

GUÍA COMPARATIVA DE FRAMEWORKS PARA LOS LENGUAJES HTML 5,  
CSS Y JAVASCRIPT PARA EL DESARROLLO DE APLICACIONES WEB

ÁNGELA MARÍA VALBUENA APONTE

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERÍAS  
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
PEREIRA  
2014

GUÍA COMPARATIVA DE FRAMEWORKS PARA LOS LENGUAJES HTML 5,  
CSS Y JAVASCRIPT PARA EL DESARROLLO DE APLICACIONES WEB

ÁNGELA MARÍA VALBUENA APONTE

MONOGRAFÍA PARA OPTAR AL TÍTULO DE:  
INGENIERO DE SISTEMAS Y COMPUTACIÓN

DIRECTOR  
CARLOS AUGUSTO MENESES ESCOBAR

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERÍAS  
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
PEREIRA  
2014

Notas de Aceptación

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma jurado

---

Firma jurado

## DEDICATORIA

Agradezco a mi madre María del Rosario Aponte Morales y a mi padre Alfonso Valbuena González por su apoyo en mi formación como persona y como profesional y a mis abuelos.

## AGRADECIMIENTOS

Gracias a Dios por permitirme culminar satisfactoriamente otra etapa de mi vida y dar inicio a un nuevo reto en el campo laboral, a mi familia, a todos los docentes que durante la carrera me guiaron y transmitieron su conocimiento formándome no sólo como profesional sino contribuyendo para mi crecimiento como persona.

## Resumen

Las tecnologías web han evolucionado hasta permitir que los desarrolladores puedan crear nuevas e increíbles experiencias web, esta evolución surgió por el continuo esfuerzo de una comunidad web abierta que ayuda a definirlas y, a su vez, garantizando la compatibilidad con los navegadores.

En el desarrollo de aplicaciones web se presentan gran variedad de herramientas que reducen el trabajo y que cuentan con características particulares que facilitan la construcción en algunos tipos de aplicaciones, con el objetivo de optimizar la construcción de una aplicación web, surge el siguiente trabajo que ofrece una comparación de los frameworks de los lenguajes HTML5, CSS y JavaScript, evaluando características, ventajas, desventajas y limitaciones.

La primera parte de recopila la información de los frameworks para cada lenguaje, luego se procede a seleccionar los frameworks más conocidos y con mayor documentación, para luego concluir que herramienta ofrece más ventajas a los desarrolladores, y por ultimo un ejemplo de una aplicación básica desarrollada en los frameworks seleccionados.

El trabajo permite a los desarrolladores la facilidad de elegir una herramienta de trabajo que cumpla con los requerimientos del cliente y optimizando el trabajo de construcción de la aplicación web.

## Abstract

Web technologies have evolved, permitting developers to create new, incredible web experiences. This evolution emerged out of the constant effort of the open internet community, which helps define browsers, and at the same time guarantees compatibility with them.

In the development of web applications, there are great variety of tools that minimize work and have proprietary characteristics that make developing some types of applications more feasible. With the objective to optimize building of web applications, this work- a comparison between different language frameworks, such as HTML5, CSS and JavaScript, and an evaluation of their different characteristics, advantages, disadvantages and limitations.

In the first part, information is gathered on the frameworks of each language. The work then proceeds to select the frameworks best known and with the most documentation, so that later it can be concluded which tools offer more advantages to developers. Finally, an example of a basic application developed with the selected frameworks will be given.

The work provides ease to developers to select a tool to work with that has all the requirements of the client and optimizes the building of web applications.

## Palabras Claves

- ✓ HTML5: HyperText Markup Language, version 5
- ✓ CSS: Cascading Style Sheets
- ✓ JavaScript
- ✓ Freelance
- ✓ Aplicación Web
- ✓ Responsive Design
- ✓ Layout



## CONTENIDO

	Pág.
INTRODUCCIÓN	10
1. GENERALIDADES	11
1.1. Título del proyecto	11
1.2. Planteamiento del problema	12
1.3. Justificación	13
1.4. Objetivos	14
1.4.1 Objetivo general	14
1.4.2 Objetivo específicos	14
2. ESTADO DEL ARTE	15
2.1 Marcos	15
2.1.1 Marco conceptual	15
2.1.2 Marco referencia	19
2.1.3 Marco histórico	20
2.2 Frameworks del lenguaje HTML5	21
2.2.1 Twitter Bootstrap 2.3.1	23
2.2.2 Foundation 4	26
2.3 Frameworks del lenguaje CSS	28
2.3.1 960 Grid Systems	29

2.3.2	Blueprint CSS	31
2.4	Frameworks del lenguaje JavaScript	33
2.4.1	Backbone.js	34
2.4.2	Ember.js	36
2.4.3	Angularjs	38
3.	APORTES	41
3.1	Comparativa de Frameworks HTML5	41
3.2	Comparativa de Frameworks CSS	42
3.3	Comparativa de Frameworks JavaScript	43
3.4	Funcionamiento del framework de HTML5, CSS y JavaScript	44
3.4.1	Twitter BootStrap 2.3.1	44
3.4.2	Blueprint CSS	50
3.4.3	Backbone.js	55
3.5	Aplicación Web con los 3 frameworks integrados	58
4.	CONCLUSIONES	59
5.	BIBLIOGRAFÍA	60

## TABLAS

Tabla 1: Tabla comparativa de Frameworks HTML5	20
Tabla 2 comparativa 2: Frameworks HTML 5	41
Tabla 3 comparativa: Frameworks CSS	42
Tabla 4 comparativa: Frameworks JavaScript	43

## INTRODUCCIÓN

En sus inicios las páginas web solo contenían información estática, luego estas evolucionaron siendo más dinámicas y convirtiéndose en aplicaciones web que permitían ejecutar las mismas tareas que una aplicación de escritorio. Estos avances crearon la necesidad de optimizar el trabajo surgiendo los frameworks como ayuda en la creación de aplicaciones web.

La utilización de los frameworks permitió que los desarrolladores consumieran menos recursos y tiempo en la construcción de las aplicaciones; en la actualidad existen gran variedad de estas herramientas de trabajo creados por la comunidad. Las aplicaciones web se han convertido en una ventaja muy competitiva para las compañías debido a que estás agilizan el trabajo, por otra parte en la generación de empleo para los desarrolladores debido a la gran demanda que ha generado el auge de la computación en la nube.

Por último, las ventajas que tiene desarrollar una aplicación web en comparación con una de escritorio son: los virus alojados en los equipos no son un peligro, la información se puede trabajar desde cualquier lugar con conexión a internet, la información se puede acceder más fácil, son algunas de las mejoras que han llevado a que grandes aplicaciones estén en proceso de migración a aplicaciones web.

## 1. GENERALIDADES

### 1.1 Título del proyecto

Guía comparativa de frameworks para los lenguajes HTML5, CSS y JavaScript para el desarrollo de aplicaciones web.

## 1.2. Planteamiento del problema

La constante evolución de los sistemas y la globalización han permitido que la accesibilidad a las diversas herramientas informáticas que utilizamos en nuestras labores pueda llevarse a cabo tan solo con una computadora que tenga conexión a la red. Esto ha generado el desarrollo de numerosos lenguajes con múltiples opciones para cada usuario y de acuerdo a sus preferencias; así mismo se han creado varios framework (Herramientas de trabajo) que facilitan y reducen el tiempo en la ejecución de las tareas, sin embargo la cantidad de framework ha aumentado considerablemente haciendo necesario su clasificación y estructuración en guías comparativas.

Para este caso se analizarán alternativas de frameworks en lenguajes HTML 5, CSS y JavaScript, que en la actualidad se utilizan bastante en el desarrollo de aplicativos web, y además no es común encontrar ayudas guiadas que le permitan al desarrollador determinar cuál es la mejor opción para crear su propia aplicación web. Lo anterior ha permitido identificar ciertos problemas como:

- ✓ No contar con una guía detallada y clara que permita brindar soluciones combinando los tres lenguajes mencionados anteriormente para construir una aplicación básica.
- ✓ No se puede determinar cuáles son las ventajas y desventajas de cada framework.
- ✓ No se puede tomar una decisión objetiva sobre cual framework es la mejor opción de acuerdo a los conocimientos de los desarrolladores.
- ✓ Se invierte demasiado tiempo en actividades que podrían agilizarse y tener menos durabilidad

Estos problemas evidencian la falta de una guía comparativa de los frameworks, que permita que los resultados obtenidos sean consistentes; ¿será posible tener una guía comparativa de frameworks para los lenguajes HTML 5, CSS y JavaScript que permita conocer al menos una forma adecuada de cómo hacer desarrollos de aplicaciones web?

### 1.3 Justificación

La elaboración de este proyecto podrá generar beneficios en diferentes contextos que fomentaran el desarrollo económico, tecnológico, cultural y social del país.

Desde el contexto económico crea una nueva oportunidad de empleo que contribuye a la generación de ingresos trabajando como freelance de aplicaciones web.

Desde el contexto tecnológico, permite evaluar la compatibilidad de la aplicación además las ventajas que tiene al desarrollar en un framework específico y como esta compite en el mercado.

Desde el contexto social brinda posibilidad de mejorar la funcionalidad de las diversas aplicaciones web al servicio de los usuarios ofreciéndoles numerosos beneficios como agilidad y facilidad.

Desde el contexto internacional otorga la posibilidad de profundizar en los diferentes framework a los desarrolladores expandiendo su conocimiento y haciéndolos competitivos en el mercado internacional, además de aportarles herramientas para disminuir la complejidad en sus labores.

## 1.4 Objetivos

### 1.4.1 Objetivo General

Elaborar una guía comparativa de frameworks para los lenguajes HTML 5, CSS y JavaScript que sirva para el desarrollo de aplicaciones web.

### 1.4.2 Objetivo Específicos

Identificar los frameworks para los lenguajes HTML 5, CSS y JavaScript.

- ✓ Identificar las ventajas, desventajas, alcances y limitaciones de utilizar los diferentes frameworks.
- ✓ Establecer la comparación de los frameworks para los lenguajes HTML 5, CSS y JavaScript para aplicaciones web.
- ✓ Seleccionar un framework de cada lenguaje y explicar su funcionamiento.
- ✓ Construir un ejemplo con el uso de frameworks en los lenguajes HTML 5, CSS y JavaScript.



## 2. ESTADO DEL ARTE

### 2.1 Marcos

#### 2.1.1 Marco Conceptual

##### ✓ Framework:

El término framework, hace referencia a una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones, en la actualidad existen frameworks para lenguajes como html5, css y JavaScript. [1]

##### ✓ Framework Web:

Un framework Web, se puede definir como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. [1]

##### ✓ HTML:

HTML es una implementación del standard SGML (Standard Generalized Markup Language), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.

Es un metalenguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones. [2]

✓ HTML 5<sup>1</sup>:

Es una colección de estándares para el diseño y desarrollo de páginas web. Esta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella.

HTML 5 permite una mayor interacción de las páginas web con contenidos de reproducción digital (Videos, audio, juegos, entre otros) y facilidad a la hora de maquetar una página web. [3]

✓ CSS<sup>2</sup>:

Hojas de Estilo en Cascada, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. [4]

✓ JavaScript:

JavaScript es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML.

Hasta entonces ya se usaba HTML y JAVA, pero la aparición del JavaScript produjo una importante revolución, ya que dio al usuario la posibilidad de crear aplicaciones "on-line. [5]

---

<sup>1</sup> HTML 5: HyperText Markup Language, version 5

<sup>2</sup> CSS Cascading Style Sheets

✓ Aplicación Web:

En la Ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web, a la independencia del Sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. [6]

✓ Desarrollador Web:

Es la persona responsable de la naturaleza técnica al crear un sitio web. Su objetivo es entender cuáles son las necesidades funcionales que precisa una página web y debe implementar las acciones de desarrollo oportunas para poder cumplir con dichas necesidades.

El desarrollador web trabaja en aquello que no se ve de una web, pero que sí se siente y que se necesita para que todo funcione adecuadamente y tenga un sentido. Sus tareas diarias se basan en la programación en diversos lenguajes, tales como CSS, HTML, PHP, SQL, Perl entre otros para poder llevar a cabo el desarrollo de aplicaciones web diversas. [7]

✓ Freelance:

Es un trabajo en el que el trabajador realiza su actividad de forma autónoma o por cuenta propia. El trabajador ofrece sus servicios a terceros, que son sus clientes.

Los trabajadores freelance, a diferencia de los empleados por cuenta ajena, no suelen cobrar por horas, sino por trabajo realizado, por volumen de trabajo o por resultados. No obstante, aquí hay variaciones que dependen de las especificidades del oficio en cuestión y del tipo de servicio ofrecido. [8]

✓ API:

Interfaz de programación de aplicaciones, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las "librerías". Es una fuente de código basado en la especificación destinada a ser utilizada como una interfaz de componentes de software para comunicarse entre sí. [9]

### 2.1.2 Marco de Referencia

Algunos estudios permiten entender el uso de frameworks en el desarrollo de soluciones web. Entre estos tenemos:

- a) *“Estudio comparativo de alternativas y frameworks de programación, para el desarrollo de aplicaciones móviles en entorno Android”*, Ricardo José Iskandar Morine – Universitat Politècnica de Catalunya. El proyecto evalúa el alcance de las aplicaciones al ser desarrolladas en API 's diferentes al nativo, confrontando las ventajas y desventajas de utilizar los API 's según el conocimiento del desarrollador y el tiempo del desarrollo además estudiando el funcionamiento de los frameworks para aplicaciones móviles, para lograr lo anterior se tomaron como muestra nueve alternativas que permiten crear aplicaciones en Android para luego del estudio seleccionar cuatro alternativas y sobre estas se analizó que elementos y componentes pueden acceder cada uno de los frameworks. [10]
- b) *“Frameworks y herramientas para el desarrollo de aplicaciones orientadas a dispositivos móviles”*, Johanna Fernanda Rengifo Briñez, Carlos Alberto Betancourt Pérez – Universidad Tecnológica de Pereira, el proyecto reúne los frameworks y herramientas para el desarrollo de aplicaciones móviles con el objetivo de que al crear una aplicación se puede evaluar la compatibilidad de la misma con otros sistemas. [11]
- c) *“Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java”*, Ángel Hernández Cruz – Universidad de la Sierra Juarez, en el estudio se analizaron los frameworks Spring Roo 1.2.1, Struts 2.3.1.2 y Tapestry 5.3.3 del lenguaje java determinando de estos el tiempo del desarrollo, la documentación del framework y las dificultades encontradas al desarrollar una aplicación web. [12]
- d) *“Estudio comparativo de frameworks de instrumentación dinámica de ejecutables”*, Juan Antonio Artal Lozano – Universidad Zaragoza, El estudio evalúa el rendimiento de diferentes frameworks de (DBI) instrumentación dinámica de ejecutables, esta comparación se obtiene comparando el rendimiento de la misma aplicación ejecutada de forma nativa y con diferentes frameworks de DBI. [13]

### 2.1.3 Marco Histórico

Las aplicaciones Web interactivas poco a poco han revolucionado la forma de utilizar internet, aumentando el contenido de las páginas con texto a un contenido rico e interactivo, por lo tanto escalable.

Uno de los primeros lenguajes de programación para el desarrollo de aplicaciones web es el "Perl". Fue inventado por Larry Wall en 1987 antes de que internet se convirtiera en accesible para el público en general. Pero fue en 1995 cuando el programador Rasmus Lerdorf puso a disposición el lenguaje PHP con este el desarrollo de aplicaciones web realmente despegó.

El uso de frameworks en la construcción de aplicaciones pequeñas se generó con la "revolución web 2.0", una compañía de 5 personas lanzó BaseCamp, un servicio de administración de proyectos basado en web y construido sobre Rails, un framework de desarrollo web para el versátil lenguaje "Ruby". [14]

Los frameworks se originaron a partir de los patrones de diseños, estos son contruidos en base a lenguajes orientados a objetos. Esto permite una mejor modularización de los componentes y óptima reutilización de código. Además, en la mayoría de los casos un framework implementará uno o más patrones de diseño de software que aseguren la escalabilidad del producto. Un patrón de diseño es un set de metodologías probadas para resolver problemas comunes en el diseño de aplicaciones.

Los frameworks, entonces, suelen ser implementaciones de patrones de diseño conocidos, aderezados con funciones que asisten al desarrollador. Pero un framework, a veces, puede ser más de lo que se necesita para ciertos proyectos. [15]

## 2.2 Frameworks del lenguaje HTML5

En la actualidad existen gran variedad de frameworks HTML5, algunos con enfoque a diseñadores y otros a programadores, se seleccionaron los frameworks más utilizados con el fin de evaluar las características, ventajas, desventajas, alcances y limitaciones. Basados en esta información se comparan los frameworks, con el objetivo de seleccionar el que más ventajas aporte a los usuarios y se estructura un ejemplo básico de su funcionamiento.

Para seleccionar los frameworks se tomó como referencia la siguiente tabla:

	License	Size (MB)*	CDN	Contr-ibutors	Twitter	Founder	Repo	Start Date
<b>Twitter Bootstrap</b>	Apache License v2.0	3.15	cdnjs BootstrapCDN	235	@twbootstrap 69,923 followers	Mark Otto, Jacob (fat)	GitHub	Aug 2011
<b>Foundation</b>	MIT License	2.20	None	237	@foundationzurb 11,326 followers	ZURB	GitHub	Sep 2011
<b>Skeleton</b>	MIT License	0.03	jsDelivr	14	@dhg 6,676 followers	Dave Gamache	GitHub	May 2011
<b>HTML5 Boilerplate</b>	Multiple open source licenses	.09	None	157	@h5bp 28,221 followers	Paul Irish	GitHub	Apr 2010
<b>HTML KickStart</b>	MIT License	0.30	jsDelivr	11	@htmlkickstart 308 followers	Joshua Gatcke	GitHub	Feb 2013
<b>Montage</b>	BSD License	0.89	None	24	@montagejs 83 followers	Multiple	GitHub	Jul 2012
<b>SproutCore</b>	MIT License	10.7	None	126	@SproutCore 2,999 followers	Strobe Inc.	GitHub	2010
<b>Zebra</b>	LGPL	6.82	None	4	@sandtube 0 followers	Andrei Vishneuski	GitHub	Nov 2012
<b>CreateJS</b>	Unknown	Various sizes	CreateJS CDN Libraries	14	@CreateJS 2,972 followers	Multiple	GitHub	Mar 2012
<b>Less Framework</b>	MIT License	0.007 MB	None	2	@lessframework 1,461 followers	Joni Korpi	GitHub	Jun 2011

Tabla 1: Tabla comparativa de Frameworks HTML5. Tomada de [21]

La lista siguiente corresponde a los frameworks seleccionados para este estudio teniendo en cuenta su facilidad de manejo y su popularidad de uso:

✓ Twitter Bootstrap 2.3.1

✓ Foundation 4

Para la elección de estos frameworks se procedió a buscarlos todos en internet y cuantificar la información encontrada.



### 2.2.1 Twitter Bootstrap 2.3.1

Twitter Bootstrap es una colección de herramientas de software libre para la creación de sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript, además permite la adaptación de la interfaz dependiendo del tamaño del dispositivo en el que se visualice sin que el usuario tenga que hacer nada, esto se denomina Responsive Web Design.

El Framework trae varios elementos con estilos predefinidos fáciles de configurar e integración jQuery para ofrecer ventanas y tooltips dinámicos. Es el proyecto más popular en GitHub. [16]

#### a) Características de Twitter Bootstrap 2.3.1

- ✓ Ofrece grids fijados y líquidos – 724px, 940px, 1170px
- ✓ Sistema de parrilla de 12 columnas.
- ✓ Se gestiona a través de GitHub.
- ✓ Compatible con Chrome, Firefox, Safari, IE 7 y Opera con tabletas y teléfonos inteligentes.
- ✓ Usa HTML5 y CSS3. Además de integrar librerías JavaScript y Less <sup>3</sup>.
- ✓ Es open source.
- ✓ Su descarga ocupa pocos Kilobytes. [17]

---

<sup>3</sup> Less: un lenguaje de hojas de estilo dinámico, que permite introducir funciones y variables en las CSS.

### b) Alcance de Twitter Bootstrap 2.3.1

Este framework se puede utilizar en cualquier tipo de aplicación pero se recomienda en estos tipos:

- ✓ Diseño de Panel de Control (Dashboard) de una aplicación Web: Básicamente el Dashboard es el punto de entrada a la administración de una aplicación web, generalmente de acceso restringido.
- ✓ Diseño de Aplicaciones Móviles: Twitter Bootstrap se apoya en HTML5 y CSS3, que son soportados por los dispositivos móviles. [18]

### c) Ventajas de Twitter Bootstrap 2.3.1

- ✓ Utiliza componentes y servicios creados por la comunidad
  - HTML5 shim
  - Normalize.css
  - OOCSS
  - jQuery UI (implementa 12 plugins)
  - LESS
  - GitHub
- ✓ Es un compendio de buenas prácticas:
  - Implementa los nuevos estándares: HTML5 + CSS3
  - Cross-browser
  - IE 7/8/9, Firefox, Chrome, Safari y Opera 11
  - Grid system
  - 12 columnas por defecto. Fijas (px) o fluidas (%).
- ✓ Comunidad GitHub de Twitter Bootstrap.
- ✓ Herramienta ágil para construir interfaces.
- ✓ Tiene un tema por defecto. [19]

#### d) Desventajas de Twitter Bootstrap 2.3.1

- ✓ Aunque la curva de aprendizaje es liviana, se debe familiarizar con su estructura y nomenclatura.
- ✓ El diseño gráfico debe estar adaptado a las 12 columnas.
- ✓ Por defecto ya tiene anchos, márgenes y altos de línea.
- ✓ Es complicado cambiar de versión cuando se han realizado modificaciones profundas sobre el core.
- ✓ Cuando se necesite añadir componentes que no existan, se debe personalizar para que mantenga la coherencia con el diseño. [19]

#### e) Limitaciones de Twitter Bootstrap 2.3.1

- ✓ Framework orientado a desarrolladores.
- ✓ Se debe tener conocimientos en CSS para utilizarlo, el no tenerlos generaría que todas las páginas se vean igual y problemas al intentar modificarlas. [20]

### 2.2.2 Foundation 4

Es un framework front-end que consta de herramientas útiles para la toma de respuesta, primero los sitios web móviles, construido con HTML, CSS y jQuery, utiliza tecnologías y prácticas modernas.

Foundation 4 cuenta con clases de marcador de posición y un nuevo marcado semántico, la principal diferencia con la última versión es un enfoque en ser móvil-primero, lo que significa que los contenidos móviles se priorizan sobre contenido habitual, permite a los usuarios móviles tener tiempos de carga tan cortos como sea posible. [23]

#### a) Características Foundation 4

- ✓ Tiene grids fijados en varias plataformas.
- ✓ Ofrece 1-16 columnas con personalizador.
- ✓ Tiene un conjunto de herramientas de interfaz de usuario que no se adhieren a cualquier estilo de conjunto.
- ✓ Se gestiona a través de GitHub.
- ✓ Compatible con navegadores Chrome, Firefox, IE +, Safari con OS, iPad, Android 2, 4 tableta y teléfono, y Windows Phone 7 +. [17]
- ✓ Es open source.

#### b) Alcance de Foundation 4

Este framework se utiliza para aplicaciones móviles, Foundation 4 proporciona tiempos de carga mínimos para aquellos usuarios que naveguen desde sus móviles.

#### c) Ventajas de Foundation 4

- ✓ Es compatible con cualquier dispositivo y también ofrece ordenamiento, nesting, offsets con soporte de presentación de dispositivo para facilitar la creación de diseños.
- ✓ Función de navegación que permite crear el estilo de la barra y con la navegación desplegable de 3 niveles para barras simples, barras laterales y soporte de subnav pills.
- ✓ Soporte de UI Elements para incluir elementos útiles tales como información sobre herramientas, etiquetas, clases de visibilidad para que se pueda ocultar o mostrar las cosas de acuerdo con la orientación de la pantalla o el tamaño, alertas, paneles y muchos más. [17]

#### d) Desventajas de Foundation 4

- ✓ No se cuenta con una comunidad oficial.
- ✓ Orientado a aplicaciones móviles.

#### e) Limitaciones de Foundation 4

- ✓ Se debe tener conocimiento en los estándares de aplicaciones móviles.

### 2.3 Frameworks del lenguaje CSS

Debido a que las estructuras CSS son complicadas surgen los frameworks que ayudan a facilitar el desarrollo de las aplicaciones.

En la actualidad existen gran variedad de frameworks de CSS. Se seleccionaron los frameworks más utilizados con el fin de evaluar las características, ventajas, desventajas, alcances y limitaciones.

La lista siguiente corresponde a los frameworks seleccionados para este estudio teniendo en cuenta los artículos encontrados:

- ✓ 960 Grid System
- ✓ Blueprint CSS

Para la elección de estos frameworks se tomó como referencia los artículos “Frameworks de CSS para el desarrollo web [24]” y “22 Responsive frameworks CSS para el diseño web [25]”.

### 2.3.1 960 Grid System

960 Grid es un framework de CSS que se creó por la necesidad de tener un “estándar” en el ancho de los sitios web, el cual facilita la maquetación a un estándar de 960px de ancho, tiene configuraciones de 12 y 16 columnas para poder combinar entre sí y así crear layout de una forma simple y rápida. [26]

#### a) Características de 960 Grid System

- ✓ Diseño basado en retícula, utiliza columnas de tamaño fijo para establecer una retícula que sirva de base en la creación de los sitios web. Puede ser una combinación de 12 o 16 columnas.
- ✓ Compatibilidad, se creó contemplando desde el inicio la compatibilidad con los diferentes navegadores de Internet, utiliza etiquetas estándar de HTML y CSS.
- ✓ Administración independiente de las tipografías, mediante un archivo externo llamado text.css.
- ✓ Tipografía definida en pixeles, facilita el diseño ya que se pueden manejar tamaños absolutos en el diseño.
- ✓ Estructura básica sin diseño. [27]
- ✓ Alcance de 960 Grid System
- ✓ Aplicaciones web de estructura orientada a columnas. [29]

#### b) Ventajas de 960 Grid System

- ✓ Estructura previa, el sistema ofrece una estructura para la diagramación de los sitios web.
- ✓ Experiencia visual, ofrece una retícula que mejora el balance, la alineación y el espacio para lograr una mejor experiencia visual.

- ✓ Reemplaza el uso de tablas.
- ✓ Compatibilidad. El 960 Grid System es compatible con la mayoría de los navegadores más utilizados como el Mozilla Firefox, Google Chrome, Safari, Internet Explorer y Opera.
- ✓ Mayor velocidad de desarrollo. El uso de un sistema como el 960 Grid System agiliza el proceso de creación de sitios web. [28]

#### c) Desventajas de 960 Grid System

- ✓ Es una estructura rígida, el sistema ofrece una retícula rígida para la diagramación de los sitios web que no permite variantes para lograr otras formas de mostrar el contenido.
- ✓ Limita la creatividad, el uso continuo de una retícula, limita la creatividad en el diseño y la diagramación de sitios web.
- ✓ Nuevas dimensiones de pantalla, funciona muy bien con resoluciones de hasta 1024 pixeles; sin embargo, cada vez las pantallas permiten mayores resoluciones y el sistema no se adapta satisfactoriamente a resoluciones superiores.
- ✓ Difícil implementación. [28]

#### d) Limitaciones de 960 Grid System

No es fácil de implementar por cualquier diseñador web, se deben tener conocimientos de HTML y CSS. [28]



### 2.3.2 Blueprint CSS

Blueprint es un marco CSS que sirve de apoyo en el diseño y maquetación de webs 2.0, que apunta a reducir tiempo de desarrollo ofreciendo una base sólida para construcción de un proyecto, con una tabla fácil de usar, tipografía sensible, plugins útiles y una hoja de estilo para la impresión. [31]

#### a) Características de Blueprint CSS

- ✓ Una rejilla que permite crear cualquier estructura de página web, se trata de un conjunto de clases CSS para posicionar cualquier elemento en un espacio de 950 píxeles de anchura, dividido en 24 columnas.
- ✓ Una definición de tipografía predeterminada, de manera precisa para todos los elementos de la página que pueden tener texto.
- ✓ Estilos para formularios, con los que mejorar las interfaces de usuario.
- ✓ Estilos para impresión, con las definiciones CSS más útiles para que las páginas se lean bien cuando se imprimen en papel.
- ✓ Estilos específicos para IE, con los que resolver algunos de los problemas más comunes de Internet Explorer. [30]

#### b) Alcance de Blueprint CSS

Cualquier aplicación se puede desarrollar utilizando el framework.

#### c) Ventajas de Blueprint CSS

- ✓ La curva de aprendizaje es corta.
- ✓ Muy robusto.
- ✓ Se integra bien con otros frameworks.
- ✓ Su nivel de Cross-browser es del 99%.
- ✓ No es restrictivo.
- ✓ Hoja de estilos para impresión.
- ✓ Clases fáciles de recordar.
- ✓ Te puedes crear tu propia rejilla.
- ✓ Gran comunidad. [32]

#### d) Desventajas de Blueprint CSS

- ✓ En un principio se puede percibir como un sistema rígido.
- ✓ Énfasis en la estructura por encima del contenido de los objetos. [33]

#### e) Limitaciones de Blueprint CSS

- ✓ Conocimientos previos en CSS

## 2.4 Frameworks del lenguaje JavaScript

Con la llegada del web 2.0 y lo que se conoce como AJAX, JavaScript está presente en los navegadores, los frameworks para JavaScript han permitido que el desarrollo sea más dinámico pues basta con llamar un conjunto de elementos objetos como controles, widgets, entre otros para crear una aplicación web del tipo RIA (Rich Internet applications).

En la actualidad existen gran variedad de frameworks de JavaScript, se seleccionaron los frameworks más utilizados con el fin de evaluar las características, ventajas, desventajas, alcances y limitaciones.

En la decisión de cuales frameworks comparar se tomó como referencia el artículo “los 5 mejores frameworks MVC de JavaScript” [34], se seleccionaron los frameworks:

- ✓ Backbone.js
- ✓ Ember.js
- ✓ Angular.js

### 2.4.1 Backbone.js

Backbone.js es un framework que permite al desarrollador estructurar la aplicación, delegando ciertas labores a elementos específicos con el fin de hacer aplicaciones escalables, robustas y que brinden al usuario una gran experiencia de uso. [35]

#### a) Características de Backbone.js

- ✓ Permite la programación atendiendo al paradigma MVC (Modelo Vista Controlador).
- ✓ Ayuda a crear estructuras bien definidas para los datos de la aplicación y facilita la creación de eventos cuando los datos cambian.
- ✓ Simplifica el uso de vistas, ayuda a pintar interfaces de usuario en la página.
- ✓ Permite el uso de colecciones de modelos, en los que se cuenta con acceso a diferentes operaciones sobre los mismos.
- ✓ Simplifica y ayuda a ordenar el código de las peticiones AJAX para realizar solicitudes al servidor. [36]

#### b) Alcance de Backbone.js

Aplicaciones que se pueden crear utilizando Backbone.js

- ✓ Para aplicaciones web de lado cliente y de una sola página cuya complejidad exija el apoyo de una estructura definida para un proceso de desarrollo ordenado. [39]
- ✓ Aplicación web del tipo RIA (Rich Internet applications).

#### c) Ventajas de Backbone.js

- ✓ La comunidad es fuerte y está creciendo bastante.
- ✓ Comunidad Github [34]
- ✓ Carga la funcionalidad por defecto, la librería se encarga de todo lo que hay por debajo. [37]
- ✓ No hay necesidad aprender una sintaxis nueva, si se tiene experiencia con jQuery y Underscore.js.
- ✓ Se gana control ya que se puede programar y cambiar el comportamiento de prácticamente cualquier. [38]

#### d) Desventajas de Backbone.js

- ✓ Delega la responsabilidad al programador a la hora de estructurar el código una app de tamaño mediano/grande, se debe tener cuidado y refactorizar frecuentemente el código. [38]

#### e) Limitaciones de Backbone.js

- ✓ La curva de aprendizaje es lenta si no hay conocimientos previos en jQuery.

### 2.4.2 Ember.js

Ember.js es un framework JavaScript para crear aplicaciones web del lado del cliente. Está basado en la arquitectura modelo-vista-controlador (MVC).

Está catalogado como uno de los principales framework en el mundo de JavaScript ya que permite a los desarrolladores crear aplicaciones de una sola página (single-page) escalables. [40]

#### a) Características de Ember.js

- ✓ Ember.js representa un cambio de paradigma con respecto a otros frameworks ya que para funcionar correctamente exige que se respete totalmente las convenciones de nomenclatura para nombrar los objetos de la App.
- ✓ El framework tiene la posibilidad de definir rutas anidadas, que son rutas que comparten un mismo recurso.
- ✓ Una única página puede tener varios controladores activos a la vez.
- ✓ Ember.js está muy integrado con el motor de plantillas Handlebars.
- ✓ Capa de persistencia de datos llamada Ember Data. [41]

#### b) Alcance de Ember.js

Se pueden desarrollar aplicaciones del tipo SPA (Aplicaciones Single Page). [42]

#### c) Ventajas de Ember.js

- ✓ Convención sobre configuración, estandariza el desarrollo de aplicaciones del lado del cliente, utilizando reglas muy estrictas para el desarrollo de las aplicaciones.
- ✓ Enlace de datos. Ember.js está muy unido a Handlebars.js, es una librería que permite definir templates o plantillas semánticas con las cuales podemos representar data en nuestras vistas de manera sencilla.
- ✓ La automatización de cambios, es decir, cuando un objeto JavaScript que este enlazado cambia automáticamente se actualiza DOM (Document Object Model) de la App y viceversa, cuando el DOM cambia se actualiza automáticamente el objeto de JavaScript.
- ✓ Mayor enfoque en la creación – Ember.js permite enfocarse en el desarrollo de la aplicación. [40]

#### d) Desventajas de Ember.js

- ✓ Poca información y recursos para aprender y practicar que estén actualizados y de forma gratuita.
- ✓ Curva de aprendizaje un tanto abrupta ya que es indispensable que respetemos todas las convenciones y el flujo de eventos del framework para aprovechar toda su potencia. [41]

#### e) Limitaciones de Ember.js

- ✓ El framework Ember.js no tiene buenas herramientas de aprendizaje.

### 2.4.3 Angularjs

Angular.js es un framework JavaScript Open Source desarrollado por Google, sirve para crear Webapps en lenguaje cliente con JavaScript ejecutándose con el conocido single-page applications (aplicación de una sola página) que extiende el tradicional HTML con etiquetas propias. Es totalmente extensible y funciona bien con otras bibliotecas. Cada función puede ser modificada o reemplazada para satisfacer las necesidades de flujo de trabajo de desarrollo y características únicas. [43]

#### a) Características de Angularjs

- ✓ No oculta el HTML y el CSS, toma sus fortalezas y las extiende volviéndolas adecuadas para la descripción de vistas dinámicas. El resultado es un flujo de trabajo que resulta bastante familiar para cualquier desarrollador web y con un estilo de programación de JavaScript sorprendentemente conciso, claro y enfocado.
- ✓ Fácil comprensión para los que comienzan a usarlo pues ofrece características sofisticadas para desarrolladores con necesidades complejas.
- ✓ El código de aplicaciones creadas con Angular.js siempre está organizado Modelos, Vistas, Controladores y opcionalmente Servicios. [44]

#### b) Alcance de Angularjs

- ✓ Ideal para declarar documentos estáticos.



### c) Ventajas de Angularjs

- ✓ Potente sistema de templating incluido en el mismo framework.
- ✓ Sincronización entre vistas y modelos para crear páginas one-page.
- ✓ Uso de “directives” para la creación de nuevos atributos o nuevas etiquetas html.
- ✓ Uso de filtros para alterar la presentación de datos.
- ✓ Uso de Servicios que se encargan de la comunicación con el servidor para la consulta de datos. [45]

### d) Desventajas de Angularjs

- ✓ Funciona bien para aplicaciones REST en donde la mayor carga del desarrollo está en el lado del cliente, pero no para aplicaciones donde el manejo de datos es muy grande y se deba atribuir un tanto de esa carga al servidor.
- ✓ Sintaxis de templating común por muchos frameworks de este tipo incluso en el lado del servidor. Se corre el riesgo de crear conflictos, pero afortunadamente esto se puede alterar en la configuración de Angular.js.
- ✓ La directiva ng-cloak permite ocultar elementos hasta que Angularjs los tiene listos, sin embargo en algunas ocasiones tus hojas de estilo pueden interferir.
- ✓ Alta segmentación de nuestro scripting y estructura de archivos, se deben tener separados los views, los models, los modules, los directives, los filters y los directives. [45]

e) Limitaciones de Angularjs

- ✓ Conocimientos previos en HTML

### 3 APORTES

#### 3.1 Comparativa de Frameworks HTML5

La tabla resumen las características de los frameworks de HTML5.

<b>Características</b>	<b>Twitter Bootstrap 2.3.1</b>	<b>Foundation 4</b>
Multi-Navegador	Firefox, Chrome, Safari , IE 7+ y Opera	IE 7+, Firefox, Chrome, Safari y Opera 11
Diseño adaptable web	✓	✓
Open Source	✓	✓
Tamaño	140 KB	249 KB
Integración con otros frameworks	✓	x Limitada algunos frameworks
Soporte a versión	✓	✓
Actualización versión	x Problemas al actualizar versión	✓
Comunidad	✓	x No cuenta con una comunidad oficial
Documentación	✓	x Muy poca documentación
Curva de aprendizaje	✓	✓

Tabla 2 comparativa 2: Frameworks HTML 5

### 3.2 Comparativa de Frameworks CSS

La tabla resumen las características de los frameworks CSS.

Características	960 Grid System	Blueprint CSS
Modificación del código	✓	✓
Soporte de SASS	✓	✓
Diseño web adaptable	✓	✓
Diseño móvil	✓	✓
Multi-Navegador	✓	✓
Documentación	✓	✓
Comunidad	x No cuenta con una comunidad oficial	✓
Curva de aprendizaje	x Se deben tener conocimientos previos en CSS	x Se deben tener conocimientos previos en CSS
Integración	✓	✓
Creatividad en el diseño	x El uso de la retícula limita la creatividad	✓
Dimensiones de Pantalla mayores a 1024 Píxeles	x No se adapta a estas resoluciones	✓

Tabla 3 comparativa: Frameworks CSS

### 3.3 Comparativa de Frameworks JavaScript

La tabla resumen las características de los frameworks JavaScript

Características	Backbone.js	Ember.js	Angularjs
Multi-Navegador	IE 6+, Firefox 2+, Chrome, Opera, Safari	IE 9+, FF 11+, Chrome 17+, Safari 5+	IE8 + Chrome, Safari, FireFox
Diseño web adaptable	✓	✓	✓
Open Source	✓	✓	✓
Integración con otras frameworks	✓	✓	
Soporte a versión	✓	✓	✓
Tamaño	51 KB -120 KB	308 KB	81 KB
Estándares HTML5	✓	✓	✓
MVC	✓	✓	✓
Comunidad	✓	X No se cuenta con una comunidad oficial	✓
Documentación	✓	X Muy poca documentación	✓
Curva de aprendizaje	✓	X Curva de aprendizaje abrupta	X Curva de aprendizaje abrupta
Navegador Móvil	✓	✓	✓

Tabla 4 comparativa: Frameworks JavaScript

### 3.4 Funcionamiento del framework de HTML5, CSS y JavaScript.

Las tablas realizadas en el punto anterior nos permitieron comparar las características de los frameworks, obteniendo como resultado los frameworks que aportan más beneficios a los usuarios.

Se seleccionaron los frameworks:

- ✓ Twitter Bootstrap 2.3.1
- ✓ Blueprint CSS
- ✓ Backbone.js

Para estos frameworks se detallara un paso a paso de su funcionamiento.

#### 3.4.1 Twitter Bootstrap 2.3.1

##### a) Descargando Bootstrap

Existen varias formas de empezar con Bootstrap, cada una orientada a un tipo de público en función de su nivel técnico.

- ✓ Descargar el código CSS y JavaScript compilado, que es la forma más sencilla de empezar a utilizar Bootstrap. La desventaja es que esta versión no incluye ni los archivos originales ni la documentación. Para descargar esta versión, accede a [getbootstrap.com](http://getbootstrap.com) y pulsa el botón Download Bootstrap.
- ✓ Descargar el código fuente, contiene todos los archivos Less, y JavaScript originales de Bootstrap. La desventaja es que requiere un compilador de archivos Less y cierto trabajo de configuración. Descarga la versión más reciente en el sitio [www.github.com/twbs/bootstrap/releases](http://www.github.com/twbs/bootstrap/releases).
- ✓ Descargar el código fuente en formato Sass, se trata de una variante de la versión anterior y que se ha creado para facilitar la integración de Bootstrap en las aplicaciones Ruby On Rails, Compass o cualquier otro proyecto

basado en Sass. Descarga la versión más reciente en el sitio [www.github.com/twbs/bootstrap-sass/releases](http://www.github.com/twbs/bootstrap-sass/releases) .

#### b) Utiliza la CDN de Bootstrap

La empresa NetDNA aloja de forma gratuita en su CDN una copia de los archivos CSS y JavaScript de Bootstrap. Para utilizar estos archivos, modifica los siguientes dos enlaces en las páginas (cambia el valor 3.0.0 por la versión específica de Bootstrap que quieras):

```
<!-- Versión compilada y comprimida del CSS de Bootstrap -->
<link rel="stylesheet"
href="//netdna.bootstrapcdn.com/bootstrap/3.1.0/css/bootstrap.min.css">

<!-- Tema opcional -->
<link rel="stylesheet"
href="//netdna.bootstrapcdn.com/bootstrap/3.1.0/css/bootstrap-theme.min.css">

<!-- Versión compilada y comprimida del JavaScript de Bootstrap -->
<script
src="//netdna.bootstrapcdn.com/bootstrap/3.1.0/js/bootstrap.min.js"></script>
```

#### c) Instala Bootstrap con Bower

Utiliza este gestor de dependencias para gestionar los archivos CSS y JavaScript originales de Bootstrap y así mantener una copia local de su documentación. Para ello, ejecuta el siguiente comando:

```
$ Bower install bootstrap
```

#### d) La primera plantilla Bootstrap

Se puede empezar con la plantilla HTML básica que se muestra a continuación. La idea de estos ejemplos es que los utilices como punto de partida.

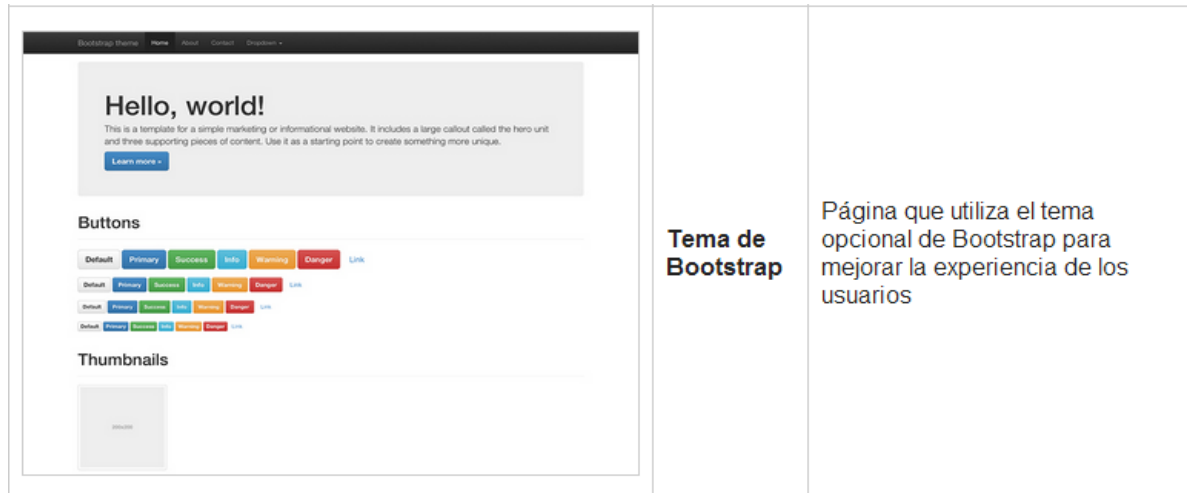
El siguiente código HTML muestra una plantilla muy sencilla creada con Bootstrap:

```
<!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">
    <title>Plantilla básica de Bootstrap</title>
    <!-- CSS de Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
    <!-- librerías opcionales que activan el soporte de HTML5 para IE8 -->
    <!--[if lt IE 9]>
      <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
      <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
    <![endif]>
  </head>
  <body>
    <h1>¡Hola mundo! </h1>
    <!-- Librería jQuery requerida por los plugins de JavaScript -->
    <script src="http://code.jquery.com/jquery.js"></script>
    <!-- Todos los plugins JavaScript de Bootstrap (también puedes
    incluir archivos JavaScript individuales de los únicos
    plugins que utilices) -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```



### e) Ejemplos de plantillas creadas con Bootstrap

Partiendo de la primera plantilla básica mostrada anteriormente se pueden crear muchos otros diseños de página.



### f) Diseñando con rejilla

Preparando la página: antes de comenzar a diseñar el layout o estructura de contenidos de las páginas, es necesario realizar algunos preparativos importantes.

- ✓ Se requiere el doctype de HTML5: Bootstrap utiliza algunos elementos HTML y algunas propiedades CSS que requieren el uso del doctype de HTML5.

```
<!DOCTYPE html>
<html lang="es">...
</html>
```

- ✓ Imágenes responsive: Bootstrap 3 ya no adapta el tamaño de las imágenes automáticamente como sucedía en Bootstrap 2. Para mantener el mismo comportamiento de antes, debes añadir la clase `img-responsive` a cada imagen que quieras que se comporte de manera responsive. Esta clase incluye las propiedades `max-width: 100%;` y `height: auto;` para que la imagen escale en función del tamaño del elemento en el que se encuentra.

```

```

## g) Formularios

Los formularios son uno de los elementos más importante de los sitios y aplicaciones web. Por eso Bootstrap 3 permite diseñar formularios con aspectos muy variados y define decenas de estilos para todos los campos de formulario.

- ✓ Bootstrap 3 aplica por defecto algunos estilos a todos los componentes de los formularios. Además se añade la clase `form-control` a los elementos `<input>`, `<textarea>` y `<select>`.

```
<form role="form">
  <div class="form-group">
    <label for="ejemplo_email_1">Email</label>
    <input type="email" class="form-control" id="ejemplo_email_1"
      placeholder="Introduce el email">
  </div>
  <div class="form-group">
    <label for="ejemplo_password_1">Contraseña</label>
    <input type="password" class="form-control"
id="ejemplo_password_1"
      placeholder="Contraseña">
  </div>
  <div class="form-group">
    <label for="ejemplo_archivo_1">Adjuntar un archivo</label>
    <input type="file" id="ejemplo_archivo_1">
    <p class="help-block">Ejemplo de texto de ayuda.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Activa esta casilla
    </label>
  </div>
  <button type="submit" class="btn btn-default">Enviar</button>
</form>
```

## h) Visualización del formulario

---

### Email

### Contraseña

### Adjuntar un archivo

 Ningún archivo seleccionado

Ejemplo de texto de ayuda.

☐ Activa esta casilla

### 3.4.2 Blueprint CSS

#### a) Componentes de la descarga de Blueprint

- ✓ Directorio Blueprint: Tenemos los archivos del framework ie.css, print.css y screen.css. Esos tres archivos son los que se incluyen. Estos archivos están "comprimidos", es decir, sin comentarios, saltos de línea y otras cosas para que ocupen poco espacio en Kb.
- ✓ Directorio blueprint/src: Son los archivos fuente del framework, pero con el código completo, con los comentarios y formatos para que se puedan leer cómodamente. Es conveniente echar un vistazo a estos archivos, se pueden aprender muchas cosas, sobre cómo está hecho y las clases que han creado para maquetar las páginas. Tiene los siguientes archivos:
  - reset.css: Código para resetear los estilos de todos los navegadores y hacerlos compatibles entre sí.
  - typography.css: definición de tipografía por defecto para la página web, que soluciona algunos problemas cuando se trabaja con fuentes.
  - grid.css: Este es el archivo más importante, con las clases para generar la rejilla, así que merece la pena prestar atención a cómo está hecha y qué clases nos ofrece.
  - forms.css: Código para estilizar formularios y mensajes al usuario.
  - ie.css: con estilos propios para Internet Explorer.
  - print.css: Los estilos que se utilizarán cuando se imprima la página en papel.
- ✓ Directorio blueprint/plugins: Una serie de plugins que funcionan sobre blueprint para diferentes casos de uso.
- ✓ Directorio lib: Con algunos scripts en el lenguaje de programación Ruby para trabajar con Blueprint.

- ✓ Directorio templates: Plantillas para empezar a diseñar en un editor gráfico, con guías para ajustarse al espacio por columnas definido en la rejilla.
- ✓ Directorio test: Contiene ejemplos de páginas creadas para probar las características y posibilidades de Blueprint.

#### b) Incluir los archivos de Blueprint en la página

El primer paso para empezar a usar Blueprint en una página web es incluir los archivos CSS con el código comprimido del framework. Esto lo tenemos que hacer con la conocida etiqueta LINK, enlazada con las hojas de estilo.

```
<link href="css/blueprint/screen.css" type="text/css" rel="stylesheet"
media="screen, projection">
<link href="css/blueprint/print.css" type="text/css" rel="stylesheet"
media="print">
<!--[if lt IE 8]><link rel="stylesheet" href="css/blueprint/ie.css" type="text/css"
media="screen, projection"><![endif]-->
```

El primer archivo screen.css contiene todos los estilos que se aplican cuando la página se visualiza en una pantalla. Como se puede ver, eso se indica en la etiqueta LINK en el atributo media="screen, projection". En ese archivo está la mayoría del código del framework, combinando varios de los archivos de estilos CSS que hay en los archivos.

El segundo archivo que se incluye contiene el código para la versión para imprimir de la página web, con los estilos del framework que facilitarán la lectura de las webs cuando se impriman. Este archivo lo podemos además editar para hacer que no se vean elementos innecesarios en la impresión, por ejemplo con código como este:

```
#cabecera{display: none;}
#navegador{display: none;}
```

El último archivo que se incluye es el ie.css, que además se inserta dentro de un comentario, para que sólo sea visible en la familia de navegadores de Internet Explorer. Contiene, como podremos imaginar, código CSS para solucionar

algunas diferencias de interpretación de los estilos en los navegadores de Microsoft.

En el código HTML anterior para incluir Blueprint sólo tenemos que asegurarnos que la ruta en el atributo src esté dirigida al lugar donde hemos guardado los correspondientes archivos del framework.

c) Contenedor principal de la página Blueprint: Veamos ahora una página básica que usa Blueprint. Para ello primero tenemos que saber que todo el código HTML de nuestra página tenemos que meterlo en un contenedor.

```
<div class="container">  
  <p>Contenido de la página</p>  
</div>
```

Como vemos, para el contenedor principal usamos la clase "container".

d) La parte más importante de este framework CSS es la rejilla, permite situar los elementos de la página de una manera precisa y sencilla.

```
<div class="container">  
  <div class="span-24 last">  
    Cabecera  
  </div>  
  <div class="span-16">  
    Cuerpo izquierda  
  </div>  
  <div class="span-8 last">  
    Lateral derecha  
  </div>  
  <div class="span-24 last">  
    Pie  
  </div>  
</div>
```

#### e) Ejemplo

- ✓ En el HEAD incluimos los archivos con el CSS de blueprint.
- ✓ En el BODY comenzamos con un "container".
- ✓ Colocamos una cabecera, una fila con una única columna de span-24 y last.
- ✓ En el cuerpo dividimos los espacios en dos columnas principales, de span-18 y span-6 (18+6=24).
- ✓ En el cuerpo, dentro del contenedor de span-18, anidamos tres columnas de span-6 (como estamos en un span-18 tenemos espacio para 3 espacios de span-6). Es importante fijarse que el último span-6 tiene también la clase "last", porque termina una fila dentro del span-18.
- ✓ El pie también es span-24 y last

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd"
```

```
>
```

```
<html>
```

```
<head>
```

```
<title>Ejemplo de Blueprint</title>
```

```
<link rel="stylesheet" href="css/blueprint/screen.css" type="text/css"
media="screen, projection">
```

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

```
<!--[if lt IE 8]><link rel="stylesheet" href="css/blueprint/ie.css" type="text/css"
media="screen, projection"><![endif]-->
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="span-24 last">
```

```
<h1>Probando Blueprint</h1>
```

```
</div>
```

```
<div class="span-18">
```

```

<p>Cuerpo...</p>

<div class="span-6">
  Destacado 1
</div>
<div class="span-6">
  Destacado 2
</div>
<div class="span-6 last">
  Destacado 3
</div>
</div>
<div class="span-6 last">
  Lateral
</div>
<div class="span-24 last">
  Pie...
</div>
</div>

</body>
</html>

```

f) Ejemplo:

## Probando Blueprint

Cuerpo...

Lateral

Destacado 1  
Pie...

Destacado 2

Destacado 3



### 3.4.3 Backbone.js

Requisitos básicos: navegador compatible con herramientas de desarrollador para ver qué está pasando en la consola. Cualquier navegador moderno servirá.

#### ✓ Descarga

La única dependencia que tiene Backbone es underscore.js, una librería llena de utilidades y funciones JavaScript de propósito general creada por la misma gente de Backbone, y entre las funcionalidades que ofrece se encuentran la manipulación de arrays, el enlazado de funciones (binding) o templates JavaScript. Para realizar manipulación del DOM y poder gestionar persistencia RESTful completa, necesitas también json2 y jQuery.

#### ✓ Enlazando Backbone

```
<script type="text/javascript" src="jquery/jquery.min.js"></script>
<script type="text/javascript" src="backbone/json2.min.js"></script>
<script type="text/javascript"
src="backbone/underscore.min.js"></script>
<script type="text/javascript" src="backbone/backbone.min.js"></script>
```

#### ✓ Componentes

##### a) Modelo:

Los modelos son los encargados de almacenar los datos de la aplicación, así como proporcionar un conjunto de funcionalidades comunes y conectarlos al sistema de eventos Backbone para notificarnos cuando un modelo ha sido creado, modificado o borrado.

Para crear un modelo Backbone utilizaremos el método Backbone.Model.extend():

```
var Cliente = Backbone.Model.extend ({
  initialize: function (){
    alert('Esta funcion se llamará en la creación de cada instancia')
  }
});
```

```
var cliente = new Cliente();
```

#### b) Vista:

Las vistas en Backbone son clases de control que nos ayudarán a representar nuestros modelos dentro de la interfaz de usuario de nuestra aplicación, detectando eventos de modificación para realizar las actualizaciones correspondientes.

Lo primero que debemos hacer para crear una vista es definirla utilizando el método `extend` de `Backbone.View`:

```
FichaCliente = Backbone.View.extend ({  
  initialize: function(){ /* ... */},  
  render: function(){ /* ... */}  
});  
var ficha = new FichaCliente();
```

#### c) Eventos:

La librería Backbone se apoya en el concepto de evento como método principal de comunicación entre componentes. Cada uno de los elementos de Backbone será capaz de emitir y capturar eventos para conectar los distintos elementos de la aplicación.

Backbone tiene un pequeño módulo de eventos que puede utilizarse para extender cualquier objeto de la aplicación, dotándole de esta forma de la posibilidad de enviar y capturar eventos arbitrarios.

Para que un objeto posea la capacidad de capturar y enviar eventos, debe extender `Backbone.Events` con la función `_.extend` de `underscore.js`:

```
var mi_objeto = {};  
MiObjeto=_.extend(mi_objeto, Backbone.Events);
```

Con la función `bind` podemos enlazar un evento cualquiera con una función callback que se ejecutará cuando este evento ocurra en este objeto

```
mi_objeto.bind("un_evento", function(msg){
```

```
    alert('Vaya, ha ocurrido un_evento con mensaje' + msg);  
  });
```

Un objeto puede disparar un evento en el momento que desee utilizando la función trigger

```
mi_objeto.trigger('un_evento','de_ejemplo');
```

#### d) Colección:

Se puede crear una colección extendiendo el objeto Backbone.Collection donde especificaremos el modelo que asociaremos con él:

```
Cientes = Backbone.Collection.extend(  
  model: Cliente  
);
```

#### e) Router:

Se puede definir un router a través del método extend del objeto Backbone.Router. En él especificaremos el objeto hash routes que contendrá todas las rutas que se atenderán, debiendo contener al menos una. En este objeto hash introduciremos la ruta como key y una función asociada como valor que será la que se ejecutará cuando estemos en dicha ruta.

```
Enrutador = Backbone.Router.extend({  
  routes:{  
    "" : "index"  
    "help" : "ayuda",  
    "tag/:tagid" : "muestraEtiqueta"  
    "tag/:tagid/p:page" : "muestraEtiqueta"  
    "download/*file" : "descargar"  
  },  
  index: function() { /* ... */},  
  ayuda: function(){ /* ... */},  
  muestraEtiqueta: function(tagid, page){ /* ... */},  
  descargar: function(file){ /* ... */}  
});
```

### 3.5 Aplicación Web con los 3 frameworks integrados

Con la información recopilada de los frameworks se construyó una aplicación integrando los frameworks:

- ✓ Twitter Bootstrap 2.3.1
- ✓ Blueprint CSS
- ✓ Backbone.js

La página se encuentra almacenada en un servidor gratuito, la disponibilidad de la página en el mismo está sujeta a las condiciones de la empresa que brinda este servicio.

URL: <http://frameworksweb.hostei.com/#>



El proceso de integración de los frameworks se realizó descargando primero el framework de backbone.js, luego desde las vistas se asoció el framework Twitter Bootstrap para posteriormente adicionar el archivo css los componentes de Blueprint CSS.

## CONCLUSIONES

La elección de un framework de desarrollo web está sujeta, a las necesidades de la aplicación debido a que los frameworks se especializan en un tipo de aplicación. Identificar las características de la aplicación permite explotar todas las ventajas que el framework adecuado ofrece.

El desarrollador o diseñador debe hacer una autoevaluación para identificar su conocimiento, debido a que algunos frameworks tiene una curva de aprendizaje compleja y requieren de conocimientos previos en otros lenguajes.

Para este proyecto la elección del framework se fundamentó en las características del framework y en la construcción de una aplicación básica.

No existe un framework que permita desarrollar todo tipo de aplicaciones, debido a que esta decisión está sujeta a variables como el costo, las limitaciones y las facilidades que ofrece.

Los frameworks están en constante evolución debido a las nuevas tendencias de la web, es de gran importancia seleccionar un framework actualizado.

El proyecto deja abierta la posibilidad de que se realice un nuevo proyecto, debido a que solo fueron estudiados algunos de los frameworks, posteriormente se podrían incluir los que no se consideraron.

## 4 BIBLIOGRAFÍA

- [1] Javier J. Gutiérrez. ¿Qué es un framework? [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf) [Citado el 27 de Febrero de 2014].
- [2] Pedro Rufo Martín. HTML < <http://www.asptutor.com/zip/cbhtml.pdf> > [Citado el 27 de Febrero de 2014].
- [3] Víctor Martínez. HTML5: ¿Qué es? Y ¿Cómo usarlo? <<http://www.trazos-web.com/2010/02/01/html5-que-es-y-como-usarlo/>> [Citado el 27 de Febrero de 2014].
- [4] W3C España. Guía Breve de CSS <<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>> [Citado el 6 de Marzo de 2014].
- [5] UNIVERSIDAD DEL AZUAY. Manual de JavaScript <[http://www.uazuay.edu.ec/estudios/sistemas/lenguaje\\_iii/MAnualJavaScript/introduccion.htm](http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MAnualJavaScript/introduccion.htm)> [Citado el 6 de Marzo de 2014].
- [6] EcuRed. Aplicación web <[http://www.ecured.cu/index.php/Aplicaci%C3%B3n\\_web](http://www.ecured.cu/index.php/Aplicaci%C3%B3n_web)> [Citado el 6 de Marzo de 2014].
- [7] BaseKit. Diferencias entre un desarrollador y un diseñador web <<http://www.basekit.com.mx/basekitema/diferencias-entre-un-desarrollador-y-un-disenador-web>> [Citado el 6 de Marzo de 2014].
- [8] Gerencie. Trabajo Freelance <<http://www.gerencie.com/trabajo-freelance.html>> [Citado el 6 de Marzo de 2014].
- [9] Rene Camacho. Interfaz de programación de aplicaciones (API) <<http://rcmdispmoviles.blogspot.com/2012/04/interfaz-de-programacion-de.html>> [Citado el 6 de Marzo de 2014].

- [10] ISKANDAR MORINE Ricardo José. Estudio comparativo de alternativas y frameworks de programación, para el desarrollo de aplicaciones móviles en entorno Android. Barcelona, 2013, 127 págs. Trabajo de grado (Ingeniero de Telecomunicaciones). Universitat Politècnica de Catalunya. Disponible en el catalogo en línea: <  
[http://upcommons.upc.edu/pfc/bitstream/2099.1/18249/5/Proyecto\\_Final\\_de\\_Carrera\\_UPC\\_-\\_Ricardo\\_Iskandar.pdf](http://upcommons.upc.edu/pfc/bitstream/2099.1/18249/5/Proyecto_Final_de_Carrera_UPC_-_Ricardo_Iskandar.pdf) >.
- [11] RENGIFO BRÍÑEZ Johanna Fernanda, BETANCOURT PÉREZ Carlos Alberto. Frameworks y herramientas para el desarrollo de aplicaciones orientadas a dispositivos móviles. Pereira, 2011, 48 págs. Trabajo de grado (Ingeniero de Sistemas y Computación). Universidad Tecnológica de Pereira. Disponible en el catalogo en línea:<  
<http://repositorio.utp.edu.co/dspace/bitstream/11059/2490/1/00416R412.pdf> >.
- [12] HERNÁNDEZ CRUZ Ángel. Análisis comparativo de frameworks para el desarrollo de aplicaciones web en java. Ixtlán de Juárez, 2013, 197 págs. Trabajo de grado (Licenciado en Informática). Universidad de la Sierra Juárez. Disponible en el catalogo en línea:<  
<http://www.unsij.edu.mx/tesis/digitales/tesis%20Angel%20Hernandez%20Cruz.pdf> >.
- [13] ARTAL LOZANO Juan Antonio, Estudio comparativo de frameworks de instrumentación dinámica de ejecutables. 2012, 92 págs. Trabajo de grado (Ingeniero en Informática). Universidad Zaragoza. Disponible en el catalogo en línea:<  
[http://webdiis.unizar.es/~ricardo/files/PFCs/Estudio.Frameworks.DBI/Memoria\\_PFC\\_EstudioDBI.pdf](http://webdiis.unizar.es/~ricardo/files/PFCs/Estudio.Frameworks.DBI/Memoria_PFC_EstudioDBI.pdf)>.
- [14] Rafael Barzanallana. Universidad de Murcia. Servicios en internet Historia del desarrollo de aplicaciones Web. <  
<http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html> > [Citado el 9 de Marzo de 2014].
- [15] Enrique Buono .Los patrones de diseño son el Origen de los Framework?<  
[http://kuainasi.ciens.ucv.ve/red\\_educativa/blogs/43?language\\_id=1](http://kuainasi.ciens.ucv.ve/red_educativa/blogs/43?language_id=1)> [Citado el 9 de Marzo de 2014].

- [16] Álvaro Fontela. ¿Qué es BootStrap? < <http://openwebcms.es/2013/que-es-bootstrap/> > [Citado el 13 de Marzo de 2014].
- [17] Templatemonster blog. ¿Bootstrap o Foundation? ¿Cuál HTML5 Framework es más Sensible para el Desarrollo de Sitios Web? <<http://www.templatemonsterblog.es/2013/06/05/bootstrap-o-foundation-cual-html5-framework-es-mas-sensible-para-el-desarrollo-de-sitios-web/>> [Citado el 13 de Marzo de 2014].
- [18] Patricio. Desarrollando Webs Dinámicas <<http://desarrollandowebsdinamicas.blogspot.com/2014/02/que-es-twitter-bootstrap.html>> [Citado el 13 de Marzo de 2014].
- [19] Bruno García Echegaray. ¿Qué es Bootstrap? <<http://bruno.garciaecheagaray.com/Bootstrap.The.Melee/presentation/#more>> [Citado el 13 de Marzo de 2014].
- [20] Alan Chávez. Si eres desarrollador web, debes utilizar Bootstrap y punto <<http://alanchavez.com/si-eres-desarrollador-web-debes-utilizar-bootstrap-y-punto/>> [Citado el 15 de Marzo de 2014].
- [21] Jacob Gube. 10 Best Responsive HTML5 Frameworks and Tools <<http://designinstruct.com/roundups/html5-frameworks/>> [Citado el 19 de Marzo de 2014].
- [22] GNU GPLv2. lineadecodigo <[https://code.google.com/p/lineadecodigo/source/browse/trunk/lineadecodigo\\_bootstrap/WebContent/basicos/holamundo-bootstrap.html](https://code.google.com/p/lineadecodigo/source/browse/trunk/lineadecodigo_bootstrap/WebContent/basicos/holamundo-bootstrap.html)> [Citado el 19 de Marzo de 2014].
- [23] Enrique Serrano. Framework de Diseño Web Adaptativo Foundation 4 <<http://webadaptativo.blogspot.com/2013/03/framework-de-diseno-web-adaptativo.html>> [Citado el 25 de Marzo de 2014].
- [24] Roger Romero. Frameworks de CSS para el desarrollo web <<http://www.regoremor.com/desarrollo-web/css/frameworks-de-css-para-el-desarrollo-web/>> [Citado el 29 de Marzo de 2014].



- [25] Sauvage. ¿Cuáles son los Frameworks? Mejores 22 Responsive frameworks CSS para el diseño web <<http://sauvagedesign.com/es/cuales-son-los-frameworks-mejores-22-responsive-frameworks-css-para-el-diseno-web/>> [Citado el 29 de Marzo de 2014].
- [26] Jepser Bernardino. 960 grid, un framework para CSS <<http://jepserbernardino.com/idea/960-grid-un-framework-para-css/>> [Citado el 29 de Marzo de 2014].
- [27] Nathan Smith. 960 Grid System <[http://wiki.ead.pucv.cl/index.php/960\\_GRID\\_SYSTEM](http://wiki.ead.pucv.cl/index.php/960_GRID_SYSTEM)> [Citado el 29 de Marzo de 2014].
- [28] Marco Sánchez. 960 Grid System [http://observatoriotecedu.uned.ac.cr/media/960\\_grid.pdf](http://observatoriotecedu.uned.ac.cr/media/960_grid.pdf) [Citado el 29 de Marzo de 2014].
- [29] Nacho Rodríguez. 960 Grid System, un framework para CSS <<http://www.genbetadev.com/desarrollo-web/960-grid-system-un-framework-para-css>> [Citado el 30 de Marzo de 2014].
- [30] Miguel Angel Álvarez. Blueprint, Framework CSS <<http://www.desarrolloweb.com/articulos/blueprint-frameworks-css.html>> [Citado el 30 de Marzo de 2014].
- [31] Cristiano Montoya. Blueprint <<http://www.blueprintcss.org/>> [Citado el 2 de Mayo de 2014].
- [32] Andrés Bayona. Taller de maquetación web. <<http://www.slideshare.net/webmastergranada/taller-maquetacion-web>> [Citado el 2 de Mayo de 2014].
- [33] Ganarce. Blueprint CSS framework <<http://www.slideshare.net/ganarce/blueprint-css-framework>> [Citado el 2 de Mayo de 2014].
- [34] Carlos Santana Roldán. Los 5 mejores frameworks MVC de JavaScript <<http://www.codejobs.biz/es/blog/2013/04/11/los-5-mejores-frameworks-mvc-de-javascript#sthash.5gvsiSlx.dpbs>> [Citado el 4 de Mayo de 2014].

- [35] Oscar Andrés Granada Baquero. Tutorial de Backbone.js (I) <<http://notoquesmicodigo.blogspot.com/2014/03/tutorial-de-backbonejs-i.html>> [Citado el 4 de Mayo de 2014].
- [36] Erick Ruiz de Chavez. Qué es BackboneJS <<http://www.desarrolloweb.com/articulos/que-es-backbonejs.html>> [Citado el 4 de Mayo de 2014].
- [37] Damian Serrano Thode. Crear una aplicación sencilla con Backbone.js <<http://dsthode.info/2012/09/crear-una-aplicacion-sencilla-con-backbone-js/>> [Citado el 4 de Mayo de 2014].
- [38] NitsnetsBackbone vs Angular vs Ember <<http://www.lostiemposcambian.com/blog/javascript/backbone-vs-angular-vs-ember/>> [Citado el 4 de Mayo de 2014].
- [39] Mariano Dimaggio. Backbone.js: otro enfoque para desarrollar aplicaciones en JavaScript <<http://www.4rsoluciones.com/backbone-js-otro-enfoque-para-desarrollar-aplicaciones-en-javascript/>> [Citado el 4 de Mayo de 2014].
- [40] Carlos Picca. Ember.js desde Cero: Introducción e Instalación <<http://codehero.co/ember-js-desde-cero-introduccion-e-instalacion/>> [Citado el 5 de Mayo de 2014].
- [41] Introducción al framework Ember.js <<http://www.lostiemposcambian.com/blog/javascript/intro-framework-ember/>> [Citado el 5 de Mayo de 2014].
- [42] Daniel Díaz Suarez. Nuestra Primera App con Ember.js <<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=PrimeraAplicacionEmber>> [Citado el 5 de Mayo de 2014].
- [43] Nitsnets. AngularJS. Framework JavaScript para Webapps <<http://www.lostiemposcambian.com/blog/javascript/angularjs-framework-javascript-para-webapps/>> [Citado el 5 de Mayo de 2014].
- [44] Activ. Comenzando con AngularJS <<http://activ.com.mx/comenzando-con-angularjs/>> [Citado el 5 de Mayo de 2014].

- [45] Brian Serrano. AngularJs vs. BackboneJs<<http://foreground.com.co/blog/index.php/component/content/article/79-desarrollo-web/73-angularjs-vs-backbonejs.html>> [Citado el 5 de Mayo de 2014].