralizar en esa herramienta a todos los usuarios móviles. Tanto en *desktops* como en *tablets* se visualiza el diseño completo, aunque escalado en resoluciones menores, ya que no se encuentra diseñado responsivamente.

Un caso similar presenta el diario digital La Nación. El sitio, cuando es visto desde una computadora de escritorio, puede comprobarse que no es adaptable. A su vez, en *tablets* se muestra la misma versión. Caso contrario ocurre con los *smartphones*, ya que para ambos se ofrece una versión acotada del contenido, que se ajusta a sus límites.

El siguiente portal es también un diario digital, Clarín. Éste ha cambiado radicalmente su diseño de manera reciente. Anteriormente, presentaba prácticamente los mismos lineamientos que los dos antecesores. El sitio no se encontraba diseñado de manera adaptable, si bien desde su desarrollo presentaba algunas técnicas de esta práctica, como el uso de *media queries* para mostrar y ocultar contenido condicionalmente. La diferencia con los previos es que solamente en el *smartphone* más pequeño se ofrecía un diseño adaptado a sus características, mientras que en el más grande, como también en la *tablet* y la *notebook* se podía apreciar el *full-site*, obteniendo, en los dos primeros casos, una experiencia limitada ya que los tamaños no eran ideales para una correcta lectura de la información. Si bien el enfoque que proponían era claramente limitado – considerando los objetivos planteados en la hipótesis de este capítulo –, la nueva interfaz presentada aún no hace uso del diseño web responsivo para su mejora. La misma es líquida, es decir, no utiliza anchos fijos y estáticos, lo cual permite que el diseño fluya en pantallas de escritorio de tamaños variados. Sin embargo, éste detecta el punto de acceso y sirve una interfaz adaptada las características de ese dispositivo. Si bien no redirecciona a una página de inicio distinta, es fácilmente reconocible que no hace uso de técnicas de diseño web responsivo *per se*.

El siguiente sitio, Infobae, también es un portal de noticias y su diseño sí es responsivo. A todos los dispositivos les sirve exactamente el mismo contenido, el cual automáticamente se adapta al contenedor físico en el cual se lo visualiza. En todos ellos se observa una experiencia óptima de lectura. Este sitio es retomado más adelante, ya que se utiliza como caso de estudio por sus propiedades responsivas.

El portal de intercambio de información Taringa!, también viró el eje en el planteo de su interfaz. Al comienzo de este análisis, presentaba un único diseño que se replicaba en los tres dispositivos más grandes, el cual estaba originalmente destinado para *desktops*. En el *smartphone* más chico, sin embargo, detectaba que se trataba de un aparato móvil, y realizaba entonces un redireccionamiento a un nuevo sitio optimizado para *mobile*. Pese a esto, el proceso de detección que realizaba no era muy eficaz ya que dejaba afuera dos plataformas a las cuales debería haber provisto de una alternativa, en donde la visualización del diseño original era defectuosa. Recientemente, como se puede observar en el apartado de Taringa! del Cuerpo C, este portal cambió su diseño por uno fluido que se adapta para pantallas grandes y medianas, es decir, la misma interfaz aplica tanto en computadoras de escritorio, como en las *tablets* en ambas orientaciones. Empero, no se utilizan los principios de diseño web responsivo a su máxima capacidad. Es posible observar que el mismo no escala bien en medidas menores. Por esta razón, siguen manteniendo el uso de un detector de dispositivos, que en caso de identificar el acceso desde un móvil, dirige al usuario a una versión exclusiva para ese tipo de plataformas, la cual sirve otro contenido condicionado y otro diseño acotado.

Olé es también un portal periodístico, aunque deportivo. La estructura en el diseño es similar a la que se observó en los diarios digitales anteriormente analizados. Sin embargo, el enfoque que aquí realizan es deficiente en comparación a sus pares. En pantallas de computadoras de escritorio se provee la versión completa del mismo, la cual no se adapta en absoluto a medidas menores de las que establece el diseño. Es decir, no es fluido, sino que de ancho fijo. No obstante, en dispositivos con resoluciones menores

realiza un proceso de detección de acceso, para luego redirigir al usuario a un sitio aparentemente optimizado para las características del aparato. En las pruebas llevadas a cabo, todos los dispositivos fueron detectados de la misma manera – a todos los identificó como *Android* – dirigiéndolos a una misma interfaz para móviles que no contempla distinciones de tamaños entre sí. Por esta razón, es posible observar en las capturas presentadas en el Cuerpo C como la visualización del sitio en *tablets* es deficiente, ya que se sirve exactamente el mismo diseño que a dispositivos de menor porte, salvo que de manera estirada.

El sitio web de TN, si bien también es un portal de noticias, se despegó del diseño clásico y quizás más conservador de los casos analizados anteriormente. El caso que presentan en relación a lo aquí analizado, es particular, ya que el sitio sí se encuentra diseñado responsivamente, pero a su vez presenta un sitio apartado para algunos móviles. En resoluciones superiores, el diseño es de ancho completo y líquido, es decir se adapta a los límites, tanto de pantallas de escritorio y *notebooks* como en *tablets*. A pesar de ser responsivo y escalable, al ingresar al sitio desde dispositivos móviles más pequeños, como es el caso de los *smartphones*, éste redirige al usuario a una versión diferente especialmente optimizada para aquellos contextos. Es un caso especial, ya que el diseño original posee la capacidad de adaptarse a esta clase de resoluciones, y esto es comprobable mediante pruebas de escalabilidad. Sin embargo, dada la estructura de diseño poco convencional que utilizan, es entendible que el mismo no degrade bien en esos casos, y sea necesario ofrecer una alternativa mejor optimizada cuando éste detecta que el punto de acceso no es favorable a su propuesta.

El sitio web del banco Santander Río, si bien fue rediseñado recientemente, no presenta ningún tipo de solución a usuarios de dispositivos móviles. En pruebas con diferentes plataformas se pudo comprobar que provee el mismo diseño y contenido en todos los casos, el *full-site*, o diseño pensado para resoluciones estándares de máquinas de escritorio y *laptops*. El navegador en los móviles fue capaz de reconocer esta

complicación, por lo que habilita el *zoom* para que la estructura pueda visualizarse en su totalidad. No obstante, esto únicamente dificulta aún más su navegación, ya que la lectura de la información es tediosa. Si bien únicamente se analiza el portal principal, es pertinente aclarar que uno de los usos más corrientes del sitio es el *home-banking*, el cual tampoco se encuentra optimizado para su navegación en resoluciones limitadas.

Ciudad.com, un portal de noticias de entretenimiento, repite el patrón observado en algunos de los sitios ya analizados. La versión *full-site* del mismo está diseñada con una medida de ancho fija, la cual no escala en caso de que la resolución sea menor a la establecida arbitrariamente. Esto puede observarse en la *tablet*, en ambas orientaciones, donde se sirve exactamente el mismo diseño, el cual no se adapta al contenedor, sino que el navegador del aparato hace *zoom* sobre éste para que pueda observarse en su totalidad. Si bien como ya se ha mencionado, esto suele perjudicar la lectura en algunos casos, aunque los tamaños predefinidos ayudan a no perder legibilidad en esta oportunidad. Cuando se accede desde dispositivos más pequeños, sin embargo, el sitio realiza un proceso de detección y redirecciona al usuario a una versión del mismo, acotada y optimizada para la visualización en dichos contextos.

El último sitio de esta lista, Minutouno.com, está planteado de la misma manera que la mayoría de los casos precedentes: ignorando a las *tablets* como dispositivos móviles, sirviéndoles el mismo diseño y contenido que a pantallas de resoluciones estándares de escritorio – el cual también es de ancho fijo y estático –, para luego solamente detectar plataformas de resolución inferior como móviles, redirigiendo a estos usuarios a una página principal distinta, configurada para sus contextos.

Partiendo del análisis sobre cómo los sitios que manejan mayor tráfico de usuarios en Argentina tratan de remediar la problemática móvil, es posible concluir que, en términos generales, las soluciones provistas no son ideales. Un solo caso utiliza diseño web responsivo en su totalidad, mientras que algunos otros intentan un acercamiento al mismo, aunque limitado. Se ha demostrado que el aislamiento de usuarios de

smartphones es un común, ya que muchos de estos sitios lo realizan y aparentemente lo encuentran viable. Como ya se ha planteado anteriormente, esta práctica no es recomendable dado que su correcta aplicación requiere una inversión mayor de recursos y es aún más rebuscado su mantenimiento. Además, como se puede observar en los casos planteados, no siempre esta solución se encuentra bien implementada, lo que suele generar una pobre o mala experiencia en los usuarios.

También se ha observado que a pesar de manejar tráfico que lo avale, aún hay sitios y portales que no consideran pertinente una migración hacia la expansión, y siguen relegando a sus usuarios móviles a tener una experiencia deficiente, la cual probablemente generará descontento de ellos hacia con el producto o sitio en cuestión.

5.1.1. *Responsive web design* en las grandes ligas: Infobae

Considerando las conclusiones obtenidas de las observaciones previas sobre los sitios más visitados en la Argentina, es meritorio resaltar al diario digital Infobae, el cual se encuentra a la vanguardia en su campo ya que provee a todos sus usuarios la mejor experiencia posible adaptada a sus contextos y necesidades. Como ya se ha desarrollado, la puesta en práctica de los conceptos de diseño web responsivo ayudan considerablemente no solamente al usuario, el cual se ve beneficiado y es incluido sin importar su dispositivo de acceso, sino también a los desarrolladores y quienes deban mantener el sitio, los cuales ven su porción de trabajo disminuida y facilitada. Es por eso que se selecciona este caso para desglosar de qué manera se aplica la técnica en detalle.

Desde el punto de vista estructural, el diseño se divide en columnas, sobre las cuales se establecen los elementos de manera fluida. Estas columnas no necesariamente escalan proporcionalmente cuando cambia la resolución, sino que mediante declaraciones de estilo en bloques de *media queries* se reestablecen los anchos porcentuales de las mismas para que se reacomoden de manera acorde. Si bien al definir los puntos de

quiebre sobre los cuales se realizarán los ajustes en CSS no es posible apuntar a toda clase de dispositivos existente de manera que cada uno obtenga un diseño puntual, es viable establecer una serie de valores comprobados como límites, para que luego los elementos – que tienen anchos fluidos y adaptables – se acomoden líquidamente a ellos. Los diseñadores de *Lúcuma Labs*, responsables del diseño del sitio de Infobae, sientan base en esa fórmula para conseguir una estructura que se rearma agradidamente.

A su vez, dada la cantidad de información que es posible mostrar en pantallas que lo permiten, se hace uso constante de la muestra condicional de contenidos. Es decir, tanto en la pantalla principal, como en noticias y secciones interiores, cuando la cantidad de datos excede a la que se puede visualizar de manera clara y entendible, se recurre a suprimir información y textos complementarios que entorpecen el flujo de lectura. Esto se define fácilmente desde CSS en un bloque de *media query* que apunte a cierta resolución de pantalla, donde luego se oculta el elemento en cuestión mediante el uso de la propiedad *display: none*.

Hay, además, conceptos que se desprenden de la partida estética del diseño que ayudan a una mejor aplicación del diseño web responsivo. Considerando que Infobae es un portal de noticias, en donde el contenido es *el rey*, la elección de la paleta de colores beneficia ampliamente a poder generar un diseño limpio, legible, y a su vez adaptable, ya que el correcto uso del blanco y la moderación en cantidad de colores da al usuario una sensación de buen uso de los espacios, en donde no ve el contenido *aplastado* en sus márgenes.

Un aspecto importante en cualquier sitio web, que además toma mayor valor cuando se trata de interfaces para móviles, es la navegación. En el caso de Infobae, la versión de escritorios, además de otras barras complementarias, ya presenta la misma navegación que luego se replica en resoluciones más pequeñas: un enlace a Secciones precedido por el ícono de tres líneas apiladas, generalmente reconocido como botón de menú. Al seleccionarlo, se abre una solapa del lado izquierdo que lista todos los hipervínculos

pertinentes. Generalmente, esta clase de menú solapado se utiliza únicamente en versiones móviles, dado que el espacio es reducido. Igualmente, los diseñadores de la interfaz de Infobae decidieron establecerlo en todas las versiones. En el sitio completo, igualmente, aparecen otros accesos de navegación que son luego ocultados condicionalmente cuando el sitio se escala, dependiendo de la permisividad del espacio disponible.

En conclusión, se puede apreciar que este diseño se ha realizado de manera consciente sobre la forma de uso que hacen de él la totalidad de sus navegantes, y que un exhaustivo análisis de UX ha sido llevado a cabo y luego puesto en práctica en el armado de la propuesta final. Sería provechoso que más *gigantes* del mercado local tomaran los aportes realizados por Infobae y su impronta de inclusión equitativa, de manera que todos los usuarios del país puedan disfrutar de una mejor experiencia a la hora de navegar.

5.1.2. Pensando un portfolio: rediseño del sitio web personal de la autora

Como se ha mencionado anteriormente, la autora de este Proyecto se desempeña de forma profesional actualmente como maquetadora, por lo cual su entorno de trabajo permanente es la creación y diseño de interfaces efectivas y usables para la Web actual. Esto quiere decir que procura que sus proyectos cumplan con los máximos estándares de calidad para que los usuarios de los mismos puedan disfrutar de la mejor experiencia que se les pueda ofrecer. Esta línea de pensamiento ha sido plasmada a lo largo del presente Ensayo, y es a su vez implementada en sus creaciones.

Durante el transcurso del mes de marzo de 2014, se realizó el rediseño y posterior actualización de contenidos del sitio web personal de la autora, ubicado en <http://www.amireynoso.com.ar>. Esta clase de sitio suele ser conocido en el ámbito del diseño como portfolio, ya que es una exhibición de los trabajos más destacados realizados por el o los diseñadores.

El principal objetivo para plasmar en esta transición era la correcta aplicación de los conceptos de diseño web responsivo, ya que se consideraba importante destacar la creatividad impuesta en la utilización de esta técnica, teniendo en cuenta que es una cualidad muy valorada por reclutadores laborales a la hora de seleccionar candidatos para puestos en el área.

Desde el punto de vista estético, se buscó mantener una coherencia visual desde la utilización de una paleta de colores limitada pero dinámica. A su vez, se realizó la elección de una familia tipográfica que fuese maleable para ser utilizada tanto para grandes cantidades de texto como para titulares, y que a su vez escalase bien en tamaños reducidos sin perder legibilidad.

Partiendo desde una concepción minimalista del diseño, en donde el contenido debe destacarse por sí mismo, se comenzó a desarrollar el sitio en términos técnicos. Para esto se utilizó como base estructural el *framework Bootstrap*, anteriormente mencionado, dado que ofrece numerosos recursos pre-armados para ahorrar pasos en el flujo de trabajo. Empero, tal como se recomendó también, estas bases fueron manipuladas para adecuarse al partido gráfico propio del diseño propuesto.

Es posible apreciar que se ha sacado provecho del sistema de grillas, sobre el cual se construyen los cimientos de la estructura. Los elementos se encuentran en ella establecidos, y la misma, al ser flexible, permite que éstos se reacomoden dinámicamente.

La página principal cuenta con dos aspectos que ya han sido planteados. Por un lado, los ítems del portfolio que se muestran por debajo de la introducción tienen una cuadrícula propia, que sigue sus propias reglas de adaptación a diferentes pantallas. Es decir, no escala igual que el resto de las columnas, sino que fueron diseñadas especialmente para que su contenido, el elemento de mayor relevancia para el diseñador, se muestre siempre de la mejor manera posible. Por otro lado, es de suma importancia resaltar que las imágenes de estos ítems son a su vez elásticas, flexibles y que se adaptan a su

contenedor, es decir, la columna que las contiene. Sin embargo, no se tiene extremada consideración por el peso de las mismas, lo cual puede llegar a ocasionar demoras en su carga en dispositivos con conexión limitada.

Se puede observar que si bien se ha intentado que el diseño funcione de la mejor manera posible para todos los dispositivos, hay una falla de planificación a la hora de mostrar el contenido extendido de cada ítem del portfolio. Esta información se muestra en forma de ventana modal, una clase de *pop-up* que se posiciona por delante del contenido y que se carga de manera dinámica. Aquí se presenta una galería de imágenes, y en los casos que corresponde, un video, además de texto en dos columnas. En condiciones de visualización amplias, es decir, en computadoras de escritorio y hasta en *tablets*, este contenido se ve correctamente ya que tiene espacio disponible para acomodarse cómodamente. No obstante, en parámetros de visualización más reducidos, éste comienza a *enlatarse*, es decir, reducirse al punto de mostrarse comprimido, de manera que no permite una lectura fluida. Además, tanto las imágenes como los videos reducen su ancho al del dispositivo, por lo cual se muestran en tamaños minúsculos.

Si bien el sitio no cuenta con una cantidad extensiva de páginas interiores, cada una de ellas ha sido trabajada con especial cuidado para que cumplan su objetivo bajo distintos parámetros de visualización.

La navegación adopta un formato diferente en dispositivos pequeños, el cual facilita al usuario *clickear* sobre sus enlaces. Para mantener la consistencia, se siguió la misma lógica para mostrar las categorías del portfolio en la página principal, como los enlaces a las secciones del *curriculum vitae*. Otro componente en esta página que toma una morfología diferente a la planteada originalmente es la línea de tiempo. Sus datos se reacomodan y se suprimen elementos gráficos meramente decorativos para optimizar la lectura de la información.

En el desarrollo de este proyecto personal se obvió un aspecto que ha sido tratado en este Ensayo y cuya utilización le hubiese dado un nivel invaluable de funcionalidad y

usabilidad en móviles: los gestos táctiles. Siguiendo el ejemplo que se desarrolló en la explicación de los mismos, el del carrusel de imágenes, dicha implementación podría haber sido aplicada para navegar entre las diferentes ventanas modales de los items del portfolio. Éstas tienen ya un botón que permiten ir hacia adelante y hacia atrás, pero se ha observado en pruebas reales de usuarios sobre este mismo diseño que ellos entienden que esa funcionalidad le es innata. Se sugiere entonces una futura actualización del mismo para incluir esta capacidad.

En términos generales, entonces, este sitio aplica correctamente las técnicas de diseño web adaptable, ya que la experiencia de usuario en la mayoría de los casos se encuentra optimizada.

Para la comparación de visualizaciones en diferentes dispositivos, se han utilizado exactamente los mismos aparatos mencionados anteriormente para las pruebas de análisis en los sitios más visitados de Argentina.

5.2. Ganan todos: beneficios del uso de esta práctica

Durante el proceso de este trabajo se han ido remarcando diferentes razones por las cuales el diseño web responsivo es una alternativa viable y eficiente para democratizar el acceso a los contenidos y recursos en la Web.

Desde un comienzo se ha tratado de dar cuenta que por más que la filosofía propuesta no sea el camino a tomar, es primordial establecer en cualquier proyecto web, nuevo o en la actualización de uno pre-existente, la necesidad de incluir a los usuarios que accedan a él desde plataformas móviles en la experiencia que se les brindaría a quienes entren desde dispositivos con parámetros de visualización considerados normales. Si bien se intenta dejar conocimiento que esta técnica es ideal en la mayoría de los casos, es importante resaltar que cualquier intento de mejora para esta clase de usuarios es mejor que ninguno.

El diseño web responsivo ha demostrado ser, mediante su aplicación en la gran cantidad de proyectos que la han tomado como propia, una solución tan maleable y líquida como la Web misma. Al tomar de ella sus capacidades de adaptación, es posible realizar, con mínimo esfuerzo por parte del desarrollador como con bajo peso de carga en relación a otras técnicas, una experiencia de usuario cuasi-perfecta para cada situación, si se aplica con conciencia y creatividad.

La tecnología y el desarrollo en la Web se encuentran en constante evolución, lo cual significa que en un futuro puedan generarse otras prácticas para atacar esta problemática que sean aún más sencillas y convenientes. Tomando en consideración la realidad presente y hasta un futuro cercano, el diseño web responsivo puede ser considerado un escalón estable en una larga escalera de crecimiento y constante mejora a la que comúnmente se la llama *World Wide Web*.

Conclusiones

Durante el transcurso del desarrollo de este Proyecto se han desglosado temáticas que no deberían faltar en la maleta de recursos de un diseñador web. Comenzando por el marco teórico que envuelve a la práctica en general, y más adelante ahondando en cuestiones más específicas relativas a la actualidad digital, tomando como base datos reales que cimientan la hipótesis aquí planteada.

Tanto la teoría como la información estadística han demostrado sustentar la validez del pedido de soluciones por parte de los usuarios móviles, por lo cual este trabajo apunta a desmitificar la problemática y traerla a la agenda vigente de quienes encaren el diseño y desarrollo para la Web como profesión.

En términos generales, se considera que los contenidos desarrollados en este Ensayo presentan pertinencia académica y profesional, además de éstos estar actualizados a las últimas vanguardias y tendencias que se presentan en el campo del diseño web de manera casi constante.

A través del proceso de creación y desarrollo de este trabajo, la autora consiguió ampliar, mejorar y actualizar su saber sobre la materia aquí tratada. Para poder fundamentar la teoría presentada a lo largo de *Multiplicidad digital*, fue necesario hacer una recopilación extensa de textos publicados en el tema, sobre la cual, luego de analizar cada ejemplar propuesto en profundidad, se seleccionaron los títulos cuyo contenido podía aportar más y mejor sustento al mismo. La mayor parte de la bibliografía se encuentra únicamente publicada en idioma extranjero, por lo que hubo que, además, realizar un desglose de manera aún más consciente, para poder rescatar lo que fuese realmente de calidad en relación a las temáticas específicas planteadas.

Este proceso de deconstrucción y posterior selección de información valiosa significó una parte importante del desarrollo del Proyecto en sí, ya que para poder obtener datos demostrativos fue necesario leer cada título prácticamente en su totalidad, de manera reiterada. Esta práctica, lejos de ser tediosa, significó para la autora, una actividad

enriquecedora que le proveyó de conocimientos invaluable que son armas fundamentales en su propio arsenal, que como profesional del área representa un salto cualitativo sobre otros pares de la misma.

A su vez, se considera que dicho proceso de aprendizaje se ve reflejado en el relato del presente escrito. El mismo se ha desarrollado durante un transcurso prolongado de tiempo, en el cual, dada la vorágine con la que la Web avanza, los contenidos en él tratados han ido mutando y a su vez perfeccionándose. Es por eso que la temática planteada requirió de una actualización dinámica de información, para poder transmitir siempre datos válidos a la realidad digital actual. Uno de los casos en donde esta variación inmediata pudo observarse en este trabajo, es en el análisis de los casos de estudio en el quinto capítulo. En un momento incierto durante el transcurso del desarrollo de aquella investigación, algunos sujetos cambiaron su morfología, por lo que fue necesario realizar concesiones y permutas en las conclusiones del análisis, para que se adapten a las novedades.

La formulación del contenido desarrollado fue articulado de manera que el lector sea participe de un proceso de aprendizaje, dependiendo claramente de los conocimientos con los que él cuente previamente. Esta estructura fluye desde lo general hacia lo particular, pudiendo así abarcar el recorte de la temática habiéndolo hecho primero un repaso de los fundamentos que sostienen al diseño web. Siguiendo esta lógica, se busca cumplir uno de los objetivos de *Multiplicidad digital*, de servir como material de consulta.

Se considera que la pregunta disparadora que inspiró la realización del PG, ¿cómo puede el diseño web para dispositivos móviles potenciar cualquier sitio?, ha sido ampliamente respondida. Por medio del desglosamiento de datos e información estadística sólida se logró delinear un panorama general sobre el futuro cercano en relación al uso de dispositivos portátiles. Este pronóstico funciona como justificativo de la importancia que debe atribuírsele a este mercado. Tomando como punto de partida este análisis, junto con un recorrido por la tecnología que construye el mundo móvil, se presentaron diversas

técnicas que facilitan la conversión de un sitio web estático – hablando de la adaptabilidad del mismo – hacia uno dinámico que cambie su morfología en función del contexto de visualización del usuario.

A partir del tercer capítulo en adelante, se hace foco en una práctica que permite portar un sitio web a variados formatos de tamaño, de manera sencilla y fluida, el *responsive web design*. Considerando que para llevarlo a cabo no se requieren extensivos conocimientos de programación, ni herramientas externas al propio trabajo del autor en principio, se sugiere que el diseño web responsivo es la manera más adecuada de hacer frente a la problemática planteada, para la mayoría de los casos. Si se realiza de manera correcta, prestando particular atención a la optimización en cada caso, es posible generar un producto poderoso y universal. Éste, entonces, será capaz de generar una sensación de simpatía en los usuarios móviles, tan frecuentemente olvidados e ignorados, lo cual probablemente signifique en un aumento en la posibilidad de conversión de los mismos hacia lo que el sitio trate de comunicar y transmitir.

Los aportes que se desean plasmar, tanto en la sociedad como en la articulación curricular y pedagógica de la materia, es la relevancia del recorte de la temática y la problemática que presenta la carencia de su aplicación en la realidad digital del momento, por lo menos en la escena nacional.

Este Proyecto busca entonces convertirse en una herramienta con la cual hacerle frente, y poder servir como material de aprendizaje y actualización a profesionales que se desempeñen hoy por hoy en el área, a generaciones venideras de estudiantes y a entusiastas de la disciplina que desean ampliar sus conocimientos. Funciona, a su vez, como una demostración viva sobre los beneficios que puede generar dar el salto final hacia el terreno móvil, adelantándose al futuro.

En conclusión, se considera entonces que *Multiplidad digital* no sólo cumple con sus objetivos, sino que los trasciende, transformándose en un manual de consulta, además de una justificación en sí mismo, sobre la necesidad de una Web más justa e inclusiva.

Imágenes seleccionadas

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Ejemplo de página web</title>
5     <link rel="stylesheet" href="estilos.css">
6   </head>
7   <body>
8     <p>Presionando el <strong>siguiente enlace</strong>,
9     usted podrá visitar el buscador
10    <a href="http://google.com">Google</a>.</p>
11  </body>
12 </html>
```

Figura 1: Anatomía de una página web.

Fuente: Captura de pantalla realizada por la autora.

```
1 p {
2   font-family: Arial, Helvetica, sans-serif;
3   font-size: 11px;
4   color: black;
5   line-height: 200%;
6   text-align: justify;
7 }
```

Figura 2: Declaración de estilo en CSS.

Fuente: Captura de pantalla realizada por la autora.



Nombre:
Ami Reynoso

E-mail:
amireynoso

Por favor ingresa una dirección de e-mail válida.

Asunto:
Contacto

Mensaje:
Probando el funcionamiento de Javascript en el formulario.

Enviar

Figura 3: Formulario de contacto en sitio de juegos creado por la autora dentro del contexto universitario. Fuente: <http://www.artelpinguino.com.ar/contacto.php>

| Unidades absolutas en CSS | |
|----------------------------------|---|
| <i>px</i> | Unidad relativa al dispositivo de visualización. Para la visualización en pantalla, por lo general un píxel es un punto de la pantalla. |
| <i>mm</i> | Un milímetro |
| <i>cm</i> | Un centímetro (100 milímetros) |
| <i>in</i> | Una pulgada (2,54 centímetros) |
| <i>pt</i> | Un punto (1/72 de pulgada) |
| <i>pc</i> | Una pica (12 puntos) |


Tabla 1: Unidades absolutas en CSS. Fuente: Mozilla (2014). *Mozilla Developer Network*. <length>.

Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS/length>

| Unidades relativas en CSS | |
|---|---|
| <i>Relativas al tamaño de fuente</i> | |
| <i>em</i> | Representa el tamaño de fuente calculado del elemento. Si se utiliza en la propiedad <i>font-size</i> en sí, representa el tamaño de fuente heredado del elemento. |
| <i>ex</i> | Representa la altura de la x de la fuente del elemento. En fuentes con la letra 'x', <i>ex</i> es generalmente la altura de las letras minúsculas; $1ex \approx 0.5em$. |
| <i>ch</i> | Representa el ancho, o más precisamente la medida de antelación del carácter cero, 0, en la fuente del elemento. |
| <i>rem</i> | Representa el tamaño de fuente del elemento raíz (por ejemplo, el tamaño de fuente del elemento <html>). Cuando se utiliza en el tamaño de fuente para este elemento raíz, representa su valor inicial. |
| <i>Relativas al tamaño de la pantalla gráfica</i> | |
| <i>vh</i> | 1/100avo del alto de la pantalla gráfica |
| <i>vw</i> | 1/100avo del ancho de la pantalla gráfica |
| <i>vmin</i> | 1/100avo del valor mínimo entre la altura y la anchura de la ventana gráfica |
| <i>vmax</i> | 1/100avo del valor máximo entre la altura y la anchura de la ventana gráfica |

Tabla 2: Unidades relativas en CSS. Fuente: Mozilla (2014). *Mozilla Developer Network*. <length>.

Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS/length>

 **Aplicación nativa**
de Twitter (v. 5.7.0) para Android (v. 4.4.2)
en LG Google Nexus 4




 **Aplicación web**
de Twitter (Mayo 2014) accedida desde
Firefox (v. 29.0) en LG Google Nexus 4



Figura 4: Aplicaciones nativas vs. web.

Fuente: Capturas de pantalla realizadas por la autora.



Orientación horizontal
de la aplicación Candy Crush Saga (v. 1.29.0)
para Android (v. 4.4.2) en LG Google Nexus 4



Orientación vertical
de la aplicación Candy Crush Saga (v. 1.29.0)
para Android (v. 4.4.2) en LG Google Nexus 4

Figura 5: Orientaciones de pantalla.

Fuente: Capturas de pantalla realizadas por la autora.

Media queries

definidas dentro del atributo *media* en la etiqueta *link* dentro de un documento HTML

```
<link rel="stylesheet"
      media="screen and (min-width: 701px) and (max-width: 900px)"
      href="estilos.css" />
```

Media queries

definidas en un bloque *@media* dentro de un documento CSS

```
@media screen and (min-width: 701px) and (max-width: 900px) {
  body {
    font-size: 16px;
  }
}
```

Figura 6: Declaraciones de *media queries*.

Fuente: Capturas de pantalla realizadas por la autora.

Lista de referencias bibliográficas

- Alexa (2014). *Top sites in Argentina*. Recuperado el 08/06/2014 de:
<http://www.alexacom/topsites/countries/AR>
- Arhippainen, L. y Tähti, M. (2003). *Empirical Evaluation of User Experience in Two Adaptive Mobile Application Prototypes*. *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia, 10–12 Diciembre de 2003, Norrköping, Suecia*. Citado en: Hassan Montero, Y. y Martín Fernández, F. J. (2005). *La Experiencia del Usuario*. Disponible en:
http://www.nosolousabilidad.com/articulos/experiencia_del_usuario.htm
- Armada de los Estados Unidos (2013). *NIAPS Glossary of Terms*. Disponible en:
http://www.public.navy.mil/spawar/PEOEIS/NAVY311/Pages/NIAPS_Glossary.htm
- Ballard, B. (2007). *Designing the Mobile User Experience*. Chichester: John Wiley & Sons, Ltd.
- Boulton, M. (2009). *A Practical Guide to Designing for the Web*. Penarth: Mark Boulton Design Ltd.
- Budiu, R. (14 de Septiembre de 2013). *Mobile: Native Apps, Web Apps, and Hybrid Apps*. Disponible en: <http://www.nngroup.com/articles/mobile-native-apps/>
- CERN (2014). *The birth of the web*. Disponible en: <http://home.web.cern.ch/topics/birth-web>
- Cisco Systems (2014). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018*. Disponible en:
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html
- Firtman, M. (2013). *Programming the Mobile Web* (2^{da} ed.). Sebastopol: O'Reilly Media, Inc.
- Frain, B. (2012). *Responsive web design with HTML5 and CSS3*. Birmingham: Packt Publishing.
- Frost, B. (22 de Agosto de 2012). *Separate Mobile Website Vs. Responsive Website* [posteo en blog]. Disponible en:
<http://www.smashingmagazine.com/2012/08/22/separate-mobile-website-vs-responsive-website-presidential-smackdown-edition/>
- Gabriel-Petit, P. (2013). *UX Matters: Glossary*. Disponible en:
<http://www.uxmatters.com/glossary/>

- Google (2013). *Google HTML/CSS Style Guide*. Disponible en: <http://google-styleguide.googlecode.com/svn/trunk/htmlcssguide.xml>
- Grannell, C., Sumner, V. y Synodinos, D. (2012). *The essential guide to HTML5 and CSS3 web design*. Nueva York: friendsofED, Apress Media.
- Gustafson, A. (2011). *Adaptive web design: crafting rich experiences with progressive enhancement*. Chattanooga: Easy Readers, LLC.
- Hassan Montero, Y. y Ortega Santamaría, S. (2009). *Informe APEI sobre usabilidad*. Gijón: Asociación Profesional de Especialistas en Información. Disponible en: <http://www.nosolousabilidad.com/manual/>
- International Organization for Standardization (1998). *ISO 9241-11:1998. Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability*. Ginebra: ISO. Citado en: Usability Professionals' Association (2013). *About Usability. What is usability?*. Disponible en: http://www.upassoc.org/usability_resources/about_usability/definitions_of_usability.html
- Itzkovitch, A. (10 de Agosto de 2012). *Designing For Device Orientation: From Portrait To Landscape* [posteo en blog]. Disponible en: <http://www.smashingmagazine.com/2012/08/10/designing-device-orientation-portrait-landscape/>
- jQuery (2013). *What is jQuery?*. Disponible en <http://www.jquery.com/>
- Kadlec, T. (2012). *Implementing responsive design: building sites for an anywhere, everywhere web*. Berkeley: New Riders.
- Krug, Steve (2000). *Don't Make Me Think! A Common Sense Approach to Web Usability*. Berkeley: New Riders. Citado en: Usability Professionals' Association (2013). *About Usability. More Definitions of Usability*. Disponible en: http://www.upassoc.org/usability_resources/about_usability/definitions.html
- Kyrnin, J. (2013). *What is web design. An introduction to the basics of web design*. Disponible en: <http://webdesign.about.com/od/webdesign/qt/what-is-web-design.htm>
- Lara Navarra, P. y Martínez Usero, J. A. (2006). *La accesibilidad de los contenidos web*. Barcelona: Editorial UOC. Citado en: Pardo Vargas, J. F. (2011). *Accesibilidad web y discapacidad. Una web sin barreras*. Proyecto de Graduación. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo.
- Lawson, B. (2012). Why We Shouldn't Make Separate Mobile Websites. En Smashing Magazine (Ed.). *Essentials of mobile design* (p. 16-25). Friburgo de Brisgovia: Smashing Media GmbH.

- Luzardo Alliey, A. M. (2011). *Diseño de la interfaz gráfica web en función de los dispositivos móviles. Caso de estudio: diarios digitales*. Tesis de Maestría. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo.
- Marcotte, E. (2011). *Responsive web design*. Nueva York: A Book Apart, LLC.
- McLellan, C. (3 de Marzo de 2014). *The History of Tablet Computers: A timeline* [posteo en blog]. Disponible en: <http://www.zdnet.com/the-history-of-tablet-computers-a-timeline-7000026555/>
- Mozilla (2014). *Mozilla Developer Network*. <color>. Disponible en: https://developer.mozilla.org/en-US/docs/Web/CSS/color_value
- Net Market Share (2014). *Mobile/Tablet Operating System Market Share*. Recuperado el 20/04/2014 de: <http://www.netmarketshare.com/report.aspx?qprid=8&qptimeframe=M&qpsp=182&qpch=350&qpmr=100&qpdt=1&qpct=2&qpcustomd=1&qpcid=fw511749&qpf=1>
- Niederst Robbins, J. (2012). *Learning web design* (4^{ta} ed.). Sebastopol: Littlechair, Inc.
- Nielsen, J. (10 de Abril de 2012). *Mobile Site vs. Full Site*. Disponible en: <http://www.nngroup.com/articles/mobile-site-vs-full-site/>
- Nielsen, J. (1999). *Designing web usability*. Berkeley: New Riders.
- Raasch, J. (2012). How To Build A Mobile Website. En Smashing Magazine (Ed.). *Essentials of mobile design* (p. 26-44). Friburgo de Brisgovia: Smashing Media GmbH.
- Reeves, M. (5 de Abril de 2013). *How To Sell The Value Of Mobile To Clients* [posteo en blog]. Disponible en: <http://www.smashingmagazine.com/2013/04/05/how-to-sell-the-value-of-mobile-to-clients>
- Simpson, J. (7 de Agosto de 2013). *Gesture design with Javascript*. Disponible en: <http://www.inserthtml.com/2013/08/gesture-design-javascript/>
- StatCounter (2014). *StatCounter Global Stats: Top 10 Desktop Screen Resolutions from Mar 2009 to April 2014*. Recuperado el 19/04/2014 de: <http://gs.statcounter.com/#desktop-resolution-ww-monthly-200903-201404-bar>
- Toff, J. (2009). *What is a web browser?* [video de YouTube]. Recuperado el 06/06/2013 de: <http://googleblog.blogspot.com.ar/2009/10/what-is-browser.html>
- World Wide Web Consortium (2013a). *Browser Statistics*. Recuperado el 08/06/2013 de: http://www.w3schools.com/browsers/browsers_stats.asp
- World Wide Web Consortium (2013b). *What is HTML?*. Disponible en: <http://www.w3.org/html/>

World Wide Web Consortium (2014). *Browser Statistics*. Recuperado el 20/06/2014 de:
http://www.w3schools.com/browsers/browsers_stats.asp

Wroblewski, L. (2011). *Mobile first*. Nueva York: A Book Apart, LLC.

Bibliografía

- Ballard, B. (2007). *Designing the Mobile User Experience*. Chichester: John Wiley & Sons, Ltd.
- Barry, N., Clark, J., Frost, B., Kardys, D., Keith, J., Koch, P., Nudelman, G., Olsen, D., Rieger, S., Toledo, A., Van Der Merwe, R., y Walton, T. (2012). *The mobile book*. Friburgo de Brisgovia: Smashing Media GmbH.
- Beaird, J. (2010). *The principles of beautiful web design* (2^{ta} ed.). San Fransisco: SitePoint Pty. Ltd.
- Boulton, M. (2009). *A Practical Guide to Designing for the Web*. Penarth: Mark Boulton Design Ltd.
- Fling, B. (2009). *Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps*. Sebastopol: O'Reilly Media, Inc.
- Frain, B. (2012). *Responsive web design with HTML5 and CSS3*. Birmingham: Packt Publishing.
- Grannell, C., Sumner, V. y Synodinos, D. (2012). *The essential guide to HTML5 and CSS3 web design*. Nueva York: friendsofED, Apress Media.
- Gustafson, A. (2011). *Adaptive web design: crafting rich experiences with progressive enhancement*. Chattanooga: Easy Readers, LLC.
- Hassan Montero, Y. y Ortega Santamaría, S. (2009). *Informe APEI sobre usabilidad*. Gijón: Asociación Profesional de Especialistas en Información. Disponible en: <http://www.nosolousabilidad.com/manual/>
- Kadlec, T. (2012). *Implementing responsive design: building sites for an anywhere, everywhere web*. Berkeley: New Riders.
- Luzardo Alliey, A. M. (2011). *Diseño de la interfaz gráfica web en función de los dispositivos móviles. Caso de estudio: diarios digitales*. Tesis de Maestría. Buenos Aires: Facultad de Diseño y Comunicación. Universidad de Palermo.
- Marcotte, E. (2011). *Responsive web design*. Nueva York: A Book Apart, LLC.
- Niederst Robbins, J. (2012). *Learning web design* (4^{ta} ed.). Sebastopol: Littlechair, Inc.
- Nielsen, J. (1999). *Designing web usability*. Berkeley: New Riders.
- Shedroff, N. (2009). *Design is the problem. The future of design must be sustainable*. Brooklyn: Rosenfeld Media.

Smashing Magazine (Ed.) (2012). *Essentials of mobile design*. Friburgo de Brisgovia: Smashing Media GmbH.

Smashing Magazine (Ed.) (2012). *Responsive design*. Friburgo de Brisgovia: Smashing Media GmbH.

Villamor, C., Willis, D. y Wroblewski, L. (2010). *Touch gesture reference guide*. Disponible en: <http://static.lukew.com/TouchGestureGuide.pdf>

Wroblewski, L. (2011). *Mobile first*. Nueva York: A Book Apart, LLC.