



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**PROCESO DE MAQUETACIÓN DE UN SITIO WEB
MEDIANTE TWITTER BOOTSTRAP 3**

TESIS

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTAN:

**GUSTAVO ADOLFO LUQUE CONTRERAS
MARIO FRANCISCO CORRO GONZÁLEZ**

**DIRECTOR DE TESIS
CARLOS ALBERTO ROMÁN ZAMITIZ**

CIUDAD UNIVERSITARIA mayo 2015.





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Gustavo:

No podría estar más agradecido con Dios por haberme puesto a la familia que tengo; en especial a mis padres, que sin su apoyo incondicional nada de esto sería posible.

A todos esos profesores que forjaron mi desarrollo académico en la institución más maravillosa de todas, la UNAM.

A esos grandes amigos que conocí a lo largo de la carrera, Belman, Corro. El mejor equipo de todos. #teamDonker.

Y a todas esas personas que se han cruzado en mi camino y que han sido de gran bendición para mi vida.

Francisco:

Agradezco a mis padres por su incondicional apoyo a lo largo de todo el tiempo en mi formación personal y académica a pesar de los errores que cometí.

A todas las instituciones involucradas en mi formación profesional en especial a mi alma mater, la UNAM.

A todos mis amigos que han estado en difíciles momentos y me han mostrado apoyo incondicional y especialmente a los que me acompañaron en esta etapa final, la Universidad, Gustavo y Roberto #teamDonker.

A todos los que indirecta e directamente han influido para este logro.

Índice

1. Introducción.....	1
2. Definición del problema.....	2
3. Marco teórico – práctico.....	9
3.1. Internet y la Web.....	9
3.1.1. La arquitectura cliente servidor.....	11
3.1.2. <i>HTTP</i> : El idioma del cliente y del servidor.....	16
3.1.3. La Web y sus inicios.....	17
3.1.4. La evolución de la Web, sus estándares y tecnologías.....	19
3.1.5. La competencia de los navegadores web.....	21
3.1.6. Un acercamiento al presente.....	26
3.1.7. Lo que aporta el uso de estándares para la Web.....	29
3.1.8. ¿Cuáles estándares, exactamente?.....	32
3.2. Los modelos de negocio en Internet.....	37
3.2.1. Los negocios electrónicos (<i>e-Business</i>).....	38
3.2.2. El comercio electrónico (<i>e-Commerce</i>).....	43
3.2.3. Tipos de modelos de negocio en <i>Internet</i>	48
3.2.4. El impacto del buen diseño de un sitio web en el comercio electrónico.....	51
3.3. El diseño web.....	53
3.3.1. Planificación del sitio.....	56
3.3.2. Disciplinas en el desarrollo web <i>frontend</i>	58
3.3.3. Capas en el diseño y desarrollo web <i>frontend</i>	62
3.3.4. Otras tecnologías que operan en la Web.....	72
3.3.5. Interacción, experiencia e interfaz gráfica para el usuario.....	79
3.3.6. Documentos de diseño.....	80
3.3.7. Malas prácticas en el diseño web.....	83
3.4. Diseño web responsivo.....	93

3.4.1. Introducción al diseño web responsivo.....	94
3.4.2. Estructura y uso de los <i>media Queries</i>	104
3.4.3. Otras formas de usar los <i>media Queries</i>	111
3.4.4. Diseñando de manera responsiva.....	116
4. Análisis del cliente: Alindis.....	129
4.1. Breve análisis de mercado.....	129
4.2. Análisis de requerimientos del cliente (No funcionales).....	131
5. Propuesta de solución.....	147
5.1. ¿Por qué <i>Bootstrap 3</i> ?.....	147
5.2. Metodología ágil <i>RAD</i>	155
6. Implementación de la solución.....	157
6.1. Creación del prototipo de la vista frontal del sitio mediante <i>Twitter Bootstrap 3</i>	157
6.2. Integración y pruebas.....	185
6.3. Resultados de pruebas.....	189
6.4. Los beneficios aportados al proceso de maquetación del sitio.....	195
7. Conclusiones finales.....	199
Referencias.....	205

“Pienso que hay una profunda y perdurable belleza en la simplicidad, en la claridad, en la eficiencia. La verdadera simplicidad es derivada de mucho más que solo la ausencia de ornamentación excesiva; ésta consiste en traer orden a la complejidad”.

Jonathan Ive, 2013.

1. Introducción

La Web es la plataforma virtual que representa una de las muchas formas de acceder a la información en *Internet*. Ha crecido constantemente y cada día más y mejores sitios y aplicaciones la conforman. Desde la aparición de los primeros sitios con fines informativos hasta la creación de grandes portales con propósitos educativos, la Web evoluciona constantemente y ha dado lugar a la aparición de nuevas tendencias.

El objetivo del presente trabajo es exponer la forma en la que se desarrolló la primer etapa de la solución a una necesidad muy fundamental para las pequeñas y medianas empresas de hoy: Contar con presencia en Internet, específicamente en la Web. Dicha primer etapa consistió en la creación de un prototipo o maqueta de sitio web para una pequeña empresa que vende diversos sistemas y productos relacionados con la seguridad para hogares, edificios, fábricas, etcétera; léase, cámaras de seguridad, interfonos, CCTV y cercas electrificadas.

La tecnología avanza a pasos agigantados y las nuevas funcionalidades son aprovechadas al máximo. Tal es el caso que hace algún tiempo era difícil creer que sería posible tener a la mano información de cualquier índole gracias a un teléfono celular; dispositivo que, con mucha influencia del crecimiento de la Web y sus tendencias, evolucionó hasta lo que hoy en día se reconoce como uno de los más grandes valores tecnológicos de la historia para el ser humano: el *Smartphone*.

En el capítulo 3.2 se realiza un preámbulo sobre el impacto que en la actualidad los dispositivos móviles han tenido en la *Internet* y específicamente en la *Web*; al grado de cambiar las tendencias en el comercio electrónico. Se abordan los temas relacionados al comercio electrónico, para más adelante, en el capítulo 3.3, dar un salto a uno de los temas pilares de esta tesis: El diseño web.

Más adelante, partiendo del capítulo 3.4, se ahonda en otro tema crucial para este trabajo de tesis: el diseño web responsivo. Como se menciona el dicho capítulo; en un inicio, las páginas web eran diseñadas con el pre-conocimiento de que sólo a través de computadoras se accedía a ellas. Tuvieron que ser “rediseñadas” para ser cómodamente accedidas desde pantallas más pequeñas. Hoy, la manera en que se maquetan sitios web debe cubrir el importante aspecto del diseño adaptativo o responsivo; tanto así que *Google*, una de las empresas líderes en tecnología, ha anunciado que su motor de búsqueda (el más popular y usado hasta hoy, desde su creación), dará prioridad a sitios con esta característica contemplando que en un futuro muy probable se dejen de considerar aquellos que no la cumplan, pasando a ser así sitios olvidados y prácticamente inútiles.

A partir del capítulo 4 se pone en marcha todo lo necesario para abordar la solución que se propone. Se realiza un breve análisis del perfil y las necesidades del cliente, para que en el capítulo 5 y 6 se exponga sobre la herramienta utilizada, el proceso, así como la metodología que se llevó a cabo durante este proyecto.

2. Definición del problema:

El desarrollo y diseño web es ya considerado un arte [1]. La vista que proporciona un sitio influye en gran medida en el impacto que tendrá sobre los usuarios que navegan sobre éste [2]. Diseñar un sitio web no es una tarea sencilla; durante este proceso suelen surgir distintos problemas relacionados con la inconsistencia y el mantenimiento; pues cada uno de los ingenieros, diseñadores y programadores involucrados en esta etapa del proyecto tiene un estilo distinto de solucionar los problemas y de escribir código; por ende, las inconsistencias son inevitables y las dificultades de mantenimiento y actualización de la interfaz del sitio se tornarían bastante complicadas [3]. Para reducir este tipo de complicaciones existen los *frameworks*¹, los cuales son marcos de trabajo que nos trazan un esquema o patrón para el proceso de desarrollo, diseño y/o implementación de una aplicación.

La problemática fundamental se presenta cuando el cliente, una pequeña empresa que vende sistemas de vigilancia y seguridad requiere un solución para contar con presencia en la Web. Sin embargo, el simple hecho de contar con un sitio web para una pequeña empresa, no garantiza que dicho medio represente un importante factor para el incremento de clientes potenciales y mucho menos para que dichos clientes se conviertan en activos [5]. Según estudios realizados en 2002 [6], el 75% de las personas que visitan un sitio de ventas en *Internet* juzgan la credibilidad del negocio o comercio basándose en su diseño o aspecto visual; lo

¹ O marco de trabajo, conjunto de conceptos y prácticas estandarizadas para lidiar con un tipo de problema en común [4].

cual significa que un buen diseño o maquetación del sitio web se relaciona de manera directa con la tasa de conversión² para el negocio que representa [7].

La calidad de un sitio comienza con su propio diseño y demás atributos relacionados a la usabilidad. Lo anterior se desglosa en seis aspectos fundamentales relacionados al proceso de diseño web [8]:

- La multitud y variedad de dispositivos con acceso a la Web
- Los estándares Web
- Mejora continua de los sitios web
- Diseño web responsivo
- Accesibilidad
- Desempeño del sitio

Un sitio que satisfaga los puntos mencionados tiene muy altas probabilidades de ser uno de los principales factores de éxito para el individuo o negocio [9]. La palabra 'sitio' en el contexto de la *Internet* hace referencia a la presencia de una entidad, organización, negocio, etcétera, en la Web, que en términos prácticos es un subconjunto de esta inmensa red mundial de computadoras interconectadas. Los sitios web son ampliamente usados por corporaciones o negocios (independientemente de su tamaño) para proveer información acerca de sus productos y/o servicios, atraer visitas y, en el mejor de los casos, convertir dichas visitas en compras o transacciones exitosas [7].

² Tasa de conversión es el cociente del número de logros (compras) realizadas por los visitantes entre el número de visitas al sitio. En otras palabras, de todos los visitantes del sitio, cuántos se convierten en compradores [7].

Al día de hoy, la competencia en el mercado electrónico ha dado un giro violento y resulta vital para toda empresa acaparar con diversas estrategias de mercado la participación del cliente (en el inglés, *Customer share*). Un buen diseño del sitio, buenas estrategias de mercado y técnicas de optimización del posicionamiento en buscadores son actividades que juegan un papel crucial en el impulso del individuo o negocio para alcanzar sus metas y objetivos. La presencia en línea (*on-line*) ha derribado las barreras que se presentan para aquellos que apenas inician su propio negocio y ha provisto a éstos de un muy importante alcance global sin costos elevados [10].

Respecto a la multitud y variedad de dispositivos que hoy en día existen y que se conectan a la Web, los sitios deberían siempre contar con un diseño responsivo [10], el cual viene siendo una de las partes más importantes que la problemática expuesta engloba.

Cuando en el año 2007 fue introducido el *smartphone*³ comercialmente conocido como *iPhone* [12], se dio un giro revolucionario para el diseño web. Los desarrolladores y diseñadores contaban con un "lienzo", región o espacio de trabajo definido, el cual estaba regido por el tamaño estándar de los monitores de los equipos de cómputo; sí había variaciones en sus tamaños, pero eran mínimas. Los desarrolladores entonces comenzaron a elaborar sitios optimizados para equipos móviles como el *iPhone*; éstos eran aparte del sitio para monitores de tamaño "estándar" (tamaño de pantalla de una computadora fija o de escritorio).

³ Teléfono celular con una computadora integrada y funciones características que lo distinguen de los teléfonos celulares tradicionales, tal como un sistema operativo, navegación web y entorno de ejecución de aplicaciones de software [11].

Dar mantenimiento a dos sitios por separado quizás no parecía tan tedioso o complicado; sin embargo, más variedad de tamaño en dispositivos móviles comenzaron a surgir tras el primer *iPhone*, incluyendo *tablets*⁴ (y más adelante equipos de cuyo tamaño oscilaba entre un *smartphone* y una *tablet*, llamados *phablets*⁵). Fue entonces cuando los desarrolladores se dieron cuenta que crear una maquetación específica del sitio para cada tamaño de dispositivo posible resultaba realmente complicado e incluso prácticamente imposible. Era necesaria una solución a este problema; es decir, una manera en la que un solo sitio funcionara con cualquier tamaño de pantalla posible o dicho de otra forma, que el sitio se adaptara de manera automática al tipo de dispositivo y la manera de desplegar el contenido en pantalla dependiera de ello [8].

Desarrolladores y diseñadores de todo el mundo unieron esfuerzos para crear el Diseño web Responsivo; una técnica para desarrollar páginas web flexibles y que no dependen de un tamaño definido de monitor o pantalla, sino que son capaces de detectar el tipo de dispositivo, así como las medidas que posee en pantalla y acomodar el diseño (sin perder coherencia) para proveer al usuario final una experiencia de navegación óptima desde su dispositivo móvil [10].

Realizar un prototipo de sitio no es una tarea complicada por sí sola; sin embargo, hay estadísticas muy importantes por considerar antes de poner manos a la obra. Estudios por parte de la compañía *Quantcast* [13] demuestran que cuatro de cada cinco medios digitales tienen una web (o sitio) optimizada para dispositivos móviles. Del total el 33% de su tráfico proviene de estos dispositivos.

⁴ Tableta electrónica, dispositivo computacional móvil con pantalla de interfaz táctil. [14]

⁵ Dispositivo que se caracteriza por ser la combinación entre un *Smartphone* y una *Tablet* [14].

En general, el 31% del tráfico de la web le atañe a estos aparatos. Con todo lo anterior resulta casi imprescindible el uso de técnicas para desarrollo web responsivo.

Al realizar el análisis de las necesidades del cliente, resultó fácil percatarse que el diseño en cuanto a componentes gráficos tendrían que ser cuidadosamente elaborados y diseñados. Sumado a esto, los sitios de hoy en día presentan cierta interactividad desde que el usuario realiza acciones como presionar un botón, posicionar el cursor sobre algún elemento, recorrer el contenido de la página, redistribuir los elementos de acuerdo con el tamaño de pantalla, etcétera. Desde luego, dicha interactividad y funcionalidad también tendrá que ser incorporada. Todo esto por supuesto, demanda tiempo y esfuerzo. Dado que no se cuenta con la experiencia necesaria en el campo de diseño gráfico para satisfacer al cliente en ese rubro, ¿Existe alguna herramienta que agilice la tarea de maquetación del sitio solicitado y que aporte además un paquete de elementos y plantillas flexibles para incorporar componentes, bloques e interactividad de una manera más ordenada, ágil y coherente? Afortunadamente sí [4]. *Twitter Bootstrap* es un potente marco de trabajo (*framework*) que simplifica este proceso. ¿Cómo podría beneficiar el uso de *Bootstrap* en su versión más actual al momento de la elaboración de esta tesis, para que, sin ser especialistas en diseño gráfico para sitios web, se puedan cubrir adecuadamente los requerimientos del cliente respecto a la vista de su sitio, y a su vez crear una interfaz responsiva?

3. Marco teórico - práctico.

3.1 *Internet y la Web.*

Internet se ha convertido en uno de los canales de comunicación y distribución de información más importantes (si no es que el más importante). Entender cómo y por qué funciona no resulta tan sencillo para todos; sin embargo, se abordará un poco sobre lo que es y cuáles son los fundamentos que hacen de este medio masivo de comunicación lo que hoy se sabe que es [8].

Se trata de una red enorme de computadoras interconectadas. Todo lo que ahora es *Internet* fue construido con base en estándares. Dado que esta red global no es gobernada por una persona u organización, las decisiones concernientes a cómo lograr la implementación de una funcionalidad o característica específica han sido tomadas a través de esfuerzos cooperativos (invención, discusión y adopción) para finalmente establecer la manera en la dicha característica o funcionalidad se llevará a cabo. Sin embargo, existe un grupo de personas que se encargan de la operatividad y crecimiento funcional de *Internet*. Desde antes del surgimiento de la Web, el Grupo de Trabajo de Ingeniería de *Internet* (*IETF*, por sus siglas en inglés) es el responsable de ello [8].

Técnicamente, *Internet* es el espacio abstracto (infraestructura de red global) operado por un sistema de estándares y reglas. El propósito de conectar millones de computadores de todo el mundo es compartir información. Hay muchas formas en las que las computadoras conectadas en esta red pueden intercambiarse información, los métodos para lograrlo se encuentran estandarizados y son conocidos como protocolos. El fundamento o base para lograr la operatividad

completa de la *Internet* es la familia de protocolos centralizados *TCP/IP*⁶, los cuales permiten el envío y recepción de información a través de dispositivos de comunicación desde un punto a otro a lo largo y ancho del globo terráqueo [16].

La Web es la plataforma en la cual el modelo de intercambio de información se basa en documentos de hipertexto, los cuales son visualizados por medio de un navegador de *Internet* y se accede a ellos por medio de hiperenlaces. En términos prácticos, la Web es un subconjunto de *Internet* [15].

Para que todo lo anterior resulte posible, se utiliza el protocolo *HTTP*⁷ (*Hypertext Transfer Protocol*) el cual forma parte de la familia de protocolos *TCP/IP*. *HTTP* se encarga de la comunicación entre el servidor web y el cliente (navegador web), por lo que, se utiliza dicho protocolo cada vez que se realice una petición al servidor; con lo cual, éste responderá desplegando su respuesta en una página web (documento de hipertexto) que el cliente podrá visualizar [16].

La Web está compuesta por una inmensa cantidad de clientes (*Mozilla Firefox, Safari, Google Chrome, Internet Explorer, Opera*; los cuales son aplicaciones de software conocidas como navegadores web) y de servidores web, conectados. Lo que hace de la Web un sitio más y más rico en contenido e interesante son las aplicaciones y páginas web que despliegan la información y nos permiten interactuar con ella [7]. En la figura 3.1 se puede apreciar una forma sencilla de ver cómo está constituida la Web.

⁶ *TCP/IP* (*Transmission Control Protocol - Internet Protocol*, por sus siglas en inglés) es el conjunto de protocolos que operan en *Internet* y que permiten la comunicación entre los distintos dispositivos interconectados [16].

⁷ *HTTP* (*Hyper Text Transfer Protocol*) protocolo encargado de llevar a cabo la comunicación entre el navegador web (cliente) y un servidor web (servidor) [16].

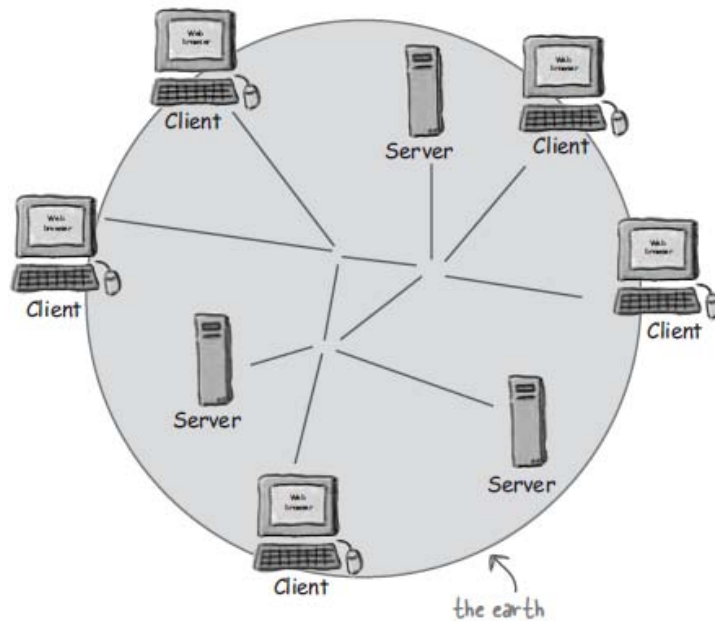


Figura 3.1 Una manera sencilla de ver la Web. Conformada por un conjunto de servidores y clientes y su comunicación se realiza a través de protocolos establecidos [17].

3.1.1 La arquitectura cliente servidor.

Dentro de la terminología de *Internet* es muy frecuente escuchar el término “arquitectura cliente-servidor”; dicho término se refiere al modelo de red en el cual un dispositivo conectado a *Internet* (o un proceso que genera) funge como el cliente o servidor. El servidor consiste en uno o más ordenadores (computadoras) generalmente con importantes capacidades de procesamiento y almacenamiento, dado que comúnmente procesa peticiones de múltiples clientes a través de una red de área amplia (*WAN*⁸) [17].

⁸*WAN* proviene del término en inglés *Wide Area Network*. *Internet* cae en esta categoría o clasificación de red [17].

¿Qué hace el cliente?

El término “cliente” por lo general hace alusión tanto al usuario como a la aplicación web (léase, navegador web) que éste está utilizando. El cliente no es más que un componente de software (navegador web) que sabe cómo debe comunicarse con el servidor, pero para que éste pueda mostrar el resultado de tal forma que el humano detrás del dispositivo pueda entender la respuesta recibida, se utiliza *HTML*⁹; que no es más que el lenguaje basado en etiquetas que los documentos de hipertexto utilizan para presentar o imprimir la información de tal forma que el usuario no tenga problema alguno en visualizar el contenido. En otras palabras, es la “forma humana” de presentar en pantalla la respuesta del servidor al cliente [17]. En la figura 3.2 se ilustra a modo de pasos, cómo se lleva a cabo una comunicación *HTTP* básica.

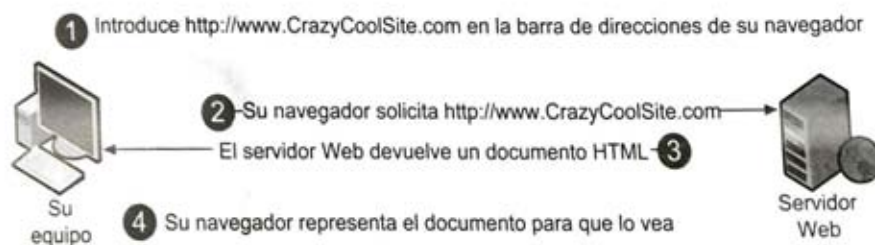


Figura 3.2 Los pasos esenciales que muestran cómo funciona una petición *HTTP* básica [18].

⁹*HTML (Hypertext Markup Language)* es el lenguaje de marcado de hipertexto que se encarga de estandarizar la manera en la que el navegador web deberá mostrar la información en pantalla para que el usuario la entienda [18].

¿Qué hace el servidor?

Los servidores en las redes informáticas son equipos que generalmente poseen un poco más en cuanto a capacidades de procesamiento y almacenamiento; cuentan con una dirección *IP*¹⁰ (como cualquier otra máquina conectada), la cual debe ser fija¹¹. Los servidores, por lo general, y como lo indica su nombre, corren servicios a través de procesos denominados “procesos demonios” que son simplemente procesos ejecutándose en segundo plano listos para recibir conexiones (en terminología informática, “escuchar conexiones”). Algunos ejemplos de este tipo de servidores son [17]:

- Servidor web *Apache*
- Servidor de archivos *FTP*¹²
- Servidor de conexiones con *Windows Samba*¹³
- Servidor de acceso remoto *SSH*¹⁴

Para conectarse a estos servicios, los usuarios deben, desde sus computadoras (clientes) utilizar “software cliente” para consumir el servicio correspondiente, por ejemplo, un usuario utiliza el navegador *Internet Explorer* para acceder al servicio web (*HTTP*) *Apache*. Sin embargo, de lo anterior se deriva la siguiente interrogante: ¿Cómo se comunican los clientes con el servidor si éste tiene

¹⁰La dirección *IP* (de *Internet Protocol*) es un número que identifica a un dispositivo dentro de una red (ya sea red interna o global), léase una computadora, un teléfono, una cámara *IP*, etcétera [16].

¹¹Los servidores informáticos necesariamente deben contar con una dirección *IP* fija o estática (ésta no cambia con el tiempo), ya que de esta forma se permite su localización en *Internet* [19].

¹²*File Transfer Protocol*, o Protocolo de Transferencia de Archivos, como su nombre indica, es un protocolo para transferir archivos (mediante la comunicación cliente-servidor) en *Internet* [16].

¹³Protocolo de *Microsoft* que permite que sistemas *UNIX* (ver nota 16) puedan fungir como servidores o clientes en redes del sistema *Windows* [19].

¹⁴*Secure Shell*, protocolo que permite el acceso a la terminal de comandos de otra computadora conectada en la red y poder controlarla de manera remota [19].

muchos servicios disponibles (Sí, un servidor generalmente corre más de un servicio) y una sola dirección *IP*? Esto se logra mediante puertos de acceso. Cada servicio tiene asignado un puerto de acceso y los clientes, para conectarse y utilizar el servicio ingresaran a través de éste. Un ejemplo muy común para ilustrar esto sería cuando un usuario “A” hace uso de un navegador web para conectarse a determinado sitio, se tomará el sitio *elpais.com* como ejemplo. El servidor al que “A” se está conectando tiene la dirección *IP* 91.216.63.241. Gracias a los sistemas de nombres de dominio (*DNS*¹⁵) no es necesario introducir en el navegador esta dirección, basta con ingresar el “nombre de dominio” que le corresponde (en este caso, *elpais.com*). El servicio web *Apache* funciona por defecto en el puerto 80; de esta manera, el navegador siempre incluye el número de puerto al final de la dirección *IP* de manera implícita para indicarle al servidor al que se conecta que accederá al servicio web: *91.216.63.241:80* [19].

Posteriormente, el servidor se encarga de recibir la petición por parte de “A”, la procesa; encuentra el recurso solicitado y le devuelve una respuesta. La mayoría de las ocasiones, esta respuesta es una página *HTML* (en otras ocasiones podría ser un fichero). La respuesta enviada al cliente o usuario “A” no siempre será la que se espera; ya que, por ejemplo, si el recurso no fue hallado, se devolverá un código de error (404) indicando que dicho recurso no pudo ser encontrado y el dispositivo cliente (la máquina o computadora usada por “A”) se encargará de interpretar dicho error desplegando un mensaje entendible para el usuario [19].

¹⁵ *DNS* o *Domain Name System*. Sistema que permite cambiar nombres a direcciones *IP* [19].

Para que todos los servidores web alrededor del mundo puedan comunicarse con todos los clientes de manera correcta, ambos (cliente y servidor) deben conocer o “hablar el mismo idioma” o “hablar con un mismo sistema de reglas” (ésta es la finalidad de un protocolo de comunicación) [19].

Dado que las computadoras que operan en *Internet* se encargan de proveer de documentos de hipertexto al cliente por cada petición *HTTP*, son conocidos como “servidores”. Sin embargo, este término puede adquirir mayor consistencia si se aclara que el servidor es tanto el equipo físico (hardware) como la parte lógica-operacional intangible (software) que opera en el mismo. Este software operativo es el que permite al equipo de cómputo que funge como servidor comunicarse con otras computadoras que toman el rol de clientes [19].

Un servidor puede ser desde una computadora hogareña con características básicas hasta una potente máquina con software operativo *UNIX*¹⁶, ya que es el software el que se encarga de que las tareas correspondientes se lleven a cabo. Desde luego, el sistema operativo que se ejecuta en la máquina servidor deberá contar con software específico para el manejo y procesamiento de peticiones *HTTP*, es por ello que a los servidores web también se les conoce como “servidores *HTTP*” [19].

¹⁶*UNIX* es la marca registrada del sistema operativo multitarea y multiusuario desarrollado originalmente en los laboratorios Bell de AT&T por Ken Thomson, Dennis Ritchie (creador también del Lenguaje C, que es la base del sistema Unix) y otros [19].

3.1.2 HTTP: el idioma del cliente y del servidor.

Las “conversaciones” que tanto clientes como servidores sostienen dentro de la web se realizan bajo *HTTP*, dichas conversaciones consisten simplemente en una secuencia “petición-respuesta”. El cliente pide y el servidor responde. Este protocolo corre sobre la capa más alta del modelo *TCP/IP* (Capa de aplicación). El protocolo *TCP* es fundamental en la comunicación en *Internet* dado que maneja comunicación orientada a la conexión, con lo cual garantiza la entrega de información íntegra y ordenada siempre que exista un nodo o dispositivo del otro lado de la comunicación. Es por lo tanto, responsable de asegurar que un fichero (archivo) que se envía de una red a otra llegue de manera completa e íntegra (aún cuando dicho archivo es enviado en partes a través de paquetes de datos, lo cual sucede en la mayoría de las veces). El protocolo de *Internet* (*IP*, por sus siglas en inglés) se encuentra en la capa más baja (Capa de Acceso a la Red) y es el que se encarga de mover o encaminar los paquetes de datos entre distintos nodos o dispositivos (también conocidos como *hosts*¹⁷) hasta hacerlos llegar a su destino como un fichero completo [17]. La figura 3.3 ilustra una “conversación” *HTTP* de una manera más entendible.

¹⁷ *Host*, en terminología informática, se refiere a cada uno de los equipos de cómputo que interactúan y se comunican en una red [19].

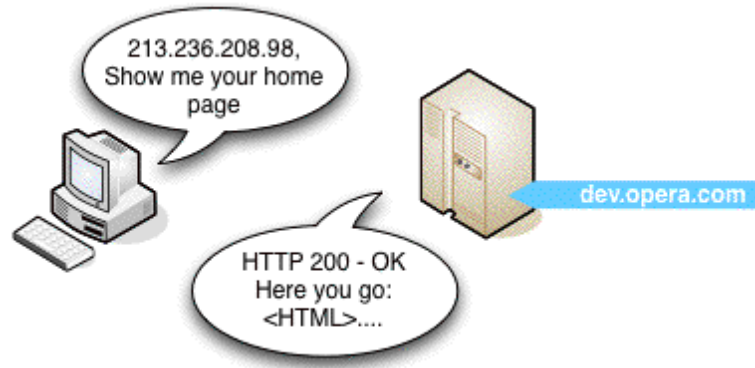


Figura 3.3 Ejemplo de una “conversación” HTTP entre un cliente y un servidor web de un modo más amigable [20].

3.1.3 La Web y sus inicios.

Fue en el año de 1989 cuando Tim Berners-Lee inventó la *World Wide Web* (*WWW*). Sin embargo, fue 20 años atrás cuando se dio inicio a esto. Tim fungía como ingeniero de software en CERN¹⁸ cuando detectó la necesidad de muchos científicos alrededor del mundo de intercambiar información y resultados concernientes a sus múltiples investigaciones y trabajo de muchos años. Tim entendió la necesidad y se percató del potencial tan grande que se puede alcanzar tendiendo miles de millones de computadoras interconectadas en la *Internet*. Lo anterior hizo que Tim documentara su propuesta de la *WWW* en CERN. Dicha propuesta especificaba un conjunto de tecnologías que harían del *Internet* un lugar verdaderamente accesible y útil para todas las personas en el mundo. Al inicio dicha propuesta fue rechazada; Tim perseveró, y fue en 1990 que especificó las

¹⁸La Organización Europea para la Investigación Nuclear (CERN, por sus siglas en francés) es el laboratorio más grande e importante a nivel mundial que se encarga de la investigación en el ámbito de la física de las partículas [21].

tres principales tecnologías que son la base para la Web de hoy: *HTML*, *URI*¹⁹ y *http* [22].

T. Berners-Lee junto con otros ingenieros se percataron del potencial de la Web y acordaron convertir dichas tecnologías en estándares. Fue en 1994 cuando Tim fundó el consorcio mundial de la Web, *W3C*²⁰ (*World Wide Web Consortium*); a través de éste, se establecen especificaciones, directrices y recomendaciones para asegurar el funcionamiento adecuado de la Web para todo el mundo, así como su evolución de manera responsable. Dichas recomendaciones y estándares establecidos han habilitado con el paso del tiempo una web rica en contenido y cada vez más accesible a la personas alrededor del mundo. La segunda generación de esta plataforma (Web 2.0) especifica una web más colaborativa y activa, la Web 3.0 presenta características para una web semántica y con presencia en las cosas (teléfonos, *tablets*, autos, relojes, etcétera) [23]. La figura 3.4 muestra la forma en la que las primeras páginas web eran visualizadas.

¹⁹URI (*Uniform Resource Identifier*) es un término que se utiliza para identificar de manera única un recurso en la Web [19].

²⁰La *W3C* es la principal organización mundial que estandariza las tecnologías de la Web al mismo tiempo que controla su evolución y desarrollo [23].

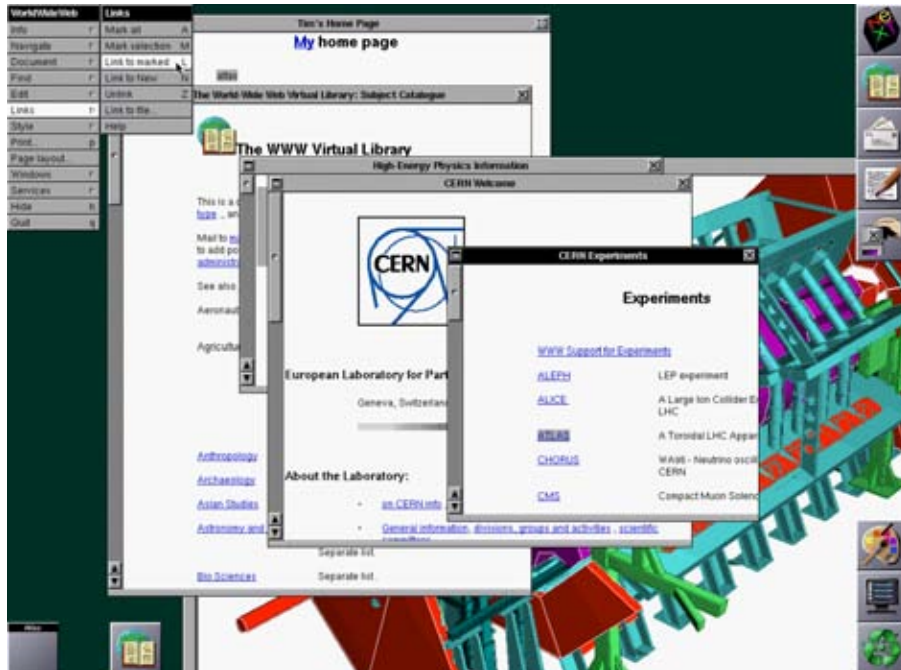


Figura 3.4 (1990) Tim Berners-Lee crea el primer cliente web (*World Wide Web*). Las páginas web eran mostradas como se ilustra [22].

3.1.4 La evolución de la Web, sus estándares y sus tecnologías.

En un inicio (1990 – 2003), la Web presentaba sitios dirigidos al contenido estático (Web 1.0); eran unidireccionales, es decir, el usuario no tenía forma de interactuar con el sitio ni influir en el crecimiento del mismo. Las páginas mostraban contenido en forma de texto y eran los administradores de éstas los responsables de actualizar su información. El objetivo primordial de esta web era la divulgación de información y el usuario sólo tenía el papel de espectador [18].

Las tendencias y hábitos de los usuarios al usar la web, así como las necesidades del mercado fueron factores de suma importancia para el surgimiento de la Web 2.0 ó la “Web Social” (2004). Esta web va más allá del contenido

estático y el objetivo principal era contar con una web dinámica, interactiva y colaborativa. La comunicación ya no era unidireccional; los usuarios ya estaban cansados de su rol pasivo y surgen necesidades de participación activa. El usuario ya podía interactuar con el sitio y por ende influir en el crecimiento y actualización. Aquí surgen los sitios tipo *blog*²¹, *wiki*²², foros y redes sociales [25].

El surgimiento de la Web 3.0 (a partir del 2006) trajo mejoras en muchos aspectos. Los desarrolladores web ya contaban con herramientas más avanzadas para crear toda clase de sitios. Ésta es la Web inteligente, semántica; aquella que cuenta con sitios y aplicaciones disponibles desde cualquier dispositivo móvil; la web 3D, la web de las tiendas electrónicas, la web que explota el potencial del cómputo en la nube²³, la web de los mapas interactivos y de la geo-localización²⁴; en otras palabras, la web donde el usuario disfruta de innumerables servicios y modos de entretenimiento [25].

La Web ha sido sujeta al mismo arduo proceso de desarrollo que cualquiera de los protocolos que conforman *Internet*. En un principio hubo serios problemas relacionados con la incorporación de estándares para la Web. Fue tal el impacto que ocasionó la primera generación de esta plataforma que el desarrollo y rápida

²¹ Sitio web que sirve de lugar para publicar actualizaciones (noticias, eventos, etc.) acerca de un tema específico o de cuestiones personales relacionadas al autor del mismo [24].

²² Sitio web colaborativo que comprende una colección perpetua de varios autores para temas específicos [24].

²³ La palabra “nube” se emplea metafóricamente para referirse a *Internet*. El cómputo en la nube consiste en la gestión y uso de información, aplicaciones y demás recursos informáticos acorde a las necesidades del usuario en un momento específico pero de manera remota; es decir, desde cualquier equipo de cómputo con acceso a *Internet*, ya que dichos archivos, aplicaciones o recursos no necesariamente se encuentran almacenados o instalados en el equipo de cómputo que se esté empleando, sino en centros de datos para este propósito [24].

²⁴ Capacidad de un sistema informático para detectar y registrar la ubicación de una persona con el uso de alguna tecnología de comunicación específica con conexión a *Internet* [24].

evolución de *HTML* y otras tecnologías web ocasionó que se rebasara el ritmo de aprobación por estándares que se tenía para ese entonces; en otras palabras, comenzaron a evolucionar y adoptarse las distintas tecnologías web sin que antes fueran aprobadas como estándares; de manera que, mientras *W3C* comenzó a trabajar en los estándares *HTML* (1994), las compañías que desarrollaban navegadores web no esperaron a su finalización [25].

3.1.5 La competencia de los navegadores web.

Para ganar terreno en el mercado de los navegadores web, la compañía Netscape, que ya contaba con su propio navegador (llamado ciertamente, *Netscape*) apareció en la escena y estableció su propio conjunto de etiquetas *HTML* que desde luego mejoraron de manera considerable el aspecto visual de las páginas web. Microsoft, empresa de software fundada en 1975 por Bill Gates y Paul Allen, no se quiso quedar atrás y de igual manera diseñaron sus propias etiquetas para su navegador *Internet Explorer (IE)*. El grave problema de esta “guerra” entre navegadores web fue sin duda el impacto a largo plazo. El diseño web evolucionó y comenzó a abarcar tecnologías más allá de *HTML* (como las hojas de estilo en cascada y *Javascript*²⁵ principalmente). Los resultados fueron devastadores. Existen etiquetas y/o tecnologías que solo funcionan en uno u otro navegador; se tienen elementos (etiquetas) *HTML* que en ninguna forma describen la estructura del documento; lo cual rompe con el propósito original del lenguaje: describir la estructura y distribución del contenido de la página, no su aspecto o presentación visual [15].

²⁵Lenguaje extensivo para *HTML* para dar interactividad a los componentes de una página web [18].

No pasó mucho tiempo hasta que la comunidad de desarrolladores sumara esfuerzos y demandara que los dueños de los navegadores web se apegaran a las recomendaciones W3C. Fue el Proyecto de Estándares Web llamado *Web Standards Project*²⁶ (conformado por una comunidad de expertos en la Web) el que destaca en esta suma de esfuerzos. Trabajaron arduamente para que tanto desarrolladores, diseñadores y creadores de herramientas se mantuvieran “en la misma línea”. Sin duda dichos esfuerzos han dado resultado debido a que al día de hoy podemos ver una web robusta en cuanto a la estandarización de las tecnologías que la constituyen [15].

La historia del surgimiento de los navegadores provee un contexto muy importante a la manera en la que hoy se visualizan y desarrollan los sitios web. A continuación un pequeño bosquejo histórico a modo de línea de tiempo del surgimiento de los navegadores más destacados de la historia de la Web [15].

- 1993: *NCSA Mosaic*. Navegador creado por Marc Andreessen, estudiante del Centro Nacional de Aplicaciones en Supercomputación (NCSA). Se popularizó especialmente por su capacidad multi-plataforma²⁷. Adicionalmente permitía el uso de gráficos en las páginas (No fue el primer navegador en implementar esta funcionalidad), que sin duda fue un factor crucial para la rápida evolución de la Web. Contaba también con soporte de sonido, video, marcadores y formularios. Para ese entonces todas las páginas web eran mostradas en texto negro y fondo gris.

²⁶ *WaSP*, www.webstandards.org

²⁷ Capacidad de una aplicación de software de ejecutarse de manera idéntica en distintas plataformas o sistemas operativos gracias a la generación de archivos binarios compatibles [26].

- 1994: Lanzamiento de la versión 0.9 de *Netscape*. Marc Andreessen forma la corporación Mosaic Communications, que más tarde se renombró *Netscape Communications* y lanzó el navegador *Netscape* 0.9. con el fin de lograr atraer usuarios, *Netscape* creó su propio conjunto de etiquetas *HTML* sin tomar en cuenta que el proceso de estandarización que estaba en curso. En este mismo año la empresa multinacional de telecomunicaciones de origen noruego, Telenor comenzaba el desarrollo de un navegador que sirviera como “antídoto” a la excesiva ornamentación de *Internet Explorer*. *Opera*. Durante años, se retrasó dado que se buscaba una versión libre de publicidad, pero esto implicaba costo. Actualmente se encuentra disponible de forma gratuita y libre de anuncios (como el resto de navegadores) para las tres plataformas de escritorio más populares (*Linux*²⁸, *Mac*²⁹ y *Windows*).
- 1996: *Internet Explorer 3.0*. Microsoft entra al mundo web con su primer navegador competitivo junto con su “juego” de etiquetas y características distintivas. Fue el primer navegador en soportar hojas de estilo en cascada (*CSS*³⁰).
- 1996 – 1999: Comienza la batalla entre Microsoft y Netscape por tomar el liderazgo en el mercado de los navegadores web.

²⁸ *GNU/Linux* es el nombre del sistema operativo que nace gracias a la unión de dos esfuerzos separados: Richard Stallman y su equipo de trabajadores para crear la estructura y esqueleto del sistema y un estudiante finlandés, Linus Torvalds que trabajó en el núcleo del mismo. Se encuentra bajo la Licencia Pública General (*GPL*, por sus siglas en inglés) y surge por la necesidad de contar con un sistema basado en *UNIX* pero con el esquema completo del software libre [19].

²⁹ Conocido por el nombre *Mac*, *Mac OS X* es el nombre del sistema operativo diseñado por Apple Inc. para sus equipos de cómputo [19].

³⁰ *CSS (Cascading Style Sheets)* es el estándar para la presentación de las páginas web. Mientras que *HTML* representa la estructura del sitio, *CSS* establece el formato y presentación del mismo para dar un aspecto visual atractivo. [18].

- 1998: Netscape publica el código de su aplicación *Communicator* (Sucesor de *Netscape Navigator*). Miles de desarrolladores participaron en la mejora de la aplicación; al final, optaron por comenzar desde cero su desarrollo.
- 2000: *Internet Explorer* para *Mac*. Éste fue el primer navegador en soportar por completo *HTML 4.01* y el conjunto de recomendaciones *CSS1*; lo cual resultó un paso importantísimo para la incorporación de estándares en otros navegadores. Además fue el primer navegador en contar con soporte para el formato *PNG*³¹ con transparencia *alpha*, la cual define la capacidad de transparencia de un pixel en una imagen.
- 2000: AOL³² adquiere Netscape. Considerada por muchos como la derrota de Netscape ante Microsoft en la batalla de navegadores. *Internet Explorer* era el enemigo perfecto dado a que se encontraba “atado” a cada computador personal con el sistema operativo oficial de Microsoft, *Windows*. Netscape sufrió pérdidas importantes en el terreno de navegadores al haber lanzado aplicaciones “todo en uno” (conocidas como *suites* en inglés), como *Netscape Communicator*, y tomarse varios años para rediseñar y reescribir sus aplicaciones para el lanzamiento de *Netscape 6*. Como resultado; para el año 2006, *Netscape* se encontraba instalado (combinando todas sus versiones disponibles) apenas en el 1% de los ordenadores mientras que *Internet Explorer* tenía acaparado el

³¹*Portable Network Graphics*, (Gráficos de red portátiles) es un formato de archivos de gráficos de mapa de bits. Es la alternativa gratuita al formato *GIF*, que a diferencia de *PNG* está patentado [24].

³²Conocida anteriormente como America Online es una empresa proveedora de servicios de *Internet* y telecomunicaciones con sede en Nueva York, Estados Unidos [24].

90% del terreno (combinando también todas las versiones de *Internet Explorer*).

- 2002: *Safari*. Navegador diseñado por la compañía Apple³³ es el navegador que viene incorporado en sus sistemas operativos, tanto móvil (*iOS*) como de escritorio (*Mac OS X*). Entre sus características destacan que es fluido y cuenta con una elegante interfaz. Existe en dos versiones: *Mac* y *Windows*, siendo la primera, la más popular de ambas.
- 2003: Se forma la fundación Mozilla: El código abierto de *Mozilla* continuó su desarrollo bajo esta nueva fundación, de la cual AOL es parte importante.
- 2005: Se publica la primera versión del navegador *Mozilla: Firefox 1.0* recibió una calurosa bienvenida por parte de la comunidad de desarrolladores dado que contaba con un elevado soporte de los estándares web junto con una robusta seguridad implementada (considerablemente mejor que la de *Internet Explorer*). *Firefox* es considerado uno de los navegadores web de más influencia, ya que fue el primero en arrebatarse una buena parte de la cuota de mercado que presentaba *Microsoft* sobre el ámbito web. A pesar de la fuerte competencia al día de hoy, sigue ocupando un lugar muy importante gracias a su increíblemente flexible sistema de complementos (extensiones o *plugins*).

³³Apple Inc. Empresa multinacional de origen estadounidense que diseña y produce equipos de cómputo y software. Tiene su sede en Cupertino (California, E.U.A). Es mayormente conocida por exitosos productos como el *Macintosh* (1981), *iPod* (2000), *iPhone* (2007) y el *iPad* (2010) [24].

- 2008: Aparece *Google Chrome*. Es considerado (y con mucha razón) el navegador con el crecimiento más grande y rápido de la historia en cuanto a su cuota de mercado. Una de sus principales características es su interfaz limpia, seguridad robusta y ligereza en cuanto a que consume poca cantidad de recursos del procesador del equipo donde se ejecuta.

3.1.6 *Un acercamiento al presente...*

Internet Explorer es un navegador que por su historia se podría considerar controversial. Por muchos años fue el navegador más utilizado en todo el mundo (y por un margen amplio). Sin embargo, las estadísticas no mienten y se ha demostrado que en los últimos años ha perdido importancia entre los usuarios que navegan por la Web [15].

Para el año 2005, *IE* acaparaba casi el 75% de la participación del usuario, lo cual deja por entendido el por qué los diseñadores web prácticamente no pensaban en otro navegador al momento de maquetar sus sitios. Para 2008, su cuota de mercado desciende en a un 55%. Para fines del 2011 ya no alcanzaba el 40% de dicha cuota de navegadores. Pero de manera vertiginosa, su liderazgo ha ido en declive a lo largo de los últimos años. Para enero del presente año *IE* bajó su cuota de mercado a menos del 8% [15]. La figura 3.5 muestra una tabla estadística para representar lo anterior.

2015	Chrome	IE	Firefox	Safari	Opera
April	63.9 %	8.0 %	21.6 %	3.8 %	1.5 %
March	63.7 %	7.7 %	22.1 %	3.9 %	1.5 %
February	62.5 %	8.0 %	22.9 %	3.9 %	1.5 %
January	61.9 %	7.8 %	23.4 %	3.8 %	1.6 %

Figura 3.5. Se muestra que para el año 2015 *Internet Explorer* ya alcanza ni el 10% de la cuota de mercado de los navegadores web [27].

La Web entonces, ha pasado por un arduo proceso de madurez, y a lo largo del mismo, la diversidad y número de dispositivos que acceden a ella ha crecido de manera exponencial, lo cual va haciendo las tareas de diseño y desarrollo de sitios en la web una tarea más compleja, ya que aunque se habla de “web móvil”, no existe tal; es una sola web a la que es posible acceder desde casi cualquier tipo de dispositivo conectado a *Internet*. En otras palabras, es una sola web vista desde distintas perspectivas. El reto como diseñadores y desarrolladores web es sin duda, cómo lidiar con esta diversidad de dispositivos [28]. Y desde luego, el primer paso es mediante el apego a estándares o recomendaciones del *W3C*: *HTML*, *CSS* y *Javascript*. Adherirse a dichos estándares es el arma principal que posee el desarrollador para lograr consistencia en sus sitios al momento de ser “leídos” por los navegadores que tengan conformidad a los estándares *W3C* (aproximadamente el 99% de los navegadores actuales). Más adelante se cubrirán los aspectos fundamentales y disciplinas que envuelven el desarrollo web

*frontend*³⁴. La figura 3.6 que se muestra a continuación se trata de la captura de pantalla de la publicación en su cuenta de *Twitter*³⁵ del desarrollador Stephen Hay acerca de la inexistencia de la “web móvil”.



Figura 3.6 Captura de pantalla de la publicación de enero de 2011 a través de la red social *Twitter* acerca de la controversia de la “Web Móvil” por parte del diseñador web Stephen Hay [28].

El ritmo evolutivo de la Web es sorprendente; tal es éste que ya podríamos estar hablando de la “Web de las cosas”; es decir, aquella plataforma en *Internet* a la que toda clase de dispositivos (automóviles, relojes inteligentes, refrigeradores, pantallas, etc.) tendrán acceso. Es por ello que como desarrolladores web, es preciso considerar dos recomendaciones importantes [8]:

- ✓ Cumplir con los estándares web: Esto para que todo el mundo disfrute de una experiencia positiva al momento de navegar por el sitio. Se debe crear entonces, un código *HTML* sólido y basado en estándares, y asegurarse de

³⁴ Se refiere al desarrollo web que involucra las tecnologías relacionadas al aspecto visual del sitio (interfaz gráfica y de usuario); es decir, *HTML*, *CSS* y *Javascript* principalmente [15].

³⁵ Plataforma social creada en el año 2006 que funciona bajo el concepto de micro-blogging; es decir, pequeñas publicaciones que realizan los usuarios llamadas “*tweets*” y que se caracterizan por ser de 140 caracteres como longitud máxima [24].

que todas las páginas incluyan una definición del tipo de documento (ésta se conoce como *doctype*³⁶).

- ✓ Probar compatibilidad del sitio en múltiples navegadores: Una vez determinado el diseño inicial (maquetación completa del sitio), se recomendaría probar la compatibilidad con tres navegadores esencialmente: *Internet Explorer*, *Firefox* y *Chrome*.

3.1.7 Lo que aporta el uso de estándares para la Web:

Aún no llega el día en el cual todos los navegadores del mercado se han adherido por completo a las recomendaciones del W3C, sin embargo, como desarrolladores web es importante considerar seriamente el apegarnos y no permitir que el hecho de que existan navegadores web no tan conformados a estándares sea un motivo de desaliento [15].

- **La accesibilidad** es sin duda una de las razones más fuertes para apegarse a estándares web. El contenido del sitio podrá ser visto por una variedad de navegadores tanto móviles como de escritorio, así como en dispositivos de lectura para personas con problemas visuales. La creación de documentos bien estructurados y con una distribución lógica de los elementos permite a los usuarios obtener una mejor experiencia de navegación.
- **Compatibilidad futura**. De los estándares que se tienen hoy se han derivado, o se derivarán otros. Si desde el inicio de desarrolla con apego

³⁶ *Document Type Declaration (doctype)*. Se trata de la declaración del documento HTML que se especifica al inicio del mismo para indicar al navegador web el tipo de documento que abrirá [29].

a los estándares, el trabajo realizado disfrutará de longevidad y cuando llegue el día en que los elementos y atributos ya no aprobados dentro de los documentos, aún el sitio creado tendrá la capacidad de ser correctamente interpretado por los navegadores hasta un tiempo considerable.

- **Desarrollo más simplificado, rapidez y eficacia.** En un inicio, los desarrolladores tenían que crear diferentes versiones de su sitio para conseguir la compatibilidad entre los distintos navegadores. El uso adecuado de *HTML*, es decir, para establecer el marcado de la estructura del sitio y las hojas de estilo para la presentación habilita al sitio para ser accedido desde múltiples dispositivos y navegadores. Al contar con una separación lógica entre la parte visual (presentación) y el contenido, el trabajo en edición de información y rediseño puede suceder conjuntamente, lo cual reduce los tiempos de producción y desarrollo que sin duda resulta siempre benéfico desde el punto de vista del negocio. El uso de etiquetas y elementos en las páginas que están desaprobados o que simplemente no concuerdan con el propósito de *HTML* suele traer efectos negativos al momento de hacer uso del ancho de banda³⁷ y descarga de archivos desde el sitio web; un ejemplo podría ser el uso de etiquetas (mal orientadas) *HTML* para presentar la tipografía. Al eliminar dichos elementos de la estructura y contenido y apegarse al uso de hojas de estilo para el control de la presentación, las páginas del sitio cargarán

³⁷Cantidad de datos y/o recursos de comunicación disponibles o consumidos expresada en bits por segundo de acuerdo con la tasa de transmisión o transferencia (velocidad de descarga) con la que se cuente en determinado instante [24].

más rápidamente en las pantallas de los dispositivos lo que por obvias razones dará mayor satisfacción a los usuarios.

Existen diversas maneras de mejorar considerablemente el desempeño del un sitio, las más importantes caen dentro de dos amplias categorías del ámbito de la optimización en la arquitectura cliente-servidor: limitación del tamaño de los archivos que conforman la aplicación o sitio, y reducción de la cantidad de peticiones al servidor. Más específicamente, las técnicas y/o recomendaciones más generales son [15]:

- ✓ Optimizar el tamaño de las imágenes sin impactar de manera negativa la calidad de las mismas
- ✓ Minimizar los documentos *HTML* y *CSS* eliminando espacios y saltos de línea innecesarios.
- ✓ No abusar del uso de *scripts*³⁸ (especialmente de *Javascript*)
- ✓ Agregar los *scripts* de tal manera que cargue de manera paralela a otros recursos de la página sin bloquear el proceso de interpretación y despliegue de la misma.
- ✓ No cargar recursos que no sean necesarios (imágenes, librerías *Javascript* y otros *scripts*).
- ✓ Reducir lo más posible el número de veces que se realizan peticiones al servidor (peticiones *HTTP*).

³⁸El propósito de los *scripts* en el ámbito web originalmente era minimizar la carga de procesamiento de los servidores web y permitir que parte de éste se distribuya del lado del cliente. A diferencia de códigos de programación tradicionales, los *scripts* funcionan en ambientes con intérpretes, que se encargan de ejecutar y automatizar tareas específicas que le dan interactividad a la página y que ayudan a cumplir con propósitos específicos de manera eficiente [8].

3.1.8 ¿Cuáles estándares, exactamente?

El W3C crea y supervisa el desarrollo de las tecnologías que operan en la Web, así como las múltiples aplicaciones de éstas [23].

Como tal, el consorcio mundial de la Web, no es un cuerpo de estándares oficiales, sino la unión de una serie de esfuerzos por parte de expertos en el ámbito web para proporcionar y establecer un orden al progreso y evolución de la Web a través del desarrollo de diversas tecnologías. Para cada una de sus especificaciones, el W3C establece distintos niveles o etapas de madurez. El proceso de formación de un estándar³⁹ W3C se encuentra definido en un documento oficial (*W3C Process Document*). Éste especifica cuatro etapas de madurez (reflejadas cada una en documentos separados) para que una especificación se convierta en recomendación (estándar) [23]:

- **Working Draft (WD)**: Una vez recopilada información de los borradores del editor y diversas discusiones el estándar ya podrá ser publicado como *WD* para ser revisado por la comunidad. Este documento es la primera forma en la que un estándar se encuentre disponible al público. En esta etapa, el documento puede presentar variaciones respecto a su versión final.
- **Candidate Recommendation (CR)**: Una versión más firme que el *WD*. En este punto el W3C ha determinado que el estándar ha sido ampliamente revisado y satisface los requerimientos técnicos de los

³⁹Los estándares del W3C son publicados en documentos conocidos como “Recomendaciones” [23].

grupos de trabajo responsables de dicho estándar. El propósito del *CR* es recabar información relacionada a la experiencia de implementación, es decir, de acuerdo con la comunidad, qué tan posible es implementar el estándar.

- ***Proposed Recommendation (PR)***: Se trata de un documento de reporte de madurez técnica que tras una larga y exhaustiva revisión de implementación y características técnicas, el W3C ha enviado al comité responsable de asesoría (*W3C Advisory Committee*) para su aprobación final y definitiva.
- ***W3C Recommendation (REC)***: Es la etapa de mayor madurez. Se trata de la especificación y serie de directrices que, después de extensas revisiones y pruebas bajo condiciones teóricas y técnicas, ya ha sido aprobada por la dirección y miembros del W3C y en consecuencia reconocida como recomendación; lo cual significa que ya está lista para su completa utilización.

Además de las recomendaciones emitidas por el W3C, existen diversos estándares emitidos por organismos específicos que influyen en *Internet* y la Web [15]:

- *ISO (International Organization for Standardization)*: La Organización Internacional de Normalización es la que se encarga de promover el desarrollo de normas internacionales para la fabricación de productos y entrega de servicios, el comercio y comunicación en todas las índoles industriales. Tiene como principal objetivo buscar la estandarización normas

de productos y seguridad para las empresas y organizaciones (tanto del sector privado como público) a nivel internacional. La aprobación *ISO* ayuda a mantener la compatibilidad del comercio y las tecnologías de comunicación en todo el mundo.

- *IETF (Internet Engineering Task Force)*: El Grupo de Trabajo de Ingeniería en *Internet* es la comunidad internacional de ingenieros, investigadores y expertos en redes de datos cuyo objetivo es cuidar la evolución y desarrollo de la *Internet* en el ámbito de su ingeniería, abarcando áreas como transporte, encaminamiento y seguridad en ella. Es la organización responsable de las publicaciones *RFC*⁴⁰, que indican cómo funcionan los diversos protocolos que rigen *Internet* (*HTTP*, *TCP/IP*, *FTP*, etcétera). El proceso comienza de la siguiente manera: Cuando se identifica una necesidad para cierta funcionalidad específica (léase, adjuntar un archivo a un correo electrónico) un grupo de ingenieros y demás profesionales en el área proponen un sistema para hacerlo posible. Tras varias discusiones y acuerdos, el sistema o tecnología propuesta se convierte en un estándar. Es desde luego, un proceso mucho más complejo de lo que se describe aquí; pero para fines prácticos se puede llegar a la conclusión de que esta es la forma en la que todas aquellas “cosas” que como usuarios activos de *Internet* podemos hacer y que nos facilitan las tareas de comunicación e intercambio de información comenzaron como una propuesta y terminaron siendo publicados en los ya mencionados *RFC*.

⁴⁰*RFC: Request for Comments*, serie de publicaciones que describen detalladamente los aspectos de funcionamiento de algún protocolo o estándar específico en *Internet* [26].

- Ecma international: Anteriormente conocida como ECMA (*European Computer Manufacturers Association*), es la organización internacional para la estandarización de sistemas de información y comunicación. *Ecma International* gestiona los estándares de las tecnologías de la información, incluyendo *ECMAScript*, la versión estandarizada de *Javascript*.
- The Unicode Consortium: Es el consorcio que maneja el estándar *Unicode* para las tipografías de múltiples lenguajes.
- ANSI (American National Standards Institute): Cubre una amplia variedad de estándares, tales como *ASCII*, que es el código estándar americano para el intercambio de información.

Gracias a todos los esfuerzos que ha formado *Internet*, es posible disfrutar de infinidad de servicios a través del mismo; pero una vez más, todos los servicios que se ofrecen en *Internet* tienen como propósito fundamental el acceso a la información e intercambio de ésta sin importar la distancia física que separa a los usuarios de esta red [15].

3.2 Los modelos de negocio en Internet

Como ya se ha mencionado, *Internet* hoy en día es una compleja infraestructura de computadoras interconectadas entre sí alrededor de todo el mundo que es operada bajo estándares con el fin de compartir información. La evolución de la Web ha dado lugar a innumerables sitios de todo tipo y por ende ha causado que todos quieran sacar provecho de esta tecnología. A lo largo de este capítulo se enfatizará en unos de los principales usos que se le ha dado a la Web. De los usos más frecuentes de esta plataforma actualmente se encuentra la compra-venta de productos/servicios [30]. El aumento de los usuarios en la Web aunado a la evolución de la misma (mediante el surgimiento de sitios cada vez mas dinámicos) dio lugar para que las empresas o negocios adaptaran estas nuevas tecnologías, en las que vieron un gran potencial, para llegar a sus clientes de una manera más sencilla a las tradicionales y conocidas hasta antes del *Internet* (catálogos, publicidad en radio, TV o revistas, etcétera) [31].

Los negocios tradicionales que se habían basado en las ventas por catálogo tuvieron un nuevo sector de ventas. Otros visualizaron que la Web era un buen lugar para poner la información de servicio al cliente de una manera sencilla y económica (que a través de catálogos) o simplemente para plasmar la imagen de una empresa a la vanguardia con presencia en *Internet*. Estos nuevos enfoques son los que dieron lugar a el desarrollo de nuevos modelos de negocios, así fue como se hizo necesario crear los modelos de negocios basados en *Internet* conocidos como negocios electrónicos [31].

3.2.1 Los negocios electrónicos (*e-Business*⁴¹)

Para entender el porqué surgieron los modelos de negocios en *Internet* es necesario dar una breve explicación de cómo evolucionaron los negocios y el comercio a partir de la aparición de la Web.

El uso temprano de la Web para hacer negocios

Las empresas comenzaron a utilizar sitios web con fines de comercialización poco después de que el diseño web basado en gráficos estuviera disponible desde principios de 1990. La mayoría de estos sitios web sirvieron para ofrecer a los visitantes la información básica sobre los productos y servicios de la empresa e incluían información de contacto (teléfonos, email, etc.), proveer una información sencilla del negocio. Solicitar servicios a través de la web se produjo tan pronto como los departamentos de mercadotecnia⁴² se dieron cuenta de que los sitios web de la compañía estaban a disposición de millones de personas. Las ventas en línea comenzaron en 1994 con la capacidad de cifrar los datos de tarjetas de crédito [31].

Las primeras ventas en línea

Con la esperada llegada de la Capa de Conexión Segura *SSL*⁴³, desarrollado por Netscape en 1994, los sitios web desarrollaron la capacidad de cifrar las sesiones, con lo que las transacciones de tarjetas de crédito a través de *Internet* se hicieron

⁴¹ Del término inglés *electronic Business*. Se refiere a la conducción de un negocio a través de *Internet* [24].

⁴² Mercadotecnia es el análisis del comportamiento de los mercados y consumidores. Analiza la gestión comercial de las empresas con el objetivo de captar y retener a los clientes [32].

⁴³ *Secure Socket Layer*, por sus siglas en inglés; protocolo que funge como una capa de seguridad que se agrega al protocolo *HTTP* para que la transferencia de información entre el cliente y el servidor web sea de manera cifrada [24].

más seguras. Con una conexión cifrada entre el servidor de una empresa y un equipo cliente, el número de las tarjetas de crédito puede ser enmascarado de manera que no puede ser interceptado por un tercero, haciendo que el robo de información de la tarjeta fuera menos probable. Esta seguridad llevó a un mayor número de empresas a ofrecer sus productos para la venta a través de la web [33].

Nacimiento de las ventas web modernas

Los avances en la tecnología de servidores, incluyendo la capacidad para construir sitios web a partir de las bases de datos de productos, dieron lugar al surgimiento de grandes empresas en *Internet* como eBay y Amazon, cada uno de los cuales con sus respectivas tiendas virtuales [33]. En las anteriores, cada producto tenía que ser publicado de forma manual en una página web. Con los sitios web con infraestructura de bases de datos, las empresas podrían utilizar plantillas de páginas web para mostrar decenas de miles de productos de forma automática. A medida que el número de productos disponibles aumentaban, también aumentaba el tráfico y las ventas en estos sitios web [31].

Los avances del sistema de pago

Las primeras implementaciones *SSL* eran buenas, pero muchas personas aún no confiaban en éstas para proporcionar la información de sus tarjetas de crédito para hacer el pago. Como resultado, un número de sitios de micropagos (pagos por mínimas cantidades) iban y venían. Existe un sitio web que se mantuvo y que le ha ido muy bien debido a su capacidad para transferir dinero a través una variedad de fuentes de financiación, incluidas las tarjetas de crédito y las cuentas

bancarias, sin revelar la información de la tarjeta de crédito del pagador al comerciante. Esa compañía es PayPal⁴⁴ [31].

Paypal ha permitido el procesamiento de tarjetas de crédito de muchas empresas pequeñas que de otro modo no hubieran podido ser elegibles para una cuenta de tarjeta de crédito de comerciante tradicional.

Burbuja de las “punto com” de 2001

Los problemas con la confianza de los clientes comenzaron a finales de 1990. Los ataques notables a la negación de servicio (*DoS*⁴⁵, por sus siglas en inglés) a los sitios web destacados hicieron que los clientes se preocuparan de que sus datos de tarjeta de crédito podrían no estar seguros. A lo largo de este período, los negocios en línea recibieron grandes inversiones de capital a través de ofertas públicas iniciales, y vieron su venta de acciones a precios muy por encima del valor real de sus empresas. Muchas empresas tenían buenas ideas, pero planes de negocios pobres, y los especuladores hicieron subir los precios de las acciones en las empresas de *Internet*. Los primeros golpes vinieron ya que algunas empresas en línea comenzaron a reportar grandes pérdidas y los inversores comenzaron a examinar la viabilidad de los planes de negocio en línea. Los inversionistas temerosos empezaron a vender sus acciones, haciendo que los precios de las acciones cayeran por debajo de su inflado valor real. Un número de compañías bien conocidas cerraron, tales como eToys [31]. Muchas otras empresas que carecían de sólidos planes de negocios fallaron entre 2001 y 2002.

⁴⁴ Empresa *e-Commerce* que permite pagos y transferencias de dinero por *Internet* [26].

⁴⁵ Se trata de un ataque en el ámbito de redes informáticas que consiste en hacer que un recurso o equipo de cómputo servidor en la Web se vuelva inalcanzable (no disponible) para sus usuarios alrededor del mundo [24].

El estado actual del e-Business

En la actualidad, el *e-Business* abarca desde simples sitios que ofrecen información corporativa hasta aquellos sitios que ofrecen productos y servicios para la venta en línea. Los usos innovadores de las nuevas tecnologías de la información incluyen servicios en línea como: clases de idiomas, grandes repositorios de información comercial y el uso de *Internet* para la investigación. Las ventas en línea de tiendas basadas en la Web continúan creciendo. Las ventas de la información digital, en forma de libros electrónicos y archivos de música digital, son las ofertas más recientes de los negocios electrónicos como Apple, Amazon y Barnes & Noble [31].

Resumiendo todo el proceso de evolución en 4 fases [31]:

Fase 1: Correo Electrónico (*e-Mail*)

Fase 2: *Sitio web* Corporativo

Fase 3: Comercio Electrónico (*e-Commerce*)

Fase 4: Negocios Electrónicos (*e-Business*)

Se puede entonces definir que el *e-Business* o negocios electrónicos como la aplicación de las tecnologías de la información para realizar compra-venta de productos/servicios mediante *Internet* que, como ya sabemos, está basada en determinados estándares de comunicaciones y la forma más común de acceder a la información que aloja dicha red es la Web. Constituye una nueva definición de los modelos de gestión empresarial, basados en nuevas tecnologías y una fase superior al *e-Commerce* [34], aquí es donde estará enfocada la maqueta del sitio solicitado por el cliente que originó la propuesta de solución que abarca esta tesis.

Las aplicaciones basadas en el concepto de *e-Business* se caracterizan por agregar más interactividad, lo que favorece el aumento del volumen de las transacciones. A su vez, permiten optimizar procesos críticos en la organización tales como gestión, comunicación, y distribución, y realizar mediciones constantes para la mejora continua del proceso comercial [34].

Además de comprar y vender, las empresas realizan muchas otras actividades que las mantienen en el negocio, por ejemplo, el vendedor de un producto debe identificar la demanda, promover su producto ante los compradores potenciales, aceptar pedidos, entregar y apoyar el uso de su producto, facturar y recibir el pago de su producto; todo este tipo de procesos se realizan después de la venta por lo que los modelos *e-Business* se encargan de mantener una relación óptima entre la empresa o negocio y sus clientes con el fin de lograr su fidelidad [34].

Dicho de otra manera *e-Business* se enfoca más en el proceso interno de una empresa:

- Utiliza la Web tanto como un medio, como un elemento de mercadotecnia.
- Comprende la integración de sistemas internos de negocio (*e-mail*, *chats*, video conferencias, sistemas de colaboración y comunicación empresarial).
- Incluye al *e-Commerce*, y abarca determinados procesos internos como la producción, el inventario y el desarrollo de productos, las finanzas, el desarrollo de estrategias de negocios y la administración de los recursos humanos.

- Su foco principal está en los costos y la eficiencia en las operaciones.

El *e-Business* abarca a toda la organización, con el objetivo de optimizar todo el proceso interno y utilizando *Internet* como el principal medio de comunicación. Su énfasis está en la planificación del proceso. En cambio, el *e-Commerce* o Comercio Electrónico es la realización de transacciones o negocios a través de medios electrónicos, complementando el intercambio de bienes y servicios con el manejo de información a través de *Internet*. El *e-Commerce* es un componente fundamental del *e-Business* y no debe confundirse con este. Ya se mencionó que en la actualidad la competencia en el mercado es muy alta y las empresas que quieran sobresalir deben utilizar las tecnologías de la información. El cliente que detrás de este proyecto de tesis expresa su necesidad de entrar a la competencia mediante su sitio para así tener una mayor participación en el mercado [34].

3.2.2 El comercio electrónico (*e-Commerce*⁴⁶)

El comercio electrónico consiste en realizar electrónicamente transacciones comerciales, es cualquier actividad en la que las empresas y los consumidores interactúan y hacen negocios entre sí usando medios electrónicos. Mediante transmisión de datos (texto, imágenes y video) comprende actividades como son el comercio de bienes y servicios, suministro de contenidos digitales, subastas o cualquier otra interacción entre empresas y sus clientes [34].

⁴⁶ Del término inglés *electronic Commerce*. Se refiere a la actividades de compra-venta de productos, servicios u otra clase de bienes a través de *Internet* [24].

En el año de 1993 las grandes empresas perciben la importancia y comienzan a crear sus sitios web, empezaron a realizar catálogos en la red, las páginas son estáticas y el modo de comunicación era a través del correo electrónico. Inmediatamente las empresas vieron la posibilidad de emplear páginas web para sus negocios: surgen las tiendas virtuales e incluso algunas empresas rentaban espacios para otras tiendas que estuviesen interesadas en dar a conocer sus productos, el medio de pago se realizaba a través de tarjetas electrónicas las cuales consisten en transferencias de dinero a través de una tarjeta bancaria en la red [22] [31].

Posterior a esa etapa se pretende automatizar el proceso de envío de datos acerca de los productos comprados, surgen las primeras implementaciones de bases de datos junto con aplicaciones web dinámicas y de fácil interacción con el usuario, surge la publicidad “la mercadotecnia en la red” y aparecen los primeros protocolos de pago seguro a través de las tarjetas electrónicas. El contenido ya es completamente dinámico, se cuida el diseño y aspecto del sitio empleando diseñadores gráficos especializados en su creación e informáticos para la programación del sitio web, se mejora la seguridad en el sitio y se implementan diversos nuevos mecanismos de seguridad [22] [34].

El comercio electrónico ha ido evolucionando, de ser un simple catálogo de productos o servicios, construido a partir de una sitio estático, a convertirse en uno de los medios más populares para realizar ventas y generar ganancias, el comercio electrónico pasa a ser un medio más para llegar al cliente que en las empresas tradicionales.

Sin duda alguna el *e-Commerce* ha evolucionado de una manera exponencial y hoy en día es una manera de hacer negocio en la red sin necesidad de realizar grandes inversiones y poder hacerlo directamente desde casa u oficina siempre y cuando se cuente con una conexión a *internet*.

En el ámbito geográfico, México también ha sido alcanzado por la evolución del *e-Commerce* y desde que los primeros negocios incluyeron estos modelos no ha parado de incrementarse los ingresos derivados de las operaciones comerciales realizadas electrónicamente [35].

En el sitio web de AMIPCI⁴⁷ pueden consultarse las estadísticas en el estudio de comercio electrónico que se realiza anualmente. Los resultados de este análisis nos permiten tener una mejor perspectiva sobre la evolución del comercio electrónico en México [35]. La figura 3.7 ilustra gráficamente lo anterior.

⁴⁷(AMIPCI) Asociación Mexicana de *Internet*, es una asociación civil encargada de realizar estudios anuales en relación a las tendencias en *Internet* [35].

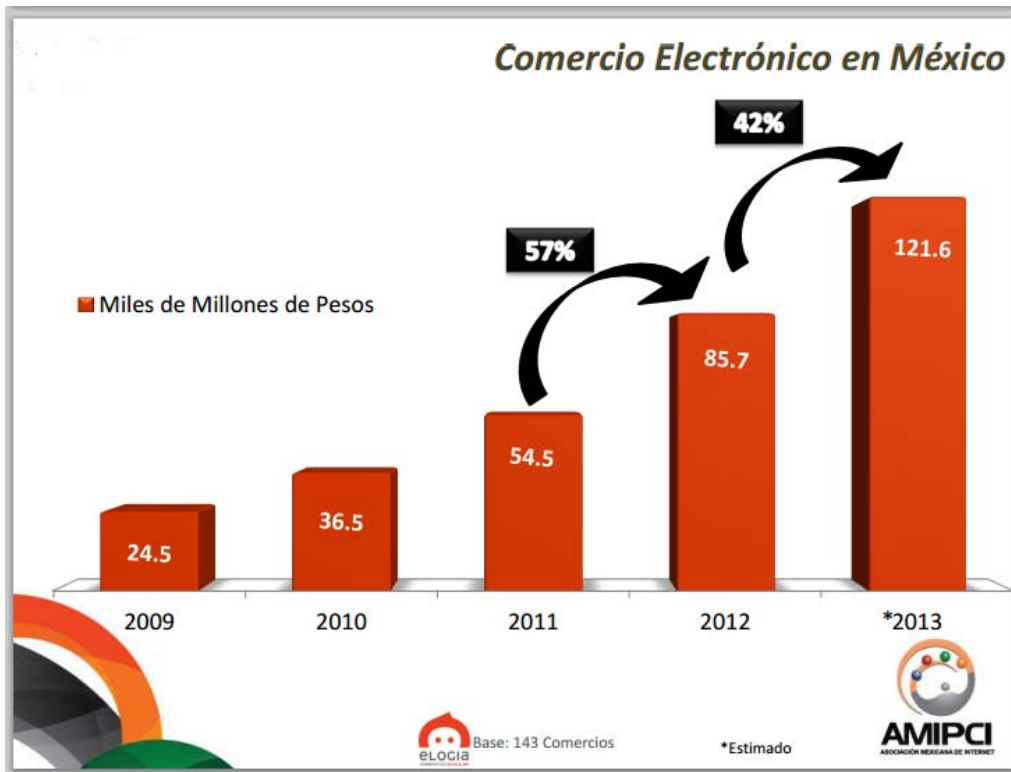


Figura 3.7 Se muestra la comparativa anual desde el año 2009 hasta el 2013 de gasto en comercio electrónico en miles de millones de pesos y el porcentaje de aumento de los últimos 3 años [35].

Reportes más actuales indican que para el año 2014 siguió el crecimiento del comercio electrónico y aunque la mayoría de las transacciones en línea fueron para comprar o pagar servicios, como viajes, entradas al cine, pagos de impuestos, etcétera. Muchas empresas de productos de consumo vieron en la Web una opción válida para extender sus negocios. Durante ese año firmas como Vicky Form, Walmart, Home Depot, Nike, entre otras empresas departamentales y de servicios, apostaron al comercio electrónico en el mercado mexicano, el cual reporta un valor de 121 mil 600 millones de pesos [36], de acuerdo con la

Asociación Mexicana de *Internet*, esto representa un crecimiento de 20% en relación con 2013 [35].

Price Shoes, empresa de ventas por catálogo, que con un crecimiento de entre 15 y 18 por ciento en promedio anual, se integró al e-Commerce en 2014,

Por su parte The Home Depot recientemente inauguró su servicio de venta *online* en el Distrito Federal, Estado de México, Jalisco y Nuevo León, con el objetivo a crecer alrededor de 3% de sus ventas totales en el país al cierre de 2014 [36].

Walmart de México lanzó nuevo portal para la compra de más de 30 mil artículos de abarrotes, frutas, verduras y productos de primera necesidad a través de la implementación de su nueva plataforma en línea [36].

El estudio de comercio electrónico, realizado por la Asociación Mexicana de *Internet* anualmente, del 2013 revela que al menos 92 por ciento de los mexicanos utiliza sus dispositivos móviles para investigar o comprar productos varias veces al mes. Dentro del territorio nacional una de cada tres compras se realiza con un teléfono móvil; un dato proveniente también del mismo estudio de 2014 [35]. Obsérvese la figura 3.8 para mayor claridad.

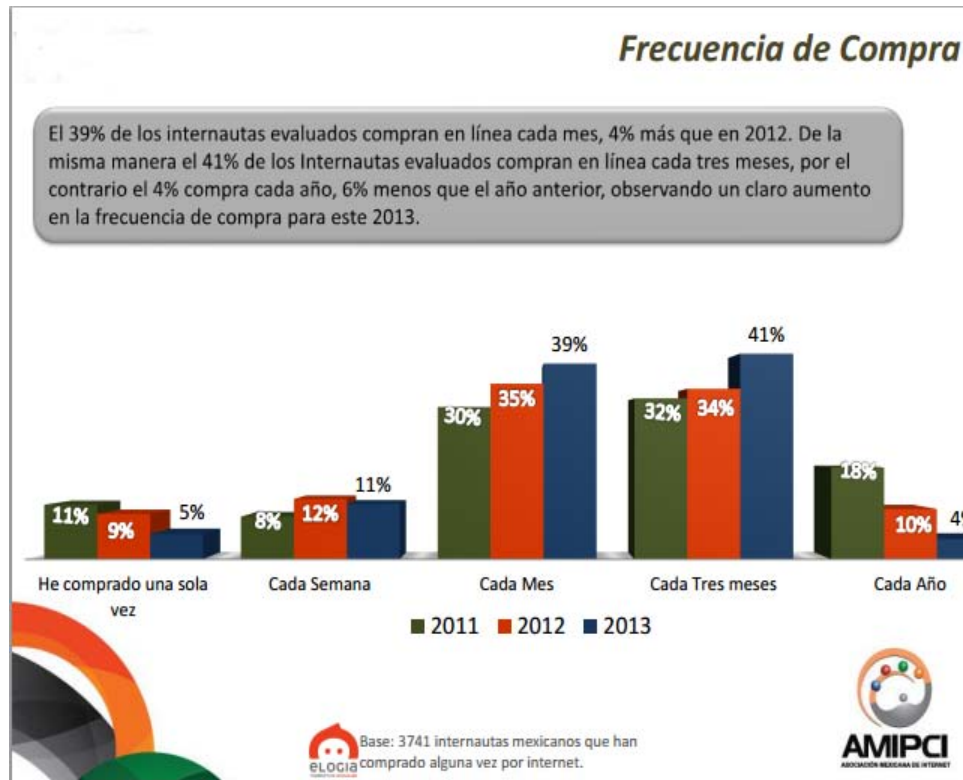


Figura 3.8 Grafica comparativa de la frecuencia con la que los Mexicanos compran en línea. Se observa que cada vez es más frecuente el tiempo en que los usuarios compran. Se demuestra una participación cada vez más activa año tras año.

3.2.3 Tipos de modelos de negocio en Internet

Las tendencias actuales indican que cada vez más empresas en México, nacionales o extranjeras, están incursionando en el comercio electrónico por los variados beneficios que ofrece, reducir costos y ampliar la participación de mercado a prácticamente cualquier parte del mundo [36].

Existen varios tipos de modelos de negocios. Una breve explicación de que casos son y algunos ejemplos:

B2B (Business to Business⁴⁸)

Negocio a negocio, se considera toda la gama de acciones que se puedan ocurrir en dos organizaciones, como por ejemplo las compras, la administración de proveedores, administración de pagos, abastecimientos, y tareas como servicio y soporte. Esta modalidad representa el 80% del comercio electrónico. En los últimos años. Se han llevado a cabo asociaciones entre compradores y vendedores. Sus ventajas son la reducción de costos por transacción, y la reducción de tiempo, la eficiencia en las operaciones puede aumentar [33] [34].

Este modelo es, en otras palabras, para corporaciones grandes que tiene como clientes empresas y con las cuales hacen transacciones, compra-venta de servicios o de productos pero en grandes cantidades destinados a la pequeña y mediana empresa (*PyME*) o emprendedores iniciales.

B2C (Business to Consumer⁴⁹)

Esta modalidad de negocio a consumidor, se refiere al intercambio entre empresas y consumidores finales, es decir, el comercio tradicional a través de medios electrónicos. Esta modalidad fue la de mayor desarrollo inicial ya que como se ha mencionado anteriormente sus inicios datan desde que se empezó utilizar el correo electrónico para sustituir la venta por catálogo [31], provocando una reducción de costos, ya que el vendedor ahorra el costo de producir, imprimir y enviar sus catálogos, otra ventaja es que toda persona que tenga computadora

⁴⁸ Se refiere al modelo de negocio característico por las operaciones comerciales entre dos organizaciones o empresas [24].

⁴⁹ Se refiere al modelo de negocio característico por las operaciones comerciales entre una organización o empresa y consumidores (público en general) [24].

con acceso a *Internet*, pueda ser tentado a realizar compras. Hay mayores posibilidades en que se realicen compras de software, video, fotos, y música que como en los años recientes se ha incrementado [33] [34].

Existen numerosos ejemplos de empresas en esta modalidad porque es la más común debido a que los compradores son los usuarios finales. Amazon.com es un sitio de los más populares en la actualidad.

C2C (Costumer to Consumer⁵⁰)

El comercio electrónico C2C engloba aquellas transacciones en las que tanto como el vendedor y el comprador, son consumidores finales. Generalmente se trata de asociaciones de consumidores con intereses comunes, que apoyándose en las infraestructuras existentes, crean entornos que le permiten intercambiar ideas, conocimientos o productos [33] [34].

Por lo tanto en el comercio electrónico C2C los usuarios actúan tanto como compradores y vendedores a través de una plataforma de intercambio. Por ejemplo: arte, antigüedades, joyerías, aparatos electrónicos, servicios profesionales y un sin fin de artículos. La comunidad en línea eBay, sitio web donde los usuarios ponen en venta artículos o buscan otros para su compra, en la actualidad cuenta con más de 30 millones de usuarios; es sin duda, un buen ejemplo de un comercio electrónico C2C [37].

⁵⁰ Se refiere al modelo de negocio característico por las operaciones comerciales entre dos personas consideradas consumidores finales [24].

3.2.4 El impacto del buen diseño de un sitio web en el comercio electrónico.

Se ha recalcado la importancia en la actualidad del comercio electrónico para cualquier empresa debido a que cada vez más aumentan los usuarios en *Internet* y sus muchos servicios que ofrece. La evolución continua en la Web provoca el incremento de clientes potenciales. Las estadísticas indican que las transacciones comerciales a través de la Web se incrementan exponencialmente año con año dado las grandes ventajas que ofrece hoy en día la tecnología [35] [36]. Sin embargo, como en todo, existen algunas desventajas: algunos procesos de negociación tal vez nunca se ajusten al uso del comercio electrónico, como es el caso de los alimentos perecederos o algunos artículos de un alto valor económico tales como joyas o antigüedades; puesto que es imposible que puedan inspeccionarse adecuadamente por una localización entre el vendedor y el comprador muy remota [33]; además de la seguridad que aún después de mucho tiempo sigue siendo un factor importante del porqué muchos usuarios aún sean temerosos de dejar datos personales en bases de datos como pueden ser números de tarjetas de crédito, direcciones y demás información de carácter confidencial. Los beneficios que ofrece el comercio electrónico son el motivo de que siga incrementándose el número de personas que utilizan la Web para obtener productos/servicios para satisfacer sus necesidades. Hace posible una igualdad entre clientes, sin importar su nacionalidad o lugar de residencia, es un medio abierto las 24 horas del día y los 7 días de la semana, por decirle de algún modo. Para los compradores acceder a ofertas mucho más amplias de productos; también permite acceder a productos que no se puedan encontrar dentro de la

localidad, una visualización del catálogo de cualquier sitio web para una mayor información detallada de los productos, comodidad en el proceso de compras.

Por parte de la empresa que incursiona en el comercio electrónico existen beneficios como lo es la reducción de costos, lo cual representa un ahorro en tiempo y recursos económicos en el momento de presentar sus productos y la posibilidad de llegar a ofrecer estos productos prácticamente alrededor de todo el mundo en donde exista conexión a *Internet*. El gran potencial de crecimiento con la que cuenta cualquier empresa con presencia en la Web debe ser cuidadosamente explotada para de esta forma obtener el mayor beneficio reflejado en más ingresos, situación que desde luego, busca cualquier empresa [34].

Aquí es donde convergen los intereses de ambas partes, mientras el vendedor/empresa/negocio busca ofrecer sus productos de una manera económica y fácil, el comprador buscará ofertas en diversos sitios; entonces un sitio web es el lugar en donde ambos estarán interactuando. El vendedor quiere mostrar sus productos y causar la mejor impresión posible para atrapar al comprador. Un buen aspecto visual de un sitio es un factor importante aunado a que debe ser intuitivo y de fácil navegación. Estos factores son muy importantes para poder causar una imagen de negocio exitoso. En este punto es donde recae la importancia de un buen diseño web; por supuesto, adaptándose a todo tipo de dispositivo a través del cual puede accederse a la Web (aludiendo así al diseño responsivo; técnica que se cubrirá en capítulos posteriores) para tener siempre un sitio agradable e intuitivo no importando desde qué perspectiva se esté visualizando.

3.3 El diseño web.

Muchas empresas al día de hoy tienen el diseño de su sitio web como parte primordial para su éxito y desempeño en el mercado, especialmente cuando se habla de comercio electrónico. Mientras la estrategia de comercialización de una empresa (*marketing*) puede captar la atención de las personas, el diseño del sitio web muy a menudo es el determinante para la realización de una compra [38].

Según estudios por parte de la universidad de *Stanford* [6], el 75% de las personas juzgan la credibilidad de un negocio o comercio basándose en el aspecto de su sitio web. Desde luego, dicha credibilidad influye en la decisión de las personas por comprar o adquirir un producto o servicio, o simplemente buscar alternativas; lo cual significa que la calidad del diseño del sitio web de un comercio está directamente relacionada con la tasa de conversión del mismo [7]. Aunque muchas veces no parezca tan importante, un diseño pobre del sitio que representa el negocio puede ser el causante de la pérdida de muchos clientes. Los puntos clave que se deben considerar al momento de diseñar un sitio web son [15]:

- **Navegación:** Uno de los puntos más importantes a considerar. La navegación por las diferentes páginas de un sitio debe ser sencilla y fácil de comprender para el usuario. Simplificar la experiencia de navegación al máximo suele ser una sabia decisión sobre todo cuando se trata de un sitio extenso. Los diseñadores más experimentados recomiendan mantener los niveles de navegación más importantes divididos en 10 pestañas de acceso.

- **Consistencia de la marca:** El logo de la empresa y los colores asociados a ella deberían verse reflejados de una manera sobria en el diseño del sitio. Eso habilita la comunicación visual de los clientes con la empresa al momento de navegar por el sitio.
- **Patrones de lectura y optimización de búsqueda:** La mayor parte de los usuarios en *Internet* leen las páginas web como si se tratase de cualquier libro; de arriba abajo y de izquierda a derecha. Es importante tomar este en consideración al momento de diseñar las páginas del sitio y colocar la información más relevante en una columna superior izquierda dentro del cuerpo de la página. Esta forma de distribución mejora la optimización del motor de búsqueda (*SEO*⁵¹, por sus siglas en inglés). Al incluir determinadas palabras clave relacionadas con la empresa dentro del código *HTML*, los motores de búsqueda como *Google* y *Bing* serán capaces de hacer un “barrido” mucho más efectivo y fácil a lo largo de las páginas *HTML* incrementado el posicionamiento del sitio en la indexación de estos buscadores y por ende, dando mayor visibilidad a los usuarios al momento de realizar búsquedas relacionadas con la empresa, negocio o comercio.
- **Contenido:** Es tarea del diseñador entregar un contexto visual que envuelva el contenido del sitio, sin embargo; dicho contenido deberá ser corto pero con un mensaje concreto, conciso y sobre todo, amigable. Cuando las páginas del sitio están atascadas de contenido, la lectura para el usuario se vuelve tediosa. Es aquí donde se vuelve primordial la

⁵¹*Search Engine Optimization*. Conjunto de técnicas llevadas a cabo para que se incrementen la cantidad de visitas hacia un sitio web apareciendo en los primeros resultados en los buscadores [26].

colocación y distribución racional del texto y que este armonice con el diseño y estilo de las páginas.

- **Confianza:** Primordial al momento de elaborar la maquetación (diseño) del sitio. Diseñar un sitio con características que reflejan hospitalidad y confianza incrementará en gran medida el tráfico en el sitio. *Internet* suele ser un lugar peligroso para aquellas personas que están acostumbradas a cerrar negocios cara a cara. Desde luego, al estar en *Internet*, dichas personas estarían buscando un sitio que proyecte características confiables (buena usabilidad, interfaz de usuario atractiva y contenido bien distribuido), lo cual incrementaría las probabilidades de que estos visitantes regresen al sitio y se conviertan en clientes.

Puede verse entonces, el impacto tan grande que un buen diseño tiene sobre el negocio o comercio. En diseño web abarca todo lo relacionado al uso de tecnologías para el desarrollo de la vista de un sitio web (la parte frontal, lo que el usuario final percibe a primera vista); en otras palabras, desarrollo del lado del cliente (desarrollo web *frontend*). Muchas ocasiones al diseño web se le asocia únicamente a los colores y el aspecto visual, sin embargo, el diseño web supone el dominio de un conjunto de tecnologías, habilidades y disciplinas para entregar al usuario final una experiencia completa y agradable a la hora de navegar por las páginas que conforman el sitio creado [8].

3.3.1 Planificación del sitio:

Es difícil creer que un sitio web tendrá éxito si no se ha decidido para qué va a servir. A continuación una lista de los principales tipos de sitio que existen en la Web [18]:

- **Sitios personales:** Sitios que “hablan” sobre uno mismo. La Web ha alcanzado una popularidad enorme; es muy frecuente ya encontrar una gran variedad de sitios personales o, por llamarlos de algún modo “hogares en línea”. Al momento de diseñar un sitio de esta índole, la ambición es la meta. Si dicha meta es solo compartir momentos mediante fotos o videos y mantenerse en contacto con amigos, se puede recurrir entonces a una red social. Si se tiene el deseo de mostrar una crónica de nuestra vida, mediante texto e imágenes se puede hacer uso de las herramientas en línea para la creación de un *blog*. Pero si lo que se quiere es un sitio extremadamente personalizado o que englobe varios intereses personales, entonces, una página web personalizada serviría como “tarjeta de presentación” para el usuario en cuestión
- **Curriculum Vitae:** Estos sitios pueden ser de mucho potencial en el ámbito profesional de la persona. Sería más fácil distribuir una tarjeta de visita que apunte al *Curriculum Vitae* (historia de vida) alojado en línea que fotocopiar muchos ejemplares y distribuirlos a distintas personas. Además aspectos como la inclusión de enlaces a empresas donde se laboró simplifica la planificación y extiende el valor de este documento de presentación profesional.

- **Temáticos:** Como lo indica su nombre, se centran en un tema fundamental, ya sea música, arte, historia, ciencia, política, religión, etcétera. Dado que la web está llena de información, es importante tomar en cuenta qué tanto interés podría existir en otras personas sobre el tema que se quiere plasmar.
- **Noticias:** Estos sitios son vistos como “Portales” y su propósito, desde luego es el de informar de los más actuales acontecimientos que ocurren ya sea localmente (dependiendo del país de origen del sitio informativo) o alrededor del mundo. Es importante que dichos sitios se mantengan actualizados en todo momento y que provengan de agencias de noticias con credibilidad y prestigio para asegurar un buen tráfico de usuarios.
- **Promoción:** Los sitios promocionales están diseñados para ampliar la noticia sobre un artículo (producto) específico. En ocasiones, estos sitios crecen hasta convertirse en pequeños negocios en línea donde pueden venderse dichos artículos.
- **Comercio electrónico:** Ya se habló del comercio electrónico en el capítulo 3.2. Estos sitios presentan el uso que más éxito tiene en la Web: vender algo.

Ya se mencionó que la Web de hoy no es la misma que la de hace 10 años o incluso que la de hace un mes. Hay una verdad fundamental relacionada a la vida útil de los sitios en *Internet*: “Los mejores sitios son los que mejoran constantemente”. Los creadores de éstos añaden soportes para nuevas funciones de navegación, ajustan su apariencia para adecuarse a las nuevas tendencias e

incorporan contenidos nuevos. Empero, muchos sitios web que fueron considerados estupendos en su tiempo hoy están “derrumbados” debido a la negligencia de sus diseñadores [18].

3.3.2 *Disciplinas en el desarrollo web frontend:*

Las siguientes destacan como las disciplinas fundamentales en el desarrollo web *frontend* [8]:

- **Autoría (Marcado):** En cuanto a la autoría en el ámbito del desarrollo web, hacemos referencia al proceso de preparación del contenido para ser entregado en forma de páginas web; es decir, utilizar la tecnología de marcado (HTML) para estructurar y definir la funcionalidad del contenido mediante el marcado o uso de etiquetas estandarizadas. Como ya se mencionó con anterioridad, la tecnología HTML está en un continuo proceso evolutivo controlado por su puesto por el W3C, y como desarrolladores web tenemos que estar siempre al día en cuanto a las nuevas especificaciones definidas para el lenguaje. La estructura de las páginas es el cimiento de los sitios web, esencial para las técnicas de optimización de motor de búsqueda y vital para lograr la conexión con las capas superiores (presentación e interactividad); todo esto para conseguir una funcionalidad completa para el usuario final.
- **Estilo:** Todo lo que concierne al estilo pertenece a la capa de presentación. El estándar de hojas de estilo en cascada es fundamental en el desarrollo del lado del cliente pues proporciona el esquema visual de las páginas, el cual habilita la usabilidad, facilidad de navegación, lectura clara y cómoda

de los contenidos, etcétera. El diseño web, la manera en la que luce un sitio y el cómo se encuentra estructurado son partes fundamentales que van de la mano para el desarrollador/diseñador. Otro aspecto importante en el estilo es la consistencia de la visualización en los diferentes navegadores, lo cual no resulta tan trivial en un inicio.

- **Compatibilidad con distintas plataformas, navegadores y dispositivos:**

Aunque los navegadores instalados en los ordenadores no portátiles (de escritorio) tendrán aún una larga vida como las más potente aplicaciones cliente-servidor para acceder a la Web, esto no significa que mirar hacia los navegadores portátiles sea poco importante; por el contrario, el crecimiento vertiginoso del tráfico en la web por medio de dispositivos móviles han dado pie a que se tome muy en serio la portabilidad en la Web. Según reportes por parte de *Walker Sands Digital* [39], el tráfico web desde dispositivos móviles ha aumentado un 133% desde el primer cuarto del año 2012. Las tecnologías web que tenemos hoy son bastante poderosas y flexibles para lograr objetivos de portabilidad y gracias a los estándares conseguir el mayor nivel de portabilidad posible.

Dado que esta nueva generación de la Web engloba sitios semánticos y aplicaciones accesibles a través de cualquier dispositivo móvil inteligente (léase una *Tablet*, o un *Smartphone*), cada navegador en el mercado ahora compite por ser el mejor en potencia de carga de páginas y en variedad de extensiones para aportar la mejor experiencia de navegación.

El meollo del diseño web es que en realidad los desarrolladores-diseñadores nunca sabrán con exactitud cómo se verán las páginas que crean. No podrán saber cuáles de los cientos de navegadores serán usados para visualizar sus sitios (léase navegadores para ordenadores de escritorio o para dispositivos móviles); ni cuál será el tamaño de la ventana de dichos navegadores, ni las fuentes tipográficas que se tiene instaladas; no se sabrá tampoco si está habilitada la funcionalidad para ejecutar *Javascript* en el navegador, ni la velocidad de la conexión a *Internet* con la que el usuario cuenta o si se está usando un lector de pantalla. En otras palabras, son aspectos imprescindibles para el desarrollador *web frontend* [8]:

- La multitud y variedad de dispositivos con acceso a la Web
- Los estándares Web
- Mejora continua de los sitios web
- Diseño web responsivo
- Accesibilidad
- Desempeño del sitio

Accesibilidad.

El propósito de la accesibilidad en la Web es hacer de las páginas de un sitio accesibles o navegables desde distintas puntos o perspectivas para el usuario final. En otras palabras nulificar lo más posible las barreras de acceso para todas las personas, independientemente de sus limitaciones físicas o incluso, limitaciones en cuanto a recursos de tecnología para ingresar y navegar por la Web. Cuando los diseñadores aplican adecuadamente los principios para hacer de

su sitio un lugar con mayor accesibilidad, paralelamente están atacando los problemas tecnológicos que se presentan en algunos usuarios, uno de los más comunes es sin duda la velocidad de conexión [8].

Para lograr esto, primero como desarrolladores debemos abandonar la engañosa noción que muy a menudo se tiene acerca del conocimiento del cómo los usuarios utilizan nuestro sitio. Es necesario que se comprenda el medio en el que se trabaja, y abrazar las técnicas y códigos para habilitar la accesibilidad.

Existen cuatro principales áreas de discapacidad que tienen impacto en la manera en la que los usuarios navegan por nuestro sitio web. [8]

- Problemas visuales: Personas ciegas o con poca capacidad ocular utilizan una variedad de tecnología de ayuda (término conocido en el inglés como *Assistive Technology*) para captar el contenido del sitio, tales como los lectores de pantalla, dispositivos *Braille*⁵², amplificadores de pantalla, entre otros.
- Problemas motrices: Éstos son los problemas relacionados a la incapacidad para mover partes del cuerpo, por ejemplo, incapacidad para utilizar las manos. La variedad de dispositivos para proporcionar ayuda a personas con esta condición incluye “ratones” y teclados adaptados o modificados, pedales, palancas de mando, etcétera.
- Problemas auditivos: Usuarios con limitada o nula capacidad auditiva no podrán disfrutar de los contenidos multimedia en cuanto al audio que

⁵² *Braille* es el término usado para referirse al sistema de lectura y escritura para personas con discapacidad visual [26].

producen. Para ello resulta necesaria la implementación de alternativas para ellos. Entre ellas destacan las transcripciones de las pistas de audio o subtítulos para los videos.

- Problemas cognitivos: Aquellos usuarios con limitadas capacidades de atención, retención de información, lectura y/o solución de problemas se verán beneficiados al usar un sitio claro y bien diseñado (diseño limpio y contenido simplificado). En realidad estos aspectos benefician a toda clase de usuarios finales.

3.3.3 Capas en el diseño y desarrollo web frontend.

Cuando hablamos de diseño y desarrollo web, comúnmente se hace uso del término “capas”. Así como modelos que describen internamente la operación de intercomunicación entre sistemas en una infraestructura de red (*OSI*⁵³ y *TCP/IP*) podemos emular tres capas para aterrizar el término en el ámbito web. Primeramente la capa estructural es representada por los documentos de marcado (páginas *HTML*); esta capa es la base para las demás capas. Sigue la capa de presentación, que es conformada por las hojas de estilo en cascada, que proveen las “instrucciones” o indicaciones de cómo deben verse las páginas en la pantalla del usuario. Por encima de las dos anteriores tenemos la capa de comportamiento, que es definida por la interactividad y dinámica del sitio que aporta el uso de tecnologías de programación y *scripting* (como *Javascript*) [15].

⁵³ *Open Systems Interconnection*, modelo conceptual que estandariza las funcionalidades internas de comunicación entre dispositivos mediante capas de abstracción. OSI está formado por 7 capas de abstracción [26].

La capa estructural:

Las siguientes son las principales tecnologías que han conformado esta capa [15]:

- *HTML* 1.0 (1991): Presentaba muchas limitaciones, el lenguaje tan solo permitía dar formato y presentación a documentos con texto. Muy pocas personas estaban involucradas en el desarrollo de páginas web en ese entonces.
- *HTML* 2.0 (1994): Incorporó algunas etiquetas para mejorar la presentación de las páginas. Fue el estándar para el desarrollo web hasta 1997.
- *HTML* 3.0 (1995): Incluía ciertas características que proporcionaban mayor flexibilidad a los *webmasters*⁵⁴ para el diseño de sus sitios; sin embargo, resultó difícil para los navegadores web de ese entonces implementar dichas mejoras para el despliegue visual de las páginas. Las complicaciones fueron tales que esta versión fue abandonada.
- *HTML* 3.2 (1997): Muchas de las etiquetas que la empresa *Netscape* incorporó en *HTML* 3.0 se hicieron oficiales en este lanzamiento.
- *HTML* 4.0/4.01 (1998): Incorporó características de internacionalización y muy importante, el soporte de un nuevo lenguaje de presentación: *CSS*. Tiempo después, tras una revisión exhaustiva de la documentación de la especificación *HTML* 4.0 recibió oficialmente el título *HTML* 4.01.
- *XHTML* (del término inglés *Extensible Markup Language*): es el lenguaje de marcas diseñado con el objetivo de sustituir a *HTML*. Algunos expertos ven a *XHTML* como la versión *XML* de *HTML*, contando con sus mismas

⁵⁴Los *webmasters* son las personas con conocimientos técnicos específicos para el diseño, creación, mantenimiento y administración de sitios web [26].

funcionalidades y alcance pero incorporando las especificaciones más estrictas de *XML*. El objetivo de esta recomendación *W3C* es que las páginas web ya tengan una clara y contundente separación entre su contenido e información y la presentación; siendo entonces *XHTML* un lenguaje semántico, lo que significa que define el significado de las cosas mas no su aspecto.

Con *XHTML* lo que se busca principalmente es un cambio respecto a *HTML* en el concepto y la forma de estructurar los documentos; así como una sintaxis coherente dentro del mismo, donde los distintos elementos se encuentran por regla correctamente anidados, todas las etiquetas en minúsculas, todos los elementos cerrados correctamente, etcétera. De esta forma se lograría reducir notablemente la existente anarquía que los códigos de muchos sitios actuales presentan [29].

Las etapas por las que ha pasado el proceso evolutivo de *XHTML* son dos esencialmente [29]:

- *XHTML* 1.0 (1999): El *W3C* determinó que el lenguaje de marcas de hipertexto extensible formaría parte de sus recomendaciones. Desde el 26 de enero del 2000 se mantiene como un estándar ensamblado a *HTML* 4.01. *XHTML* es simplemente *HTML* 4.01 reescrito con un conjunto de reglas sintácticas más estrictas. La llegada posterior de *XHTML* 1.1 se deshace de elementos y etiquetas antiguas y desaprobadas y fue especificado en módulos para facilitar su futura expansión.
- *XHTML* 2.0 (2002) El grupo de trabajo en control de *HTML* comenzó con una nueva propuesta para *XHTML*. Dicha versión ofrecía características

para dispositivos móviles e internacionalización. Sin embargo, el rumbo que estaba tomando *XHTML* estaba siendo muy monótono e incluso inalcanzable para algunos. Se requería de un enfoque distinto y refrescado para el lenguaje de marcas de hipertexto.

- *HTML5*⁵⁵ (2004): se trata del último estándar de *HTML*. El desarrollo de *XHTML* 2.0 fue discontinuado y el grupo de trabajo *Web Hypertext Application Technology Working Group (WHATWG)* comenzó a poner manos a la obra para el sucesor de *HTML* 4.01. la aparición de *HTML5* trajo un importante cambio al lenguaje de marcas; nuevas etiquetas que daban un sentido semántico a la estructura de los documentos, nuevos controles para el manejo de formularios y elementos para el manejo de gráficos y multimedia. Las características y capacidades que aporta esta nueva recomendación *W3C* representa una importante competencia frente a las tecnologías de extensión (también conocidos como *plugins*) para los navegadores, tales como *Adobe Flash*⁵⁶. *HTML5* ya comienza a ser ampliamente utilizado y por su extensa e innovadora *API*⁵⁷ *HTML5* representa el lenguaje de marcado del presente y del futuro de la Web [15] [25].

Como tal, *HTML5* posee características, que como ya se mencionó en el párrafo anterior, dan un sentido semántico al contenido de las páginas del sitio. Se verá a continuación cuáles son las etiquetas que incluye *HTML5* como recomendación

⁵⁵A partir del 28 de octubre de 2014, *HTML5* ya forma parte del grupo de recomendaciones (estándares) del *W3C* [40].

⁵⁶Plataforma multimedia para añadir interacción, animación y video en las páginas web [26].

⁵⁷*API* son las siglas de *Application Program Interface* y se trata del conjunto de herramientas, reglas y protocolos para el desarrollo de aplicaciones de software [26].

W3C y que serán de plena utilidad para el desarrollo del prototipo de sitio que se expuso en la introducción de este proyecto de tesis [25].

- *Header*: Etiqueta que nos permite identificar el encabezado a modo de introducción del contenido.
- *Nav*: Define un conjunto de enlaces a través de los cuales se puede navegar. Esto no significa que todos los enlaces del sitio deberán estar dentro de esta etiqueta, más bien, *nav* define un bloque largo para enlaces o *links* de navegación.
- *Section*: Define una sección en el documento o página. Para entenderse mejor, la W3C en la documentación de la recomendación HTML5 establece que una sección como tal representa el agrupamiento de contenido de acuerdo a su temática.
- *Article*: Como tal, especifica contenido, independiente. La etiqueta diferencia contenido que tiene sentido por sí solo (una publicación en un foro o en un blog, o el artículo de un portal de noticias).
- *Aside*: Establece aquel contenido que va a un lado de aquel con el cual tiene cierta relación; en otras palabras, se recomienda que todo contenido marcado con esta etiqueta deberá relacionarse al principal.
- *Footer*: Marca el pie del documento o página; por lo general, se incluye el auto, información de derechos de autor, términos de uso y de privacidad, información de contacto, etcétera.

La figura 3.9 ilustra la estructura gráfica de las anteriores etiquetas semánticas *HTML5*:

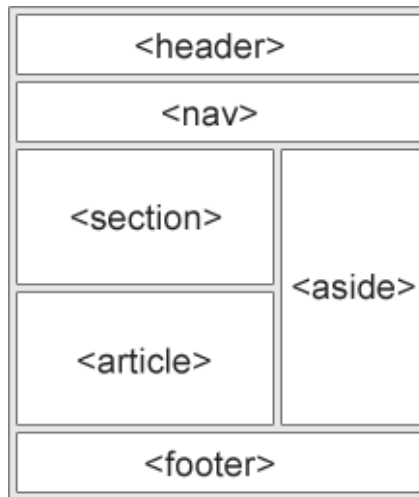


Figura 3.9 El esquema básico de un documento HTML5 con sus principales etiquetas semánticas [25].

La figura 3.10 muestra la estructura básica de un documento XHTML:

```

<html>
<head>
<title>Mi documento</title>
</head>
<body>
.....
</body>
</html>

```

Figura 3.10 Estructura básica y obligatoria de un documento XHTML. Utiliza un *doctype* o declaración de documento más estricta [18].

La capa de presentación:

Ahora que en las últimas recomendaciones del W3C para HTML han sido removidas por completo las etiquetas relacionadas con la presentación del documento (las cuales, como ya se había mencionado, iban en contra del

propósito original de *HTML*) [15], dicha información es responsabilidad exclusiva del estándar para hojas de estilo en cascada o *CSS* por sus siglas en inglés. Este estándar ha sido desarrollado en fases (niveles) [8] [15] [41]:

- **CSS nivel 1** (1996): Este estándar es una recomendación W3C desde 1996 y es soportado en su totalidad por los navegadores actuales. El nivel 1 consiste de reglas para el despliegue del texto, márgenes y bordes.
- **CSS2** (1998): la segunda especificación del lenguaje CSS por parte del W3C. Se incluyen nuevas capacidades para el lenguaje. La más destacable son los *media types*⁵⁸ que nos permiten especificar las reglas para el esquema visual en distintos dispositivos o medios.
- **CSS3** (2009): Sus características sin duda comenzaron a movilizar el diseño web. CSS3 está documentado por módulos; tres de ellos ya forman parte del estándar W3C: Color, Espacio de nombres y Selectores. Las características más destacables de CSS3 son:
 - Nuevas alternativas para dibujar bordes
 - Novedades en el trabajo fondos, uso de degradados e inclusión de múltiples imágenes
 - Uso de sobras para texto y elementos y efectos de opacidad
 - Incorporación de varias novedades respecto al flujo del texto dentro de la página.
 - Características relacionadas con la interfaz de usuario
 - Animaciones y transiciones

⁵⁸ Declaración en la hoja de estilo CSS que permite establecer propiedades específicas para un tipo de medio específico (por ejemplo una impresora) [15].

- Nuevas características para trabajar con múltiples columnas

La capa de comportamiento (interactividad):

La programación y generación de *scripts* proporciona la interactividad y dinámica en las páginas web.

- **Modelo de Objetos del Documento:**

Consiste de una interfaz que permite la representación del documento en el modelo de objeto (*DOM*, por sus siglas en el idioma inglés). Así, el desarrollador podrá tener acceso a los elementos *HTML* o *XML* con la posibilidad de manipularlos, modificarlos o eliminarlos. Dicho de una manera más simple, *DOM* nos permite acceder a un documento (léase una página web) representado en una estructura jerárquica de objetos, para así cambiarle sus atributos o agregarle nuevos. Se trata de la *API* que permite que los *scripts* y aplicaciones accedan y modifiquen el contenido, estructura y estilo de un documento de hipertexto al proporcionar a cada parte del mismo, junto con sus atributos, un nombre específico y tratarlos como objetos, especificando cómo pueden ser manipulados [29]. Un modelo típico de objetos del documento se ilustra en la figura 3.11.

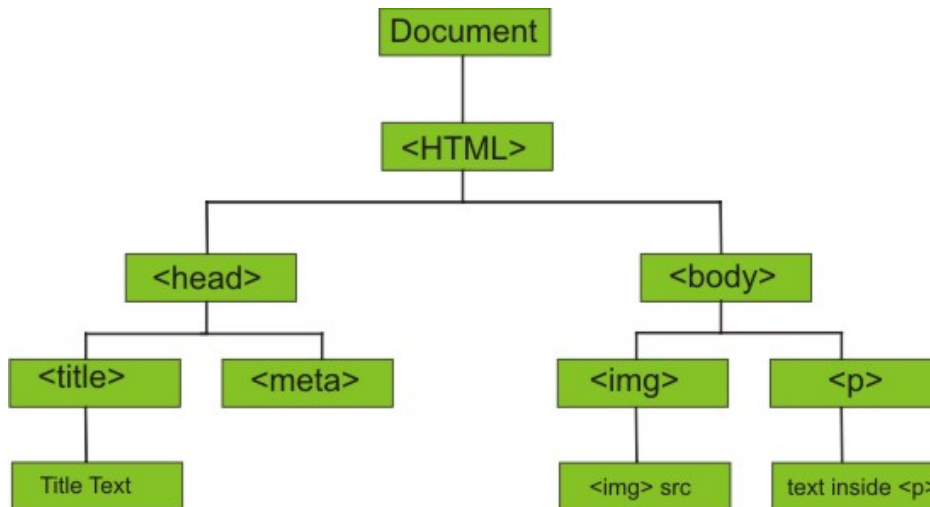


Figura 3.11 Ejemplo de representación en árbol (jerárquica) del Modelo de Objetos del Documento o *DOM*. Incluye las etiquetas o elementos del documento *HTML*.

- **Scripting:**

En tanto la Web evoluciona como una plataforma no solo de páginas sino de aplicaciones de todo tipo, la programación se vuelve imprescindible para los desarrolladores web. *Javascript* es el lenguaje que aporta interactividad entre los elementos de las páginas *HTML*. Añade comportamiento definido y funcionalidad a dichos elementos e incluso al mismo navegador web [15].

Existen otras tecnologías de programación web, que a diferencia de *HTML*, *JavaScript* y *CSS*, funcionan del lado del servidor; entre ellas, *PHP*⁵⁹, *Ruby*⁶⁰, *Python*⁶¹ y *ASP.NET*⁶² y se encargan del procesamiento y

⁵⁹PHP: *Hypertext Preprocessor*. Es el lenguaje de script que se ejecuta del lado del servidor para la creación de sitios web dinámicos e interactivos que manejan bases de datos y grandes cantidades de información [26].

⁶⁰Lenguaje de programación orientado a objetos, creado en 1993 por Yukihiro Matsumoto [26].

⁶¹Lenguaje de programación interpretado cuya sintaxis hace referencia a una filosofía de programación para hacer el código “limpio” y legible [24].

control de datos e información para luego ser enviada al cliente y que éste a su vez despliegue la información de manera adecuada para el usuario final, sin embargo, los anteriores no forman parte de la propuesta de solución que más adelante se presentará en este trabajo.

El *scripting* web y la programación definitivamente requieren de habilidades tradicionales de cómputo para ser dominadas, sin embargo se ha demostrado que a diferencia de lenguajes de programación tradicionales, la curva de aprendizaje para el dominio de las tecnologías web es menor [8].

En cuanto al surgimiento de *Javascript*, la compañía Netscape introdujo su propio lenguaje de *scripts*, originalmente llamado *Livescript*, que posteriormente por razones de *marketing* y tras su alianza con la compañía Sun Microsystems cambió de nombre a *Javascript*. Microsoft también contaba con un lenguaje de *scripts* propio (*JScript*). Era muy necesario un lenguaje de *scripts* estandarizado para todos los navegadores web [15].

JavaScript 1.5/ECMAScript 262: La versión estandarizada de *Javascript* en colaboración con *Ecma International* [15].

Javascript funciona como extensión de *HTML*. Es orientado a objetos (considerado también orientado a eventos) y diseñado para el desarrollo de aplicaciones cliente-servidor a través de *Internet*. Gracias a *Javascript* es posible desarrollar programas que se ejecutan directamente en el

⁶²Principal competidor de PHP, es una tecnología que aporta un completo marco de trabajo (*framework*) para la creación de sitios interactivos con las tecnologías tradicionales del lado del cliente. Microsoft es la compañía dueña de esta tecnología.

navegador web (aunque también es posible ejecutar esta tecnología del lado del servidor) y que sea capaz de realizar operaciones o tareas específicas sin la necesidad de comunicarse con el servidor. En otras palabras, es el lenguaje que permite que la página sea altamente dinámica e interactiva.

Javascript es un lenguaje que requiere de un intérprete para ser ejecutado. Así como los navegadores web cuentan con un motor para representar el contenido *HTML* y *CSS* de los documentos, también cuenta con un motor que permite la interpretación del código *Javascript*. Las características que posee este lenguaje facilitan en gran manera la ejecución de tareas específicas del lado del cliente (alivianando la carga de trabajo al servidor) tales como validación de campos en un formulario, mostrar y aplicar efectos específicos a los elementos visuales de las páginas y presentar anuncios y/o alertas en pantalla [8].

3.3.4 Otras tecnologías que operan en la Web:

La Web que se conoce hoy podría considerarse como el universo de aplicaciones y páginas web interconectadas lleno de vídeos, fotos y contenido interactivo. El usuario común en la Web, sin embargo, no puede ver cómo interactúan los navegadores y las tecnologías web para hacer que esto sea posible.

A lo largo del tiempo, las tecnologías web han evolucionado hasta permitir que los desarrolladores puedan crear nuevas e increíbles experiencias web. La Web actual es el resultado de los continuos esfuerzos de una comunidad web abierta que ayuda a definir estas tecnologías web. Además de *HTML*, *CSS* y

Javascript existen diversas tecnologías que han colaborado en gran manera al desarrollo potencial de esta grandiosa plataforma; algunas ya prácticamente en desuso, otras siguen creciendo [15].

- *Java* (1996): La tecnología de Java que permitía la ejecución de aplicaciones en las páginas web a través del navegador. Era necesario tener instalada la máquina virtual de java en la computadora del usuario para ejecutar dicha tecnología.
- *Flash* (1996): Previamente conocida como *Macromedia Flash*, es un formato multimedia creado especialmente para la Web. Usado principalmente para animaciones a pantalla completa, gráficos interactivos, integración de audio y *clips* de video; así como juegos y aplicaciones basados en *scripts*; todo en fichero o archivos relativamente ligeros en cuanto a su tamaño computacional. Sin embargo, en los últimos años el uso de esta tecnología ha ido decreciendo debido a diversos factores, entre los más importantes:
 - La decisión de Apple de no incluir esta tecnología en sus dispositivos *iPhone* e *iPad* a favor de *HTML5*.
 - La decisión de la compañía *Adobe* (creadora de *Flash*) de dejar de incluir su propia tecnología en los navegadores web móviles (navegadores web en teléfonos inteligentes y *tablets*).
 - La aparición del esquema *Canvas* (Lienzo de trabajo) que permite la creación de elementos interactivos mediante *HTML5* y cuya funcionalidad resultaba ser la misma que en el caso de *Flash*.

El uso de *Flash* comenzó a verse muy criticado respecto a los objetivos primordiales que tenían los usuarios finales al momento de visitar un sitio, léase el sitio web de un restaurante. En este caso, el usuario final, en vez de desear ver un video clip o escuchar alguna pista de audio, su único objetivo al entrar al sitio muy probablemente sea el conocer sus horarios de servicio [8].

Resultaba realmente molesto para los desarrolladores el hecho de saber que un *plugin* es requerido en el navegador para ejecutar la multimedia de *Flash*.

A pesar de que hoy en día resulta algo común escuchar de la voz de algunos expertos que la tecnología *Flash* está “muerta”, y por ende alrededor de toda la Web muchos lo vean como el desfavorecido, *Flash* sigue aportando ciertas ventajas en su uso [8]:

- Usa gráficos escalables, lo que facilita enormemente la reducción del tamaño del archivo sin perder precisión en los detalles de los mismos.
- Es un formato para *streaming*⁶³ y por tanto, la reproducción de videoclips o películas se realiza en un instante y continúa su reproducción mientras sigue la descarga de datos.
- Se puede hacer uso de *ActionScript*⁶⁴ para añadir cierto comportamiento e interactividad sofisticada permitiendo que *Flash*

⁶³Se trata de una técnica para la transferencia de datos de tal forma que se cuente con un flujo continuo y estable para la reproducción del contenido, generalmente audio o video.

⁶⁴Lenguaje de programación orientado a objetos diseñado para la plataforma *Adobe Flash*.

sea usada como capa de aspecto (*Frontend*) para contenido generado dinámicamente o sitios *e-commerce*.

- El soporte para toda clase de navegadores fijos (de escritorio) es altamente confiable por su ya bien conocida distribución a través del *plugin*.
- A pesar de que *HTML5* evoluciona cada vez más y es visto como el reemplazo definitivo de *Flash*, existen ciertas características de desempeño y funcionalidad que aún no igualan a *Flash*.

A pesar de todo lo que se habla en la Web respecto a *Flash*, aún no podemos asegurar que desaparecerá pronto, al contrario, todo parece indicar que *Adobe* aún está poniendo cierto esfuerzo para dificultarle el liderato a *HTML5* en cuanto a la tecnología ideal para incorporar multimedia a los sitios web [8].

- *XML* (1996): El lenguaje extensible de marcado, por sus siglas en inglés. Es el estándar que especifica un conjunto de reglas para almacenar y codificar grandes cantidades de datos e información de manera legible. Incluso, dicho conjunto de reglas permite a los desarrolladores crear sus propias etiquetas para uso específico. Al igual que HTML se deriva de SGML⁶⁵ y forma parte de las recomendaciones (estándares) del W3C. Su aplicación en *Internet* es más frecuente de lo que uno podría llegar a pensar [15].

⁶⁵SGML: Estándar de Lenguaje de Marcado Generalizado por sus siglas en inglés (*Standard Generalized Markup Language*). Especificado en la norma ISO 8870:1986. Se trata del conjunto de reglas de etiquetado de documentos para las organizaciones [26].

Es importante aclarar que en sí, *XML* no es un lenguaje sino una herramienta que nos ofrece una forma estandarizada de especificar reglas para la definición de etiquetas. Es ampliamente utilizado para el intercambio de información y su uso se ha potenciado con la llegada de *AJAX*. Una aplicación muy común de la tecnología *XML* viene siendo el formato *RSS*⁶⁶ que se emplea para publicación de información en *Internet* [25].

- *AJAX* (1999): Conjunto de métodos interrelacionados en el desarrollo web que habilitan la comunicación asíncrona⁶⁷ en las aplicaciones web. Los lenguajes y tecnologías que intervienen en *AJAX* son [25]:
 - *HTML / XHTML* y *CSS*.
 - *XML* y *JSON*⁶⁸
 - *XMLHttpRequest* (mediante este lenguaje se realiza el intercambio asíncrono de los datos que sean necesarios)
 - *DOM* (Ya se habló anteriormente del Modelo de Objetos del Documento).
 - *JavaScript*.
- *SVG* (2001): *Scalable Vector Graphics* se refiere a la familia de especificaciones de un formato de archivo basado en *XML* para la

⁶⁶*Rich Site Summary* o *Really Simple Syndication*; familia de estándares de formato para publicaciones específicas en la Web: Entradas de blog, artículos de noticias, audio y video, etcétera. Un documento *RSS* se conoce como fuente o canal web (*web feed*). Gracias a esta tecnología es posible distribuir contenido “al día” desde un sitio web a otros miles de sitios alrededor del mundo y facilita la rápida navegación y consulta de noticias y otras publicaciones en la Web [24].

⁶⁷La comunicación asíncrona en el ámbito web se refiere a la capacidad de una aplicación de interactuar con el servidor web (enviar y recibir información) sin bloqueos o tiempos de espera, haciendo que el proceso pase desapercibido y el usuario no note ninguna clase de interferencia en el despliegue de información o comportamiento de la página [26].

⁶⁸Por sus siglas en inglés, *JavaScript Object Notation* se refiere al formato ligero para el intercambio y representación de información mediante la relación atributo – valor. No requiere el uso de *XML* [24].

representación de gráficos de vectores de dos dimensiones tanto estáticos como dinámicos. Una característica destacable de esta tecnología es que los gráficos no pierden su calidad de imagen aun al hacer acercamiento hacia ellos o al redimensionarlos. SVG es una recomendación W3C (fue el 16 de agosto de 2011 cuando se anunció oficialmente como recomendación). Actualmente la versión publicada es la 1.1 (segunda edición) [15].

La figura 3.12 ejemplifica un documento XML:

```
<?xml version="1.0"?>
<menu>
  <parent title="kirupaPicks">
    <child>
      <title>kirupa.com</title>
      <link>http://www.kirupa.com</link>
    </child>
    <child>
      <title>kirupaForum</title>
      <link>http://www.kirupa.com/forum/</link>
    </child>
    <child>
      <title>kirupa's Blog</title>
      <link>http://blog.kirupa.com</link>
    </child>
  </parent>
</menu>
```

Figura 3.12 Ejemplo de una especificación de información con formato XML [43].

A continuación, la figura 3.13 muestra un pequeño ejemplo de una plantilla RSS, que sigue siendo código XML.

```

<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">

<channel>
  <title>W3Schools Home Page</title>
  <link>http://www.w3schools.com</link>
  <description>Free web building tutorials</description>
  <item>
    <title>RSS Tutorial</title>
    <link>http://www.w3schools.com/rss</link>
    <description>New RSS tutorial on W3Schools</description>
  </item>
  <item>
    <title>XML Tutorial</title>
    <link>http://www.w3schools.com/xml</link>
    <description>New XML tutorial on W3Schools</description>
  </item>
</channel>

</rss>

```

Figura 3.13 Ejemplo de documento RSS. Se especifica con las mismas reglas de un archivo XML [44].

Aunque en ocasiones parezca simple, el diseño web es un conjunto de disciplinas muy importante en el ámbito web y cuando se habla de ella es imprescindible mencionar [8]:

- Diseño visual (interfaz gráfica)
- Interfaz de usuario y experiencia de usuario.
- Generación de hojas de estilo
- Programación y *scripting*
- Contenido
- Multimedia

3.3.5 Interacción, experiencia e interfaz gráfica para el usuario.

Según Jonathan Ive, vicepresidente ejecutivo de diseño en Apple Inc., el diseño se trata de algo mucho más allá de la manera en la que algo luce; es el cómo realmente ese algo funciona en diversos niveles y aspectos; y si principal propósito consiste en que lo complejo pueda ser presentado de forma clara y ordenada [45]. El diseño define de manera significativa gran parte de nuestra experiencia como usuarios.

El término “diseño” muchas veces se asocia únicamente a la apariencia visual de algo; sin embargo, antes de decidir sobre factores relacionados a dicha apariencia visual, tales como colores y estilos de fuente, resulta más importante tomar decisiones acerca de los objetivos principales del sitio; su usabilidad, accesibilidad y el cómo los visitantes se moverán a lo largo del mismo. Todos los rubros anteriores caen sobre tres disciplinas fundamentales [8]:

- Diseño de Interacción (*Interaction design, IxD*)
- Diseño Interfaz de Usuario (*User Interface, UI*)
- Diseño de la Experiencia de Usuario (*User Experience, UX*)

En cuanto al diseño de interacción, el objetivo se centra en hacer del sitio lo más fácil, eficiente y agradable para su uso. Esta disciplina va muy de la mano con el diseño de la interfaz de usuario, la cual tiene un enfoque más preciso a la funcionalidad de la página respecto a la organización de sus páginas y la distribución de diferentes elementos interactivos que los usuarios utilizarán para navegar en las mismas (botones, menús, cajas de texto, enlaces, etcétera) [8].

La tercera disciplina es responsable de un enfoque más holístico, es decir, supervisar y hacer que la completa experiencia que genera el utilizar el sitio sea favorable. Es aquí donde se vuelve fundamental entender las necesidades de los usuarios/clientes y resulta necesario utilizar la ingeniería de software para el proceso de recolección de requerimientos [8].

3.3.6 Documentos de diseño

Entender las necesidades, deseos y limitaciones de los usuarios es parte central para el éxito en el proceso de diseño del sitio o aplicación web. El enfoque de diseñar el sitio de acuerdo con lo anterior se le conoce como Diseño Centrado en el Usuario (*User Centered Design, UCD*). Por lo general, todo inicia con una serie de investigaciones, que incluyen entrevistas y un proceso de observación con el fin de captar mejor las necesidades fundamentales de los usuarios y cómo el sitio va a cubrirlas. Para el aseguramiento de la usabilidad del diseño, los diseñadores realizan pruebas en cada etapa o fase del proceso. Todo lo anterior suele plasmarse en un documento conocido como *Investigación de Usuarios y Reporte de Pruebas* [8].

Los diagramas son muy importantes dado que representan de una manera más perceptible la idea o concepto general del sitio en cuanto a su estructura. Los esquemas o plantillas visuales de la página (*Wireframes*⁶⁹) representan la estructura de las páginas mediante un boceto o representación abstracta de los

⁶⁹ Diagramas estructurales que muestran un boceto de un sitio a modo de requerimiento o solicitud para que sea desarrollado posteriormente con las tecnologías web pertinentes [8]

diferentes elementos que las conforman [8]. En la figura 3.14 se muestra un ejemplo de ello.

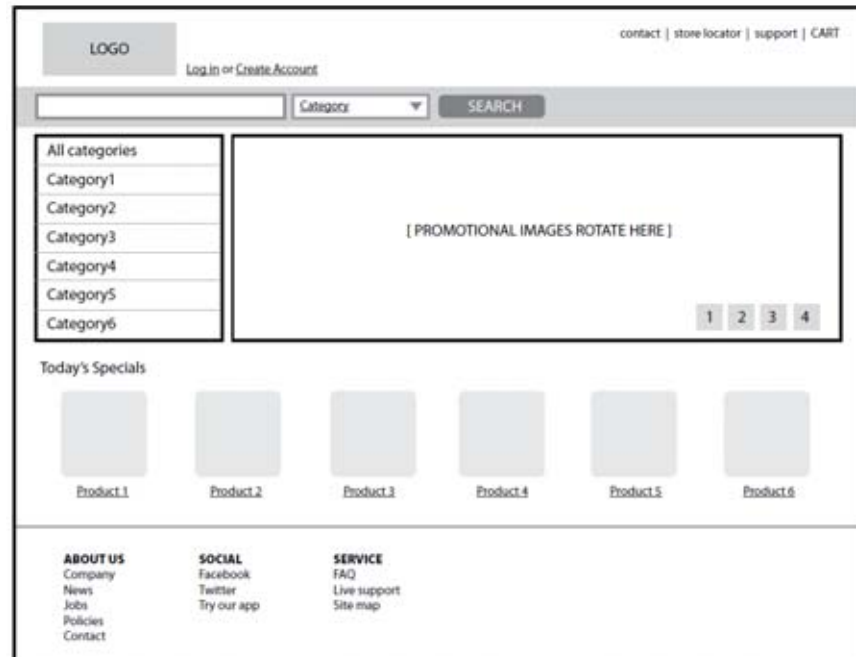


Figura 3.14 Ejemplo de un *wireframe* o boceto estructural del sitio web que se desea construir [8].

Su propósito es indicar cómo será dividida la página y en qué partes se colocan los componentes de funcionalidad y contenido (menús de navegación, barras de búsqueda, paneles para formularios, etcétera).

Los diagramas de sitio indican la estructura del sitio pero de manera completa, es decir un bosquejo de cómo se despliega dicha estructura al usuario, así como la manera en la que se interrelacionan las páginas del sitio [8]. Se ilustra un ejemplo en la figura 3.15.

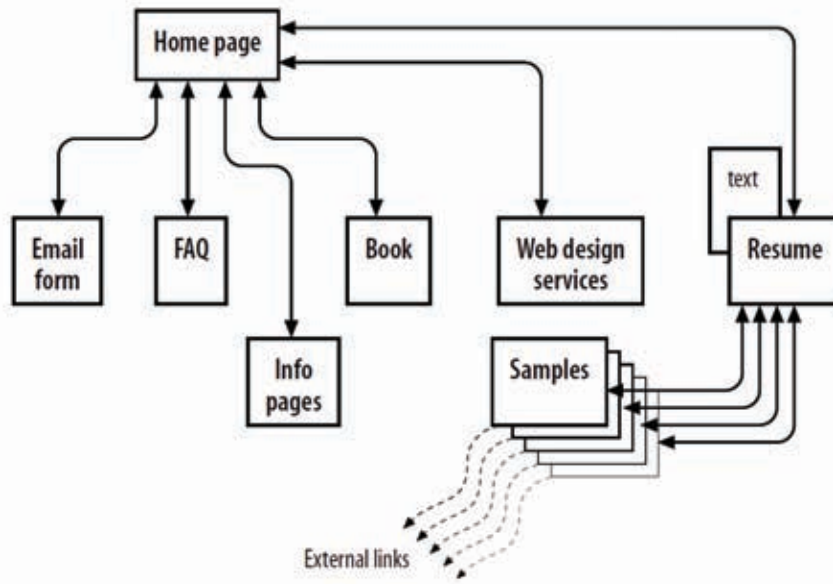


Figura 3.15 Ejemplo de un diagrama de sitio. Se especifican las secciones que conformarán el mismo [8].

Para asegurar que el sitio aporte buenas impresiones a los usuarios desde un inicio, los diseñadores gráficos crean bosquejos que muestran cómo el sitio lucirá desde una perspectiva más visual que estructural. Muchos diseñadores gráficos en el ámbito web actualmente traducen sus diseños en documentos *CSS* y *HTML* por su propia cuenta. En realidad, dentro de la terminología del mundo de la Web para llamarse “diseñador web” es necesario contar con las habilidades suficientes de crear diseños propios, así como trasladar la maquetación meramente a modo de bosquejo en código web para poder construir un prototipo real del sitio [8].

3.3.7 Malas prácticas en el diseño web

El construir un sitio web completo puede resultar sobrecogedor por todas las tareas y disciplinas que implica. Pero en realidad, el verdadero al que un diseñador/desarrollador web se enfrenta a la hora de llevar a cabo lo anterior se encuentra en hacer el sitio un lugar virtual realmente útil. En ocasiones parece que como desarrolladores web la gran mayoría de las veces olvidamos que el sitio está siendo diseñado y maquetado para usuarios finales en la Web y no para nosotros mismos. Darle más prioridad a la creatividad que a la practicidad y usabilidad suele ser un error [46].

Se hará mención de 11 puntos esenciales que reflejan errores que como desarrolladores web se suele cometer [47].

1. Se omite una “caja” de texto de búsqueda

El sitio web, en términos técnicos estrictos, es un compendio de información debidamente estructurada; ya sea que se trate del sitio de una corporación o un simple blog, una caja de texto será siempre imprescindible. El usuario final (visitante) podría estar buscando alguna información en específico y dicha información no siempre estará disponible o plena vista en la página donde se encuentre navegando. El buscador será de gran ayuda para que el usuario encuentre el recurso solicitado.

2. La legibilidad en general del sitio es pobre.

Es verdad que el aspecto visual del sitio es de gran importancia para captar la atención de los visitantes, pero qué hay de la legibilidad de los textos de cada una de las páginas que conforman el sitio. Los usuarios deberán ser

capaces de captar sin dificultad la información de desean obtener. Hay sitios que realmente representan un “dolor de cabeza” al momento de tratar de captar información de estos.

Desde luego, la sugerencia para contrarrestar este problema se centra en dos puntos básicos:

- Comparar los esquemas de colores de la mayoría de los sitios y verificar que en verdad una adecuada combinación de dos colores mejora considerablemente la legibilidad. En la Web, *Adobe Kuler* es una de las mejores herramientas para este fin.
- Utilizar una familia de fuentes sin serifas (terminales a modo de adorno ubicados en los extremos de las líneas de los caracteres). La familia de fuentes más recomendada por la vasta mayoría de diseñadores web es *sans-serif*. La figura 3.16 hace una lustración de las serifas.

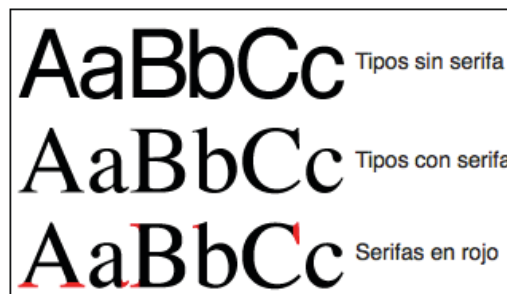


Figura 3.16 Representación gráfica de las serifas (marcadas en rojo) [48].

3. La distribución del contenido está desordenada.

Sin el contenido, un sitio web es simplemente como un libro impreso conformado solo con hojas en blanco. El contenido de un sitio es lo que le atrae el tráfico de visitantes hacia él y por tanto, el cómo este contenido está estructurado y organizado es factor clave para su éxito. La mayoría de los usuarios que visitan un sitio no leen a menos que sea muy necesario, más bien, realizan un tipo de escaneo al sitio y eligen determinados puntos de interés. Hay diseñadores web que únicamente colocan un bloque de texto a una página y omiten por completo los encabezados, sub-encabezados, viñetas, palabras clave, párrafos, etcétera. Se debe hacer uso de títulos apropiados para cada página web, de tal forma que los visitantes puedan identificar exactamente dónde se encuentran. Aunque parezca poco serio, hay diseñadores que incluso olvidan dar nombre a una o más páginas del sitio.

Sin embargo, el error más grave relacionado al contenido del sitio, sería el de colocar información imprecisa, inaccesible, carente de significado o valor (de acuerdo con los propósitos del sitio) y/o obsoleta. El contenido debe siempre coincidir con la temática global del sitio y desde luego, ser útil.

La organización del contenido del sitio deberá entonces ser establecida mediante *HTML*, mientras que *CSS* deberá emplearse para establecer el diseño de las páginas que conforman el sitio.

- Emplear márgenes para la creación de espaciado entre el texto y las imágenes.

- Actualizar y mantener la consistencia. Muchas veces se invierte presupuesto para creación de nuevos contenidos en vez del mantenimiento del sitio. En la práctica, se ha demostrado que el mantenimiento a un sitio web es una buena forma de mejorar el contenido.

4. Mala experiencia de navegación para el usuario.

La experiencia de navegación de un visitante o usuario a lo largo y ancho de un sitio debe ser prácticamente perfecta. Es verdad que no existe un estándar para crear cierta experiencia de navegación para nuestro sitio, pero es aquí donde emergen tecnologías específicas para el desarrollador web. Es demasiado importante comprender la importancia de que la navegación debe ser intuitiva y consistente.

Si el texto es parte fundamental de la navegación, éste deberá ser conciso. Los hiperenlaces deberán sobresalir del cuerpo del texto y todo aquellos hiperenlaces que ya no funcionen (conocidos también como enlaces rotos) ya no deberán aparecer más en ninguna página del sitio.

- Hacer de la navegación una experiencia agradable haciendo que los enlaces existentes sean representados con texto descriptivo. Proporcionar texto alternativo para las imágenes.
- Organizar y estructurar la navegación en conjunto con la temática visual del sitio web. Los sitios web enfocados a negocios requieren de mucha eficiencia y claridad dado que el contenido es de suma importancia.

Los supermercados y tiendas similares no están diseñadas aleatoriamente. Detrás de su diseño (es decir, la manera en la que sus productos están distribuidos en cada departamento, así como su ubicación) existe muchos factores científicos y psicológicos enfocados a hacer que el consumidor gaste más y más en productos. Todo está hecho de determinada forma por una razón. Algo similar sucede en los sitios web. Las cuatro verdades fundamentales sobre el mundo de la Web son:

- El contenido es Rey: éste siempre deberá exhibir calidad y credibilidad.
- Todos los cibernautas son lectores veloces: En realidad los visitantes de sitios web no leen, escanean la información cual lector sumamente veloz.
- La Web es un lugar para la satisfacción instantánea: Los cibernautas son impacientes, no les gusta esperar. Si en 10 segundos no encuentran una buena razón para permanecer en un sitio o si no encuentran en máximo 3 *clicks* lo que estaban buscando, lo abandonarán, así de simple.
- Los cibernautas quieren y gustan de tomar el control de su navegación por la Web.

5. Inconsistencias en el diseño de la interfaz:

Exceso en la creatividad es precisamente eso, exceso. Hay diseñadores que gustan de cambiar el diseño para cada página que conforma el sitio.

Esto solo crea confusión para el usuario. Sin importar qué tan brillante sea el diseño de un sitio, si la interfaz en términos generales no muestra consistencia, los usuarios no se sentirán en control y por ende en disconformidad con el sitio. Simplemente se irán así tan pronto como llegaron.

- Utilizar un diseño consistente para cada una de las páginas del sitio
- Crear un diseño simple pero visualmente agradable y se evitará que el usuario se confunda al momento de navegar por las páginas.

6. Resolución de pantalla poco amigable:

Un rotundo NO se debe dar a todas aquellas páginas en las que se requiera desplazarse horizontalmente para terminar de ver el contenido a lo ancho.

Un buen diseñador web maquetará sus sitios para que queden optimizados a casi cualquier tamaño de pantalla. El tamaño óptimo aceptado actualmente es de 1024 por 768px⁷⁰.

Es verdad que es difícil mantener un diseño que sea óptimo para cualquier tamaño de pantalla dado que los visitantes en la web actualmente llegan por medio de cualquier tipo de dispositivo móvil; sin embargo podemos darnos una clara idea con la lista de los navegadores web más populares usados por los cibernautas [27].

7. Formas de registro complicadas:

Ya no estamos más en los días donde se requerían demasiados detalles del usuario para que éste se registrara. Hay sitios que marcan como

⁷⁰ La abreviatura *px* se refiere a la unidad *pixel*. Pixel o píxel del acrónimo en inglés *picture element* (elemento de imagen) es el elemento más pequeño de los que componen una imagen digital; siendo una unidad de medición es así como se mide la resolución de las pantallas [26].

obligatorios ciertos campos y realizan la validación de manera tan estricta que provocan la desesperación del usuario. Es preciso recordar que los usuarios llegan a un sitio web para recolectar información relevante de éste y no al revés.

- Comparar las formas de registro de distintos sitios y comunidades populares en la web es una muy buena idea para realizar lo propio en nuestro sitio. Un formulario simple como el que muestra la figura 3.17 suele ser una muy buena opción.

Create An Account. It's Free and Relatively Easy!



The image shows a registration form with the following fields:

- First Name
- Last Name
- User Name
- Password
- Confirm Password

Figura 3.17 Se recomiendan formas simples y rápidas para evitar la “frustración” del usuario [46].

8. Uso exagerado de imágenes y/o animaciones

Las imágenes captan la atención del usuario, esa es una realidad pero lamentablemente está también demostrado que puede ser un distractor agravante. Las imágenes deben ser usadas para ilustrar y guiar apropiadamente al usuario.

En cuanto a las animaciones; éstas son un medio muy poderoso cuando son usadas adecuadamente, pero cuando están continuamente apareciendo comienzan a frustrar al usuario. Los usuarios no suelen tener la paciencia o el tiempo para ver las animaciones, por tanto es obligación de los diseñadores presentar la opción para saltar dicha animación.

9. Excesiva ornamentación.

Son muchos los diseñadores web que subestiman la importancia del uso de espacios en blanco. Se encuentran tan aferrados a sus diseños que tratan de meter todo en una sola página. El resultado salta a la vista: páginas muy amontonadas y con poca claridad. La figura 3.18 muestra un ejemplo de diseño web en columnas y ornamentación (decoración) disminuida.



Figura 3.18 Distribuir en columnas suele ser un buen diseño ya que mantiene bien estructurada y legible la información presentada [46].

10. Uso de música o sonidos de fondo.

Esto también es un rotundo NO para las prácticas y tendencias de diseño web hoy en día. Los usuarios no llegan a un sitio web para descubrir qué pueden escuchar de ahí.

- Si por alguna extraña razón, resulta muy necesario usar música de fondo, deberá entonces proporcionar al usuario de controles que le permitan al usuario detener el sonido o música de fondo.

11. Omisión de pruebas:

Probar el diseño e interfaz general del sitio en cada uno de los navegadores más populares es un paso muy importante. Las pruebas realizadas deberán responder a:

- ¿El sitio web puede ser visto en diferentes ambientes?
- ¿El diseño es consistente en los diferentes navegadores web?
- ¿Cuando se desactivan ciertas opciones (*Javascript*, música de fondo, etcétera), aún es posible navegar adecuadamente por el sitio?

Al final, todo se reduce a simplicidad y usabilidad. Sí, los mejores sitios hoy en la Web mantienen estas dos características.

3.4 El Diseño web responsivo.

Son cuatro palabras específicas las que definen la tendencia del presente año, 2015 en cuanto a la elaboración de sitios web: simplificación, concisión, claridad y armonía. Un diseño más minimalista junto con colores más claros y desde luego capacidades responsivas son los nuevos valores que no se pueden ya descartar en plena etapa de crecimiento de la Web. Gran parte de este concepto de diseño se le atribuye a *Google*, el cual adoptó una filosofía de diseño a la que llamó *Material Design*⁷¹ que se caracteriza principalmente por una tipografía clara, casillas bien ordenadas y agrupadas, colores e imágenes claros pero llamativos para no perder el foco y un sentido del orden y la jerarquía muy marcado [50].



Figura 3.19 Un diseño más simplificado en cuanto a gráficos y una gama de colores más clara y amena.

⁷¹Durante la conferencia tecnológica de las novedades de la empresa *Google*, llamada *Google I/O* en el año 2014, se introdujo este nuevo concepto de diseño que abarca desde aplicaciones para su plataforma móvil *Android* hasta sitios web; que ya casi por regla deberán contar con características responsivas [49].

3.4.1 Introducción al diseño web responsivo:

En el capítulo anterior poco se hablo sobre el aspecto técnico que involucra el proceso de diseño web. A partir de este capítulo, se hará un enfoque desde una perspectiva aún más técnica sobre dicho proceso y nos sumergiremos hacia los puntos fundamentales del diseño responsivo.

La pregunta obligada para dar inicio a la introducción sería ¿Qué es el diseño web responsivo? La mejor manera de explicarlo sería mediante la comparación entre un sitio no responsivo y uno responsivo; sin embargo ¿cuáles serían los puntos clave que nos darían las pautas para detectar las diferencias? No hay una mejor manera de detectar dichas diferencias más que usando un *smartphone* y una computadora de escritorio.

Para ejemplificar lo anterior, imaginemos un sitio con una anchura predefinida, 960px (esto en términos CSS se le conoce como *fixed-width website*, y se refiere a aquel sitio diseñado para siempre mostrar una determinada anchura en pixeles). La manera de indicar lo anterior al intérprete CSS que tienen los navegadores es mediante la directiva *HTML* [51]:

```
{ min-width: 960px; }
```

Cuando dicho sitio sea visto desde un navegador móvil (el navegador de algún *smartphone*) nos percataremos de cómo el sitio se verá desplegado de la misma manera (la misma distribución) que en el navegador de escritorio pero escalado para caber en el pequeño monitor del teléfono inteligente. El problema comienza al querer leer el contenido o los textos que aparecen en las páginas. Se tendrá que

hacer uso de la funcionalidad de aumento (zoom) hacia cada parte de la página donde se encuentre el texto que deseamos leer. La figura 3.20 ilustra cómo se vería lo anteriormente descrito.

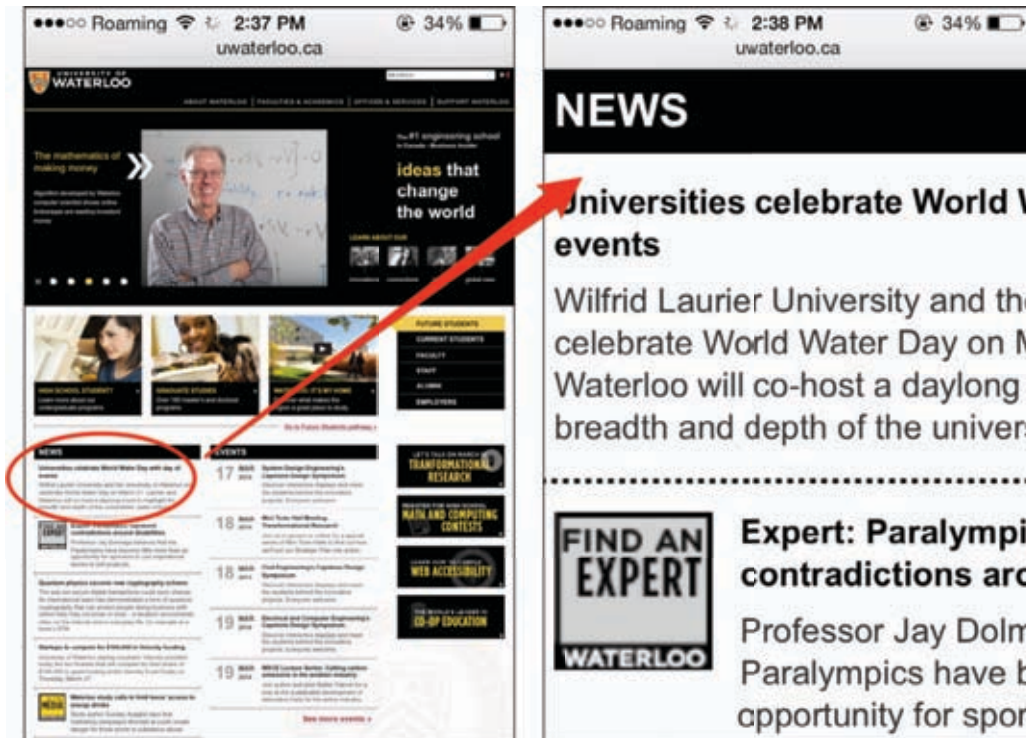


Figura 3.20 La manera en la que se visualiza un sitio sin estar optimizado para una pantalla pequeña. Es necesario hacer el acercamiento manual para leer textos específicos en la página. Esto desde luego, no representa una buena experiencia para el usuario [51].

Existen otros sitios para los cuales existe una versión separada para móviles. Este tipo de sitios despliegan el contenido de manera distinta a como se presentaría en un navegador de escritorio de tal forma que no resulte necesario hacer aumento en ciertas zonas para ver o leer contenidos específicos, sin embargo muchas

veces se presenta cierta inconsistencia en la versión móvil (ciertos contenidos fueron omitidos o cambiados). Aunado a esto, el problema con los sitios separados para móviles es que en su tiempo (tras la salida del *iPhone* en el 2007) eran generalmente adaptados por un solo dispositivo móvil, pero con el paso del tiempo han salido una enorme variedad de dispositivos móviles en el mercado. La problemática evidentemente empeora. Tener que construir múltiples versiones separadas para cada uno de los distintos tamaños de dispositivos móviles existentes y por existir es extremadamente complicado y poco razonable [51].

Diseño con anchura predefinida (Fixed-Width Design)

Hasta hace algunos años, los sitios web eran diseñados para adecuarse en los tamaños más comunes de pantalla, tomando en cuenta computadoras de escritorio y portátiles (*laptops*). A inicios del año 2000 la anchura estándar eran 800 pixeles, mientras que ésta pasó a ser 1024px a partir de mediados del 2000 [51].

Desde luego, el tamaño en monitores de computadores ha variado también bastante, los hay muy anchos y otros angostos. Los diseñadores tuvieron la iniciativa de que pese a la variedad de los tamaños en los monitores de computadoras, sus sitios tenían que mostrar una anchura fija. La anchura fija o preestablecida (*fixed-width*) más usada fue la de 960px. Sobre aquellas pantallas o monitores que presentaban una anchura mayor a los 960px (1024px por ejemplo) simplemente se mostraba el sitio con la anchura fija de 960px. A los lados el espacio en blanco era rellenado de acuerdo al diseño del sitio. Veamos a

continuación una ilustración de esto [51]. Un ejemplo de esto se presenta en la figura 3.21.

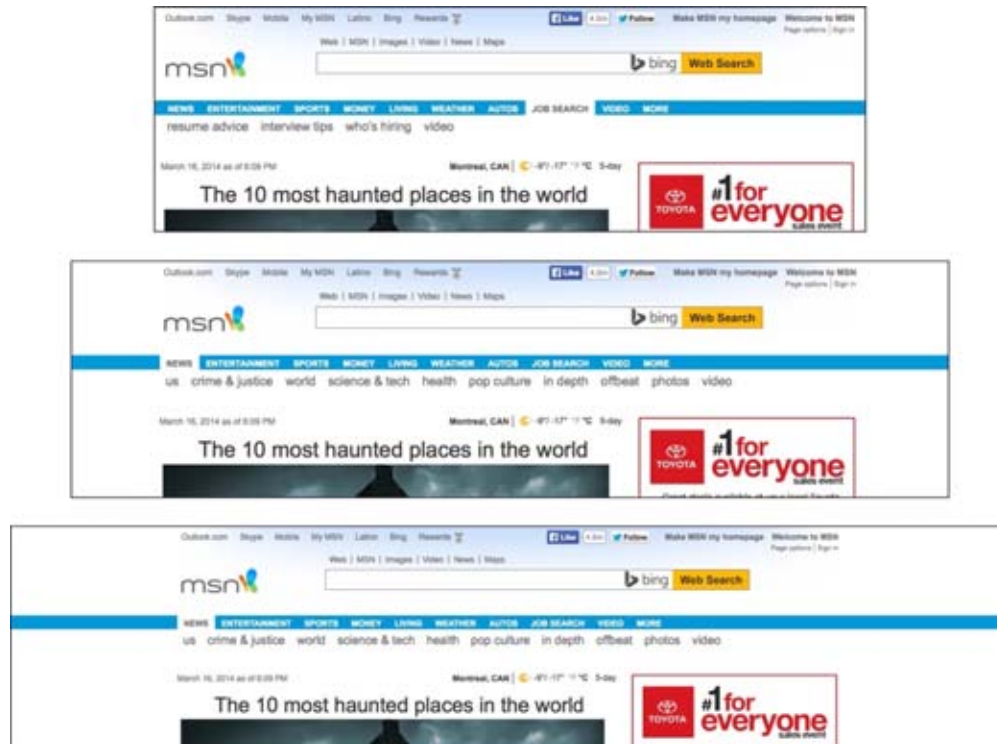


Figura 3.21 Ejemplo de un diseño con anchura predefinida (*fixed-width*). Se puede apreciar que el ancho del contenido del sitio se mantiene.

La navegación Web desde la perspectiva móvil:

El acceso a *Internet*, y más específicamente a la Web desde un dispositivo móvil se hizo realidad a partir de la mitad de la década de los 90s; sin embargo, en términos generales, dicho acceso estaba muy limitado. Los dispositivos con esta capacidad o funcionalidad no presentaban el esquema visual del sitio web como aparecía desde la perspectiva de un navegador de escritorio; en vez de ello,

mostraban únicamente los datos que podían representarse en texto. Datos como información meteorológica, reportes de movimientos en las bolsas de valores, resultados de encuentros deportivos, etcétera [51].

Fue hasta mediados de los años 2000 que los navegadores móviles comenzaron a integrar la funcionalidad que permitía al usuario visualizar una página web como es mostrado desde un navegador fijo (de escritorio) incorporando tecnologías para la traducción de código *CSS2* y *Javascript* [51].

El teléfono conocido comercialmente como *iPhone* apareció en la escena a inicios del 2007. Fue realmente un teléfono que marcó un antes y un después en la telefonía móvil y que además trajo consigo el concepto de “*smartphone*” para ese año aún no existía como ahora [52]. Sus características más destacables eran sin duda su interfaz novedosa, y su pantalla de 3.5 pulgadas (diagonalmente), y desde luego, su tecnología *multitouch*⁷² [51]. La imagen 3.22 es la representación de cómo lucía el primer *iPhone* de la historia.

⁷²Se llama así a la técnica empleada en el esquema de interacción usuario-máquina y al hardware que la aplica y consiste en una pantalla táctil (*Touchpad*) que reconoce más de un punto de contacto del dedo del usuario sobre su superficie. Cuenta con un software que funge como intérprete de dichos gestos o contactos para generar una respuesta que el usuario pueda percibir de manera inmediata. Aunque parezca muy complicado por cómo se describe, la técnica “*multitouch*” es extremadamente popular hoy en día para los usuarios de teléfonos inteligentes y tabletas electrónicas [26].



Figura 3.22 imagen que representa al *iPhone* original presentado en el 2007 por el entonces presidente de la empresa Apple, Steve Jobs [52].

Es verdad que los teléfonos con tecnología táctil ya existían desde hacía unos años atrás; sin embargo, la tecnología *multitouch* en el *iPhone* era lo que realmente lo hacía sobresalir de los demás dispositivos con pantalla táctil. Dicha tecnología permitía el reconocimiento de más de un punto de contacto sobre la pantalla de manera simultánea y disparar una respuesta inmediata cuando el usuario generaba dicho evento.

Las características más a fondo presentadas durante la conferencia conocida como “*Macworld*” llevada a cabo en la ciudad de San Francisco, Estados Unidos, hicieron al *iPhone* ser visto como uno de los productos más innovadores del años (y sin duda de la historia de la telefonía móvil)⁷³.

Sin embargo, existía un problema. En ese entonces, las páginas web estaban diseñadas para ser visualizadas en monitores de tamaño tradicional (igual

⁷³Para mayor detalle sobre el evento sostenido en la ciudad de San Francisco en los Estados Unidos, se recomienda ver el video original donde Steve Jobs hace la presentación por primera vez al público de este novedoso teléfono inteligente: <https://youtu.be/e7EfxMOEIBE>

o mayor a 960px), no en pantallas miniatura como la del iPhone de primera generación (320px). La solución provista por Apple, compañía encargada bajo el mando de Steve Jobs de crear el iPhone, fue la de achicar (presentar a menor escala) la página web completa para que cupiese en el tamaño de la pantalla de este *smartphone*, y permite que el usuario efectuara un gesto para acercar la pantalla deslizando dos dedos simultáneamente separándolos entre sí⁷⁴ o dos toques rápidos al área deseada en la pantalla⁷⁵. Visualizar pequeñas porciones de la página haciendo gestos táctiles para el acercamiento no es en lo absoluto una experiencia de usuario agradable. Los diseñadores web identificaron de inmediato que esta no era una manera óptima de visualizar o navegar por la Web para los usuarios. Se necesitaba encontrar una solución para hacer que las páginas web fueran desplegadas de una manera más “amigable” o sencilla en la pequeña pantalla del *iPhone* [51].

Para entonces, no aparentaba tanto problema para los desarrolladores crear diseños separados de acuerdo al tamaño de pantalla. La solución... Crear un otro sitio de tamaño fijo óptimo para el iPhone (320px ancho), aparte del tamaño fijo tradicional (1024px). Fue entonces que se implementó un mecanismo en el cual los usuarios eran re-direccionados a la versión “móvil” del sitio que querían visitar con su iPhone; por ejemplo, en vez de llegar a <http://youtube.com>, eran redirigidos a <http://m.youtube.com> estos sitios llegaron a ser conocidos como los sitios “*m-dot*” o “*m-dot-websites*” [51].

⁷⁴Gesto conocido en el inglés como “*pinch-to-zoom*”.

⁷⁵Gesto conocido en el inglés como “*doublé tap*”.

Parecía que ya se contaba con una solución que perduraría por mucho tiempo; y la justificación para ello fue que los usuarios usarían sus teléfonos de manera esporádica para visitar un sitio web. Vaya sorpresa cuando la movilidad de la web comenzó a crecer de manera exponencial. La gente comenzaba a usar sus teléfonos para tareas ya no tan simples o rápidas como antes, sino para hacer muchas cosas que antes solo hacían con su computadora de escritorio o portátil [51].

Lo que sucedió tras la salida del *iPhone* al mercado móvil es algo que simplemente tenía que suceder. Las demás compañías comenzaron a producir los *smartphones* que le harían competencia al teléfono inteligente de Apple [51].

Las consecuencias fueron entonces devastadoras para los desarrolladores y diseñadores web al percatarse que comenzaba a existir una amplia gama y variedad de teléfonos inteligentes y por ende distintos tamaños de pantalla; algunos con pantalla más amplia que la del *iPhone* y otros de menos [51].

Años más tarde, Apple anuncia en 2010 su *tablet iPad*, y nuevamente, este dispositivo marcó un antes y un después en las *tablets*; ya que, si bien es cierto que el concepto ya existía, aún no se tenía una *tablet* que aportara un valor real para todo el público. El *iPad* representó justo eso [51].

Fue entonces que los desarrolladores web comenzaron a notar que la variedad de tamaños de pantalla existente entre *smartphones* y *tablets* surgidas a partir del *iPad* representaban un verdadero dolor de cabeza. Se requería un

solución en la que un sitio web pudiera ser visto de manera óptima por cualquier tipo de dispositivo, y por ende, de tamaño de pantalla.

Esencialmente, debido al rango de tamaños de pantalla ya existentes, el problema era el siguiente: evitar por completo la creación de diseños separados para distintos tamaños de monitor y a su vez lograr que un sitio web pueda ser representado en una sola columna para pantallas pequeñas y en múltiples columnas para pantallas amplias. Es decir, solicitarle al navegador que haga los cambios pertinentes en el diseño, basándose en el tamaño de pantalla que detectó del dispositivo que se está usando para visualizar el sitio. La solución se encontraba en dos conceptos: *Media Queries*⁷⁶ y flexibilidad.

La regla *@media*, en la hoja de estilos permite desplegar o mostrar diferentes estilos CSS basado en las características del equipo móvil que se utiliza. Ya existía como parte del CSS nivel 2 desde hace poco más de una década, pero solo permitía *queries* relacionados con el tipo de medio a utilizar (pantalla tradicional o vista de impresión). Es por ello que el diseño web responsivo existe sólo a partir del año 2010 [51].

El uso más común era el de crear un estilo específico para la versión “imprimible” de las páginas del sitio, pero su utilidad nunca había servida para más allá de eso.

⁷⁶ Un *media query* consiste en un tipo de medio y al menos una consulta (o pregunta, por ser el significado del término inglés *query*) que limita las hojas de estilo utilizando características del medio tales como ancho, alto y color [51].

En la recomendación CSS3, se incluyen los *media queries* y se detalla su funcionalidad. Los navegadores comenzaron a tener soporte para *media queries* a partir del 2009 [51].

Básicamente, por medio de un *media query* es posible solicitar la amplitud de la pantalla en uso, y con base en ello cambiar el esquema visual del sitio; es decir, por ejemplo, desplegar el contenido del sitio de manera que éste se distribuya en dos columnas, si y solo si el tamaño de pantalla que se utiliza en ese momento es lo suficientemente ancho como para que quepan dos columnas adecuadamente.

A pesar del potencial que se descubrió de los *media queries*. Éstos por sí solos no podían completar el objetivo del diseño web responsivo. Se necesita un fundamento sólido de flexibilidad; es decir, la manera en la que se redimensionarán los elementos del sitio de acuerdo al cambio de tamaño y distribución de los mismos según las propiedades de dimensión de la pantalla. Las unidades de medida horizontal deben ser flexibles en lugar hacer uso de píxeles fijos; en el caso de la medida del tamaño de fuente, utilizar la unidad *em*⁷⁷. Así también implementar una estrategia para que las imágenes siempre quepan en el lugar deseado, independientemente del tamaño de la pantalla [51].

Los conceptos anteriores por sí mismos no eran algo nuevo para ese entonces (2010); pero fue hasta que el diseñador web Ethan Marcotte encontró la manera de juntar ambos conceptos para crear uno solo: Diseño web responsivo [51] [54].

⁷⁷ La unidad *em* hace referencia al tamaño de la fuente del elemento. Por ejemplo, *2em* significaría: dos veces el tamaño de la fuente del elemento seleccionado [55].

Resumiendo lo anterior, podemos decir entonces que el diseño web responsivo es la técnica para desarrollar páginas web que se rige por tres aspectos clave:

1. Distribución de elementos fluida y flexible: Hacer uso de porcentajes en vez de unidades de anchura fijas.
2. Imágenes flexibles: Utilizar porcentajes para el redimensionado dinámico de las imágenes.
3. *CSS Media Queries*: Permitir asignar un esquema de estilos distinto acorde al tamaño de pantalla en uso.

3.4.2 Estructura y uso de los *media Queries*

La tecnología de los *media queries* es la que le aporta a las páginas web una funcionalidad responsiva respecto al dispositivo sobre el cual se visualizan. En sí, los *media queries* nos son la totalidad de la técnica del diseño responsivo, son más bien la parte que permite establecer distintos esquemas visuales o distinta distribución de elementos visuales a lo largo y ancho de la página (lo que se conoce en el inglés como *layout*) de acuerdo con el tipo de dispositivo empleado por el usuario (computadora de escritorio, *smartphone*, *tablet*, etc.). Podemos decir entonces que los *media queries* son la base de todo lo que resta respecto al diseño web responsivo. Cómo funcionan es básicamente de la siguiente manera [51]:

El intérprete de CSS realiza la pregunta al navegador “¿El tamaño de la pantalla corresponde a 40em?”. Si la respuesta resulta afirmativa (verdadero), entonces el navegador tendrá que aplicar la hoja de estilo correspondiente sobre

la página en cuestión; de lo contrario dicha hoja de estilo será ignorada. En términos de programación, lo anterior puede verse como una simple declaración condicional *if/then*.

La figura 3.23 muestra un ejemplo de un típico *media query*.

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

@media only screen and (min-width: 40em) {
  body { background-color: blue; }
}
```

Figura 3.23 Ejemplo de *media query* para establecer color dependiendo del tamaño de pantalla. Se establece un fondo de color azul, siempre y cuando la pantalla del dispositivo en uso sea mayor o igual a 40em; de lo contrario establece un fondo de color verde.

La variedad de sabores en los media queries:

- **Columnas:** Podemos establecer el uso de una sola columna cuando se esté empleando un dispositivo de pantalla angosta, léase un teléfono inteligente en posición vertical, e ir cambiando el esquema a dos o tres, o más columnas mientras haya espacio disponible o suficiente.
- **Navegación:** Es posible elegir qué elementos se muestran y cuáles se esconden de acuerdo al ancho de la pantalla. Por ejemplo: un menú desplegable en una pantalla angosta y un menú de navegación completa en pantallas más amplias.

- **Tipografía:** Aquí elegimos si deseamos que el tamaño de la letra se cambie de acuerdo al ancho de pantalla; o ajustar el ancho de las columnas para que se despliegue un número óptimo de caracteres por línea y facilitar la legibilidad en esas áreas.
- **Imágenes:** En caso de no contar con espacio suficiente para presentar una imagen en pantalla, podemos establecer el tamaño a mostrar de la misma.

Estructura de los media Queries:

La manera más común para integrar *media queries* es haciéndolo dentro de las mismas hojas de estilo [51].

Tomemos el ejemplo de la figura 3.23 y comencemos separando el código en partes:

```
@media only screen and (min-width: 40em) {  
  body { background-color: blue; }  
}
```

Todos los *media queries* incluidos en la misma hoja de estilo siempre deberán comenzar con la directiva `@media`; lo cual desde luego, define a todo lo que le procede como un *media query*. Cuando se crea un *media query*, siempre será necesario comenzar con el tipo de medio [51].

Seguido de la directiva, contamos con una o más expresiones. Éstas pueden verse como simples “preguntas” que serán evaluadas de manera lógica (verdadero o falso). La primera expresión siempre será, como se indicó anteriormente, el tipo de medio; en este caso, pantalla (*screen*). la expresión

onlyscreen equivale a la pregunta: “¿El tipo de medio es un monitor?”. Dentro de los propósitos del diseño responsivo, el tipo de medio a utilizar siempre será *screen*.

Como se indicó en un inicio, las expresiones serán evaluadas de manera lógica; por tanto al momento de agregar la siguiente expresión, antepone un tipo de operador: *and* (También es posible emplear *or* o *not*, pero una vez más, para los propósitos del diseño responsivo, los anteriores operadores son prácticamente innecesarios). Lo que se desea, entonces es que la expresión de la izquierda del *and* y aquella del lado derecha se cumplan, ambas [51].

La expresión que está después del operador *and* es decir, la segunda expresión, resulta clave para las estrategias de diseño responsivo.

```
@media only screen and (min-width: 40em)
```

Se observa que la segunda expresión está dentro de paréntesis. Cada expresión va dentro de paréntesis excepto aquellas que constan de una sola palabra (*screen*, *print*)⁷⁸. Un navegador antiguo que lea las expresiones, al encontrarse con la primera, obtendrá un *true*, después, la expresión seguida del *and* no será reconocida y por tanto ignorada; obteniendo un resultado verdadero y aplicando el estilo sin tomar en cuenta el tamaño del ancho de la pantalla. Debido a lo anterior la palabra *only* provocará que al no reconocer dicha palabra como el

⁷⁸ La palabra *only* solo sirve para evitar que los navegadores sumamente antiguos apliquen el estilo sin tomar en cuenta la segunda expresión, debido a que éstos ignoran solo aquellas expresiones que no conocen [51].

inicio de una expresión o alguna directiva, todas las expresiones serán ignoradas y nunca recibirá una evaluación verdadera; de esta forma, el estilo no será aplicado.

La expresión *min-width: 40em* significa lo siguiente: “Si el ancho del monitor o pantalla es de *40em* o más ancho que eso, devuelve *verdadero* como resultado de la evaluación de esta expresión” [51].

Como prefijos, pudimos haber empleado *max* en vez de *min*, pero por razones lógicas el significado de la expresión sería el opuesto: “Ancho de pantalla igual o menor a *40em*”. Finalmente, las llaves de apertura y cierre son para delimitar todas las declaraciones CSS que deseamos que sean aplicadas si las condiciones se cumplieron. Por ejemplo, obsérvese la figura 3.24.

```
@media only screen and (min-width: 40em) {  
  body { background-color: blue; }  
  p { padding: 5px 5%; }  
  .example { color: red; }  
}
```

Figura 3.24 *Media query* finalizado y que indica que se aplicarán los estilos correspondientes solo si el ancho de la pantalla es igual o menor a *40em*.

Es preciso aclarar que en la figura 3.24 no significa que ya se pondrán todas las declaraciones CSS restantes dentro de las llaves que representan al *media query*. Lo correcto es comenzar con todas aquellas declaraciones CSS que aplican independientemente del tamaño de la pantalla del dispositivo presente y

posteriormente, sobrescribir de los selectores y propiedades necesarios de acuerdo con las características de amplitud de pantalla⁷⁹.

Uso de los media Queries:

Como ya se vio con anterioridad, es posible incluir dentro de una misma hoja de estilo los *media queries* necesarios. Pero qué si se desea contar un hojas de estilo por separado y aplicarlas de acuerdo con las condiciones ya vistas.

Para agregar de manera tradicional una hoja de estilo y hacer referencia a ella debido a que es externa al documento *HTML*, basta con incluir dentro de la etiqueta *head*, la etiqueta *link* [51]:

```
<link rel="stylesheet" href="styles/mainstyles.css">  
<link rel="stylesheet" href="styles/otherstyles.css">
```

Entonces, en lugar de aplicar los *media queries* dentro de una misma hoja de estilo, suele ser una buena idea contar con una hoja separada e indicarle al navegador que solo la aplique si se cumplen las condiciones requeridas:

```
<link rel="stylesheet" href="styles/mainstyles.css">  
<link rel="stylesheet" href="styles/printstyles.css"  
media="print">
```

Por ejemplo, aquí se indica que la hoja de estilo llamada "*printstyles.css*" solo se aplique cuando se presente el modo impresión en el navegador. Se agrega después del estilo normal o principal (por decirlo de algún modo) "*manistyles.css*"

⁷⁹ Las declaraciones CSS sustituyen o sobrescriben a las anteriores si se utilizan los mismos selectores o propiedades; es por ello que se llaman hojas de estilo en cascada.

ya que las propiedades en vista de impresión deberán sobrescribir a las de la hoja de estilos principal en caso de ser necesario. Recordemos que son hojas de estilo en cascada porque la interpretación de sus propiedades van de acuerdo al orden (de inicio al final del documento) a como fueron establecidas.

El código anterior muestra el uso de una simple expresión *media=print* y como ya se había mencionado anteriormente, el uso de este tipo de *media query* es aplicable a partir de la segunda versión de la recomendación CSS (CSS2). Pero como ya se sabe, es posible usar *media queries* aún más sofisticados para poder incluir características respondidas a las páginas del sitio. Véase el siguiente ejemplo:

```
<link rel="stylesheet" href="styles/mainstyles.css">
<link rel="stylesheet" href="styles/widescreeen.css"
media="only screen and (min-width: 40em) ">
```

Se puede apreciar con claridad que en la segunda etiqueta *link*, la hoja de estilo “widescreeen.css” solo es aplicable si se trata de una pantalla y su ancho de despliegue (*viewport*⁸⁰) sea igual o mayor a 40em.

Se ve entonces que es bastante sencillo incluir *media queries* o hacer referencia a ellos (en caso de que estén separados del documento *HTML* y del estilo principal). Lo anterior nos da como desarrolladores y diseñadores web una flexibilidad enorme, dejando a un lado la preocupación del modelo o marca en específico del dispositivo que se emplea para visualizar los sitios. Ya se está más en la época donde los desarrolladores hacían una maquinación separada y

⁸⁰ Término que hace referencia al tamaño del área de despliegue de la pantalla del dispositivo o de la ventana del navegador web, en su defecto [51].

específica para los teléfonos inteligentes *iPhone (iPhone websites)*, sino en la época en la que ya debería ser posible navegar con la misma facilidad por un sitio ya sea desde un navegador de escritorio o un navegador instalado en un dispositivo móvil (cualquiera que este sea).

3.4.3 Otras formas de usar los *media Queries*:

Ya se conocen hasta aquí dos formas específicas para emplear *media queries*: ya sea mediante la inclusión en la hoja de estilo principal o enlazando una hoja de estilo separada indicando mediante el *media query* que solo es aplicable si se cumplen ciertas condiciones. La elección de acuerdo al tipo de uso que se dará va de acuerdo con las preferencias y prácticas del desarrollador o diseñador web [51].

A continuación se verán otras formas un tanto interesantes en las que es posible emplear los *media queries*:

Si se decide incluir un estilo dentro del documento *HTML* mediante la etiqueta *style*, es posible incrustar el *media query* como lo muestra la siguiente figura:

```
<style media="only screen and (min-width: 40em) ">  
  ...  
</style>
```

De esta forma, se está indicando que todos los estilos declarados dentro de la etiqueta *style* serán aplicables solo según la validez de las expresiones evaluadas por el *media query*.

Sin embargo, es posible además de la manera anterior, importar una hoja de estilo al momento de declarar el *media query*. Por ejemplo:

```
@import url(styles.css) only screen and (min-width: 40em)
```

En términos generales, es aún más recomendable emplear la etiqueta de enlace *link* en vez de importar hojas de estilo.

Hay una basta posibilidad de características de medios que pueden ser utilizadas como expresiones para los *queries*, tales como ancho y alto de la pantalla, ancho y alto de la región de despliegue del navegador (*viewport*), orientación, proporción entre ancho y alto (*aspect ratio*), resolución. La mayoría de las anteriores pueden estar acompañadas de los prefijos *min-* o *max-*. Las características más usadas para ser evaluadas por los *media queries*:

Para el ancho y alto de la región de despliegue de la página (*Viewport*): *width* y *height*.

Es importante aclarar que, a pesar de que tendemos a hablar de ancho de la pantalla al momento de realizar diseños responsivos, realmente lo que nos interesa como diseñadores es el ancho de la región de despliegue de la página en el navegador web, es decir el llamado *viewport*, no el monitor o pantalla en sí. es por eso que si se reduce el tamaño de la ventana del navegador cuando se está visualizando una página responsiva, ésta dará la respuesta necesaria para redistribuir los elementos de acuerdo al tamaño de la ventana tal y como lo haría con la variación de tamaños de los dispositivos [51].

- Ancho y alto del monitor del dispositivo: *device-width* y *device-height*. Funciona de manera muy similar a las propiedades anteriores, a diferencia de que las dimensiones evaluadas no son las de la región de despliegue sino las del tamaño real del monitor del dispositivo. Esto quiere decir que modificando el tamaño de la ventana del navegador no traerá en sí ningún efecto responsivo a la página, y si el usuario no tiene su ventana maximizada, probablemente no podrá visualizar el esquema que fue diseñado para ese propósito. Un ejemplo podría ser:

```
@media only screen and (max-device-width: 40em) { ... }
```

- Para la orientación: *orientation*. De acuerdo a la orientación física del dispositivo (orientación vertical o *portrait* en inglés y horizontal o *landscape* en inglés). Es una característica muy útil y realmente tiene mucho potencial, sin embargo, no todos los navegadores la soportan. Veamos un ejemplo:

```
@media only screen and (orientation: landscape) { ... }
```

La figura 3.25 muestra los tipos de orientación para un dispositivo móvil.

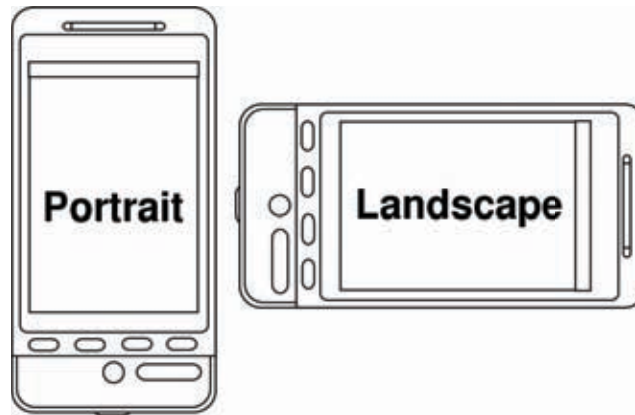


Figura 3.25 Las orientaciones físicas de un dispositivo móvil inteligente [56].

- Para la proporción ancho-alto del *viewport: aspect-ratio* y *device-aspect-ratio*. Se refiere a la proporción ancho-alto del *viewport*, por ejemplo, una área de despliegue de 1000px de ancho y una altura de 500px, su proporción ancho-alto sería 2:1⁸¹.

La proporción ancho-alto de las pantallas varían de manera considerable, a pesar de que a primera vista da la impresión de que todas las pantallas constan de un rectángulo de proporciones similares.

Algunas de las proporciones más comunes que se pueden encontrar en el mercado de las computadoras y dispositivos móviles serían 16:9, 16:10; 3:2 (como la pantalla del *iPhone* 3 y 4); 16:9 (*iPhone* 5). O las proporciones 4:3, 3:2, 16:10 o 16:9 (la mayoría de teléfonos *Android*).

⁸¹ El orden de la proporción siempre es ancho, alto [51]

```
@media only screen and (device-aspect-ratio: 16/9)
{ ... }
@media only screen and (min-device-aspect-ratio:
1920/1080) { ... }
```

- Para la resolución de la pantalla del dispositivo: *resolution*. Detecta la resolución del dispositivo. A pesar de que esto puede ser útil para presentar imágenes con resolución *ad-hoc* para la pantalla, aún no es soportado por navegadores como *Safari* y *Chrome*.

```
@media only screen and (min-resolution: 300dpi) { ... }
```

Lo que se puede evaluar con los media Queries:

Las propiedades más destacables para evaluar con los *media queries* son [51]:

- *color*: Evalúa el número de bits por cada componente de color de la pantalla siempre y cuando no se utilice el prefijo *min-*. Por sí sola, la palabra *color* evalúa se el dispositivo cuenta con una pantalla a color.
- *monochrome*: Si la pantalla del dispositivo es monocromática.
- *scan*: Evalúa si el dispositivo utiliza escaneo progresivo⁸².
- *grid*: esta expresión sirve para determinar si la pantalla o terminal de salida está basada en rejillas o mapa de bits (como es el caso de una pantalla normal que funciona mediante pixeles).

⁸²Solo aplica a los televisores por el momento y cuya resolución es de 480p, se trata de una forma en la que se despliega, almacena, o se transmiten imágenes en movimiento en las cuales cada una de las líneas del marco o *frames* son impresos o dibujados de manera secuencial [51].

3.4.4 Diseñando de manera responsiva

Ya se sabe con base en capítulos anteriores que el desarrollo web puede verse desde una perspectiva de capas. Cuando se construye un sitio web, se comienza con el contenido y *HTML* es el lenguaje que nos permitirá estructurarlo de manera semántica; ésta sería la primera capa.

La siguiente capa vendría a ser la de presentación; como se sabe, *CSS* es el lenguaje diseñado para este fin: determinar cómo se verá todo lo que las páginas del sitio incluyen (el esquema de distribución, los colores, la tipografía, etc.).

Finalmente, se tiene la capa de comportamiento; el lenguaje recomendado por la *W3C* para este propósito es *Javascript*. Es cierto que *javascript* nos permitirá dar un toque dinámico al sitio y por ende su contenido (por ser interactivo) tendrá un enriquecimiento considerable; empero, es necesario asegurarse que la funcionalidad básica puede llevarse a cabo aún con el motor *Javascript* deshabilitado en el navegador (es decir, el navegador no contará con soporte para interpretar los *scripts*). Lamentablemente, esto resultaría muy complicado para aquellos sitios que dependan mucho de su contenido interactivo [51].

El diseño con rejillas:

Cuando se comienza el desarrollo web o la maquinación de un sitio en modo responsivo, usualmente hacemos referencia a lo que en inglés se conoce como rejillas flexibles ("*flexible grids*"). Es necesario aclarar que a pesar de que la flexibilidad es inherente al diseño responsivo, las rejillas no son necesarias para

este proceso; sin embargo, éstas son parte importantísima para un buen diseño de un sitio web [51].

La idea o concepto de una rejilla es que el diseño se constituya de múltiples columnas de igual anchura, mismos márgenes entre cada columna y todo el contenido y elementos de la página se basa en esto [51].

El uso de rejillas de 12 columnas es lo más usado, pues aporta mucha flexibilidad. La razón es simple, 12 es un número divisible entre 2, 3, 4 y 6; lo cual nos da la posibilidad de establecer esquemas de cualquiera de esos valores [51]. La figura 3.26 ilustra cómo sería un diseño con rejillas.



Figura 3.26 Ejemplo característico de un esquema de diseño web basado en rejillas [51].

Desde luego, en un sitio responsivo, las celdas o cajas (como las que se muestran en la figura 3.26) pueden ser re-ordenadas para diferentes anchuras del *viewport* para que puedan caber en la misma u otra rejilla.

Diseñar de esta manera nos permite darle a las páginas del sitio cohesión y coherencia [51].

¿Comenzar el diseño para pantallas pequeñas?

Cuando se emplearán los *media queries*, que como se sabe, se hará para cambiar el esquema o diseño del sitio para distintos tamaños de pantalla; por lo general se parte de una dirección hacia la otra; es decir, ya sea comenzar por diseñar para pantallas de escritorio y portátiles y de ahí comenzar a agregar los *media queries* apropiados para agregar la característica responsiva de acuerdo a la variación de tamaño de monitor o pantalla.

Muchos expertos en diseño web aseguran que es mucho más sencillo partir desde un esquema de diseño e ir haciéndolo más grande, en vez de ir en dirección contraria (ir reduciendo el tamaño del esquema visual o de diseño). La razón es que una vez que se cuente con un esquema visual satisfactorio para pantallas angostas (pequeñas), se tendrá espacio considerable en *viewports* más amplios para mover y reacomodar los elementos de la página. Si el diseño parte en dirección contraria, se considera un tanto más complicado por el hecho de tener que, por decirlo de alguna manera, “apachurrar” el esquema completo de diseño, lo cual puede complicar un tanto las tareas de diseño o maquetación del sitio [51].

Para fines demostrativos de lo anterior, y dado que este capítulo se titula “Diseñando de manera responsiva”, se expondrá a grandes rasgos el proceso de diseño responsivo partiendo desde un esquema pequeño y ampliándolo posteriormente.

En el capítulo 3.3 (Diseño web), se dieron a conocer las principales etiquetas semánticas de *HTML5* y su propósito. Para fines de esta demostración se hará uso de ellas.

La siguiente es una breve demostración del proceso para un diseño responsivo [51].

Comenzando con la sección *article* y el número de caracteres por línea que se incluirá. Lo más sencillo es contar con un rango de 45 a 75 caracteres en cada línea; esto aumentará la legibilidad para los visitantes.

El esquema visual de partida es el que muestra la figura 3.27.

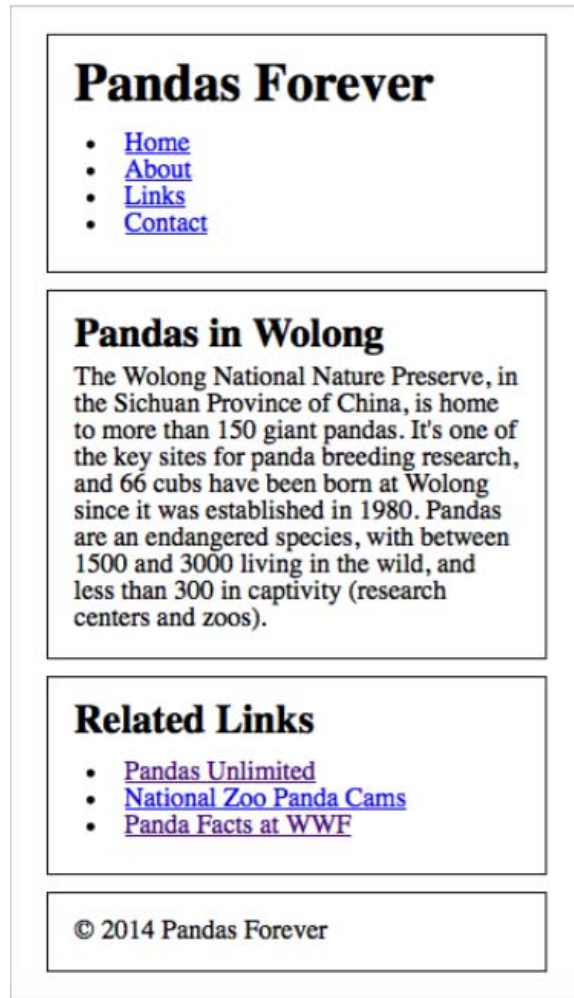


Figura 3.27 Esquema visual correspondiente a un *viewport* de un dispositivo móvil [51].

En las pantallas más pequeñas, hay 45 caracteres por línea. Cuando el sitio sea visualizado en pantallas más amplias, será deseable contar con más de 75 caracteres por línea. Para encontrar el punto de quiebre o terminación (*breakpoint*), comenzaremos a ampliar manualmente la anchura de la ventana del navegador y detenernos en el momento en que el texto exceda los 75 caracteres;

para medir la anchura de la zona de despliegue (*viewport*) podemos usar una herramienta en línea llamada *MQtest* (<http://mqtest.io>); en este caso, 36em.

Una vez que se cuenten con más de 75 caracteres por línea en el documento, se añadirá el punto de quiebre o terminación (conocido como *breakpoint*). La figura 3.28 ilustra lo anterior.

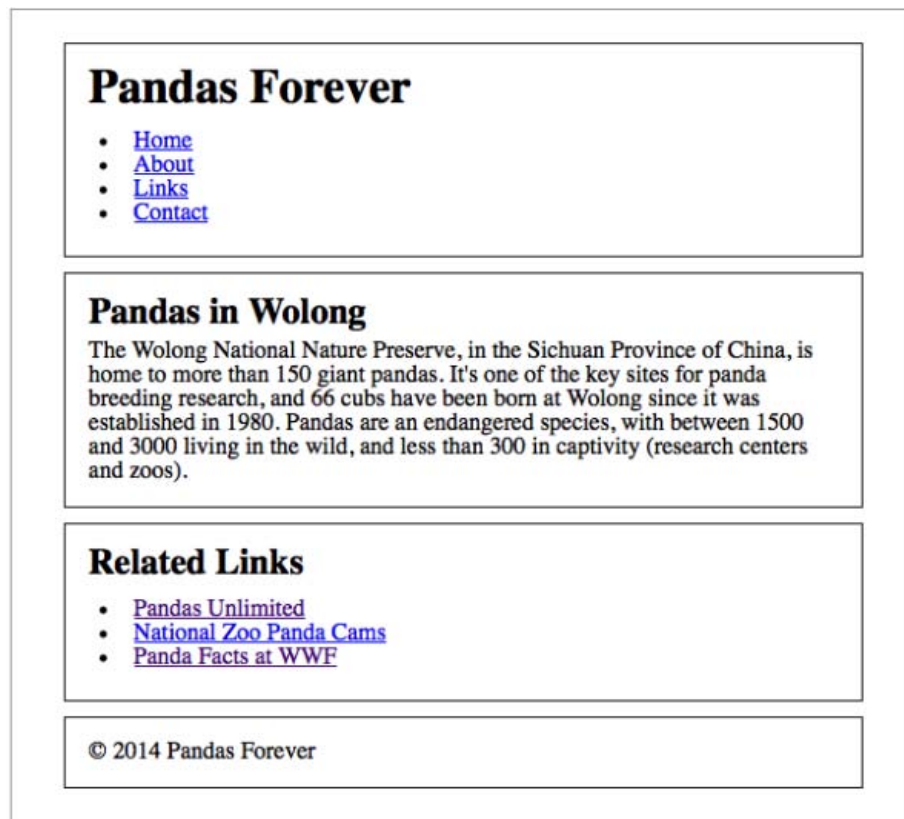


Figura 3.28 Se muestra el *breakpoint* para el esquema de esta breve demostración.

Ahora, es necesario que el esquema visual sea desplegado en dos columnas; utilizando la directiva *float* (flotante) en la hoja de estilo para esto, pero será necesario indicarle al navegador que este esquema solo será aplicable a

viewports de 36em o mayores. ¿Cómo se hace esto? Desde luego, Mediante un *media query*.

La etiqueta *article* estará del lado izquierdo y será contenido dentro de una columna más amplia que en el caso de *aside*; la cual será colocada junto con los enlaces relacionados dentro de una columna más angosta del lado derecho del documento. Se agregará la directiva *clear* al pie de página (*footer*) para “limpiar los efectos de flotación de la directiva anterior y así comenzar con una nueva línea”. Se tendrá entonces algo similar al siguiente código CSS:

```
article { float: left; }
aside { float: right; }
footer { clear: both; }
```

Hasta aquí se tienen algunos problemas. Inicialmente, no se ha asignado una anchura específica a los elementos anteriores, lo cual hace que el bloque *article* llene todo el ancho de la página y por ende el bloque de enlaces relacionados termine debajo. La figura 3.29 describe gráficamente el problema:

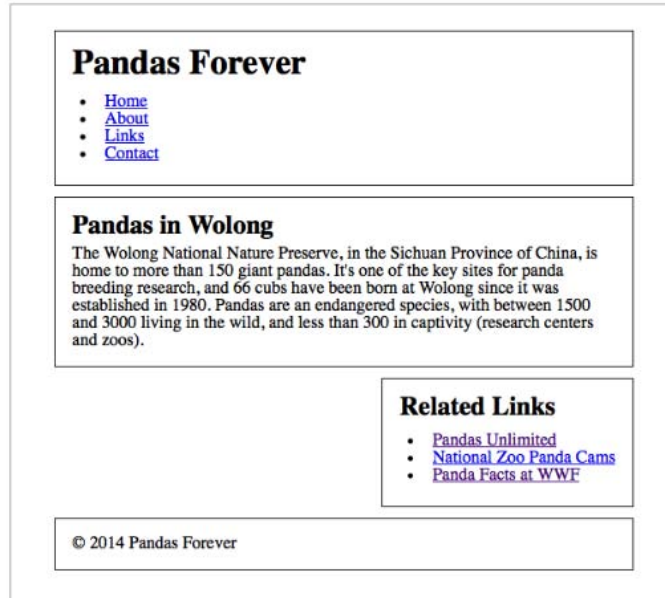


Figura 3.29 Se ilustran las consecuencias de omitir la anchura específica de los elementos contenedores.

Para resolver el problema descrito anteriormente, es necesario asignar manualmente las anchuras a los elementos. Para fines estéticos de la página, se utilizará una rejilla de cuatro columnas, 21% de ancho de cada una; márgenes de 2% entre ellos y 5% los márgenes izquierdo y derecho; completando así el 100% del ancho del *viewport*. La figura 3.30 ilustra lo anterior:

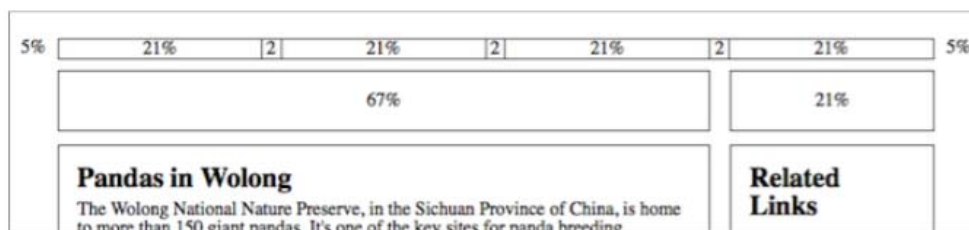


Figura 3.30 La anchura ha sido especificada y ya es posible tener un esquema como el planeado en un inicio.

El bloque *article* abarcará tres columnas a lo ancho, lo cual acumulará el 67%. la columna de la izquierda es del 21% del total de la página y contamos con un margen del 2% entre las columnas.

```
article { float: left; width: 67%; }  
aside { float: right; width: 21%; }  
footer { clear: both; }
```

A pesar de que ya está cerca el objetivo final, aún queda un ligero problema por resolver: En total los elementos junto con los márgenes hacen un total del 108%.

Es necesario eliminar el margen derecho de un bloque y el izquierdo del otro.

Quedando así:

```
article { width: 67%; float: left; margin-right: 0; }  
aside { width: 21%; float: right; margin-left: 0; }  
footer { clear: both; }
```

Una vez eliminados los márgenes sobrantes, el esquema quedaría como se muestra en la figura 3.31.

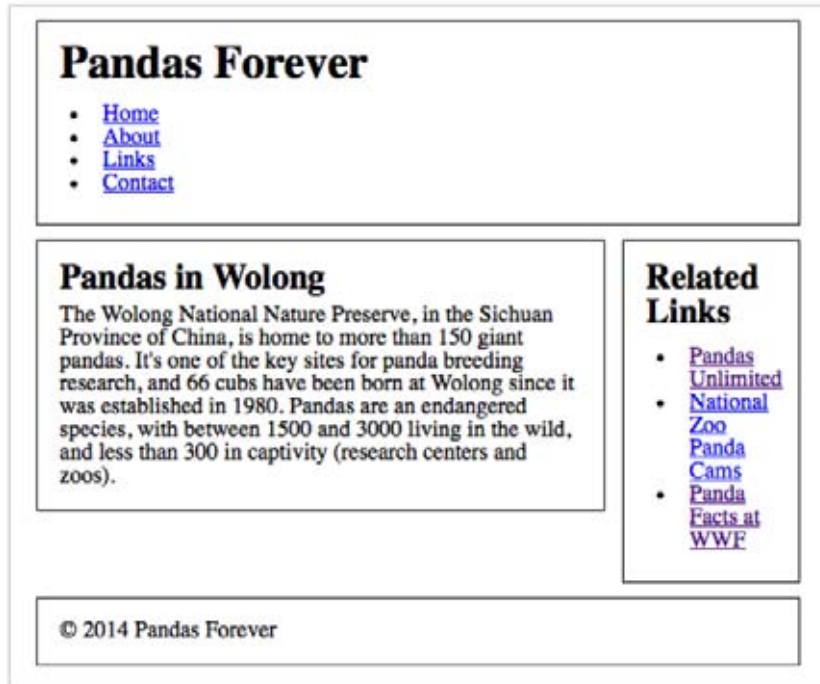


Figura 3.31 Se eliminan los márgenes sobrantes y ya se cuenta con un esquema bastante aceptable.

Agregar el *media query* necesario es bastante sencillo. Se desea que el punto de quiebre o *breakpoint* ocurra cuando se llegue a los 36em; para esto es necesario indicarle al navegador web que en caso de que el *viewport* rebase los 36em de anchura aplicará el estilo definido anteriormente para los bloques *article*, *aside*, *footer*.

```
@media only screen and (min-width: 36em) {
  article { width: 67%; float: left; margin-right: 0; }
  aside { width: 21%; float: right; margin-left: 0; }
  footer { clear: both; }
}
```

Quedando los dos esquemas visuales definidos de la manera mostrada en la figura 3.32.

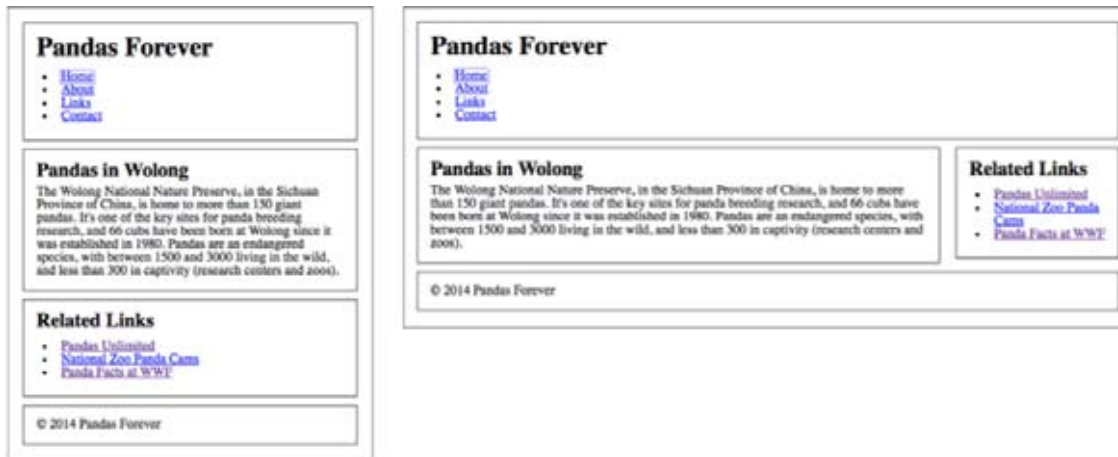


Figura 3.32 Del lado izquierdo se tiene el esquema para móviles. Del lado derecho el esquema para pantallas de escritorio.

Un último detalle. La demostración anterior consta de una pequeña página cuyo contenido no se presta para ser representado en un esquema de tres columnas; entonces, es conveniente en este caso establecer una anchura máxima para la página. Para esto, lo más viable es crear una etiqueta *div* (contenedora) para atrapar dentro de ella todo el cuerpo del documento; a dicha etiqueta que representa el bloque de toda la página sería conveniente asignarle un identificador (*id*); y como todo *id*, se el asignará un nombre: *fullpage*, para fines prácticos.

Se establecerá entonces que la anchura máxima de la página será de 63em y todo el cuerpo de la misma debería (por estética) quedar centrado. Se asigna con la directiva *margin* el margen en automático para que éste quede dividido por igual tanto del lado izquierdo como del derecho. Nos queda entonces algo como en el código siguiente:

```
#fullpage { max-width: 63em; margin: auto; }
```

Y finalmente, en la figura 3.33 se muestra el esquema con anchura fija.



Figura 3.33 Esquema con un ancho fijo que se verá igual en cualquier *viewport* ancho.

4. Análisis del cliente: Alinids

4.1 Breve análisis de mercado

A continuación se abordará sobre un sencillo pero importante análisis del perfil del cliente sobre el que yace la propuesta de solución cuyo desarrollo se plantea y expone en el capítulo subsecuente.

Alinids es una pequeña empresa que cuenta actualmente con cuatro líneas de productos⁸³: Sistemas de Circuito Cerrado de Televisión, Alarmas, Interfonos y Cercas electrificadas. Todas sus líneas de productos llegan a la tienda gracias a un proveedor específico.

En cuanto a servicios al producto, cuentan un pequeño personal técnico que se encarga de la instalación de los productos que el consumidor adquiere.

Los clientes potenciales de la empresa son por el momento, de acuerdo con el cliente, aquellas personas que residen en la misma zona donde se ubica su único establecimiento comercial. Sin duda, lo anterior representa una problemática debido a que se refleja una falta de estrategia de mercadotecnia para lograr objetivos de mejorar su posicionamiento, expandir el negocio y diversificar sus productos.

Al ser una pequeña empresa aún, Alinids tiene un objetivo a mediano – largo plazo: convertirse en una mediana empresa mediante un estrategia de expansión.

⁸³ Una línea de productos se puede definir como un grupo de artículos estrechamente relacionados entre sí [57].

Se busca un crecimiento intensivo, es decir, crecer dentro del mercado de referencia, mediante [58]:

- Una estrategia de penetración: La empresa busca aumentar las ventas para el mercado de los hogares.
- Una estrategia de desarrollo de mercado: Incrementar ventas de sus productos en nuevos mercados: establecimientos comerciales y empresas.

El medio

El medio para sus estrategias de penetración y desarrollo de mercado es la Web; es decir, contar con presencia en *Internet*.

El crecimiento y posibilidades que *Internet* ofrece a las empresas para mantenerse en contacto con sus clientes y consumidores las ha impulsado a crear blogs corporativos, apoyarse en el uso de las redes sociales, y explotar las capacidades de las nuevas tecnologías en la Web para la mejora notable y sustancial de la apariencia y usabilidad de las páginas del sitio [32]. Para lo anterior, el cliente no cuenta con los conocimientos técnicos requeridos para atacar el problema. Una maquetación de calidad (un buen diseño de sitio web) será la primer etapa de muchas que se requerirán para abordar su necesidad de mayor penetración en sus ventas.

4.2 Análisis de requerimientos del cliente (No funcionales)

Introducción

A continuación se presenta la especificación del propósito y alcance de este prototipo de sitio. Se utilizarán como base las directrices presentadas por el estándar *ANSI/IEEE 830 1998*⁸⁴.

Propósito

El objetivo de esta especificación es esclarecer todas y cada una de las características proporcionadas por el cliente en cuanto al diseño de la interfaz gráfica del prototipo de sitio solicitado.

Alcance

Esta especificación de requisitos está dirigida al cliente que a su vez fungirá como usuario del producto, con el fin de poder esclarecer las características de usabilidad que se demandan, así como el listado y distribución de cada uno de los componentes gráficos que conformarán las páginas del prototipo o maqueta.

Definiciones y acrónimos

Definiciones:

- Sitio: Para propósitos de esta especificación, cada vez que se emplee el termino "sitio" quedará por entendido que se está haciendo referencia al prototipo o maqueta del sitio web; es decir, toda la parte frontal y visible con la que el usuario tiene un interacción directa. Entiéndase prototipo o

⁸⁴ *IEEE recommended practice for software requirement specification* [59].

maqueta como todo lo relacionado a la interfaz gráfica que constituye el producto.

- *Hosting*: servicio de alojamiento de páginas web para ser accesibles desde cualquier dispositivo con conexión a *Internet*.
- *Layout*: Esquema de distribución de cada componente gráfico y texto que conforma una página web.
- *Ítem*: Elemento específico de un componente.
- Ventana Modal: Ventana flotante respecto al cuerpo o contenido de la página.
- *Iframe*: Un marco que embebe un contexto de navegación dentro de un componente específico o bloque de la página web. Uno de los usos más comunes es la de embeber un video.

Acrónimos:

- RNFXX: se utilizará en cada ocasión que se haga referencia a un requisito no funcional específico. Donde XX corresponde al número de requisito en cuestión.

Descripción general.

Descripción del producto.

En este apartado se presenta una descripción de todos los componentes gráficos que conforman al sitio, así como la forma en la que responden cuando el usuario hace uso de ellos. Así mismo, se asentarán las características de los usuarios involucrados y las restricciones presentes del sitio.

Funcionalidad

El sitio servirá como plantilla base que demuestre mediante la distribución de sus componentes gráficos (*layout*) y el comportamiento de cada uno de ellos si se ha logrado satisfacer al cliente con respecto a sus necesidad de contar con un prototipo funcional de sitio web para su negocio.

Características de los usuarios

El usuario principal de este sitio es el cliente, el cual no cuenta con conocimientos técnicos avanzados pero sí básicos para navegar por la Web desde una computadora o un dispositivo móvil.

Restricciones

No existe comunicación con algún servidor de bases de datos por lo cual la presentación del contenido será meramente estática.

Sí se contará con un servicio de *hosting* gratuito, sin embargo, será con la finalidad de la realización de pruebas de integración.

El sitio será maquetado mediante las tecnologías de *HTML5*, *Javascript* y *CSS*

Requerimientos funcionales

Este apartado quedará omitido dadas las restricciones y el alcance previamente establecidos.

Requerimientos no funcionales

En este apartado se definirán todos los aspectos relacionados con el diseño de la interfaz gráfica para poder culminar con el proceso de maquetación.

Componentes gráficos

RNF01

La página interna estará compuesta por 3 secciones :

- Sección de encabezado
- Sección de redes sociales y contacto
- Pie de página

RNF02

El sitio estará conformado por 4 páginas: Inicio, Blog, Productos y Contacto, las cuales tendrán en común la página interna descrita en el RNF01.

RNF03

La sección de encabezado estará conformada por:

- Logotipo de la empresa en miniatura.
- Menú de navegación con 4 *items*, cada uno de los cuales será un enlace a las una de las 4 páginas del según describe el RNF02.

RNF04

La sección de redes sociales y contacto estará conformados por los siguientes componentes:

- 3 enlaces gráficos a redes sociales
- Un bloque de texto con la información de contacto

RNF05

la sección de pie de pagina estará conformada por los siguientes componentes:

- Logotipo en miniatura de la empresa
- Menú de navegación secundario
- Créditos de los creadores del sitio

RNF06

El contenido de la página de inicio estará compuesta por las siguientes secciones:

- Sección de Bienvenida
- Sección de Suscripción
- Sección de Información de la empresa
- Sección de Video introductorio
- Sección de Clasificación de productos
- Sección de Clasificación de clientes

RNF07

La sección de bienvenida estará conformada por:

- El logotipo de la empresa
- Un bloque de texto de bienvenida
- 3 imágenes para productos destacados con sus respectivas etiquetas de precio

- Un enlace tipo botón a la página de productos

RNF08

La sección de suscripción tendrá los siguientes componentes:

- Texto informativo
- Un enlace tipo botón a una ventana modal

RNF09

La sección de Información de la empresa estará conformada por:

- Texto de encabezado
- 2 párrafos correspondientes a la visión y misión de la empresa

RNF10

La sección de video introductorio solo estará compuesta por un marco de video (*iframe*).

RNF11:

La sección de Clasificación de productos estará conformada por:

- Una imagen de encabezado
- 4 imágenes descriptivas para cada tipo de producto con sus respectivos enlaces.

RNF12

La sección clasificación de clientes estará compuesta de 3 imágenes descriptivas para cada tipo de clientes de la empresa.

RNF13

La pagina Blog estará conformada por 2 secciones

- Sección de encabezado que indica el nombre de la página
- Sección del contenido general del blog.

RNF14

La sección del contenido general del blog estará conformada por los siguientes componentes:

- Un número indefinido de bloques para cada publicación del blog
- Un bloque con un enlace a una ventana modal para dar a un usuario de alta para recibir futuras publicaciones del blog.
- Un bloque con una caja de texto para búsqueda de contenido específico del blog
- Un bloque para mostrar a modo de lista las publicaciones mas recientes del blog
- Un bloque para presentar la clasificación de las publicaciones de acuerdo con su categoría.

RNF15

La página de productos estará conformada por 3 secciones:

- Sección de encabezado que indica el nombre de la página
- Bloque de texto informativo
- Sección de mosaico de productos disponibles

RNF16

La sección de mosaico de productos disponibles mostrará inicialmente 12 bloques, cada uno de los cuales mostrará a su vez la imagen del producto y texto descriptivo, así como un enlace a una ventana modal para ingresar datos e iniciar el proceso de compra.

RNF17

La ventana modal para iniciar el proceso de compra estará conformada por un formulario de datos específicos del usuario para que se proceda a completar la transacción y un botón para confirmar la compra.

RNF18

La página de contacto constará de 2 secciones:

- Sección encabezado que indica el nombre de la página
- Sección de atención y sugerencias

RNF19

La sección de atención y sugerencias estará compuesta por:

- Texto descriptivo
- Un formulario con 3 campos para entrada de texto (nombre, correo electrónico y comentario, queja o sugerencia).
- Un botón para el envío de la información ingresada

RNF20

El sitio deberá ser compatible con los navegadores *Firefox* (sin restricciones de versión), *Chrome* (sin retracción de versión) e *Internet Explorer* (a partir de la versión 8).

Layout

Todo los requerimientos relacionados a este apartado, es decir la forma en la que los componentes especificados en los requerimientos anteriores van a estar distribuidos a lo largo y ancho de todas las páginas, se representan en diagramas específicos para este propósito⁸⁵.

La figura 4.1 muestra la distribución especificada por el cliente para la pagina interna.

⁸⁵ Los diagramas (*wireframes*) fueron realizados con una aplicación web llamada *Moqups*, disponible en <https://moqups.com>.



Figura 4.1. *Layout* de la página interna. Compuesta del menú de navegación en la encabezado principal, la sección de redes sociales e información de contacto; y el pie de página con sus respectivos componentes

En la figura 4.2 se muestra el *layout* correspondiente especificado para la página de inicio.

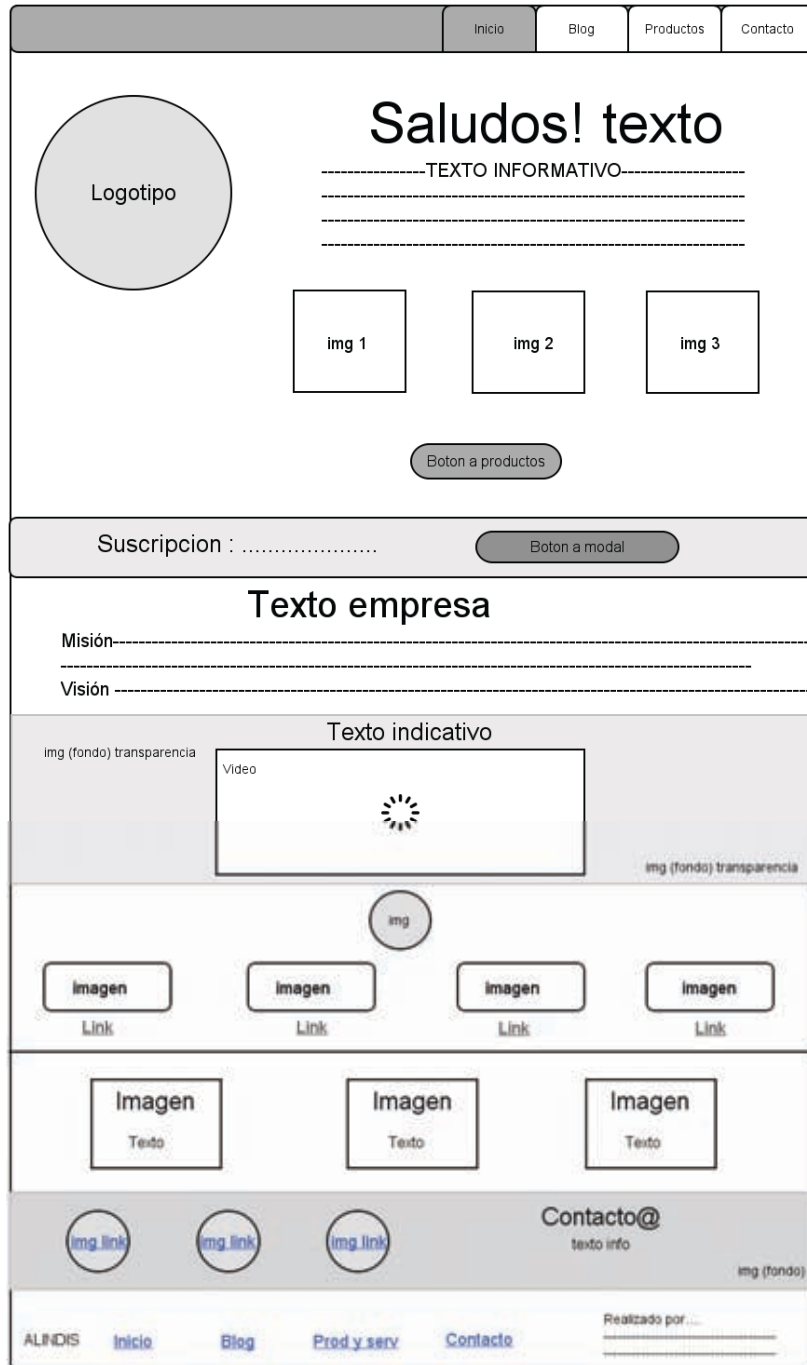


Figura 4.2 *Layout* para la página inicial con sus secciones: Bienvenida, Suscripción, Información de la empresa, Video introductorio, Clasificación de productos y Clasificación de clientes según su orden de aparición.

En la figura 4.3 se ilustra el esquema de distribución correspondiente a la página de Blog, según lo especificó el cliente.

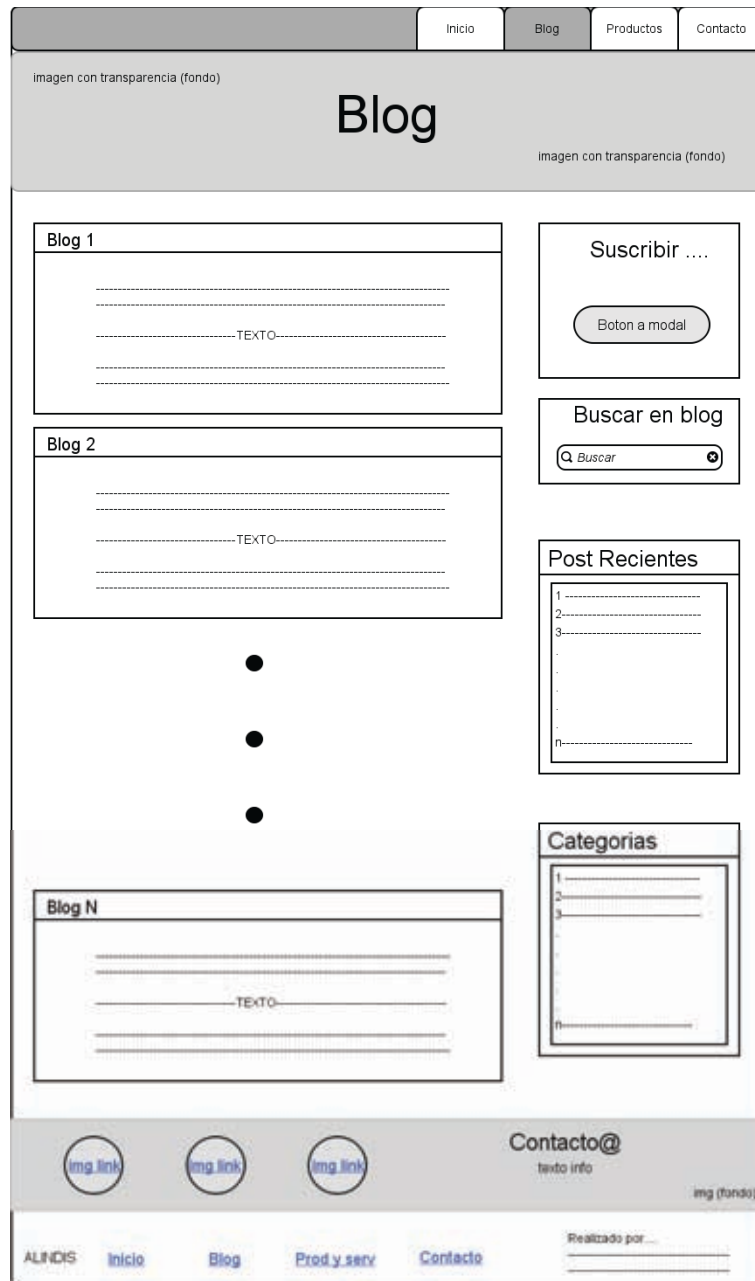


Figura 4.3 *Layout* que corresponde a la página de Blog. Una cantidad aún no definida de bloques dentro de los cuales estará el contenido de cada publicación. Y los demás bloques en la columna de la derecha

La figura 4.4 representa el *layout* correspondiente según la especificación del cliente para la pagina Productos.

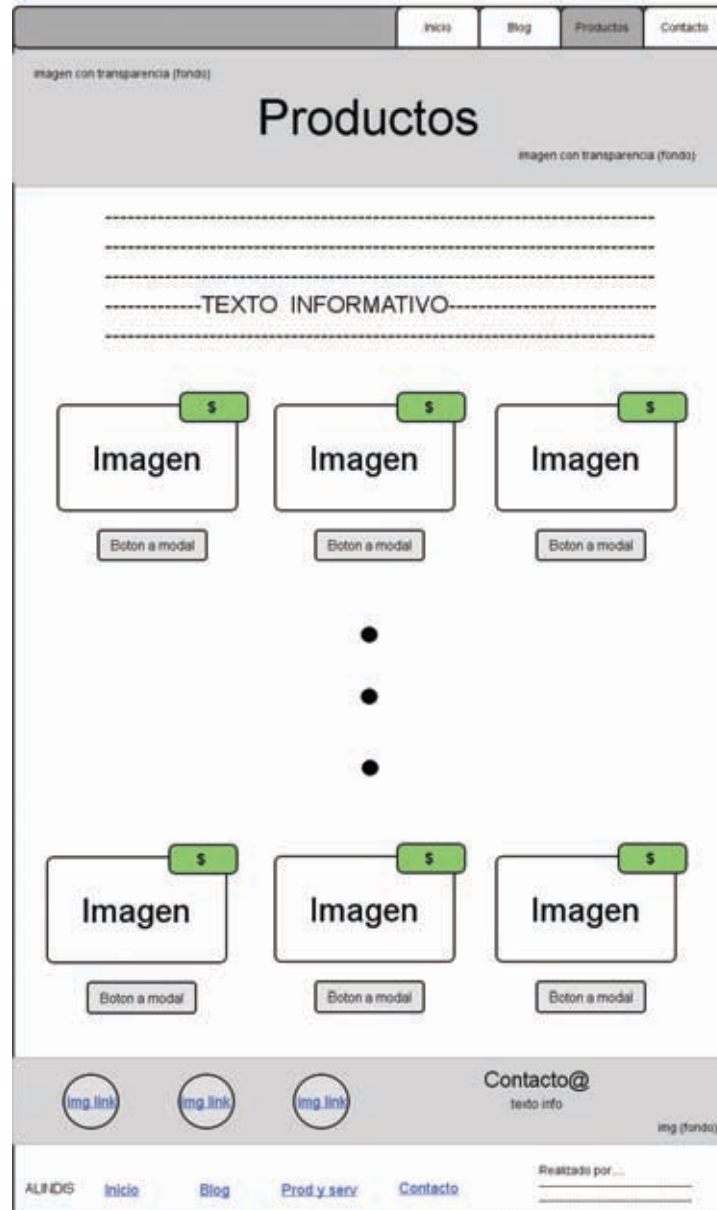


Figura 4.4 De esta manera quedaría el *layout* para la página de productos. Se cuenta con un mosaico de inicialmente 12 bloques. Dicho mosaico se convertiría en una lista (bloques apilados) si el tamaño de la pantalla comienza a ser reducido a lo ancho al grado que solo quede espacio para un bloque por fila. Esta es una de las características responsivas que la maqueta presentará.

La pagina de contacto tiene un *layout* especificado por el cliente como lo ilustra la figura 4.5.



Figura 4.5 Se muestra el *layout* correspondiente para la página Contacto, donde los visitantes y clientes activos podrán manifestar dudas, quejas o sugerencias.

Finalmente los requerimientos del *layout* iniciales para las ventanas modales que aparecerán en el sitio son ilustradas por las figuras 4.6 y 4.7

Suscribete

-----TEXTO INFORMATIVO-----

-----TEXTO INFORMATIVO-----

Nombre Email Address

-----TEXTO INFORMATIVO-----

-----TEXTO INFORMATIVO-----

Figura 4.6 *Layout* de la ventana modal que será utilizada para la subscripción a promociones o boletines de noticias.

COMPRA

 -----Texto descriptivo-----

Confirmar compra

-----Texto informativo-----

Cantidad

Nombre

Apellidos

Direccion

Delegacion

C.P.

e-mail

Telefono

Recordar mi informacion

Figura 4.7 *Layout* de la ventana modal para el ingreso de datos y realización de la compra o solicitud.

Usabilidad

La maquetación responsiva es una característica esencial [54] que se cubrirá y que paralelamente aportara características de usabilidad y accesibilidad muy importantes ahorrando costos y esfuerzo en la redefinición de la experiencia de uso para distintos tamaños de dispositivo en los que el sitio será visualizado.

Estilo

Las características relacionadas al estilo y gama de colores para el sitio no fueron proporcionados de manera explícita por el cliente, dejando este aspecto al criterio de los desarrolladores involucrados en este proyecto.

5. Propuesta de solución

5.1 ¿Por qué *Bootstrap* 3?

Como se enfatizó en capítulos anteriores, el diseño web moderno debe enfocarse en la experiencia del usuario y combinarlo con diseños adaptables, que puedan ser apreciados sin importar el dispositivo desde el que se acceda al sitio con el fin de tener la mayor cantidad de visitas y, desde el ámbito comercial, poder trascender y lograr éxito que cualquier empresa o negocio desea. Esta tarea puede hacer del desarrollo web algo más complejo, y es allí donde debemos sacar provecho a herramientas existentes que nos agilizan el trabajo y nos permiten conseguir resultados más impactantes en el menor tiempo posible. Se ha mencionado que dichas herramientas, conocidas como *frameworks*, surgieron precisamente por las necesidades que se han ido presentando a causa de la evolución en el desarrollo de la Web [4] [60]. Actualmente existen muchos y muy variados marcos de trabajo que facilitan esta tarea ahorrando escritura de código y por consecuencia tiempo al desarrollador. Un marco de trabajo para el diseño web como es el caso de *Bootstrap*, puede definirse como una colección de herramientas para la creación de sitios y aplicaciones web. Por lo general, proveen de plantillas *HTML* y *CSS* para cubrir aspectos de tipografía, formularios, elementos de navegación, botones diversos componentes de la interfaz gráfica del usuario [61].

En el año 2011, *Bootstrap* nace como un proyecto de solución a una problemática interna de inconsistencias de código en el equipo de desarrollo de *Twitter*. Básicamente no existía como tal una estructura en la codificación del

software de la plataforma social y optaron por diseñar un marco de trabajo para ello al que bautizaron con el nombre *Twitter Bootstrap* [62].

Twitter Bootstrap Es un proyecto *Open Source*⁸⁶ [63], lo cual ha permitido que muchos expertos colaboren constantemente en sus mejoras.

En agosto del 2013, dos años tras el lanzamiento oficial de la primera versión, se publica la versión 3 de este marco de trabajo. Esta última versión cuenta con nuevos esquemas de diseño, mejor manejo de errores, más y mejores opciones de personalización para los componentes visuales [62].

La idea principal detrás de *Bootstrap* es facilitar enormemente el desarrollo web *frontend* para la creación de maquetas responsivas (Sí, diseño web responsivo; tema ya cubierto en capítulos anteriores); además de contar con un prototipo de sitio web de fácil mantenimiento, coherente y cohesivo en cuanto a código y con características que permiten escalabilidad del mismo sitio o aplicación [63].

Ya sea que se busque crear alguna aplicación web, un blog, o algún sistema para la gestión de contenidos; *Bootstrap* siempre será una herramienta que se adaptará a las necesidades del desarrollador web.

Las características más destacables de *Bootstrap* se enlistan a continuación [63]:

- Para comenzar a crear el esquema de distribución de elementos (*layout*) es importante recalcar que en la versión 2 de *Bootstrap* se incluían algunas

⁸⁶ Modalidad de desarrollo de software donde el código de una determinada aplicación es liberado para estar disponible a la comunidad de desarrolladores en el mundo y poder modificarlo y mejorarlo [64].

utilidades para que las páginas se adaptaran a móviles. *Bootstrap* 3 se rediseñó por completo para tomar un enfoque 100% en dispositivos móviles; y su filosofía ahora es “*Mobile First*” o “Los móviles son primero”. Utilizar un *framework* con tal enfoque al día de hoy es una decisión sabia [60].

- La licencia de *Bootstrap* indica que podemos desarrollar y distribuir con fines comerciales cualquier aplicación sin ningún problema.
- El mantenimiento y actualización de este *framework* es realizado por *Twitter*.
- Su gran popularidad hace posible que la comunidad de usuarios puedan plantear y resolver en foros o blogs preguntas y respuesta con el fin de ayudar a usuarios que apenas inician a usar este *framework*.
- La documentación que brinda se encuentra muy bien estructura y cuenta con tutoriales y ejemplos con código para resolver dudas incluso a usuarios expertos.
- Existen cientos de tutoriales en la Web que facilitan enormemente el aprendizaje de la herramienta.
- Utiliza componentes vitales para los desarrolladores web (*HTML5*, *CSS3*, *Javascript*, *JQuery*, *GitHub* , entre otros) apegándose así a estándares *W3C*.
- Sus sistema de rejillas (*Grid System*) facilita en gran manera la maquetación por columnas siempre manteniendo la esencia del diseño responsivo.

- Es muy flexible gracias al uso de LESS. Cualquier desarrollador con conocimientos de este preprocesador podrá personalizar *Bootstrap* aún más según sus necesidades de diseño.
- La popularidad de *Bootstrap* hace que con el paso del tiempo se añadan cada vez más extensiones (*plugins*) que dan resultado a nuevas y mejores características para el marco de trabajo, haciéndolo cada vez más poderoso y flexible.
- Ofrece paquetes con muchos elementos y componentes personalizables compuestos por combinaciones de *HTML5*, *CSS3* y además de *jQuery*.
- Facilita la implementación de alternativas en código para la interpretación del sitio a través de tecnologías de apoyo para personas con capacidades especiales o diferentes.

Bootstrap respecto a otros frameworks frontend

Existe una gran variedad de *frameworks* para la maquetación web. Enunciar y comparar a *Bootstrap* respecto a cada uno es un tanto complejo; por lo cual, se elegirán 4 de los más populares hoy en día [65].

Por medio de la indagación realizada sobre las características esenciales de cada *framework* elegido se logró la creación de la tabla comparativa que se muestra en la página siguiente [61] [65] – [71].

	Sistema de Rejillas (Grid)	Elementos de Interfaz de Usuario (UI)	Unidades CSS	Soporte de Preprocesadores CSS	Soporte IE	Observaciones adicionales
Bootstrap**	Responsivo, basado en 12 columnas	Extensa cantidad de componentes de interfaz gráfica	Px	LESS y SASS	IE8+	Facilita el desarrollo rápido de prototipos web.
Foundation**	Responsivo, basado en 12 columnas	Extensa cantidad de componentes de interfaz gráfica	Rem	SASS	IE9+	Requiere mayor énfasis en el proceso de diseño.
Ink**	Responsivo, basado en 12 columnas	Pocos*	Px	SASS	IE10+	Poca documentación en la Web.*
Zimit**	Responsivo, basado en 12 columnas	Pocos*	No especificado	LESS	IE9+	Componentes con poco énfasis en el diseño simple y minimalista*
Html Kickstart**	Responsivo como característica opcional. No tan intuitivo*	Pocos*	Px	SASS	IE10+	Carece de algunas características y funcionalidades*
Pure CSS**	Responsivo basado en 5 y 24 columnas	Pocos*	Px	Sin soporte	IE7+	No tiene un enfoque móvil por defecto. La maquetación responsiva es opcional.

* En comparación con Bootstrap 3 y Foundation 5.

** En sus versiones más recientes a mayo de 2015.

Como puede observarse en la tabla comparativa, los únicos *frameworks* que cumplen con el RFN20 (Capítulo 4.2) son *Pure CSS* y *Bootstrap*; sin embargo, el principal problema respecto a *Pure CSS* es la poca cantidad de componentes gráficos (por ejemplo, no hay ventanas modales, se puede realizar mediante pero no existirá soporte para IE8) [72].

Otra de las ventajas más fuertes de *Bootstrap* respecto a los demás en la tabla es sin duda el soporte de los preprocesadores⁸⁷ *LESS* y *SASS* [61]. Ambos son considerados los preprocesadores más populares actualmente y aportan una inmensa gama de extensibilidad y flexibilidad al momento de escribir código CSS [74]. El hecho de que *Bootstrap* cuente con soporte para ambos preprocesadores le da una característica de flexibilidad aún mayor en cuestiones de mantenimiento y escalabilidad de las maquetas web que se pueden crear con la herramienta [63].

Foundation es considerado como la alternativa más fuerte frente a *Bootstrap*; empero, no cuenta con compatibilidad con IE8 y utiliza *rem* como unidad de medida CSS⁸⁸ [61]. Una de las principales razones por las que muchos desarrolladores siguen empleando *px* es que *rem* aún presenta incidencias de error cuando se utilizan navegadores antiguos para visualizar la página. La unidad *rem* no es compatible con *Internet Explorer 8* [75].

⁸⁷ Un preprocesador CSS es una herramienta que nos permite escribir pseudocódigo CSS que será compilado a CSS nativo o puro. La ventaja principal del uso de preprocesadores se centra en que nos aporta una inmensa flexibilidad para generar código CSS ya que es posible manipularlo como si se tratase de un lenguaje de programación común (usando variables, funciones, bucles, condiciones, etc.) [73]

⁸⁸ La única diferencia de *rem* respecto a *em*, es que *rem* utiliza como referencia al elemento raíz del documento; es decir el elemento (etiqueta) *html* [55].

Al finalizar este análisis de *frameworks* se llegó a la conclusión de que *Foundation* es el más fuerte competidor de *Bootstrap*; sin embargo, por razones de compatibilidad (RNF20) y dado que *Bootstrap* es considerado un *framework* más ad hoc para el proceso de creación rápida de prototipos respecto a *Foundation* [76], *Bootstrap* queda elegida como la herramienta principal de la solución propuesta.

5.2 Metodología ágil RAD

Uno de los acuerdos principales y determinantes para la elección del tipo de metodología de desarrollo sobre la cual este proyecto se tiene su base: *Rapid Application Development (RAD)* [77], es la forma en la que se establecería una comunicación diaria con el usuario para recibir constante retroalimentación en cuanto a mejoras y adiciones a los sub-prototipos (módulos) que se entregaron durante el proceso. El desarrollo de la maquetación fue modular y el código empleado para la maquetación es consistente gracias al uso de un marco de trabajo definido previamente.

La figura 5.1 ilustra el proceso general de la metodología RAD

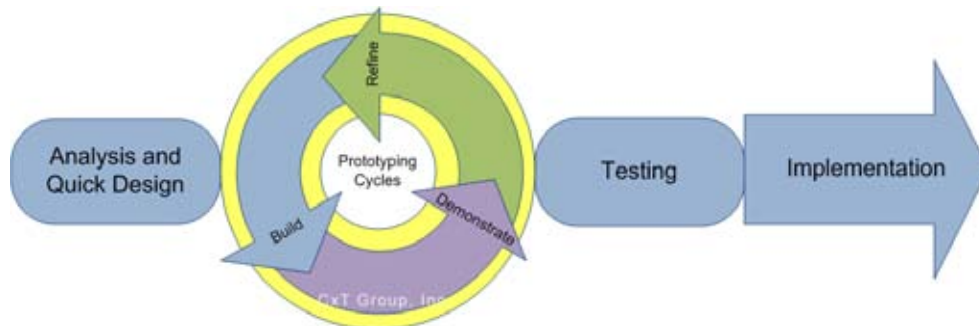


Figura 5.1 La metodología RAD emplea 4 procedimientos o etapas fundamentales: Análisis de requerimientos y diseño de la solución, Ciclo de construcción – retroalimentación, pruebas e implementación [78].

Las principales ventajas detrás de la metodología ágil RAD son [77]:

- Reducción en los tiempos de desarrollo (Proceso ágil para la creación de los prototipos).
- Revisiones surgen en las etapas tempranas del desarrollo.

- Se obtiene retroalimentación constante y sumamente valiosa por parte del cliente.
- Las integraciones desde un inicio resuelven muchos problemas que la integración final puede representar.

El primer procedimiento consta del análisis de los requisitos del cliente, así como el plasmar las especificaciones respecto a la distribución de los componentes gráficos de cada página (Capítulo 4.2). En el siguiente capítulo se abordará la etapa siguiente al proceso de análisis y diseño: Creación de la maqueta web.

6. Implementación de la solución

6.1 Creación del prototipo de la vista frontal del sitio mediante Twitter Bootstrap 3

A partir de este capítulo, se da inicio a la etapa de creación de sub-prototipos (módulos) que posteriormente fueron integrados para dar lugar a una maqueta de sitio web. Este procedimiento expuesto representa la segunda etapa del ciclo de desarrollo *RAD*, donde se construyeron los módulos, se mostraron al cliente y se recabó retroalimentación sobre estos. La constante comunicación con el cliente fue un factor crucial para lograr lo anteriormente descrito.

Las herramientas de software empleadas para el desarrollo de los sub-prototipos del proyecto de esta tesis fueron los siguientes:

- Para la maquetación del sitio:
 - *Eclipse (Kepler Service Release 2*, disponible en <https://www.eclipse.org>).
 - *Emmet v1.1* (*plugin* o extensión para facilitar la escritura de código *HTML*, disponible en <http://emmet.io>).
 - *Framework Twitter Bootstrap* versión 3.2.0 (disponible en <http://getbootstrap.com>).
 - Librería de iconos *SVG Font Awesome* v4.2 (disponible en <http://fontawesome.github.io/Font-Awesome/>).
 - Creador de esquemas de colores *Adobe Kuler* (herramienta en línea, disponible en <https://color.adobe.com>).

Preparativos

El proyecto quedó estructurado de la manera: La carpeta que representa el Proyecto completo recibe el nombre de “Alindis-Maqueta”, siendo *Alindis*, el nombre del negocio de Alarmas y otros sistemas de seguridad para el hogar. Los cuatro archivos que representan la estructura del sitio son aquellos con la extensión *html*. Para mayor orden se creó una carpeta llamada *assests*, que por cuestiones de practicidad, se nombró de esa forma ya que será la carpeta que aloja todos los “bienes” o utilidades para el enriquecimiento del sitio: Archivos o ficheros de estilo CSS, archivos SVG, imágenes y archivos Javascript (*.js*). Los archivos CSS ya compilados de *Bootstrap* se encuentran en las carpetas *css*, *fonts* y *js*, y son: *bootstrap.min.css* y *bootstrap-theme.min.css* (se antepone *min* a la extensión como parte del nombre para identificar al archivo como una versión minimizada del archivo fuente). Al mismo nivel, se cuenta con un archivo CSS con estilos propios para mayor personalización del sitio. Lo anterior se muestra en las figura 6.1 y 6.2.

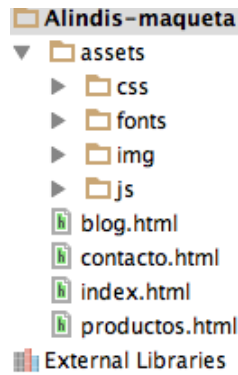


Figura 6.1 Estructura general del proyecto de maquetación. La carpeta *img* fue creada con propósitos de mantener un orden y almacenar las imágenes empleadas en el sitio.



Figura 6.2 Los archivos fuente del código de *Bootstrap 3*, escritos en *LESS*. Las plantillas *js* se encuentran en la respectiva carpeta “*js*” y todo lo archivos necesarios para la representación tipográfica en la carpeta *fonts* del proyecto.

A continuación, se ilustra en la figura 6.3 la plantilla de código base *HTML5* para la construcción de cualquier maqueta basada en *Bootstrap 3*.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head
    content must come *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
    queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js">
    </script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as
    needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>

```

Figura 6.3 Plantilla base que incluye todas las referencias a los archivos y clases que *Bootstrap* utiliza para su correcto funcionamiento.

Dentro de *assets/css* se encuentran los archivos mínimos requeridos para utilizar el framework de *Twitter*: *bootstrap.min.css*, *bootstrap-theme.min.css*; y un archivo llamado *sitio.css* donde se establecen las reglas de estilo para mayor personalización del sitio de acuerdo con los requerimientos del cliente.

Las plantillas *js* que utiliza el sitio son ilustradas por la figura 6.4.

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script src="assets/js/jquery-2.1.1.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
```

Figura 6.4. Al final de cada documento *html* que conforma el sitio, se incluyen las tres líneas de código que se muestran.

En el caso de las plantillas *js* de la figura 6.4, solo la primera corresponde a un *CDN*⁸⁹, las demás se guardan de manera local en la carpeta *assets* del proyecto.

Las librerías tipográficas *SVG Font Awesome* fueron un agregado opcional para cubrir el requisito referente a información de contacto en redes sociales de la empresa Alindis. Las siguientes líneas de código dentro de las etiquetas *head* del documento muestran la referencia a dichas librerías/clases:

```
<link href="assets/css/font-awesome/css/font-awesome.min.css"
rel="stylesheet">
<link href="assets/css/font-awesome/css/font-awesome-ie7.min.css"
rel="stylesheet">
```

La fuente tipográfica elegida, por cuestiones de preferencia es *Raleway*, disponible en <http://www.google.com/fonts>. Para esto, es necesario agregar, igualmente dentro del *head* del documento la siguiente línea *HTML*:

```
<link
href='http://fonts.googleapis.com/css?family=Raleway:400,700'
rel='stylesheet' type='text/css'>
```

⁸⁹ *Content Delivery Network*, se trata de una red de entrega de contenidos, imprescindible para dar servicio a usuarios de todo el mundo, ya sea para ofrecer la descarga de archivos o para implementar sitios y aplicaciones web [26].

Creación de la maqueta:

Los diagramas base para el proceso de maquetación se encuentran definidos en el capítulo 4.2

Una vez listos los preparativos anteriores, se puede proceder a la creación de los módulos o sub-prototipos de la maqueta en sí.

De acuerdo con el RNF02, se dividió la maquetación en cuatro páginas: *index.html*, *productos.html*, *contacto.html* y *blog.html*. Los servidores web que alojarán el sitio (aspecto excluido de esta tesis) utilizan la página cuyo nombre es *index* (*index.html*, *index.php*, etc.) para desplegarla como página inicial o por defecto cuando un usuario escribe la *URL* asociada a la tienda en línea.

La página interna del sitio es aquella que presentará los bloques que se mantendrán fijos (siempre visibles) en cualquiera de las páginas que se visite. Se presenta a continuación el código *HTML* correspondiente a las tres secciones que conforman la página interna: Encabezado, Sección de Redes Sociales e información de contacto, y el pie de página.

Las etiquetas semánticas *HTML5* serán bastante propicias dada la estructura inicial del sitio. *Bootstrap* cuenta con la clase *navbar*, que facilita la creación de menús de navegación y que, como todos los componentes de *Bootstrap* cuentan con capacidades responsivas. Las capacidades responsivas del componente *navbar* se refiere a la acción de expandir y contraer los elementos del menú (*menu items*).; esta funcionalidad requiere *Javascript*. Como ya se mencionó *Bootstrap* utiliza *LESS* y *SASS* en su código fuente; de manera que es posible personalizar

las propiedades de sus componentes. Un ejemplo claro para esto sería la variable que determina el breakpoint o punto de quiebre a partir del cual el componente navbar se contrae o se expande en su modo horizontal (no para móviles): *@grid-float-breakpoint*. Si así se desea, se puede modificar el valor de dicha variable al modo más conveniente. Así aplica para cualquier otra propiedad de los componentes. Lo anterior es uno de los factores que hace de *Bootstrap* un framework altamente *flexible*. Otro aspecto que destaca es la especificación de un rol por medio del atributo *role* facilitando la identificación de dicha región por tecnologías de apoyo (*assistive technologies*, ver capítulo). En el caso de los navbars el valor “navigation” en el atributo “role” se incluye para la finalidad descrita anteriormente.

Para la sección de encabezado (RNF03) se emplean las siguientes clases de *Bootstrap*:

- *navbar-wrapper*: Indica que el elemento contenedor tendrá que alojar un componente tipo *navbar*.
- *navbar*: Para la creación de la barra de menú en el encabezado
- *navbar-inverse*: Para invertir el color a negro de la barra completa del encabezado.
- *navbar-fixed-top*: Permite que la barra se mantenga siempre visible aunque la página sea recorrida hacia abajo.
- *navbar-brand*: Para la colocación rápida de un logotipo en el contexto de la barra de menú de navegación.
- *Navbar-right*: Para la colocación del menú del lado derecho en la pantalla

La figura 6.5 muestra el resultado de codificar y estilar la barra de menú mediante las clases de *Bootstrap* especificadas anteriormente.



Figura 6.5. La barra de menú con dos componentes específicos: el logotipo del lado izquierdo, y el menú *per se* con los cuatro enlaces para cada una de las páginas del sitio (RNF02).

Se agregó posteriormente la funcionalidad conocida como *collapse* que se consigue con la clase *navbar-collapse* y que permite, mediante *jQuery* esconder el menú de navegación de la barra superior para cuando el *viewport* sea de anchura reducida, como en el caso de la pantalla de un teléfono móvil. Lo anterior se agregó como característica de incremento y mejora la usabilidad de la barra de navegación superior.

En la sección de redes sociales y contacto (RNF04) se utilizan las clases *container* y *row*. Como ya se mencionó en el capítulo 5.1, a partir de la versión 3 de *Bootstrap*, el enfoque del *framework* pasó a ser 100% hacia dispositivos móviles. Las rejillas permiten crear un esquema de distribución de elementos (*layout*) a través de una serie de filas y columnas que encapsulan el contenido principal de la página. Las filas deberán colocarse en contenedores (elementos *div* con clase *container* o *container-fluid*). Cada fila alojará un grupo de columnas horizontales. Se pueden utilizar mixins (*LESS*) para la creación de layouts más semánticos. El sistema de rejillas de basa en múltiplos de 12, como lo ilustra la figura 6.6.



Figura 6.6 Se ilustra el sistema de rejillas en *Bootstrap 3*. Pensado primeramente para dispositivos móviles. Está basado en múltiplos de 12 para facilitar la esquematización y colocación de elementos en las páginas [62].

Las clases para la gestión de columnas tienen la forma genérica *col-xx-**. *xx* indica el punto de quiebre o *breakpoint* según el tamaño del dispositivo, mientras que el asterisco indica el valor del ancho (1 - 12).⁹⁰

- *col-xs-**: viewport < 768px de anchura
- *col-sm-**: viewport >= 768px de anchura
- *col-md-**: viewport >= 992px de anchura
- *col-lg-**: viewport >= 1200px de anchura

La sección de redes sociales fue construida utilizando este sistema de rejillas.

Consta de dos bloques como lo muestra la figura 6.7.

⁹⁰ La abreviatura *px* se refiere a *pixel* (del término *picture element*). Unidad de medida CSS que corresponde a los puntos de resolución en un monitor gráfico.



Figura 6.7 El bloque de información de contacto (lado derecho de la figura) y el bloque de redes sociales (lado izquierdo de la figura).

Se puede observar además, los pequeños íconos del lado izquierdo de la información tanto de teléfono como de correo electrónico. Estos pequeños componentes son parte de una librería llamada *GLYPHICONS* que consiste de un conjunto de pequeños íconos y símbolos monocromáticos creados con énfasis en la simplicidad y facilitar orientación en las páginas. Para agregar alguno de los *glyphicons* disponibles en <http://getbootstrap.com/components/> solo basta con crear un elemento *span* y asociarlo con la clase correspondiente al *glyphicon*. Por ejemplo, para generar el ícono correspondiente a una bocina tradicional de teléfono que se resalta en la figura 6.7 anterior solo se necesita agregar la siguiente línea de código:

```
<span class="glyphicon glyphicon-earphone">
```

Cada uno de los bloques tiene la clase *col-md-6*; lo cual indica que el *breakpoint* es para *viewports* menores al tamaño mediano (< 992px). A partir de que el *viewport* sea menor a 992px de ancho, los elementos se acomodarán en forma apilada. Mientras lo anterior no ocurra, cada uno de los bloques ocupará un ancho de 6 columnas, dando un total de 12 columnas.

Pasando ahora a la última sección de la página interna, se tiene el pie de página (comúnmente conocido como *footer*). De igual manera se utiliza el sistema de rejillas flexibles. Se colocan tres bloques flexibles en un contenedor (*container*). Los anchos son de 3, 6 y 3 columnas respectivamente. Dichos bloques tienen como *breakpoint, sm*; es decir que en *viewports* menores a 768px de ancho los elementos quedarán apilados. El bloque de texto que indica las iniciales de los creadores de la maqueta se coloca con la clase *pull-right* que permite “empujar” un bloque con el prefijo de clase *col-xx-** hacia la derecha. La figura 6.8 ilustra el *footer* después de la codificación y estilado.



Figura 6.8 El logotipo en miniatura de la empresa se coloca con la clase *col-sm-3*; *col-sm-6* para el caso de el menú de navegación secundario, y *col-sm-3* (total 12) para el bloque de texto de “Crédito” de los creadores.

Finalmente, la página interna generada tendrá un aspecto como el que ilustra la figura



Figura 6.9 De esta manera queda inicialmente, tras la codificación y estilo, la página interna del sitio.

A partir de este punto se comenzará con las siguientes páginas: *index.html* (página de inicio), *blog.html*, *productos.html* y *contacto.html*.

index.html

La página principal (RNF06) resulta ser la que más componentes *Bootstrap* utiliza. Comenzando con la sección de bienvenida, es posible observar las partes que la conforman según el boceto aprobado por el cliente (capítulo 4.2).

Según el RNF07, se procede con un logotipo grande del lado izquierdo, un bloque de texto de bienvenida del lado derecho en la parte superior y en la parte inferior del mismo lado, tres imágenes de productos con etiquetas de precios y un botón debajo de éstas.

Bootstrap cuenta con clases predefinidas para los botones. dichas clases pueden ser utilizadas en cualquiera de las etiquetas: ancla (*a*), botón (*button*) y entrada de datos (*input*):

```
<a class="btn btn-default" href="#" role="button">Link</a>
<button class="btn btn-default" type="submit">Button</button>
<input class="btn btn-default" type="button" value="Input">
```

Adicionalmente, se pueden emplear clases adicionales para mayor estilo en los botones. Las clases *btn-default*, *btn-primary*, *btn-success*, *btn-info*, *btn-warning*, *btn-danger*, *btn-link* se utilizan para este fin, haciendo más intuitiva la inclusión de botones en áreas de las páginas. La figura 6.10 muestra lo anterior



Figura 6.10 Las clases de estilos para botones en *Bootstrap 3* permiten que se tenga un aspecto visual como el que se ilustra.

Las clases específicas para establecer tamaño en los botones son: *btn-lg*, *btn-sm* y *btn-xs*. Si no se especifica ninguna de las anteriores clases se colocará un botón de tamaño normal por defecto. La clase *btn-block* establece un botón que ocupa la totalidad del ancho de su elemento padre (el elemento contenedor de se agregó el botón).

Para la colocación de los elementos como se especifica en el boceto estructural, se utiliza nuevamente un contenedor (*container*) y dos filas (*row*). La primera fila consta de dos bloques; el primero de 5 columnas de ancho con

breakpoint a partir de un *viewport* menor a los 1200px de ancho (*col-lg-5*) y el otro con el mismo *breakpoint* pero de 7 columnas (*col-lg-7*). dentro de este último se colocará otra fila (*row*) en la que se acomodan los tres sub-bloques que contendrán las imágenes de los productos destacados, dichos sub-bloques con la clase *col-mod-4*; Lo cual indica que las imágenes se distribuirán verticalmente (una encima de otra) cuando el *viewport* se encuentre debajo de los 992px de ancho. Se describe gráficamente en la figura 6.11 cómo quedaría lo anterior tras la codificación y estilado.

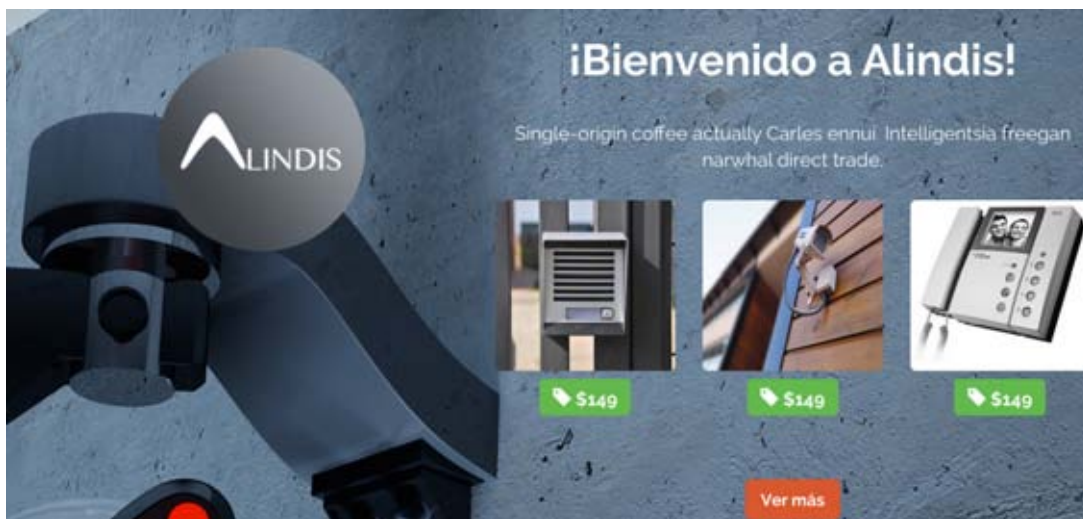


Figura 6.11 El logotipo de la empresa se visualiza del lado izquierdo, el texto de bienvenida y las 3 imágenes de productos destacados con sus respectivos precios se muestran en un bloque del derecho, así como el botón para “ver más” productos.

Respecto a las imágenes, se dispone de clases para mostrarlas de tres formas posibles: marco redondeado de las esquinas, de forma cuadrada, y de forma circular: *img-rounded*, *img-circle*, *img-thumbnail*. Se utiliza la clase *img-rounded* para las imágenes de los productos destacados.

La sección de suscripción contiene un botón a una ventana modal. *Bootstrap* provee de clases para la utilización y efecto de las ventanas modales. El botón de suscripción invoca a una ventana modal. Las clases creadas para conseguir el comportamiento apropiado de una ventana de este tipo son *modal*, *modal-dialog*, *modal-content*, *modal-header* y *modal-body*; cada una de ellas colocadas de manera jerárquica, como se indica en el listado. *Bootstrap* se encarga de asignar mediante la inclusión de estas clases el comportamiento acorde con un componente de este tipo. Para el botón, se emplean las clases *btn*, *btn-success*, y *btn-lg btn-block*. La figura 6.12 muestra gráficamente cómo se ve tras la codificación y estilado esta sección.

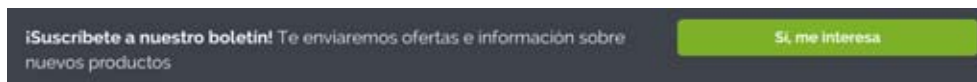


Figura 6.12. Se muestra texto informativo del lado izquierdo y el botón a la acción de suscripción.

En las secciones subsecuentes de la página principal, se continua empleando el sistema de rejillas tal y como se describió anteriormente pero usando distintos *breakpoints* y números de columnas. A partir de este momento cada vez que se requiera hacer referencia al área de despliegue o *viewport* de un dispositivo se utilizará la abreviatura *VP*, para fines prácticos.

La sección de información de la empresa es ilustrada en la figura 6.13.



Figura 6.13. Se utiliza texto simulado para rellenar los bloques de texto que se solicitaron. Un texto de encabezado, y dos párrafos correspondientes a la visión y misión de la empresa.

Los bloques de texto de misión y visión de la empresa fueron colocados con las clases `col-sm-6` para conseguir una proporción equitativa de ambas columnas a lo ancho de la `viewport`.

La sección correspondiente al video informativo o introductorio se compone de un marco (`iframe`) encerrado en un contenedor con clase `col-md-8`. Se ilustra en la figura 6.14 la forma en la que se visualiza dicha sección.



Figura 6.14 Un pequeño texto informativo seguido de un marco que envuelve a un video introductorio o informativo.

La sección clasificación de productos fue construida mediante un contenedor, la clase *row* y 4 imágenes con sus respectivas clases de *breakpoint*: *col-md-3* y *col-sm-6* (RNF11). Se ilustran gráficamente lo anterior en la figura 6.15



Figura 6.15 Cada una de las imágenes para hacer referencia a las líneas de productos que la empresa maneja.

Para la sección de clasificación de clientes, según el RNF12 se utilizan tres bloques con clase *col-sm-4* para distribuir las tres imágenes de forma proporcional y completar las 12 columnas. Se ilustra la forma en la que quedó maquetada esta sección en la figura 6.16.



Figura 6.16 Cada una de las imágenes para hacer referencia al tipo de clientes a los que la empresa tiene por objetivo, hogar, establecimientos y empresas.

La retroalimentación del cliente permitió refinar la sección de clasificación de productos añadiendo una imagen de encabezado con un texto descriptivo para dar un sentido más intuitivo y profundo a dicha sección. Dicha imagen fue colocada dentro del mismo contenedor y fuera de la sección *row*. No hubo mayor complicación para lograr esto.

Blog.html

Todo el contenido de la página *Blog* (RNF13 y RNF14) se encapsula dentro de un contenedor y se comienza a establecer las medidas de cada bloque. En su modo de escritorio, se establece un *layout* de dos columnas; del lado izquierdo se cuenta con el listado de publicaciones del *blog* (*posts*) que conforman el contenido primario; y del lado derecho (*aside*) bloques que representan el contenido secundario de esta página. La cantidad de bloques correspondientes a las publicaciones es fija dado que la maqueta representa el sitio en modo estático⁹¹ (lo mismo aplica para la página de productos).

Una vez establecido lo anterior ya solo queda plasmarlo en código *HTML* con ayuda de las clases de *Bootstrap*. Se hace uso de la etiqueta *main* para establecer junto con el atributo *role* con valor “*main*”, donde se encuentra el contenido principal o primario en esta página para que quede bien diferenciado del secundario. Acto seguido, se emplea la etiqueta *aside* para marcar el área correspondiente a los sub-bloques que conforman el contenido secundario de la página de *Blog*.

El contenido primario consiste de un bloque de 8 columnas de ancho con *breakpoint sm* (*col-sm-8*); mientras que el bloque que engloba el contenido secundario ocupa 4 columnas de ancho y el mismo *breakpoint* (*col-sm-4*). Véase la figura 6.17, la cual ilustra la maquetación de lo anteriormente descrito.

⁹¹ Toda la funcionalidad *backend* (servidor web, lógica de negocio mediante el modelado de datos y la comunicación con la base de datos) hará dinámico el sitio, sin embargo todo lo referente a dicha funcionalidad queda fuera del alcance de la tesis.



Figura 6.17 La sección de encabezado que indica el nombre o contexto para la página y todo el contenido primario y secundario del *blog* en forma de bloques: Bloque para suscripción al blog, bloque para búsqueda de contenido en el blog, y los bloques para las publicaciones recientes y categorías de temas. Se muestra la insignia de comentarios en la parte superior derecha de cada bloque correspondiente a una publicación o entrada del blog.

Cuando el sub-prototipo de la página *Blog* se mostró al cliente, no se contaba aún con una insignia que mostrara el número de comentarios por publicación. Esta solicitud fue casi inmediata tras la demostración al cliente. Para atacar el requisito entrante se agregó la un *div* estilado y se emplearon las clases *fa* y *fa-comments* de la librería *Font Awesome* para mostrar gráficamente los globos de conversación que dan un sentido más semántica a la insignia.

Productos.html

Los productos se representan en pequeños bloques (sección de mosaico de productos disponibles, de acuerdo con el RNF15 y RNF16); cada uno de los cuales tendrá asociada las clases *col-lg-3*, *col-md-4* y *col-sm-6*. Indicando una cadena de acciones responsivas: Cuando el *viewport* se mantenga 1200px o mayor (*lg*), el sub-bloque de producto ocupará tres columnas; de lo contrario, si se encuentra entre 992 y 1200px (*md*) los sub-bloques abarcarán 4 columnas, para que de este modo solo quepan tres en cada fila ($4 + 4 + 4 = 12$). Y finalmente, cabrán solo dos sub-bloques de producto en caso de que el *viewport* caiga dentro del rango 768 a 992px ya que ocuparán 6 columnas ($6 + 6 = 12$). La figura 6.18 ilustra la implementación de lo anterior.

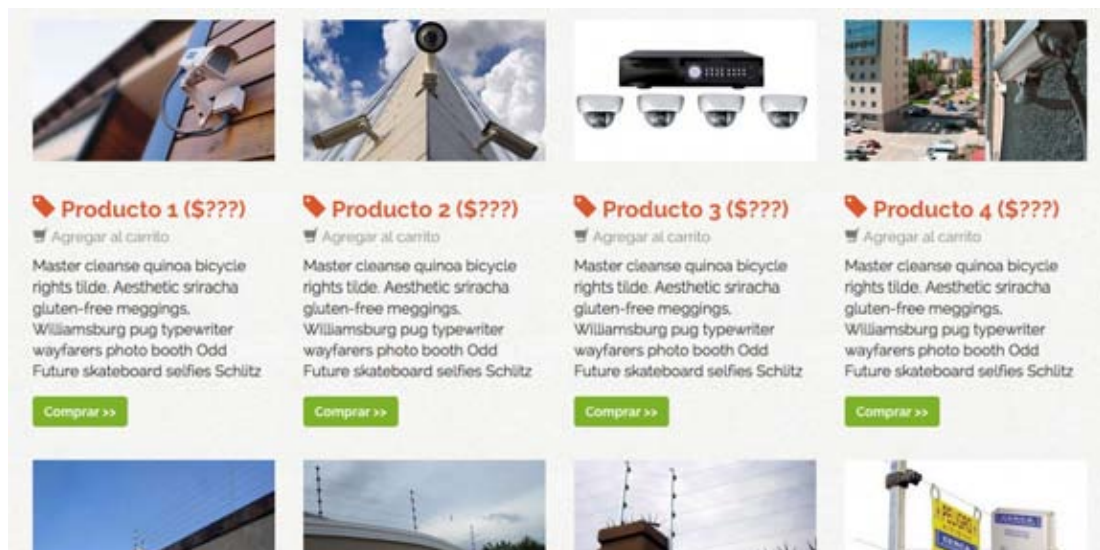


Figura 6.18 Por medio del sistema de rejillas flexibles en Bootstrap 3 fue muy sencillo el acomodo de los bloques de cada producto a modo de mosaico.

La sección de mosaico de productos se presenta inicialmente (si el viewport es suficientemente ancho) con tres filas de cuatro bloques cada una para cubrir el requisito de 12 productos mostrados.

Tras la muestra del sub-prototipo para la página de Productos, el cliente solicitó un componente que es casi imprescindible cuando se trata de presentar un conjunto de elementos cuya cantidad es indefinida, esto es, un paginador. Un componente muy importante para su inclusión en la página de productos es sin duda, un paginador. Este componente permite agrupar un conjunto de elementos en un determinado número de elementos anglobados en un contenedor. *Bootstrap*, aporta las clases necesarias para incrustar un elemento paginador. La clase *pagination* sirve para este propósito. Únicamente basta con agregar un elemento *nav* para dar un sentido semántico, ya que un paginador es al final de cuentas un elemento de navegación “multipágina”. Dentro de dicho elemento *nav* se introduce una tabla *ul* con sus respectivos *items (li)*. Gracias a las clases auxiliares *active* y *disabled* se puede establecer el estado de un enlace a un número de página: activo, en caso de ser el actual; o deshabilitado, en caso de ya no ser “clickeable”.



Figura 6.18. El paginador, que fue un requisito entrante del cliente y representa un incremento al sub-prototipo de mosaico de productos.

La ventana modal para comprar un artículo inicialmente constaba sólo de un formulario para el ingreso de datos de tarjeta bancaria. Después de observar el sub-prototipo de la ventana modal (RNF17) el cliente solicitó agregar un apartado para una nueva modalidad de pago mediante *Paypal*. Para conseguir la característica incremental anterior se emplea la clase *tabbable*, *nav-tabs*; así como *tab-content* y *tab-pane* para el contenido de la pestaña. El resultado es realmente excepcional. Sin mucho esfuerzo, se puede contar con diseño en pestañas para las modalidades de pago, como se muestra en la figura 6.19.



Figura 6.19. Ventana Modal con agrupación de modalidades de pago en pestañas.

Para la primer pestaña se utilizan las clases relacionadas a formularios. Los formularios son un conjunto de campos que se acomodan basándose en cierta lógica de contenido con el fin de recabar información específica del usuario, siendo este el que aporta dichos datos mediante el llenado del mismo.

Bootstrap también aporta un conjunto de clases predefinidas que facilitan en gran manera la construcción y colocación de formularios. En la pestaña correspondiente al pago con tarjeta de crédito, se coloca un formulario para solicitar la información de la tarjeta de crédito del usuario. El contenido correspondiente a la primer pestaña se coloca dentro de un elemento *div* asociado

con la clase *tab-pane*, el cual a su vez se encuentra dentro de un *div* con clase *tab-pane* y id *tarjeta*. El identificador sirve en este caso para que los elementos de lista (*ul*) que conforman las pestañas pueden acceder al contenido correspondiente según el id de cada panel de pestaña (*tab-pane*). Es realmente sencillo seguir la lógica que marca Bootstrap con los nombres de las clases predefinidas que aporta. De la misma manera como se describió para las ventanas modales, las clases que nos permiten construir contenido dividido en pestaña tienen un orden jerárquico (una dentro de la otra) que es fácil de comprender por la claridad en la nomenclatura de las mismas.

Se utiliza la clase *form-horizontal*, la cual permite alinear de forma lógica una etiqueta y su campo correspondiente en cada fila. Cada grupo etiqueta-campo se divide de forma lógica mediante un elemento *div* con clase *form-group* y se utilizan las clases *col-xx-** para establecer el esquema responsivo para cada grupo tal y como se hace en cualquier otro bloque o sub-bloque responsivo. Al final se obtiene un formulario tal y como lo ilustra la figura 6.20.

Método de pago

Tarjeta de Crédito **Paypal**

Pague con su tarjeta de crédito o débito **contra entrega**. El pago se realiza al personal encargado de llevar su compra a domicilio utilizando una de nuestras terminales móviles.

VISA

Cantidad:

Nombre(s):

Apellidos:

Dirección:

Delegación:

Código Postal:

E-mail:

Teléfono:

Recordar mi información

Figura 6.20 Formulario construido de manera sencilla, ordenada y muy práctica con *Bootstrap 3*. Este fue inicialmente el requerimiento del cliente excluyendo la agrupación en pestañas.

Contacto.html

La última página es la de contacto y acorde al RNF19 consta de un formulario. En este caso utilizamos un formulario sin las clases correspondientes de *Bootstrap* (como en el caso del formulario de la ventana modal de pago); en vez de ello solo se hace uso de las clases para esquema responsivo. Los campos *Nombre* e *E-mail* ocupan 6 columnas de ancho y el campo para el mensaje ocupa todo el ancho del contenedor (12 columnas). Véase el resultado de la codificación y estilo en la figura 6.21

¿Alguna pregunta? 

Puedes contactarnos por aquí. Haremos lo posible por responder en el menor tiempo.

Tu nombre

Tu e-mail

Tu mensaje

Enviar 

Figura 6.21 Tres campos de entrada de texto (nombre, correo electrónico y comentario, queja o sugerencia), así como el correspondiente botón para el envío de la información ingresada.

El símbolo que se muestra en la figura 6.21 después del texto “¿Alguna pregunta?” es otro *glyphicon* y fue agregado como requisito adicional del cliente tras la muestra del sub-prototipo correspondiente a la página de contacto. Se agregó de manera sencilla mediante la sentencia:

```
<span class="glyphicon glyphicon-comment"></span>
```

Hasta este punto, tomando en cuenta el ciclo de construcción, muestra y refinación de los sub-prototipos (mediante la retroalimentación del cliente) de la metodología *RAD*, todos los requisitos referentes a la interfaz gráfica que se construyeron y mostraron en este capítulo resultaron satisfactorios para el cliente. Lo que queda aún es la parte en la que integran los cuatro sub-prototipos (las cuatro páginas) en una sola maqueta y se realizan las pruebas de compatibilidad y usabilidad para la aprobación final del cliente.

6.2 Integración y pruebas

En lo que respecta a la integración de los sub-prototipos (páginas) que conforman el sitio, se actualizan cada uno de los enlaces contenidos en los menús del sitio (principal, en la barra de encabezado y secundario en el pie de página). De igual manera, el enlace asociado al logotipo incrustado en la barra superior de navegación (encabezado de las páginas) mediante *navbar-brand*, así como el logotipo en miniatura de la empresa en el pie de página se actualizan para referenciar a *index.html*. Los botones cuyo propósito sea el de mostrar las ventanas modales correspondientes se actualizan mediante la inclusión de los atributos *data-toggle="modal"* y *data-target="#modal-subscripcion"* dentro de los elementos *button* correspondientes, así como del enlace “suscríbete” que se incluye en el pie de página. El botón “ver más” en la sección de bienvenida también es actualizado con una referencia (*href*) hacia la página de productos.

Con relación a la etapa de pruebas que se llevó a cabo, se establecieron pruebas en dos categorías: compatibilidad y usabilidad. Las pruebas de compatibilidad se realizaron en los siguientes navegadores:

- *Mozilla Firefox v38.0 (Windows 7)*
- *Google Chrome v43.0 (Windows 7)*
- *Safari v8.0.5 (Mac OS X 10.10.3)*
- *Internet Explorer 8 (Windows 7)*
- *Navegador Safari para iPhone 5C (iOS 8)*
- *Navegador Safari para iPad Air (iOS 8)*
- *Navegador Safari para iPhone 6 plus (iOS 8)*

En cuanto a las pruebas de usabilidad, en la mayor parte de los casos se probaron las características responsivas de la maquetación desarrollada.

Para propósitos de lo anteriormente descrito, se establecieron los casos de prueba correspondientes. Se emplea el acrónimo CXX, donde XX representa el número específico del caso de prueba.

***** **Compatibilidad** *****

C01: Comprobación de la compatibilidad de los navegadores

Prueba: Comprobar la compatibilidad con *Internet Explorer 8, Google Chrome v43.0, Mozilla Firefox v38.0, Safari v8.0.5, iPhone 5C, iPhone 6 plus e iPad Air*

Resultado esperado: Se visualizan todos los elementos de manera correcta en todos los navegadores.

***** **Usabilidad** *****

C02: Comportamiento de la barra superior de navegación

Prueba: Probar que la barra de navegación del encabezado se mantenga fija y visible y el efecto *collapse* agregado a la misma.

Resultado esperado: La barra superior de navegación se mantiene fija aún cuando se recorre el contenido de la página hacia abajo y se esconde cuando el tamaño de *viewport* es reducido a tamaño estándar de pantalla de teléfono móvil y presenta el efecto *collapse*.

C03: Comportamiento de las ventanas modales

Prueba: Comprobar el comportamiento de las ventanas modales cuando se presiona sobre los enlaces que las invocan.

Resultado esperado: Las ventanas modales se visualizan de marea correcta y siempre aparecen de acuerdo con el enlace o botón utilizado para su invocación.

C04: Comprobación del iframe

Prueba: Comprobar el funcionamiento del *iframe* en la página de inicio, sección de video introductorio.

Resultado esperado: El *iframe* muestra el video (de prueba) y es posible reproducirlo.

C05: Visualización de los íconos Font Awesome

Prueba: Comprobar la forma en la que se visualizan los íconos de la librería *Font Awesome*.

Resultado esperado: Los íconos *Font Awesome* conservan su forma circular.

C06: Comprobación de la capacidad responsiva del sitio

Prueba 1: Comprobar que los componentes de la página de inicio tengan una capacidad responsiva al momento de reducir el tamaño de la ventana del navegador.

Resultado esperado: La página de inicio re-acomoda los componentes que la conforman de una manera correcta y acorde al tamaño del *viewport*.

Prueba 2: Comprobar que los componentes de la página de *Blog* tengan una capacidad responsiva al momento de reducir el tamaño de la ventana del navegador.

Resultado esperado: La página de *Blog* re-acomoda los componentes que la conforman de una manera correcta y acorde al tamaño del *viewport*.

Prueba 3: Comprobar que los componentes de la página de productos tengan una capacidad responsiva al momento de reducir el tamaño de la ventana del navegador.

Resultado esperado: El mosaico de productos acomoda los bloques de producto en el siguiente esquema ordenado: 3 filas de 4 bloques, 4 filas de 3 bloques, en 6 filas de 2 bloques y 12 filas de 1 bloque según corresponda a la reducción progresiva del tamaño del *viewport* en el navegador.

Prueba 4: Comprobar que los componentes de la página de contacto tengan una capacidad responsiva al momento de reducir el tamaño de la ventana del navegador.

Resultado esperado: La página de contacto re-acomoda los componentes que la conforman de una manera correcta y acorde al tamaño del *viewport*.

6.4 Resultados de pruebas

Los resultados de pruebas se muestran en la siguiente tabla. La categoría de resultados se clasifica de la siguiente forma:

- Éxito: El resultado cumple con el requisito esperado o especificado. Se utilizará el símbolo ✓.
- Aceptable: Se muestra un comportamiento distinto pero que no altera la usabilidad del sitio. Se utilizará el símbolo **A**.
- Error: El resultado es incorrecto y se deberá corregir la incidencia, pues afecta la usabilidad del sitio. Se utilizará el símbolo **X**.

	IE8	Chrome v43.0	Firefox v38.0	iPhone 5C	iPhone 6 plus	iPad Air
Prueba para C01	A	✓	✓	✓	✓	✓
Prueba para C02	✓	✓	✓	✓	✓	✓
Prueba para C03	A	✓	✓	✓	✓	✓
Prueba C04	✓	✓	✓	✓	✓	✓
Prueba para C05	A	✓	✓	✓	✓	✓
Prueba 1 para C06	✓	✓	✓	✓	✓	✓
Prueba 2 para C06	✓	✓	✓	✓	✓	✓
Prueba 3 para C06	✓	✓	✓	✓	✓	✓
Prueba 4 para C06	✓	✓	✓	✓	✓	✓

Para los casos de prueba con categoría A, *Internet Explorer* (versión 8) presentó las siguientes incidencias:

- Para el C01, la barra de navegación superior se traslapa ligeramente con la sección de video introductorio de la página de inicio. Esto no presenta mayor problema en la usabilidad del sitio y será arreglado posteriormente.
- Para el C03, La ventana modal correspondiente a la interfaz de pago con tarjeta bancaria, no se muestra la imagen correspondiente a los bancos, la cual es un gráfico SVG. No causa mayor problema a la interfaz de las ventanas modales y se arreglará posteriormente.

- Para el C05, Los íconos de la librería *Font Awesome* no se presentan en forma circular sino cuadrada, pero conservan sus demás características de manera íntegra. Lo anterior no es un problema mayor, ya que no interfiere con la usabilidad del sitio. Se arreglará posteriormente.

A modo de breve demostración se realizaron algunas capturas de pantalla del sitio en distintos navegadores.

Para *Google Chrome* v43.0 se ilustra con la figura 6.22.

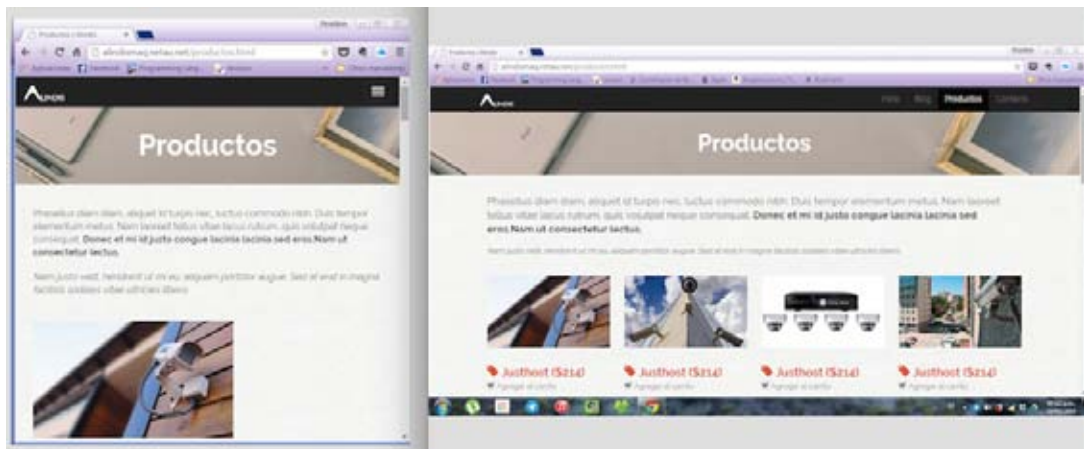


Figura 6.22 En el navegador *Google Chrome* v43.0 se muestra la captura correspondiente a la página de productos.

Para el navegador *Firefox* v38.0 la figura 6.23 se muestra con la correspondiente captura de pantalla.



Figura 6.23 En el navegador *Firefox v38.0* se muestra la captura correspondiente a la página de blog.

La figura 6.24 muestra la captura de pantalla en el navegador *Internet Explorer 8*.

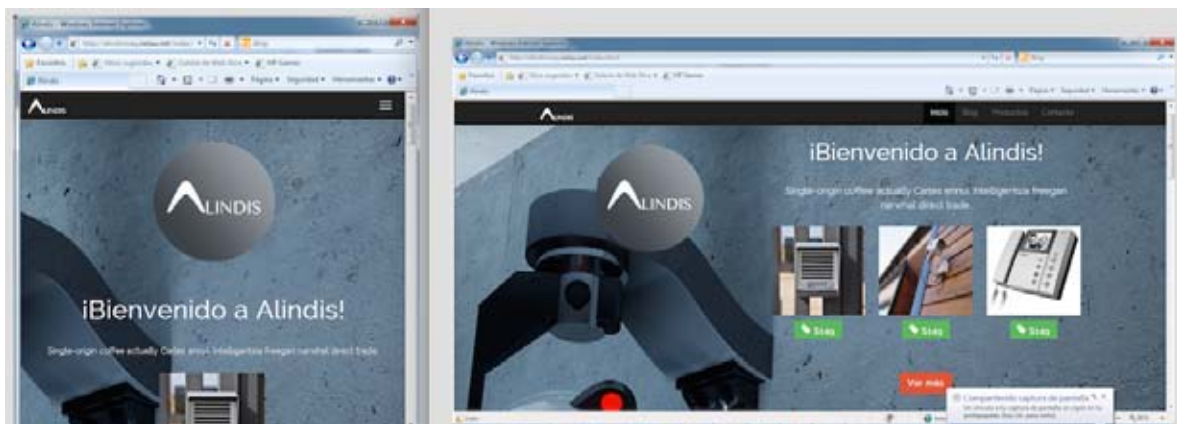


Figura 6.24 En el navegador *IE 8* se muestra la captura correspondiente a la página de inicio.

Ejemplos de cómo se visualiza el sitio desde un teléfono móvil se ilustran en la captura de la figura 6.25

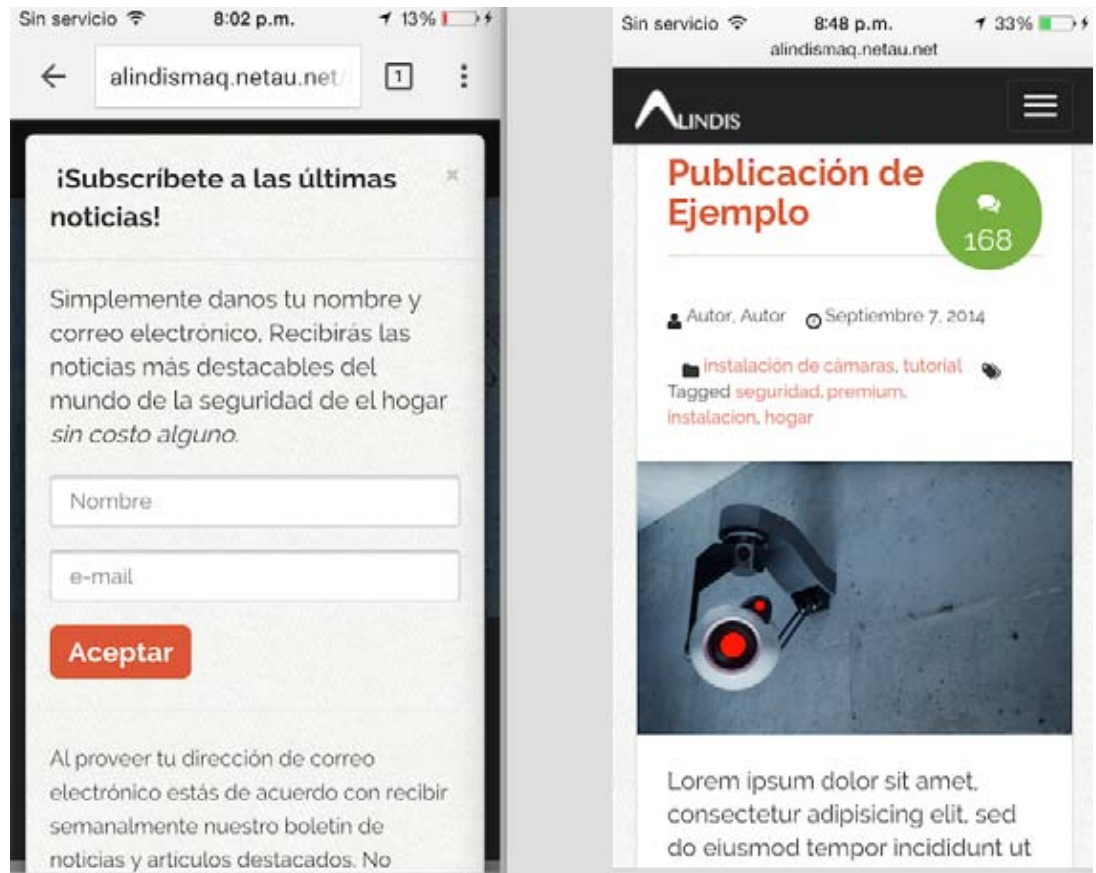


Figura 6.25 Del lado izquierdo se muestra como se ve la ventana modal de suscripción en *Chrome* versión móvil y del lado derecho la captura de la página de *Blog* desde un *iPhone 5C*.

Finalmente, en la figura 6.26 se logra apreciar cómo se visualiza la página de productos desde un dispositivo móvil tipo *Tablet*.

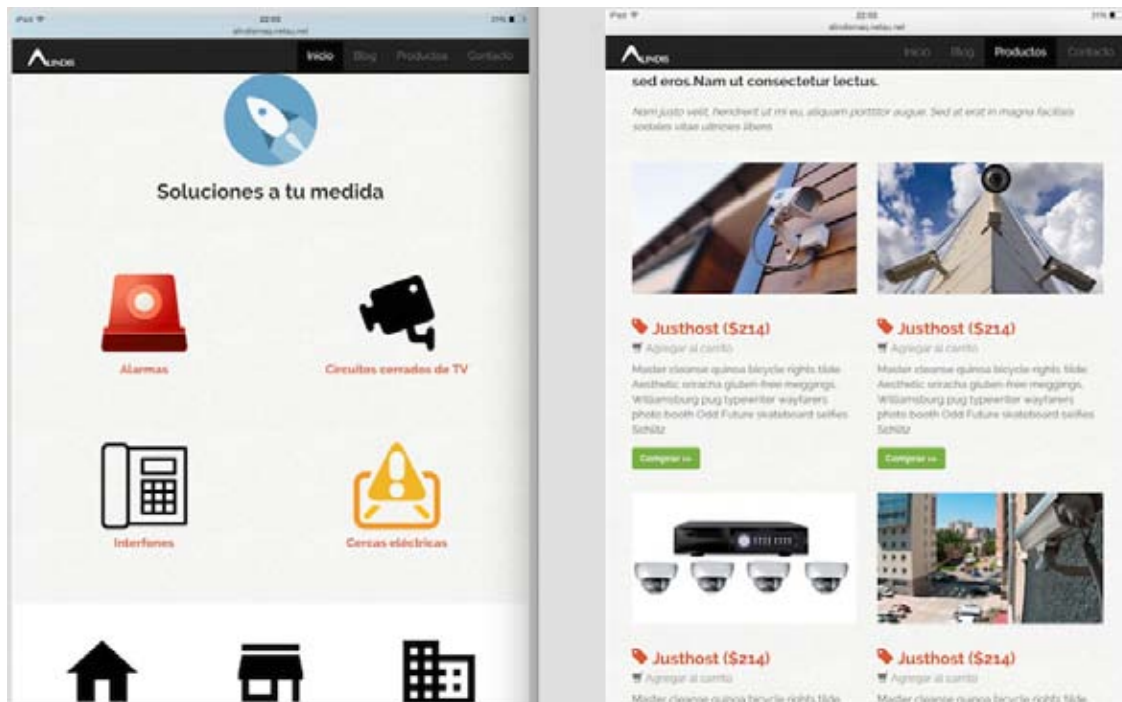


Figura 6.26. Se muestra la forma en la que se visualizan las páginas de inicio y productos desde la perspectiva de un iPad Air.

6.4 Los beneficios aportados al proceso de diseño del sitio (Creación del prototipo de vista frontal)

Bootstrap como marco de trabajo ha probado ser una herramienta muy poderosa con la que se obtienen diseños web responsivos de una manera relativamente sencilla. Sin necesidad de contar con mucha pericia en el área del diseño web, cualquier desarrollador *frontend* puede lograr maquetas muy agradables y con un alto grado de usabilidad.

Los diversos componentes que *Bootstrap 3* ofrece han facilitado el trabajo y a la vez ahorrado tiempo a la hora de escribir código. Se explicó brevemente en capítulos anteriores ejemplos de como utilizar *media Queries* para lograr aplicar estilos diferentes dependiendo del tamaño de pantalla; sin embargo crear plantillas CSS con estilos específicos para los cuatro principales tamaños de pantalla habría implicado demasiado tiempo y esfuerzo. Es aquí cuando un buen desarrollador de software debe tomar en cuenta que no es necesario “reinventar la rueda” para lograr objetivos específicos.

Hasta este punto, las páginas del sitio han quedado completas respecto a los requerimientos del cliente. Cabe señalar que el cliente en cuestión, para el momento en que se realizó la toma de requerimientos de aspecto visual y estético (en Ingeniería de Software este tipo de requerimientos caen, sin ser los únicos, dentro de la categoría de requerimientos no funcionales⁹²) aún no tenía definidas las necesidades de alto nivel; es decir, de su negocio per se. El alcance de la

⁹² Los requerimientos no funcionales de un sistema o proyecto de software describen únicamente los atributos del ambiente del mismo; en otras palabras, todo lo relacionado a sus atributos de calidad, restricciones, aspectos de seguridad, eficiencia, usabilidad y diseño de la interfaz de usuario.

especificación de requerimientos no rebasaba los límites de los visual y concerniente a la interfaz de usuario. Solicitaba un sitio accesible desde cualquier dispositivos móvil y que contara con una interfaz mínima de tienda en línea. Con el paso del tiempo se irán definiendo con más claridad las necesidades de alto nivel para esta tienda *online* y se tiene contemplado el uso de un *CMS*⁹³ para todo lo relacionado a la funcionalidad dinámica del sitio.

Considerando lo anterior, el uso de los conocimientos en Ingeniería de Software, la adopción rápida de *Bootstrap* como marco de trabajo y el uso de las herramientas ya mencionadas para el desarrollo de la maqueta web dieron como resultado un sitio con características altamente responsivas, creado con código consistente y que facilita el mantenimiento y actualización del contenido del sitio, permitiendo y facilitando a la vez la transición de maqueta o plantilla a un sitio completamente funcional que desempeñe el rol de tienda electrónica; en un inicio pequeña pero que con el paso del tiempo irá creciendo; asegurando dicha escalabilidad mediante un *CMS*.

Estos son los aspectos en los que *Bootstrap 3* contribuyó de manera sustancial al proceso de maquetación de la tienda *Alindis*:

- El esquema responsivo basado en rejillas de *Bootstrap 3* facilitó enormemente la colocación de elementos, bloques y componentes.

Evitando por completo problemas de consistencia en cuanto a la creación

⁹³ Del inglés *Content Management System*, se trata de una aplicación de software que permite y facilita la publicación, edición y modificación de contenido; organizando las tareas de actualización y mantenimiento desde una interfaz gestora centralizada.

de plantillas CSS responsivas sobre las cuales se plasmará el esqueleto del sitio. Si se requiere una modificación en cuando al esquema o distribución de algunos componentes, basta con modificar las clases a las que los componentes, bloques y/o sub-bloques están asociadas sin perder la coherencia y lógica que traza el marco de trabajo per se.

- Los componentes visuales como botones se construyen de manera casi automática. Ya cuentan con un estilo definido que se adhiere completamente a las demandas de hoy en día para las interfaces web minimalistas, donde se busca reducir la excesiva ornamentación de los elementos y/o componentes de la interfaz misma.
- La tipografía y estilos para dar formato a textos resulta muchas veces un trabajo muy tedioso para los desarrolladores. *Bootstrap* ya cuenta con clases predefinidas para la tarea concernientes a dar formato al texto sean unívocas en cuanto a su representación estética.
- El uso de la iconografía integrada de *Bootstrap 3* da un toque muy importante al proceso de hacer intuitiva la interfaz. Se cuenta con una amplia gama de íconos que enriquecen considerablemente el significado de las etiquetas de texto.
- Las clases predefinidas para explotar el potencial de librerías *Javascript* como *JQuery* facilitan en gran forma la animación visual de componentes. Específicamente la animación de los menús de navegación al momento de colocar el cursor sobre algún elemento de dicho menú. O simplemente la capacidad de contraer el menú en caso de que se esté usando un

dispositivo cuya pantalla no permita el despliegue completo de los elementos del menú. Los efectos expansión y contracción de la lista que conforma el menú, así como el efecto de desvanecimiento de una ventana modal son detalles que aunque parezcan insignificantes, enriquecen en demasía el flujo y estética de la interfaz de usuario.

- Las clases predefinidas para la colocación de formularios según las necesidades del sitio o página resultaron de gran importancia para la sección correspondiente en al que el usuario completará un pago. Es imprescindible en estos casos recabar la información necesaria para que el producto pueda ser llevado a la comodidad del hogar.
- El uso clases para contenedores básicos y aquellos que dividen el contenido en pestañas fue otro aspecto importante. La lógica en la jerarquía de estas clases simplifica en considerablemente la construcción de este tipo de componentes.
- Y en general todas aquellas clases predefinidas que aporta *Bootstrap 3* y que poseen características altamente semánticas para el desarrollador web al momento de posicionar los elementos en el esquema de distribución preestablecido fueron parte fundamental para la maquetación *frontend*.

7. Conclusiones finales

A lo largo de todos los capítulos se fueron presentando todos los factores que influyen en la actualidad para la creación de un sitio web, desde partes fundamentales para entender el funcionamiento del Internet (protocolos y estándares) hasta las tendencias que forman los usuarios al acceder a esta red global mediante la Web; desde aspectos técnicos básicos de cómo el código *HTML* estructura una página, hasta aspectos más complejos de cómo se estilan dichas páginas mediante *CSS*; y de cómo es que el diseño web ha evolucionado para adaptarse a los cambios que se han presentado a lo largo del tiempo partiendo desde las páginas planas que solo contenían texto informativo (Web 1.0) hasta llegar a ser páginas dinámicas con animaciones y flujos de interacción con los usuarios (la Web semántica).

Son muy variados los usos de la *Internet* pero se enfatizó en la forma más popular de interactuar en ella, la Web. Se destacó cómo es que la manera de usar esta plataforma para el intercambio de información llevó a las empresas a utilizarlo como medio de ventas dando lugar a nuevos modelos de negocio, incrementando así sus ingresos resultados de explotar adecuadamente el potencial enorme que es tener presencia en dicha plataforma que es al final de cuentas, un subconjunto de todo lo que es *Internet*.

Las estadísticas son claras. La creación de un sitio web debe estar a la altura de las exigencias actuales; de no ser así, son pocas las probabilidades de éxito de un sitio. Paralelamente, deben considerarse las recomendaciones establecidas por el Consorcio de la Web para la implementación de tecnologías específicas; se

debe tener especial cuidado en el diseño web, sobre todo por tratarse de realizar la maquetación de un sitio con propósitos que van acorde al modelo de negocios *B2C*.

El cliente dejó muy en claro sus necesidades en cuanto a que su negocio debería tener presencia en la Web; sin embargo, el requisito su primordial no estaba en contar con una tienda funcional; al menos no aún. Su solicitud fue clara y concisa. Requería de una maqueta o plantilla para su sitio Web. Una de las funcionalidades más recalcadas por el usuario era la de una sección tipo *blog* dentro de su sitio.

Se llevaron a cabo los procesos necesarios para recolectar a modo de diagramas y bocetos estructurales, los requisitos del cliente, que por la naturaleza de los mismos cayeron en la categoría de requisitos no funcionales. Se detectó una situación un tanto particular. El cliente requería de un sitio simple pero a la vez intuitivo y agradable; que pueda apreciarse con la misma calidad desde cualquier tipo o tamaño de dispositivo.

Hubo una importante cercanía al cliente (que a su vez tomó el rol de usuario final) gracias a las tecnologías de mensajería al día de hoy.; en otras palabras, existía una continua retroalimentación.

Parte fundamental de los acuerdos que se sostuvieron con el cliente fue la definición del alcance. Como ya se especificó, los requerimientos no reflejaban la necesidad de contar aún con un sistema completo para la compra de artículos.

Entonces, inicialmente el alcance se estableció en el contexto de un prototipo o maqueta de sitio web.

Se consideraron también las variables como tiempo e incluso costo para llevar a cabo los procesos necesarios para implementar la solución. Las herramientas elegidas para el propósito anterior no representaron costos y eran óptimas para agilizar el proceso de desarrollo de la maqueta.

Se mencionó con anterioridad los factores que condujeron a elegir a *Twitter Bootstrap* como el marco de trabajo para implementar la solución; sin embargo, es necesario también aclarar, que esta herramienta facilitó en gran proporción el trabajo por módulos. Uno de estos fue la página interna del sitio. Ésta resultó ser de suma importancia ya que representaba el “contenedor” o marco que englobaría al contenido que cada página requeriría. Cada sección de las páginas del sitio fue codificada de manera modular y se insertaron comentarios dentro del mismo código con el fin de llevar un orden y poder identificar fácilmente cada módulo.

Una vez que se concluía cada sección del sitio (comenzando por la página interna) se notificaba al cliente para que éste accediera al repositorio en el cual se alojaba el proyecto y éste pudiera visualizar dicha sección (página *HTML*) de manera local desde su navegador web.

Gracias a las actividades mencionadas en el párrafo anterior realmente se consiguió una importante interacción con el cliente/usuario final y en consecuencia una vasta retroalimentación que nos permitía llevar a cabo los cambios que se solicitaban, puesto que se tenía contemplado que los cambio en la propuesta de

maquetación podrían surgir de manera frecuente. Los cambios que proponía el cliente al final no impactaron de manera importante a la propuesta estructural que se tenía en un inicio. Cada cambio solicitado era visto desde una perspectiva de funcionalidad o característica agregada dado que *Bootstrap* es una herramienta tan versátil que facilita la agregación de componentes y funcionalidades de manera simple, coherente ordenada y sobre todo concisa.

¿Que viene después del proceso de maquetación?

Todo el proceso para la creación de este sitio web se detalló en cada fase; sin embargo, todo lo realizado nos da como resultado un sitio estático (maqueta); es decir, parte del contenido mostrado permanecerá ahí por mucho tiempo (como lo es la información de la empresa, contacto, y demás), pero la mayoría de información relevante, información de productos o toda la parte del blog, será dinámica; es decir, estará en constante cambio debido a las publicaciones nuevas que surgen cada día y los nuevos productos que salen para su venta. ¿Cómo se soluciona esto? ¿Cómo se logra la transición entre una maqueta; es decir, un sitio estático a un sitio con una completa funcionalidad *backend* (dinámico)? La respuesta más obvia sería cambiar el contenido constantemente pero el cliente no tiene los conocimientos técnicos para hacer esto y tener un desarrollador solo para actualizar el contenido no es viable. Afortunadamente existe una herramienta para ayudar en estas situaciones, usar un *CMS (Content Management System)* o sistema de gestión de contenidos por sus siglas en inglés; el cual es un programa informático utilizado para editar, gestionar y publicar contenido dentro de los sitios que lo requiere. Portales de noticias, páginas de interés social, son algunos

ejemplos de sitios dinámicos y todos utilizan esta herramienta para actualizar sus contenidos constantemente [26].

El gestor de contenidos genera páginas web dinámicas interactuando con el servidor para generar una nueva página con el formato, plantilla, tema visual, o maqueta predefinida (los documentos *HTML*, *CSS* y *JS* ya creados). Cada cambio de diseño debe ser realizado en las páginas solicitadas. Cuando se actualiza el contenido del sitio, el *CMS* cuenta con una interfaz intuitiva que permite a un administrador del sitio llenar los campos necesarios para que la nueva información sea publicada, sin la necesidad de tener conocimientos técnicos para tener acceso a los códigos fuente y realizar las modificaciones pertinentes. En resumidas cuentas, la idea principal detrás de un *CMS* es la creación de y actualización de contenidos, así como tareas de mantenimiento y de seguridad de una manera mucho más sencilla que no requiere de amplios conocimientos técnicos.

Al final, es posible concluir que todos los factores que conformaron la culminación de este proyecto de maquetación armonizaron de una manera importante y destacable. Los conocimientos con los que se cuentan en el área de la ingeniería de software fueron sustanciales para lograr un trabajo coordinado, estructurado y lleno de buenas prácticas, ya sea aquellas que se adoptaron a lo largo de una formación académica o las que vienen inherentes al uso de herramientas como los son los marcos de trabajo para el desarrollo de software.

Referencias.

- [1] A. Rodríguez. El Desarrollo Web en Los Límites Del Arte. Accedido el 4 de febrero de 2015. [Online]. Disponible: <http://abarcarodriguez.com/blog/el-desarrollo-web-en-los-limites-del-arte>
- [2] (2013, Septiembre, 23) Why Front-end Development is so Important for Business Success. [Online]. Disponible: <https://36creative.com/blog/development/1831/front-end-development-important-business-success>
- [3] Bootstrap 3 Tutorial. Accedido el 4 de febrero del 2015. [Online]. Disponible: <http://www.w3schools.com/bootstrap/default.asp>
- [4] (2015, Febrero) What Are Frameworks? 22 Best Responsive CSS Frameworks for Web Design. [Online]. Disponible: <http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html>
- [5] P. Hearn. Website – It’s Important for Business. Accedido el 7 de febrero de 2015. [Online]. Disponible: http://resources.bravenet.com/articles/profit/advertising/website_its_importance_for_business/
- [6] B. J. Fogg, C. Soohoo, L. Marable, J. Stanford, E. R. Tauber, “How Do People Evaluate a Web Site’s Credibility”, Persuasive Technology Lab, Stanford University. Report Released: 29 Octubre 2002. [Online]. Disponible: <http://risingline.com/pdf/stanford-web-credibility.pdf>
- [7] E. Del Valle (2014) Tasa de Conversión en e-Commerce: Cifras, Informes, Sueños y Realidad. Accedido el 5 de enero de 2015. [Online]. Disponible: <http://www.socialmediaycontenidos.com/tasa-de-conversion-en-e-commerce-cifras-informes-suenos-y-realidad>
- [8] J. Niederst. *Learning Web Design* , 4ta Edición, 2006, Editorial O’Reilly, 2012.[Online]. Disponible: <http://it-ebooks.info/book/2171/>
- [9] K. Soucy (2012, Junio, 8) Is Your Homepage Doing What it Should?.[Online]. Disponible: <http://web.archive.org/web/20120608071219/http://www.usableinterface.com/articles/homepage.php>
- [10] A. Nayak (2015, Enero, 8) Top 5 Reasons to Use Responsive Website Design. [Online]. Disponible: <http://tech.co/top-5-reasons-use-responsive-website-design-2015-01>
- [11] M. Rouse. Smartphone. Accedido el 3 de febrero de 2005. [Online]. Disponible: <http://searchmobilecomputing.techtarget.com/definition/smartphone>

- [12] L. Grossman (2007, Noviembre, 1) Invention of The Year: The iPhone. [Online]. Disponible: http://content.time.com/time/specials/2007/article/0,28804,1677329_1678542_1677891,00.html
- [13] S. Kirkland (2014, Mayo, 15) Quantcast: Social Drivers 34 Percent of Mobile Web Traffic, 17 Percent of Desktop Traffic. [Online]. Disponible: <http://www.poynter.org/news/media-innovation/252088/quantcast-79-percent-of-news-and-entertainment-sites-are-mobile-optimized/>
- [14] (2014, Octubre, 30) Tablets and Phablets – Know The Differences. Accedido el 8 de febrero de 2015. [Online]. Disponible: <http://www.teksmobile.com.au/blog/tablets-phablets-know-differences/>
- [15] J. Niederst. "Web Design in a Nutshell" *A desktop Quick Reference*, 3ra Edición, 2006, Editorial O'Reilly, 2006.[Online]. Disponible: <http://it-ebooks.info/book/986/>
- [16] Web TCP/IP. Accedido el 10 de febrero del 2015. [Online]. Disponible: http://www.w3schools.com/website/web_tcpip.asp
- [17] B. Bates, K. Sierra, B. Basham. "Why use Servlets & JSPs", 1ra Edicion, Editorial O'Reilly, 2004. [Online]. Disponible:
- [18] M. MacDonald, *Creating a Website: The Missing Manual*, 3ra Edición, Editorial O'Reilly Media, 2011.
- [19] H. F. Arena, *Linux*, Buenos Aires: RedUSERS, 2010.
- [20] How Does The Internet Work. Accedido el 11 de abril de 2011. [Online]. Disponible:http://www.w3.org/wiki/How_does_the_Internet_work
- [21] About CERN. Accedido el 10 de febrero de 2015. [Online]. Disponible: <http://home.web.cern.ch/about>
- [22] History of The Web. Accedido el 10 de enero de 2015. [Online]. Disponible: <http://webfoundation.org/about/vision/history-of-the-web/>
- [23] D. kumar, P. Ajinkia, A. Pfeiffer (2015, Marzo , 15) W3C. [Online]. Disponible: <https://developer.mozilla.org/en-US/docs/Glossary/W3C>
- [24] Sitio: Webopedia. Accedido en abril de 2015. [Online]. Disponible: <http://www.webopedia.com>
- [25] D. De Luca, *HTML5*, Buenos Aires: RedUSERS, 2011
- [26] Sitio: ALEGSA. Accedido en Marzo de 2015. [Online]. Disponible: <http://www.alegsa.com.ar/Diccionario/diccionario.php>
- [27] Browser Statics. Accedido el 30 de abril de 2015. [Online]. Disponible: http://www.w3schools.com/browsers/browsers_stats.asp

- [28] S. Hay (2011, enero,7) There is no Mobile Web. [Online]. Disponible: <http://www.the-haystack.com/2011/01/07/there-is-no-mobile-web/>
- [29] J. C. Orós, Diseño de Páginas Web con XHTML, JavaScript y CSS, 3ra Edición, México: Alfaomega Grupo Editor, 2011.
- [30] (2014, Mayo, 17) 15 Datos Curiosos Del Uso de Internet. [Online]. Disponible:<http://www.cnnexpansion.com/tecnologia/2014/05/17/15-datos-curiosos-del-uso-de-internet>
- [31] M. Wen. History of E-business. Accedido el 4 de febrero de 2015. [Online]. Disponible: http://www.ehow.com/about_5282606_history-ebusiness.html
- [32] Mercadotecnia. Accedido el 7 de marzo de 2015. [Online]. Disponible:<http://definicion.de/mercadotecnia/>
- [33] D. H. Torres, J. Guerra (2012) Comercio Electrónico. Accedido el 6 de marzo de 2015. [Online]. Disponible: http://www.eumed.net/ce/2012/tcgz.html#_ftnref17
- [34] D. Leon. Negocios en Internet (e-Business) y Comercio Electrónico (e-Commerce). Accedido el 6 de marzo de 2015. [Online]. Disponible: <http://www.guiadelemprendedor.com.ar/negocios-en-internet.htm>
- [35] AMIPCI (2013) Estudio de comercio electrónico México 2013. Accedido el 23 de marzo de 2015. [Online]. Disponible: https://www.amipci.org.mx/estudios/comercio_electronico/131028_-_Comunicado-Estudio_Comercio_Electronico2013-Proyecto-V3_-_REV.pdf
- [36] M. Ramirez (2014, Diciembre, 2014) 'E-commerce', Realidad de 120 Mil mdp en 2014.. [Online].Disponible:http://www.milenio.com/negocios/E-commerce-ventas_internet-comercio_internet_0_434356567.html
- [37] D. Griffin. Explicación del modelo de negocio a consumidor. Accedido el 6 de marzo de 2015. [Online]. Disponible: <http://pyme.lavoztx.com/explicacin-del-modelo-de-negocio-consumidor-8490.html>
- [38] S. Baroncini-Moe. Why a Good Web Site Matters to Your Business. Accedido el 6 enero de 2015 [Online]. Disponible: <http://www.lifehack.org/articles/featured/why-a-good-web-site-matters-to-your-business.html>
- [39] STUDY: Andorid Gains 10 Points on Apple in Q1 2014 Mobile Traffic Market Share. Accedido el 15 de Marzo de 2015. [Online]. Disponible: <http://www.walkersandsdigital.com/Walker-Sands-Mobile-Traffic-Report-Q1-2014>
- [40] (2014, Octubre,30) Finalmente HTML5 ya es Estándar del W3C. Accedido el 24 de marzo de 2015. [Online]. Disponible: <http://blogvecindad.com/finalmente-html5-ya-es-estandar-del-w3c/>

- [41] (2012, Enero, 8) CSS3 Module Status. Accedido el 16 de Marzo de 2015. [Online]. Disponible: <http://www.css3.info/modules/>
- [42] (2014, Abril, 25) What is HTML5?. Accedido el 17 de marzo de 2015. [Online]. Disponible: <http://wp2x.com/html5/>
- [43] (2007, Enero, 18) Reading XML Files Sequentially – Page 1. Accedido el 17 de Marzo de 2015. [Online]. Disponible: http://www.kirupa.com/net/readingXML_pg1.htm
- [44] Introduction to RSS. Accedido el 17 de Marzo de 2015. [Online]. Disponible: http://www.w3schools.com/webservices/rss_intro.asp
- [45] (2013, Junio, 10) Apple Unveils iOS 7. Accedido el 18 de Marzo de 2015. [Online]. Disponible: <https://www.apple.com/pr/library/2013/06/10Apple-Unveils-iOS-7.html>
- [46] D. Coffin (2012, Septiembre, 14) Learning From Bad Web Design. [Online]. Disponible: <http://webmarketingtoday.com/articles/learning-from-bad-web-design/>
- [47] B. Owoh. 11 Common Web Design Mistakes (Blunders). Accedido el 2 de marzo de 2015. [Online]. Disponible: <http://www.hongkiat.com/blog/11-common-mistakes-blunders-in-web-design/>
- [48] Serifa. Accedido el 6 de Marzo de 2015. [Online]. Disponible: <http://pt.wikipedia.org/wiki/Serifa>
- [49] R. Nurik (2014, Agosto, 5) Material Design in The 2014 Google I/O App. [Online]. Disponible: <http://android-developers.blogspot.mx/2014/08/material-design-in-2014-google-io-app.html>
- [50] Tendencias Del Diseño Web Durante 2015. Accedido el 9 de Marzo de 2015. [Online]. Disponible: <http://www.creativosonline.org/blog/tendencias-del-diseno-web-durante-2015.html>
- [51] C. Peterson. *Learning Responsive Web Design*. Editorial O'Reilly, 2014. [Online]. Disponible: <http://it-ebooks.info/book/3522/>
- [52] Steve Jobs: 14 Most Significant Inventions. Accedido el 10 de Marzo de 2015. [Online]. Disponible: <http://www.bet.com/news/fashion-and-beauty/photos/2011/10/steve-jobs-14-most-significant-inventions.html#!041811-National-Jesse-Jackson-News2>
- [53] Use AssistiveTouch on Your iPhone, iPad or iPod Touch. Accedido el 17 de marzo de 2015. [Online]. Disponible: <https://support.apple.com/en-us/HT202658>
- [54] E. Marcotte (2010, Mayo, 5) Responsive Web Design.[Online]. Disponible: <http://alistapart.com/article/responsive-web-design>
- [55] CSS Units. Accedido el 13 de Marzo de 2015. [Online]. Disponible: http://www.w3schools.com/cssref/css_units.asp

- [56] T. Eye (2012, Febrero, 20) Portrait and Landscape Screen Layouts in Android. [Online]. Disponible:<http://tekeye.biz/2012/android-portrait-landscape-screens>
- [57] (2002, Abril, 17) ¿Qué es una línea de Productos?. Accedido el 30 de abril de 2015. [Online]. Disponible:<http://www.gestiopolis.com/que-es-una-linea-de-productos/>
- [58] R. Muñiz. Marketing en el Siglo XXI. 5ta Edición. Accedido el 20 de Marzo de 2014. [Online]. Disponible:<http://www.marketing-xxi.com/Marketing-siglo-xxi.html>
- [59] *ANSI/IEEE Software Requirements Specification*, IEEE Standard 830, 1994.
- [60] S. Wilczynski (2014, Junio, 6) Front-end Frameworks. Introduction (Part 1).[Online]. Disponible: <http://www.merixstudio.com/blog/front-end-frameworks-introduction-part-1/>
- [61] Adrian (2014, Julio, 23) Bootstrap vs. Foundation: Which Frameworks Should You Choose?. [Online]. Disponible:<http://designmodo.com/bootstrap-vs-foundation/>
- [62] B. Utterback (2014, Marzo, 6) What is Bootstrap – The History and the Hype: Part 1 of 2.[Online]. Disponible: <https://www.prestashop.com/blog/en/what-is-bootstrap/>
- [63] Diseño Web Adaptativo con Bootstrap. Accedido el 24 de marzo de 2015. [Online]. Disponible: <http://www.inforede.es/disenio-web-adaptativo-con-bootstrap/>
- [64] What is Open Source?. Accedido el 26 de Abril de 2015.[Online]. Disponible:<http://opensource.com/resources/what-open-source>
- [65] J. Sathik (2014, Octubre) Top 5 Twitter Bootstrap Alternatives, Foundation, Pure, Zimit, HTML Kickstart, Ink. Accedido el 29 de abril de 2015. [Online]. Disponible:<http://www.spywhoami.com/2014/10/twitter-bootstrap-alternatives.html>
- [66] J. Kramer (2013, Agosto, 18) Comparing Twitter Bootstrap vs. ZURB Foundation. [Online]. Disponible:<http://www.slideshare.net/jen4web/comparing-bootstrap-and-foundation>
- [67] Sitio: INK. Accedido el 20 de Mayo de 2015. [Online]. Disponible: <http://ink.sapo.pt/>
- [68] Sitio: Zimit. Accedido el 20 de Mayo de 2015. [Online]. Disponible: <http://firezenk.github.io/zimit/>
- [69] Sitio: Kickstart. Accedido el 20 de Mayo de 2015.[Online]. Disponible: <http://getkickstart.com/>
- [70] Pure - A Minimalistic Responsive CSS Framework. Accedido el 29 de Abril del 2015. [Online]. Disponible: <http://pixabay.com/es/blog/posts/pure-a-minimalistic-responsive-css-framework-55/>

- [71] Sitio: Pure.CSS .Accedido el 20 de Mayo de 2015. [Online]. Disponible: <http://purecss.io/>
- [72] (2012, Mayo, 31) A Modal Cure in Pure CSS - No Wrappers, No Javascript, No BS. Accedido el 15 de Mayo de 2015. [Online]. Disponible: <http://www.backalleycoder.com/2012/05/31/a-modal-cure-in-pure-css-no-wrappers-no-javascript-no-bs/>
- [73] I. Abascal (2014. Agosto, 18) ¿Qué es un Preprocesador de CSS?. [Online]. Disponible: <http://abalozz.es/que-es-un-preprocesador-de-css/>
- [74] C. Coyier (2012, Junio, 11) Poll Results: Popularity of CSS Preprocessors. [Online]. Disponible: <https://css-tricks.com/poll-results-popularity-of-css-preprocessors/>
- [75] V. Van (2013, Diciembre, 11) Why I Dislike The Rem Unit. [Online]. Disponible: <https://vasilis.nl/nerd/dislike-rem-unit/>
- [76] T. Griffiths (2015, Febrero, 6) Front-end Frameworks Comparison - Bootstrap vs Foundation. [Online]. Disponible: <http://www.smart421.com/application-development/front-end-web-development/front-end-frameworks-comparison-bootstrap-vs-foundation/>
- [77] What Are the Software Development Models?. Accedido el 29 de marzo de 2015. [Online]. Disponible: <http://istqbexamcertification.com/what-are-the-software-development-models/>
- [78] Software Development Process Model. Accedido el 14 de Mayo de 2015. [Online]. Disponible: <http://www.cxtgroup.com/Software-Development-Model.htm>