



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

**FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN**

Adaptación del OpenSource Bootstrap basados en Ingeniería Web.

**TESIS PROFESIONAL
para obtener el título de
INGENIERO EN CIENCIAS DE LA
COMPUTACIÓN**

**PRESENTA
Roberto Fabian Morales Rodriguez**

**ASESOR:
M.C. Mariano Larios Gómez**

Ciudad Universitaria, Puebla, Abril 2016

Agradecimientos:

A mi Madre

Por ser la única que logra entender mis ideales de juventud, que sabe que quiero vivir y que está aceptando mis preferencias, que es mi cómplice en todo momento, a ella le agradezco su sacrificio.

A mis abuelos.

Por apoyarme, quererme y nunca perder la fe en mí, por ser cómplices de mis locuras y por brindarme en todo momento apoyo económico.

A mis amigos.

A ustedes que me ayudaron a olvidar los malos momentos y el estrés que cargaba con sus risas y palabras de apoyo, no importa si los tenía cerca o lejos de mí siempre supe que estaban ahí y a ustedes dedico este pequeño logro: Irvin, Poncho, Pato, Beto, Urias, Michel, Ariel, Lalo, Jesús, Xalo y demás amigos que no recordé al momento de escribir esto. Ustedes saben quiénes son.

Un poco más de persistencia, un poco más de esfuerzo, y lo que parecía irremediablemente un fracaso puede convertirse en un éxito glorioso.

Elbert Hubbard.

Índice general

AGRADECIMIENTOS:	II
CITA:	III
ÍNDICE GENERAL	IV
UNIDAD 1	1
INTRODUCCIÓN	1
1.1. ANTECEDENTES	1
1.2. DESCRIPCIÓN DEL PROBLEMA	2
1.3. METODOLOGÍA DE DESARROLLO	4
1.4. OBJETIVO DEL PROYECTO	6
1.5 RESULTADOS ESPERADOS	7
UNIDAD 2	9
MARCO TEÓRICO	9
2.1. INGENIERÍA WEB	9
2.2. HTML 5	12
2.3. CSS3	13
2.4. JAVASCRIPT	14
2.5. FRAMEWORK BOOTSTRAP	15
UNIDAD 3	17
FORMULACIÓN, PLANEACIÓN Y ANÁLISIS	17
3.1. PREGUNTAS DE FORMULACIÓN	18
3.2. RECOPIACIÓN DE REQUISITOS	21
3.3. PLANEACIÓN DEL PROYECTO	26

3.4 ANÁLISIS DE REQUISITOS	30
3.5. ANÁLISIS DE CONTENIDO	31
3.6. ANÁLISIS DE INTERACCIÓN	37
3.7. ANÁLISIS FUNCIONAL	38
3.8. ANÁLISIS DE CONFIGURACIÓN	42
UNIDAD 4	45
MODELIZACIÓN Y GENERACIÓN DE PÁGINAS.	45
4.1. DISEÑO.	46
4.2. PRODUCCIÓN DE CONTENIDO.	47
4.3. DISEÑO ARQUITECTÓNICO.	54
4.4. DISEÑO DE NAVEGACIÓN.	72
4.5. DISEÑO DE INTERFAZ Y GENERACIÓN DE PÁGINAS.	74
UNIDAD 5	80
TEST Y EVALUACIÓN.	80
5.1. MÉTODOS DE PRUEBA	81
CONCLUSIONES	84
BIBLIOGRAFÍA	85

Introducción

1.1. Antecedentes

Este proyecto de tesis comenzó con la idea de dar a conocer los productos y servicios de empresas que vieron Internet como una buena manera de difundir su marca; con aplicaciones que tengan una mejor interfaz humano-computadora se pretende cumplir con este objetivo.

El antecedente de este proyecto de tesis está basado como convenio de una consultoría de software llamada CEO-Consulting y la Facultad de Ciencias de la Computación a cargo del asesor de tesis. En primera instancia, la consultoría demandó un recurso humano para desarrollar, modelar e implementar un conjunto de aplicaciones computacionales capaces de ofrecer información de productos y servicios a diferentes usuarios.

El convenio o aportación fue el desarrollar una modificación y mejora a nuestro Framework Bootstrap, así también, de realizar una fuerte y robusta documentación de la metodología de software enfocada a web, es decir, que el proyecto esté basado sobre la ingeniería web.

1.2. Descripción del problema

Un grupo de organizaciones deseaban una aplicación que permitiera presentar sus productos a todos los usuarios que visitarán su interfaz a través de la web y en cualquier dispositivo que tuviera acceso a la red. Se tenía que resolver también el problema de crear interfaces que fueran agradables y adaptadas al tipo de usuario al que se quería llegar.

Con ayuda de la ingeniería web, se realizó un caso de uso de una aplicación para dispositivos móviles o computadoras de escritorio, que permitió dar a conocer productos y servicios requeridos por usuarios y administradores del sistema, tales medios pueden ser internet, interfaces remotas y aplicaciones móviles. Este último es hoy en día cada vez más popular como herramienta de difusión de la información, con interfaz amigable, agradable y sobre todo de fácil uso para cualquier usuario.

Otra de las características importantes que se obtuvo en este proyecto es que sea responsivo, es decir, que se pueda manejar en cualquier dispositivo móvil, así como en cualquier computadora de escritorio o laptop; sin perder la característica de facilidad de navegación.

El alcance de éste proyecto de tesis se centra en la modificación de plantillas a bajo nivel en un Framework OpenSource, además de permitirnos adecuar y mejorar futuras aplicaciones de interfaz humano-computadora, es decir que mejora la parte amigable de la aplicación web, y que nos permite informar de productos o servicios que ofrecen cada una de los dispositivos intercomunicados.

Esta modificación requiere de un uso experto del Framework Bootstrap antes mencionado, puesto que se realizaron patrones de diseño para modelar las mejoras en las plantillas y sobre todo la forma precisa de la comunicación entre todas las clases adecuadas en el Framework.

Este trabajo es sostenido con la ingeniería web, que nos permitió comprobar las mejoras en análisis y diseño, basándonos en un caso de uso específico, de esta forma se obtuvo los resultados esperados.

1.3. Metodología de desarrollo

En este proyecto de investigación de tesis utilizamos la ingeniería web por ser el método formal de organizar y elaborar un trabajo de desarrollo de aplicaciones web con buena calidad y con una documentación que lo respalde para futuros cambios o para el mantenimiento de la misma aplicación.

Utilizamos esta ingeniería por la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web. En este sentido, la ingeniería Web hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones Web complejas y de grandes dimensiones en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones. En este proyecto se utilizó las siguientes fases basándonos en la ingeniería web:

- ✓ Análisis. En ésta primera etapa se realizó o contempló el objetivo primordial de recopilar la información que se desea dar a conocer de cada WebApp.
- ✓ Planificación. Establecer tiempos de diseño, implementación y entrega por cada aplicación web.
- ✓ Formulación. Se realizaron metas de avance para la entrega de diseños, y mapa de navegación. Procesar e incorporar la información a mostrar con la plantilla Bootstrap.
- ✓ Modelización. En este punto se trabajó paralelamente con la estructura de la aplicación y la producción del contenido obtenido del análisis realizado.
- ✓ Generación de páginas. Se integró ambas partes: estructura y contenido, formalizando nuestra estructura HTML y CSS3 en conjunto con Bootstrap.
- ✓ Test. Se realizó pruebas de navegación básicas para correcciones menores de interfaz y ortografía de contenido.

- ✓ Evaluación del cliente. Se presentó la página web para su prueba final y aceptación por parte del supervisor del proyecto y así tener aprobación para subirlo a su respectivo dominio.

1.4. Objetivo del proyecto

Objetivo general:

- ✓ La modificación de un OpenSource a bajo nivel para la mejora de su interfaz humano-computadora, dando a conocer los productos y servicios que tiene cada una de las aplicaciones móviles intercomunicadas. Logrando una usabilidad adecuada para la intercomunicación de dispositivos móviles y de esta forma modelar e implementar las mejoras en las plantillas del Framework Bootstrap, respaldando el proyecto de tesis con la ingeniería web.

Objetivos específicos:

- ✓ Diseñar la interfaz de cada uno de los componentes de las clases de Bootstrap.
- ✓ Usar el lenguaje de programación HTML5, CSS3 para la modificación de las plantillas.
- ✓ Aplicar las nuevas plantillas del Framework Bootstrap para volver vistoso, amigable e interactiva la navegación de la Web App entre dispositivos interconectados.
- ✓ Llevar la aplicación a todos los dispositivos que se pueda con la mejor interfaz, es decir que sea responsivos.
- ✓ Realizar las etapas de desarrollo de aplicaciones que requiere la ingeniería web.

1.5. Resultados esperados.

Al finalizar el proyecto se terminaron cinco aplicaciones web (WebApp) funcionales que fueron Ceo Consulting, Genios, Señor Tlacuache y Cervecera Apan y que cumplen con los estándares básicos que marca la ingeniería Web que permitirán dar a conocer cada una de las actividades, productos y servicios que ofrecen las empresas para iniciar con la obtención de clientes. Los usuarios finales podrán informarse, conocer y tener contacto con la empresa para poder aclarar dudas, revisar productos a fondo e informarse sobre la ubicación de sucursales. Otro objetivo esperado es proporcionar una Web App con las mejores interfaces responsivas que ayuden a los usuarios a obtener la información que necesiten desde cualquier medio o dispositivo móvil con acceso a internet. Así también, se desea obtener como resultado el modelado y documentación más completa del proyecto bajo la metodología de ingeniería web.

La principal aportación en este proyecto de investigación de tesis es una documentación completa y robusta bajo la ingeniería web la cual contemple los puntos fundamentales, es decir, análisis, planificación, modelización, formulación, desarrollo de páginas web, test y evaluación con el cliente. Así también como un proceso de software básico como el análisis, modelado e implementación de plantillas web enfocado al área empresarial y a su vez se empleará de forma pedagógica para tener una guía en el uso de estas metodologías y herramientas que permiten el desarrollo de proyectos en la web.

El principal sector beneficiado será la parte laboral que tenga cada empresa que contará con la aplicación Web. Las ventas de producto y difusión de estas empresas mexicanas de diferentes actividades comerciales tendrán también beneficios a mediano plazo para su crecimiento, esto traerá un aumento económico para trabajadores.

Los consumidores mexicanos e internacionales que visiten los sitios mejorarán su experiencia de compra y podrán conocer el producto sin necesidad de viajar.

Esta tesis será de ayuda para todos los estudiantes que deseen profundizar en el tema de Ingeniería Web enfocados a la publicidad empresarial y por otro lado las personas que deseen conocer nuevos métodos de desarrollo utilizando como medio la web también podrán contar con este tema como guía para el desarrollo de sus proyectos.

Marco Teórico.

2.1. Ingeniería web.

En 1998 Roger Pressman moderó una mesa redonda virtual con representantes de la ingeniería de software tradicional y del desarrollo de software basado exclusivamente en Internet. La conclusión a la que se llegó fue que aplicar metodologías de ingeniería no tendría un efecto negativo si no por el contrario si se adaptaban los procesos de ingeniería se podría lograr aplicaciones web de mayor usabilidad (Gaedke, s.f.).

La Ingeniería Web (IWeb) aplica “sólidos principios científicos, de ingeniería y de administración, y enfoques disciplinados y sistemáticos para el desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basados en Web de alta calidad”.

Pressman enumera siete actividades que forman parte del proceso de la IWeb y que son aplicables a cualquier servicio web independientemente de su tamaño y complejidad. Estas actividades son:

1. La Formulación identifica objetivos y establece el alcance de la primera entrega.
2. La Planificación genera la estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.
3. El Análisis especifica los requerimientos e identifica el contenido.

4. La Modelización se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.

5. En la Generación de Páginas se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de la aplicación: las páginas.

6. El Test busca errores a todos los niveles de las siguientes formas: de contenido, funcionales, de navegación, de rendimiento, entre otras. El hecho de que las aplicaciones residan en la red, y que inter-operen en plataformas muy distintas, hace que el proceso de test sea especialmente difícil.

7. Finalmente, el resultado es sometido a la Evaluación del Cliente ya sea para resultados previos o resultados finales.

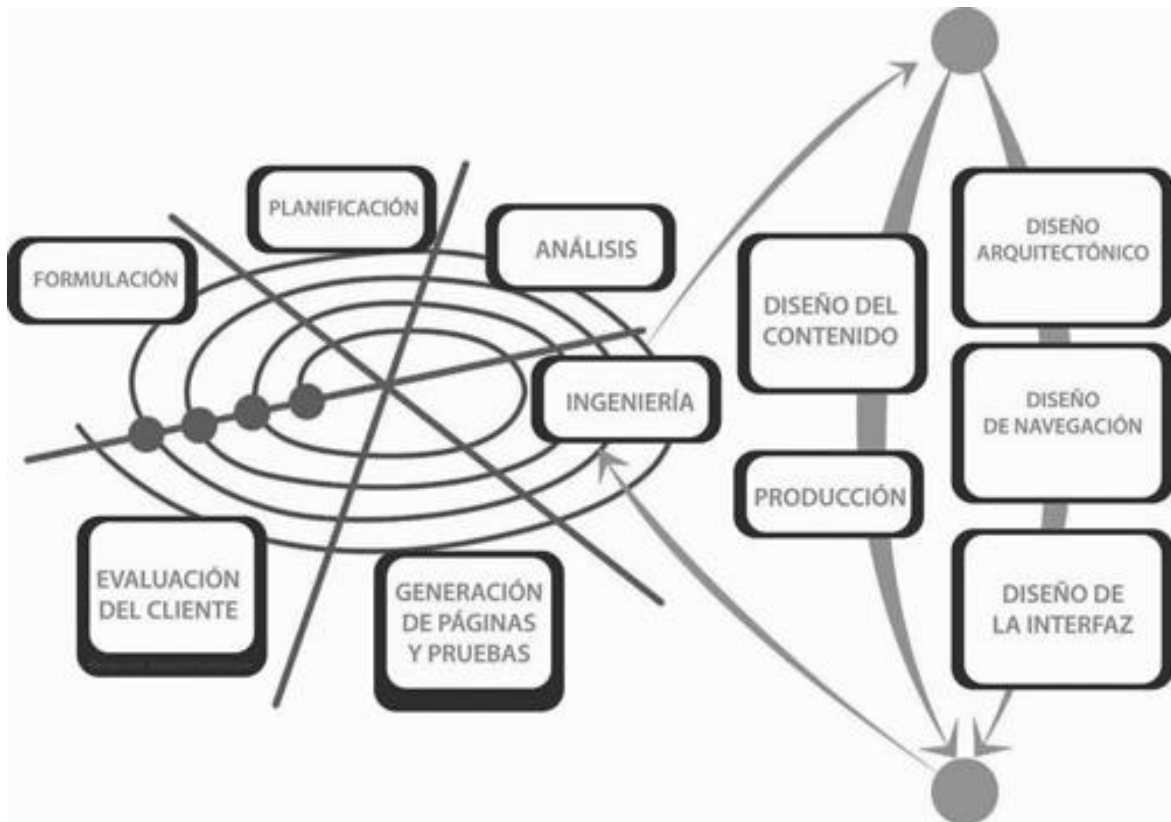


Figura 1. Modelo de proceso de la IWEB 1

En la figura 1, se representa de forma conceptual la ingeniería del software tradicional, también se representa como se manejan actividades, tales como la creación de estándares, análisis, registro de informes, etc. En la ingeniería web estos conceptos son tomados en cuenta para valorar la calidad de servicios sobre las Web App. (Pressman, 2004)

2.2. HTML5.

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

Todo comenzó mucho tiempo atrás con una simple versión de HTML propuesta para crear la estructura básica de páginas web, organizar su contenido y compartir información. El lenguaje y la web misma nacieron principalmente con la intención de comunicar información por medio de texto.

El limitado objetivo de HTML motivó a varias compañías a desarrollar nuevos lenguajes y programas para agregar características a la web nunca antes implementadas. Estos desarrollos iniciales crecieron hasta convertirse en populares y poderosos accesorios. Simples juegos y bromas animadas pronto se transformaron en sofisticadas aplicaciones, ofreciendo nuevas experiencias que cambiaron el concepto de la web para siempre. (Gauchat, 2012)

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas API's (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5.

2.3. CSS 3.

CSS no es parte de la especificación y nunca lo fue. Este lenguaje es, de hecho, un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. Al comienzo, atributos dentro de las etiquetas HTML proveían estilos esenciales para cada elemento, pero a medida que el lenguaje evolucionó, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo más satisfacer las demandas de diseñadores. En consecuencia, CSS pronto fue adoptado como la forma de separar la estructura de la presentación. Desde entonces, CSS ha crecido y ganado importancia, pero siempre desarrollado en paralelo, enfocado en las necesidades de los diseñadores y apartado del proceso de evolución de HTML.

La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web y esta es la razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a CSS3, aunque oficialmente se trate de dos tecnologías completamente separadas. (Gauchat, 2012)

El principal objetivo de CSS 3 fue el proveer la parte gráfica, la parte de estilo; colores, fondo, formas, posiciones, etc. Pero con su evolución no solo nos proporciona esa parte del estilo si no también la forma y el movimiento de la presentación de la página. Todo esto para hacer que la estructura obtenida por HTML sea la más llamativa.

2.4. JavaScript.

JavaScript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web.

JavaScript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (API's) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas API's (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores. La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web. (Gauchat, 2012)

La función de JavaScript en nuestro proyecto es completar el concepto de HTML5 con la parte de movimiento. Para brindar un servicio web con alta usabilidad, la manera en que se navegará por cada uno de los sitios será presentada de distinta manera con efectos que atraigan al cliente a repetir una y otra vez el recorrido por todo el contenido de la página.

2.5. Framework Bootstrap.

2.5.1. 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 ambientes de trabajos para lenguajes como HTML5, CSS y JavaScript. (Gutiérrez., 2014)

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. (Gutiérrez., 2014)

2.5.2. Framework Bootstrap.

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 (Diseño Web Responsivo). (Frain, 2012)

El Framework trae varios elementos con estilos predefinidos fáciles de configurar e integración jQuery para ofrecer ventanas y tooltips dinámicos.

Bootstrap maneja un desarrollo web responsivo, tiene un diseño para que el usuario pueda visualizar el contenido en un amplio rango de dispositivos. Bootstrap se ha rediseñado para adaptarse desde un principio a distintos tamaños de pantalla comunes de los móviles.

Las características principales de este ambiente de trabajo son las que se enumeran a continuación:

- ✓ Ofrece Grids fijados y líquidos – 724px, 940px, 1170px
- ✓ Sistema de parrilla de 12 columnas.
- ✓ Compatible con los buscadores o browsers Chrome, Firefox, Safari, IE 7 y Opera
- ✓ Compatibilidad con tabletas, teléfonos inteligentes entre otros dispositivos móviles.
- ✓ Se utiliza lenguajes orientados a Web como los son: HTML5 y CSS3.
- ✓ Se puede agregar o integrar librerías JavaScript y Less¹.
- ✓ Tiene la característica de ser OpenSource. (Mark Otto, 2014)

¹ Es una pequeña y gran herramienta que extiende las posibilidades del CSS añadiendo variables, mezclas, operaciones y reglas anidadas. Esto significa que con **LESS** podemos escribir código ligero y óptimo de forma muy rápida.

Formulación, Planeación y Análisis.

La formulación de sistemas y aplicaciones basados en Web representan una sucesión de actividades de ingeniería Web que comienza con la identificación de metas globales para la Web App (Aplicación web). La formulación permite que el cliente o diseñador establezca un conjunto común de metas y objetivos para la construcción de la Web App. También, identifica el ámbito de esfuerzo en el desarrollo y proporciona un medio para determinar un resultado satisfactorio.

Powell (Powell, 1998), sugiere una serie de preguntas que se deben realizar y responderse al comienzo de la etapa de formulación:

- ✓ ¿Cuál es la motivación principal para la WebApp?
- ✓ ¿Por qué es necesaria la WebApp?
- ✓ ¿Quién va a utilizar la WebApp?

Las respuestas que se establezcan implican metas específicas para el sitio Web. En general, para cada modificación de nuestras plantillas tendremos dos categorías:

Metas informativas: indican la intención de proporcionar el contenido y/o información específicos para el usuario final.

Metas aplicables: indican la habilidad de realizar algunas tareas dentro del portal web.

Aunque se desea llegar a diferentes tipos de usuarios en cada modificación de la plantilla las metas no cambian mucho puesto que todas llevan un desarrollo para el ámbito empresarial y nuestro Framework permite en su código manipulaciones detalladas.

3.1. Preguntas de formulación.

En este punto se describen las metas generales, informativas y aplicables del conjunto de Web Apps que se crearon. El código de nuestra plantilla web tendría que satisfacer a distintos usuarios, así que para tener un primer panorama de lo que trabajaremos, como lo sugiere la ingeniería web lo más práctico será hacernos las mismas preguntas en cada caso particular

✓ CEO CONSULTING

¿Cuál es la motivación principal para la WebApp?

La empresa se dará a conocer en la red con sitios atractivos y agradables al usuario a través de todos los dispositivos posibles.

¿Por qué es necesaria la WebApp?

Permitirá que el usuario conozca la empresa, sus productos y sus servicios desde cualquier medio conectado a Internet evitando costes de viaje y mejorando el margen de beneficios para usuarios.

¿Quién va a utilizar la WebApp?

Los usuarios potenciales serán todos los que cuenten con un dispositivo electrónico con acceso a Internet como una Tablet, celular, laptop o PC y que desee informarse sobre el producto que se oferta en el área de negocios.

✓ GENIOS

¿Cuál es la motivación principal para la WebApp?

Que los usuarios tengan información de las empresas que ha creado el grupo Genios. Además que se pueda dar a conocer en la web el grupo.

¿Por qué es necesaria la WebApp?

Permitirá conocer los servicios desde cualquier medio evitando costes de viaje y mejorando el margen de beneficios para usuarios.

¿Quién va a utilizar la WebApp?

Usuarios potenciales interesados en formar parte del grupo Genios que cuenten con un dispositivo electrónico con acceso a Internet y que desee informarse sobre el producto que se oferta.

✓ SEÑOR TLACUACHE

¿Cuál es la motivación principal para la WebApp?

Nuestra motivación es dar a conocernos en la web como un bazar de compra venta de artículos vintage o antiguos para poder iniciar con la captura de usuarios que compren y vendan.

¿Por qué es necesaria la WebApp?

Permitirá conocer la empresa, sus productos y sus servicios desde cualquier medio, dará a conocer los artículos que se tienen en venta y será un medio para que los usuarios puedan vender sus artículos con antigüedad evitando costes de viaje y mejorando el margen de beneficios para usuarios.

También se obtendrá mayor posición en el mercado con la ayuda del Framework Bootstrap que ayudara a la atracción de usuarios a través de sus atractivas plantillas.

¿Quién va a utilizar la WebApp?

Todo usuario que desee comprar o vender algún producto antiguo. Nuestros usuarios potenciales serán todos los que cuenten con un dispositivo electrónico con acceso a Internet y que desee informarse sobre el producto que se oferta.

✓ **CERVECERA APAN**

¿Cuál es la motivación principal para la WebApp?

Tener un WebApp que permita dar a conocer los, servicios y la ubicación de las instalaciones de la empresa para expandir sus ventas a través de la web.

¿Por qué es necesaria la WebApp?

Se necesita una aplicación en web que permita ofrecer un catálogo virtual de los productos con los que se cuenta. A los usuarios consumidores de esta clase de bebidas ofrecerá una manera interesante de ver su producto preferido

¿Quién va a utilizar la WebApp?

Nuestros usuarios que deseen informarse sobre el producto que se oferta, la manera de llegar y aquellos posibles compradores del producto.

3.2 Recopilación de requisitos.

Tenemos que tener un panorama general de lo que realizaremos. La recopilación de requisitos propuestos por la ingeniería del software se conserva también para el desarrollo web y estos se adaptan a la siguiente manera:

- ✓ Identificar requisitos funcionales.

Son declaraciones de los servicios que proveerá la aplicación web, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requerimientos funcionales de las Web App también declaran explícitamente lo que el sistema no debe hacer. En esta parte también se toman los requisitos de contenido de la aplicación.

- ✓ Identificar requisitos no funcionales.

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega la web App, sino a las propiedades de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

- ✓ Identificar requisitos dominio.

Requisitos que provienen del dominio de especificación del sistema y que reflejan las características y restricciones de ese dominio (no tienen por qué derivarse de las especificaciones del usuario). Pueden ser funcionales o no funcionales: restringir algún requisito existente, o establecer cómo se deben ejecutar cálculos particulares.

Los objetivos que se desean lograr son:

1. Que el cliente defina las categorías de usuarios y describan esta categoría.
2. Tener requisitos básicos de la WebApp.
3. Analizar la información recopilada y utilizarla para dar un seguimiento.
4. Definir casos de uso que describan escenarios de interacción.

3.2.1 Categoría de usuarios.

Cada aplicación de tipo web o de software tiene un cierto tipo de usuario al que desea llegar, este se le denomina categoría de usuario. Tomando en cuenta lo citado anteriormente tenemos que aclarar quienes serán nuestra categoría de usuarios para nuestro proyecto. Se puede argumentar que la complejidad de la WebApp es proporcional al número de categorías de usuario con el que contamos.

Para identificar estas categorías debemos preguntarnos;

- ✓ ¿Cuál es el objetivo global del usuario cuando usa la WebApp? Por ejemplo un usuario puede estar interesado en recopilar información. Otro tal vez desea comparar precios, un tercer tal vez quiere comprar. Cada uno representa una clase o categoría de usuario.
- ✓ ¿Cuáles son los antecedentes de usuario en relación con el contenido? En este caso si el usuario lleva cierto antecedente técnico sobre el tema esto podría aburrirle.

- ✓ ¿Cómo llegara el usuario a la WebApp? El usuario llegará de forma controlada o a través de un link de otra web.

- ✓ ¿Qué características de la web le gustan o disgustan al usuario? Distintos usuario pueden tener gustos distintos en la interfaz y manera de navegar.

En nuestro proyecto, para cada sitio se determinaron las siguientes categorías de usuario.

✓ **CEO CONSULTING**

Categoría de usuarios: Son todo aquel usuario con la necesidad navegar para conocer los servicios que ofrece CEO Consulting, algún usuario interesado en tener contacto con la empresa atreves de la web para contratar un servicio.

Antecedentes de usuario: Los usuarios se espera tengan conocimientos de computación y alto nivel para navegar en el sitio, por eso se espera una navegación clara con interfaz elegante.

Como se llegara a la Web App: Se podrá acceder desde la matriz Genios o desde búsqueda relacionada con negocios, consultoría, entre otros.

✓ **Genios**

Categoría de usuarios: Son todo aquel usuario con la necesidad navegar para conocer los proyectos que ha tenido GENIOS, algún usuario interesado en tener contacto con la empresa atreves de la web para contratar un servicio.

Antecedentes de usuario: Los usuarios se espera tengan conocimientos de computación y alto nivel para navegar en el sitio, por eso se espera una navegación clara con interfaz elegante.

Como se llegara a la WebApp: Se podrá acceder a la búsqueda relacionada con negocios, consultoría, etc.

✓ **Señor Tlacuache**

Categoría de usuarios: Serán los usuarios con la necesidad de conocer los servicios que ofrece EL BAZAR DEL SEÑOR TLACUACHE. El usuario interesado en tener contacto con la empresa atreves de la web para contratar un servicio. El usuario que desee vender algún artículo que entre en la clasificación de antiguo o vintage (artículos de época). El usuario que desee comprar algún artículo que se tenga en el bazar.

Antecedentes de usuario: Los usuarios se espera tengan conocimientos de computación y alto nivel para navegar en el sitio, por eso se espera una navegación clara con interfaz elegante.

¿Cómo se llegará a la WebApp?: Se podrá acceder desde la matriz Genios o desde búsqueda relacionada con bazar, compra-venta, vintage, etc.

✓ **Cervecera Apan**

Categoría de usuarios: Son todo aquel usuario con la necesidad de conocer los servicios que ofrece LA CERVECERA APAN. El usuario interesado en tener contacto con la empresa a través de la web para contratar un servicio. Usuarios consumidores y distribuidores de la marca.

Antecedentes de usuario: Los usuarios se espera tengan conocimientos de computación y alto nivel para navegar en el sitio, por eso se espera una navegación clara con interfaz elegante.

¿Cómo se llegará a la WebApp?: Se podrá acceder desde la matriz Genios o desde búsqueda relacionada con negocios, consultoría, etc.

3.3 Planeación del proyecto

Aunque los proyectos de web se ven sencillos y rápidos sería incorrecto no tener una planeación y gestión del trabajo que se tiene, por esta razón, en nuestro caso tenemos una planeación en los tiempos de trabajo que se realizará, donde se marcaran los momentos en que se tendrán entregables del trabajo.

Una vez que se tienen metas y el objetivo del porqué se desarrolla el trabajo, podemos planear nuestro trabajo para no correr riesgo de errores por mal manejo de tiempos y mal manejo de roles de cada uno de los participantes.

Un resumen de nuestra planeación fue la que se muestra en las siguientes tablas que muestran el plan de trabajo.

Hitos
Aprobación del proyecto
Plan de proyecto Revisado
Requerimientos aprobados
Especificaciones funcionales y no funcionales
Diseño aprobado
Código completo y documentado
Programa completo funcionando
Liberar recursos y traspaso de responsabilidades
Reunión de cierre

Tabla 1. Hitos del proyecto.

Como se observa en la *Tabla 1*, enlistamos los hitos² que debemos completar para poder observar avances en el proyecto.

Como ya comentamos los hitos son nuestra medida de progreso hacia el resultado final que será nuestro entregable, estos entregables se muestran en la *Tabla 2* y son el resultado físico y tangible de nuestro trabajo y es el que se mostró al supervisor del equipo para que conociera que tanto se tenía terminado.

Entregables
Planeación
Requisitos y contenido del WebApp
Modelo de Casos de uso
Diseño de contenido
Diseño arquitectónico
Página generada (1ª versión)
Revisión final

Tabla 2. Documentos entregables.

² Los hitos (milestones) son las etapas completadas dentro del plan de desarrollo de software que se usan para medir el avance del proyecto.

Riesgo	Plan de contingencia
Posible incompatibilidad de la plantilla de Bootstrap con los navegadores.	Estandarizar la plantilla de Bootstrap para la mayor compatibilidad posible.
Que el tiempo planeado para el desarrollo del código no sea suficiente.	Realizar juntas para checar avances del proyecto y conocer posibles retrasos.
Que el equipo no acepte los entregables por errores encontrados.	Realizar pruebas constantes para evitar los errores.
Que los integrantes del equipo no cumplan con sus labores en tiempo y forma.	Tener constantes evaluaciones de su trabajo.

Tabla 3. Riesgos y contingencias del proyecto.

En la *tabla 3* aclaramos de manera general posibles riesgos y algunos de los planes de contingencia para evitar que el proyecto falle o se atrase en tiempos, este trabajo debe ser previsto en la planeación del trabajo.

Actividad	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep
Formulación: Metas y objetivos del sistema	■									
Planificación de tiempos de entrega de diseño.		■								
Análisis: Identificación de contenido a incorporar			■							
Modelización: Diseño de contenido				■	■					
Modelización: Diseño Arquitectónico				■	■					
Generación de páginas: aplicación De herramientas para creación						■	■	■		
Pruebas y Evaluación del cliente							■	■		■

Tabla 4. Diagrama de Gantt 1

El proyecto que se realizó tiene que concluirse en un tiempo determinado, y como se muestra anteriormente, se deben realizar ciertos entregables de los avances que llevamos. Toda esta información debe mostrarse en tiempos, como lo muestra la tabla 4 en un diagrama de Gantt.

3.4 Análisis de requerimientos.

Una vez obtenido el panorama general de a dónde se llevó a cabo nuestro trabajo conociendo la razón por la cual se desarrolló nuestro trabajo, qué lo motiva, quienes serán los usuarios, cuáles son nuestras metas y objetivos y herramientas de trabajo, ahora tenemos que plantear un análisis poco más detallado de cuál será el contenido que se va a incorporar en nuestros sitios y que modificaciones serán necesarias de hacer en el código de nuestro Framework. También necesitamos dar un análisis de cuál será el diseño general.

El siguiente incremento en nuestro trabajo fue el análisis de contenido, este se divide o se puede hacer clasificando cuatro campos que se tienen que analizar para nuestro proyecto y estos son:

- ✓ Análisis de contenido.
- ✓ Análisis de interacción.
- ✓ Análisis funcional.
- ✓ Análisis de configuración.

3.5. Análisis de contenido.

Nuestro análisis de contenido tiene elementos estructurales que proporcionan una visión de los requisitos de contenido. Aquí tenemos los objetos de contenido que se incluyen como los textos, imágenes, sonido, video y demás contenido multimedia que se desea o puede tener nuestro trabajo.

Nuestro análisis depende de examinar los casos de uso que encontramos de cada categoría de usuario; estos describen una interacción específica entre un usuario y la WebApp.

Recordemos que los casos de uso no son parte de nuestro diseño, sino parte del análisis que se hace de nuestro proyecto. Así, es que estos nos ayudan a describir que es lo que el sistema debe hacer. Nuestros casos de uso nos dan la visión desde el punto de vista del usuario de lo que el sistema tiene que hacer.

El diagrama que se muestra más adelante es el caso de uso general del grupo de WebApps que se obtuvo después de obtener todos los requisitos para el proyecto.

Como se puede ver en la *figura 2*, muestra una sola clase de usuario. Este podrá visualizar en todos los casos un menú del cual podrá elegir otras operaciones como ver el producto, conocer la empresa, tener contacto a través de mensajes de correo con el servidor y podrá visualizar la ubicación de las instalaciones a través de la API de Google Maps.

En un análisis específico del contenido de nuestras aplicaciones web obtuvimos los siguientes casos de uso que se visualizan en las figuras siguientes.

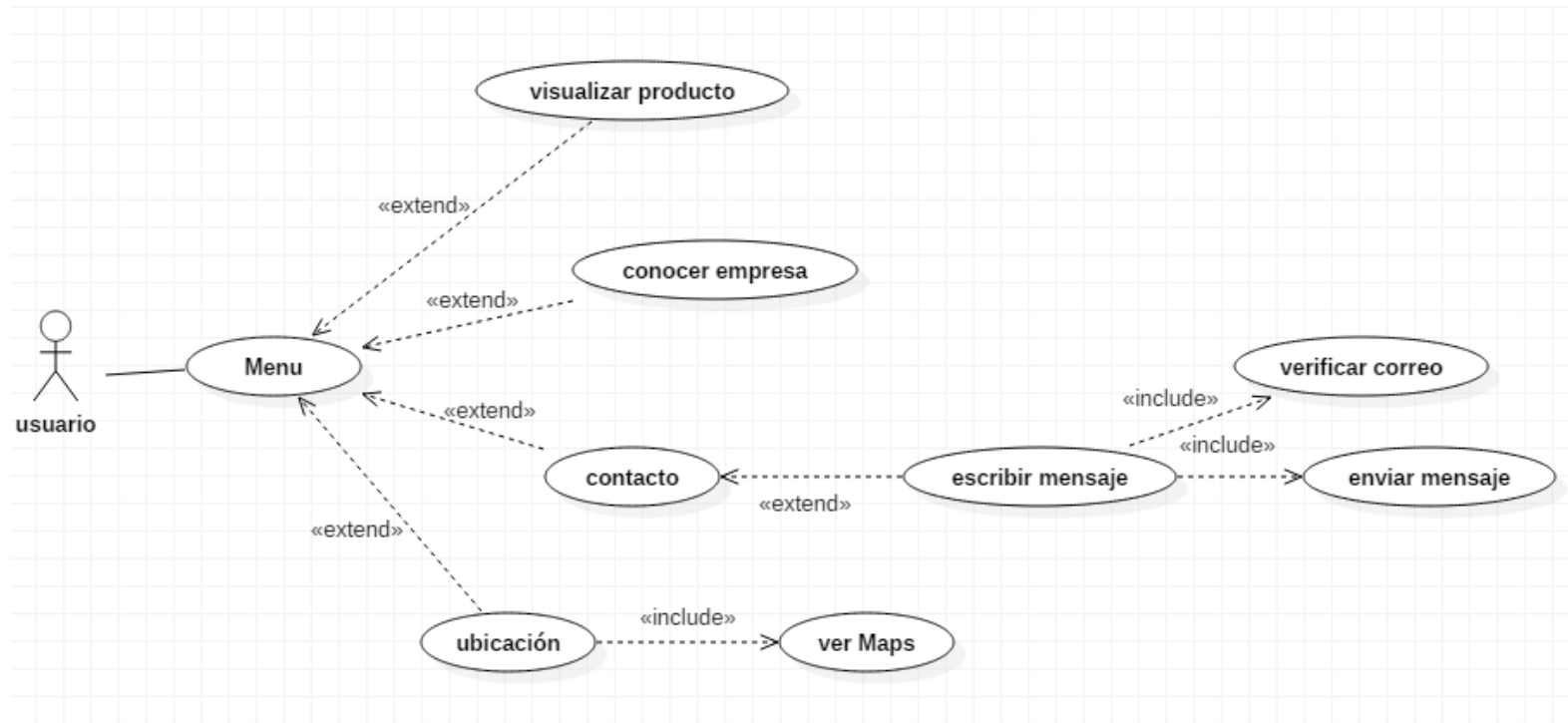


Figura 2. Caso de Uso General

✓ CEO CONSULTING

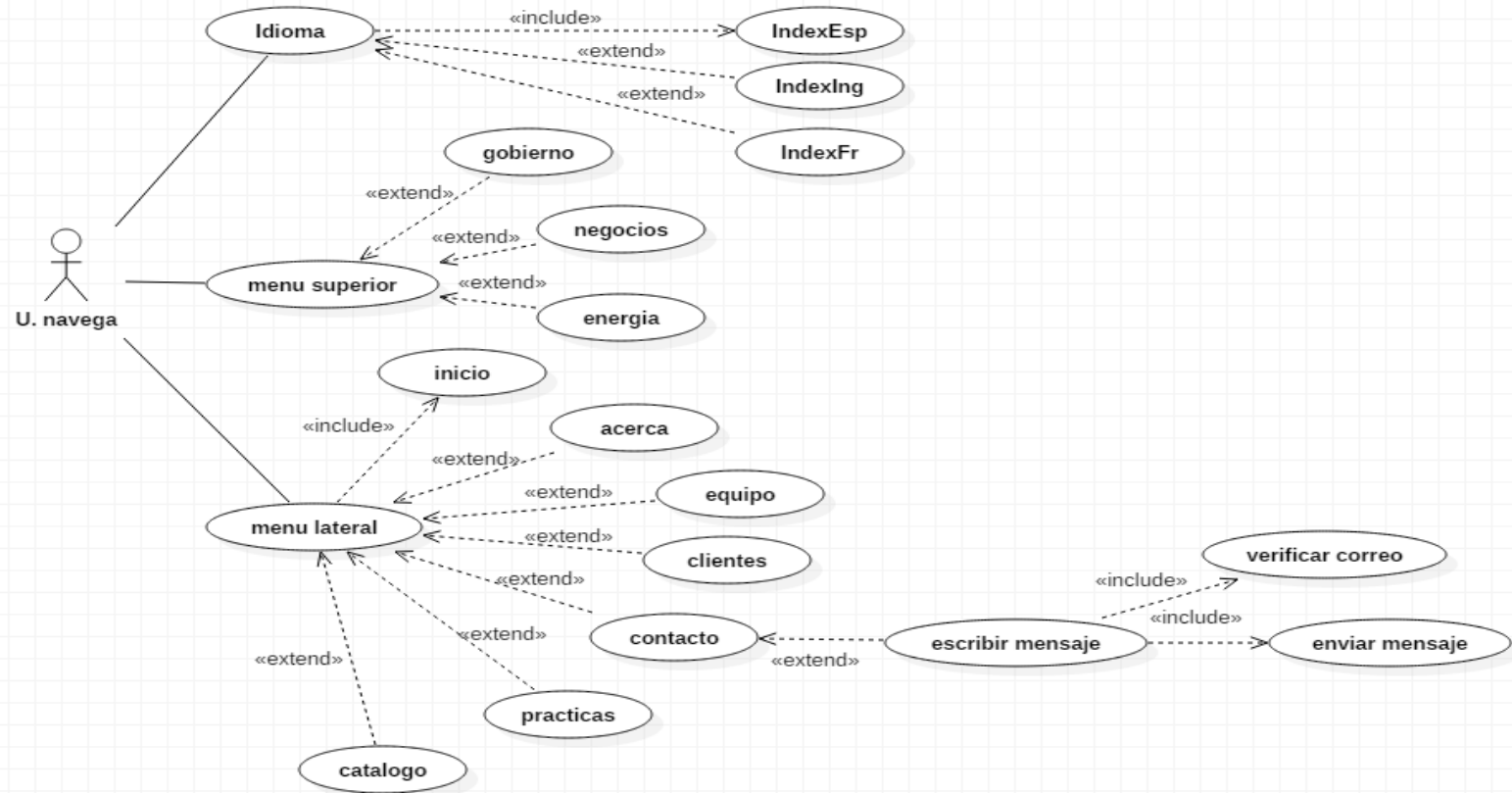


Figura 3. Caso de Uso CEO CONSULTING.

✓ GENIOS

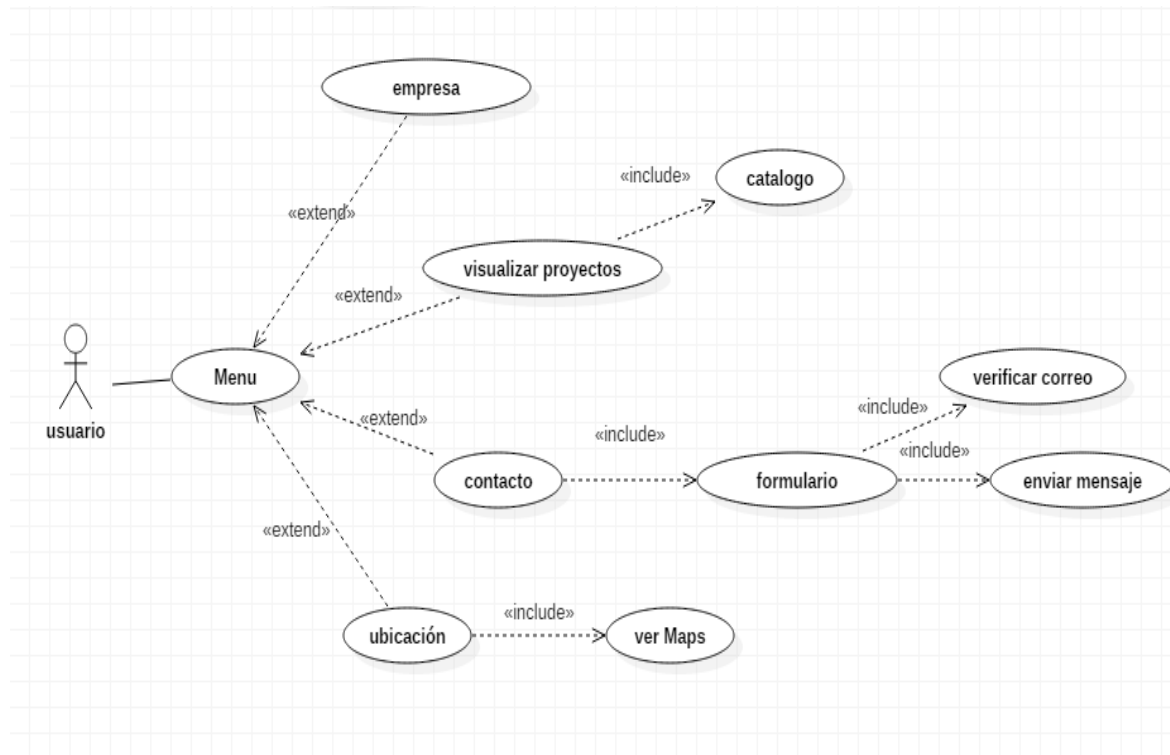


Figura 4. Caso de Uso GENIOS.

✓ SEÑOR TLACUACHE

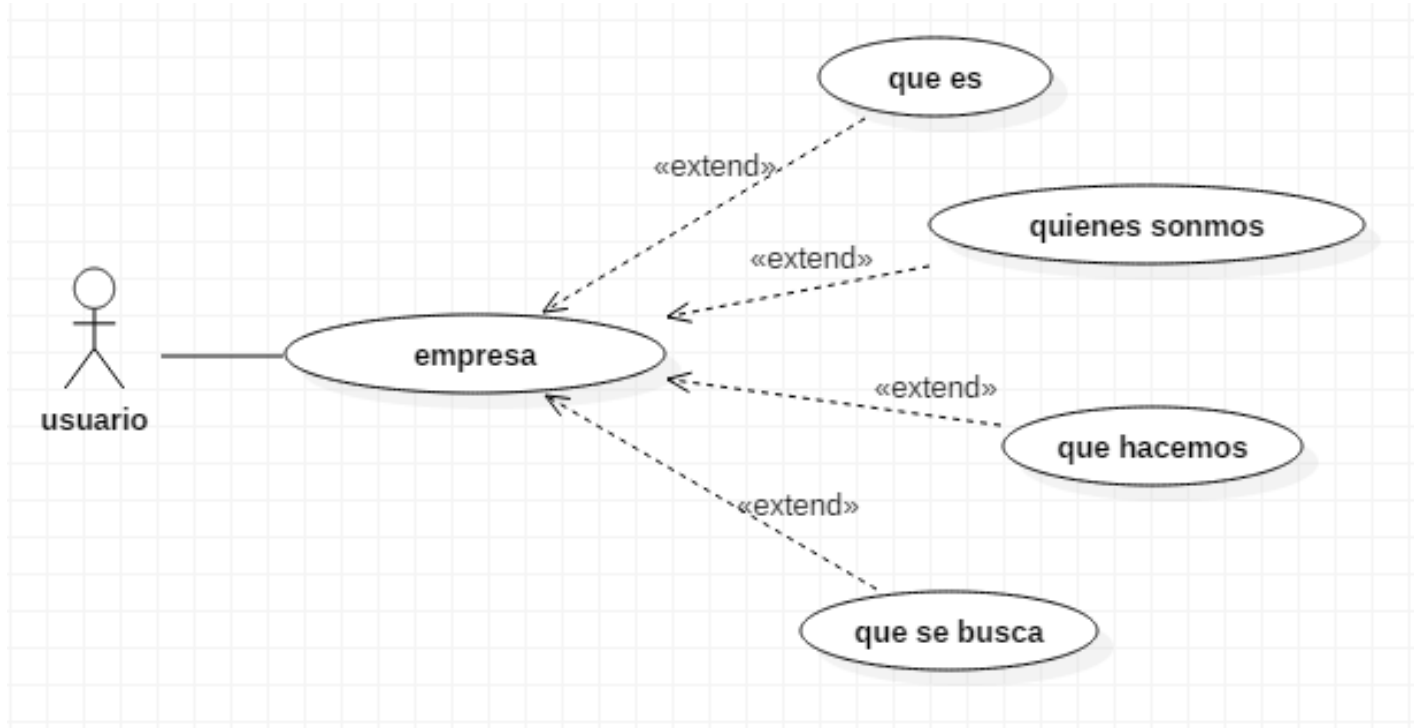


Figura 5. Caso de Uso Señor Tlacuache

✓ CERVECERA APAN

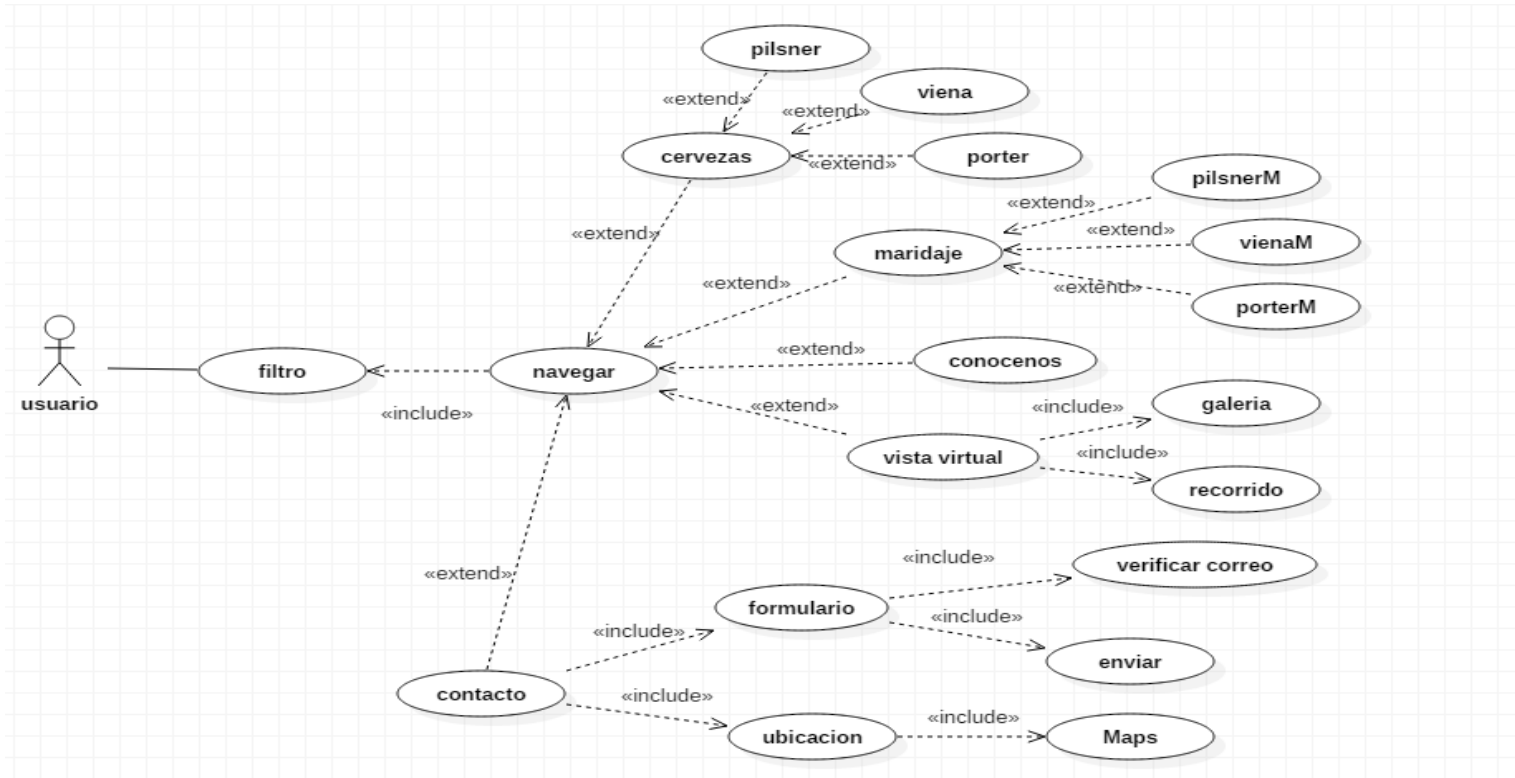


Figura 6. Caso de Uso Cervecera Apan

3.6. Análisis de interacción.

Ahora describiremos los elementos básicos de iteración del usuario, la forma de navegación y los comportamientos que se le dieron al sistema. Esto se observa en la figura 7 con un diagrama de secuencia.

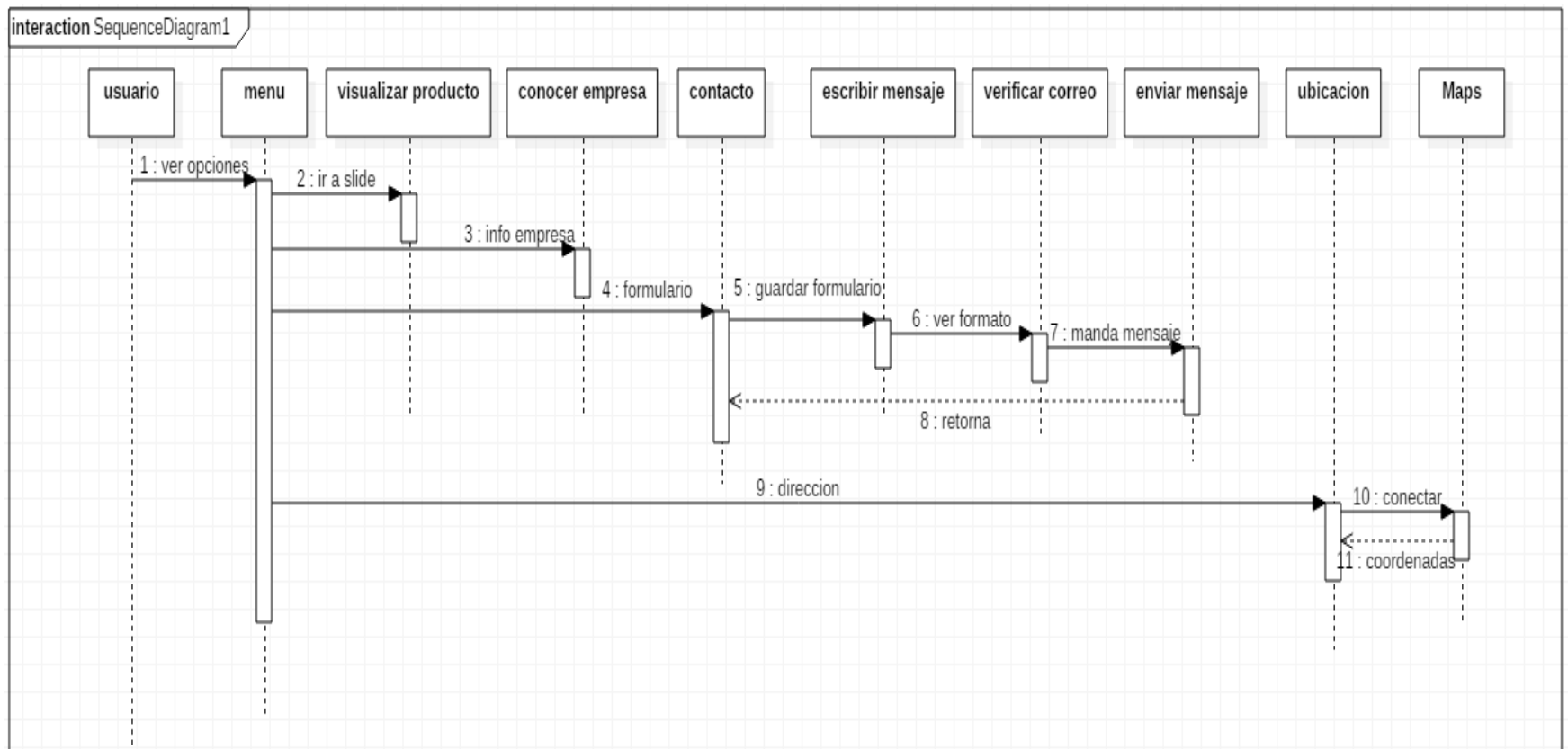


Figura 7. Diagrama de Secuencia general

En la *figura 3* vemos la distribución de actividades que puede realizar el usuario en la Web de CEO.

En la *figura 4* el CU de GENIOS nos deja ver que se cuenta con un catálogo que podrá visualizar el usuario con información de los proyectos hechos por la empresa.

La *figura 5* de la Web App del señor tlacuache muestra una sencilla acción que solo nos dejara explorar cuatro campos más en pantalla acerca de la empresa.

La *figura 6* del caso de uso de Apan es la Web App con más acciones disponibles para que el usuario navegue puesto que es una marca cervecera esta presenta tres distintos productos que se visualizarán elegantemente con la plantilla de Bootstrap además de contar con galería y un recorrido virtual.

3.7. Análisis funcional.

El análisis funcional describe las operaciones que se aplicarán al contenido de nuestra aplicación web de manera detallada. Aquí se abordan dos elementos de procesamiento de la Web:

1) Funcionalidad Observable respecto a lo que el usuario comprende, aquello que será visible y lo que entrega la Web App al usuario final.

2) Las operaciones dentro de las clases. La parte interna de nuestra Web App, aquellas clases y operaciones que permiten al usuario navegar.

Nuestra funcionalidad observable comprende cualquier función de procesamiento que el usuario inicia directamente. Será la manera de ¿Cómo? y ¿Qué? visualizará el usuario; ¿Cuáles serán los cambios que se realizó en el Framework? Son las preguntas más importantes que se deben de realizar para su respectiva funcionalidad.

Las funciones que son visibles para nuestros usuarios, en nuestro caso según el análisis de casos de uso nuestro análisis funcional nos deja ver que la manera de navegar y en forma general, la operación de envío mensajes son las funcionalidades principales en cada una de nuestras Web App.

En la *figura 8* podemos ver la funcionalidad observable del usuario, que en este caso es el formulario para poder realizar él envío de mensajes al servidor de la empresa.



The image shows a Bootstrap-style form with the following elements:

- Email:** A text input field with the placeholder text "Introduce tu email".
- Contraseña:** A text input field with the placeholder text "Contraseña".
- Adjuntar un archivo:** A section containing a "Seleccionar archivo" button, the text "No se ha seleccionado ningún archivo", and a "Ejemplo de texto de ayuda." label.
- Activa esta casilla:** A checkbox with the label "Activa esta casilla".
- Enviar:** A submit button.

Figura 8. Formulario Bootstrap

En la *figura 9* podemos ver la plantilla básica de formulario que aplica Bootstrap, en este caso las operaciones dentro de la clase, aquí se maneja la clase *.form-horizontal* que permite mantener cada *label* de nuestro formulario en forma horizontal y la clase *.form-group* que permite agrupar cada campo del formulario con el label.

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label for="ejemplo_email_3" class="col-lg-2 control-label">Email</label>
    <div class="col-lg-10">
      <input type="email" class="form-control" id="ejemplo_email_3"
        placeholder="Email">
    </div>
  </div>
  <div class="form-group">
    <label for="ejemplo_password_3" class="col-lg-2 control-label">Contraseña</label>
    <div class="col-lg-10">
      <input type="password" class="form-control" id="ejemplo_password_3"
        placeholder="Contraseña">
    </div>
  </div>
  <div class="form-group">
    <div class="col-lg-offset-2 col-lg-10">
      <div class="checkbox">
        <label>
          <input type="checkbox"> No cerrar sesión
        </label>
      </div>
    </div>
  </div>
  <div class="form-group">
    <div class="col-lg-offset-2 col-lg-10">
      <button type="submit" class="btn btn-default">Entrar</button>
    </div>
  </div>
</form>
```

Figura 9. Código de formulario Bootstrap

Más adelante se muestra como se manipularon las clases mencionadas para poder obtener el estilo deseable para cada Web App.

Como mencionamos otra funcionalidad que se encontró es la forma de navegar con respecto a la operación que se realizó para poder visualizar algunos efectos en la plantilla; por ejemplo el efecto de parallax y el efecto llamado freelancer que tienen las aplicaciones de Señor Tlacuache y Genios.

La *figura 10* tiene el código del efecto freelancer de la Web App Genios. Aquí las operaciones de la clase se encuentran en el archivo de JavaScript que es el encargado de proporcionar funciones para un efecto de cambio de secciones en forma de bloque.

```
// jQuery for page scrolling feature - requires jQuery Easing plugin
$(function() {
  $('body').on('click', '.page-scroll a', function(event) {
    var $anchor = $(this);
    $('html, body').stop().animate({
      scrollTop: $($anchor.attr('href')).offset().top
    }, 1500, 'easeInOutExpo');
    event.preventDefault();
  });
});

// Floating label headings for the contact form
$(function() {
  $("body").on("input propertychange", ".floating-label-form-group", function(e) {
    $(this).toggleClass("floating-label-form-group-with-value", !! $(e.target).val());
  }).on("focus", ".floating-label-form-group", function() {
    $(this).addClass("floating-label-form-group-with-focus");
  }).on("blur", ".floating-label-form-group", function() {
    $(this).removeClass("floating-label-form-group-with-focus");
  });
});

// Highlight the top nav as scrolling occurs
$('body').scrollspy({
  target: '.navbar-fixed-top'
})

// Closes the Responsive Menu on Menu Item Click
$('.navbar-collapse ul li a').click(function() {
  $('.navbar-toggle:visible').click();
});
});
```

Figura 10. Código JavaScript de efecto freelancer de Bootstrap

En la *figura 11* podemos ver el código del efecto parallax que tendrá el Señor Tlacuache. Aunque es un conjunto de enlaces de código entre HTML, CSS y JavaScript, este último nos da el verdadero efecto y es quien realiza la operación principal.

```
$(document).ready(function(){
    $(window).scroll(function(){
        var barra = $(window).scrollTop();
        var posicion = (barra * 0.10);

        $('body').css({
            'background-position': '0 -' + posicion + 'px'
        });
    });
});
```

Figura 11. Código JavaScript para realizar efecto parallax

3.8. Análisis de configuración.

Las WebApps se deben diseñar e implementar de manera que sean compatibles con una diversidad grande de dispositivos que tengan acceso a Internet y tengan el soporte necesario para poder descargar desde el servidor el contenido completo que se desea. En este análisis veremos el entorno e infraestructura en donde residirá nuestra página.

En nuestro proyecto, Bootstrap maneja animaciones e interfaces llamativas que deben contar con las últimas versiones de los navegadores de internet para poder ofrecer al usuario final la mejor visión de la web. En la siguiente tabla se muestra la actual compatibilidad de Bootstrap con los navegadores para PC y dispositivos móviles.

La *tabla 5* sólo funcionó de guía para conocer la compatibilidad del Framework con los navegadores, sin embargo, más adelante se verá cómo se sometió el proyecto a una amplia prueba en cada configuración de navegador.

	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	✓	X	-	X	-
iOS	✓	-	-	X	✓
Mac OS X	✓	✓	-	✓	✓
Windows	✓	✓	✓	✓	X

Tabla 5. Soporte del Framework Bootstrap

Según el análisis, Bootstrap también debería funcionar bien en Chromium (Linux) e Internet Explorer 7 (Windows), aunque no están soportados oficialmente.

Los navegadores Internet Explorer 8 y 9 también están soportados, pero se debe tener en cuenta que muchas propiedades de CSS3 y elementos de HTML5 no funcionan en estos navegadores. Además, Internet Explorer 8 requiere el uso de la librería *respond.js* para que el diseño web responsivo funcione correctamente.

Funcionalidad	Internet Explorer 8	Internet Explorer 9
border-radius	X No soportado	✓ Soportado
box-shadow	X No soportado	✓ Soportado
transform	X No soportado	✓ Soportado con el prefijo -ms
transition	X No soportado	X No soportado
Placeholder	X No soportado	X No soportado

Tabla 6. Comparativa de funciones no soportadas.

Como se puede observar en la *tabla 6*, Internet Explorer es el navegador con menos soporte en las funcionalidades del Framework, así que la manipulación de nuestro código de Bootstrap tomará en cuenta evitar estas funciones no soportadas lo más que se pueda.

Modelización y Generación de Páginas.

La etapa de modelización como se mencionó en una introducción a la ingeniería web tiene dos ramas que se deben seguir paralelamente para poder llegar a la generación de una Web App de calidad estas son la producción y diseño del contenido y por otro lado el diseño arquitectónico, de navegación e interfaz de la página.

Pero, ¿Cómo evaluar la calidad de nuestro proyecto? Esto lo podemos medir guiándonos del “árbol de requisitos de calidad” propuesto por Olsina (Olsina, 1999) y sus colegas que identifican los siguientes atributos técnicos como valorables para la calidad de una Web.

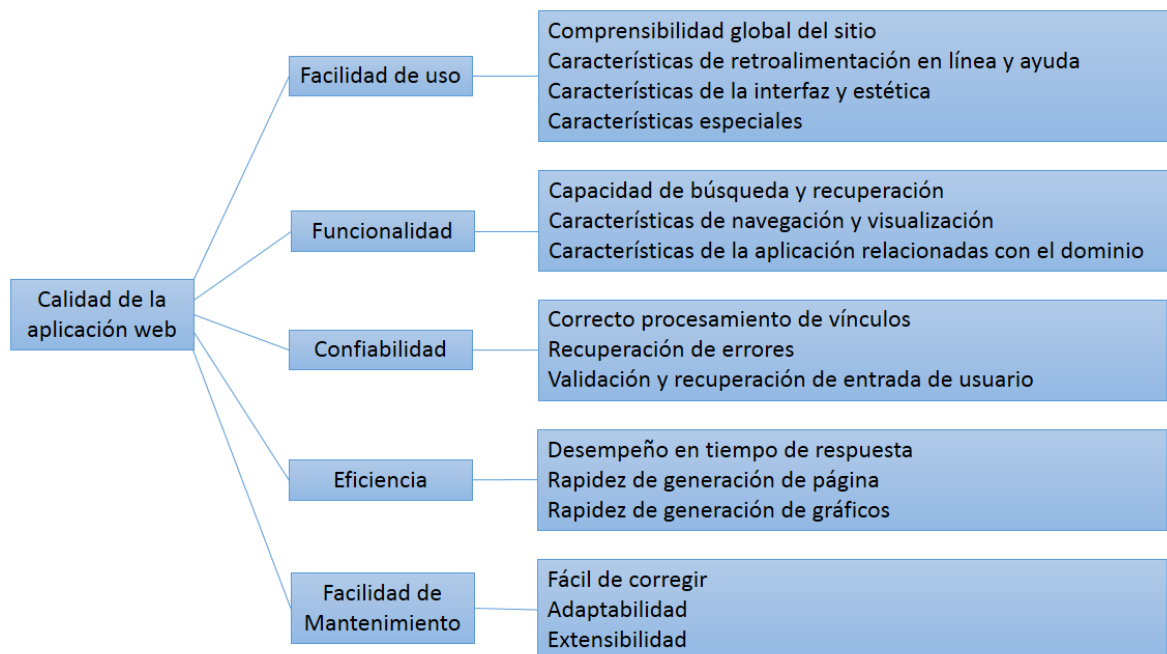


Figura 12. Árbol de requisitos de calidad

Tomando en cuenta la ilustración de la *figura 12* como guía para crear una web de calidad, se empezó a trabajar en el diseño de contenido y diseño arquitectónico.

4.1. Diseño.

El diseño para la ingeniería web es un punto que abarca no sólo el ámbito de la programación sino que también se debe tener un gran cuidado con la parte gráfica, ya que de todas las categorías de software, las WebApp son las más visuales, las más dinámicas y las más estéticas.

El Framework Bootstrap por ser una herramienta de código libre podremos manipular de una manera más rápida y sencilla la parte visual y de efectos de nuestra Web para poder cumplir con los estándares primarios de diseño que en la ingeniería web se explican de la siguiente manera. (Kaiser, 2002)

- ✓ Simplicidad: Nuestra WebApp debe encontrar un equilibrio para que no tenga un exceso de contenido o efectos visuales que puedan molestar al usuario.
- ✓ Consistencia: Se aplica virtualmente a cada elemento del modelo de nuestro sitio. Debemos cuidar que nuestro contenido lleve una consistencia en el tipo de fuente, formato de texto y arte gráfico.
- ✓ Identidad: Este punto es muy importante en nuestro proyecto, esta meta se debe cuidar mucho ya que manejaremos cinco Webs y cada una deberá tener una “identidad” propia de las demás porque cada una mostrara diferente información a un campo diferente de usuarios. Por ello la estética de la interfaz y el diseño de navegación deben ser consistentes con el dominio de la aplicación.

- ✓ Robustez: Con base a la identidad establecida, cada página tendrá que tener un contenido y funciones relevantes que espera el usuario.
- ✓ Navegabilidad: Los mejores proyectos de ingeniería web muestran trabajos finales con una interfaz intuitiva, predecible y fácil de entender para los usuarios al momento de navegar.
- ✓ Apariencia visual: Como ya se comentó debemos cuidar nuestra plantilla de interfaz, la coordinación de color, el equilibrio del texto, los gráficos y otros medios audiovisuales que se vayan a manejar en nuestro caso con Bootstrap.
- ✓ Compatibilidad: Cada uno de los efectos visuales y la interfaz en general debe poder visualizarse en todos los navegadores que el usuario pueda manejar.

4.2. Producción de Contenido.

El diseño de contenido se enfoca en dos asuntos de diseño, el primero es una representación de diseño para que los objetos de contenido, esto va a representar los mecanismo que se necesitan para que se establezcan relaciones uno con otro.

Además, el diseño de contenido se ocupa de la representación de la información dentro de un objeto de contenido específico: actividad de diseño que dirigen publicistas, diseñadores gráficos entre otros y es deber como programador aproximarse lo más que se pueda a la idea de los usuarios finales.

Con los bosquejos que se presentan en las siguientes imágenes y los casos de uso que se obtuvieron del análisis podemos hacer una representación del diseño de objetos de nuestro trabajo.

Con la descripción de nuestros componentes podemos ver cuáles serán los cambios necesarios que se desarrollarán en el código de las nuestro Framework para alcanzar la idea estética que se desea.

Con el material obtenido en todo el proceso de Ingeniería que hemos llevado ahora sólo nos queda definir un bosquejo más claro de lo que se pretende obtener visualmente de nuestras páginas web elaborando borradores con imágenes de la pantalla para cada acción de usuario, describir la plantilla para cada estado y describir el comportamiento de la interfaz.

✓ CEO CONSULTING

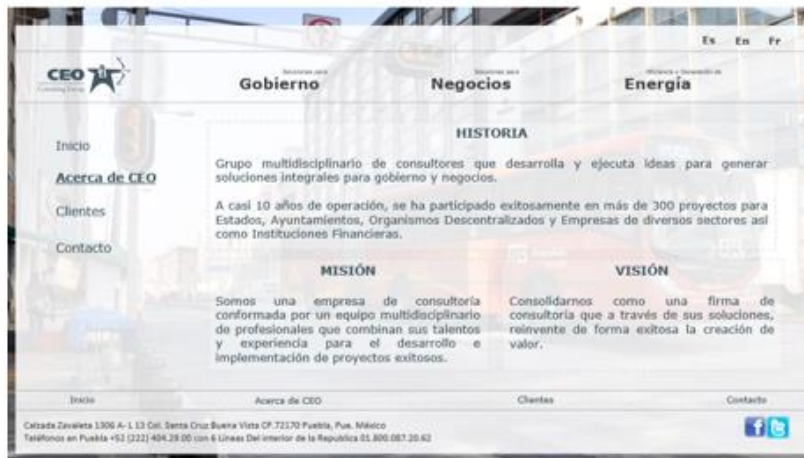


Figura 13. Bosquejo de diseño de la Web App CEO

✓ GENIOS



Figura 14. Bosquejo principal de GENIOS



Figura 15. Bosquejo del catálogo de empresas de GENIOS

La figura 14 muestra el bosquejo de la pantalla principal de la Web, mientras que la figura 15 enseña un boceto de las pantallas que se desean ver en la sección de “empresas”.

✓ CERVECERA APAN

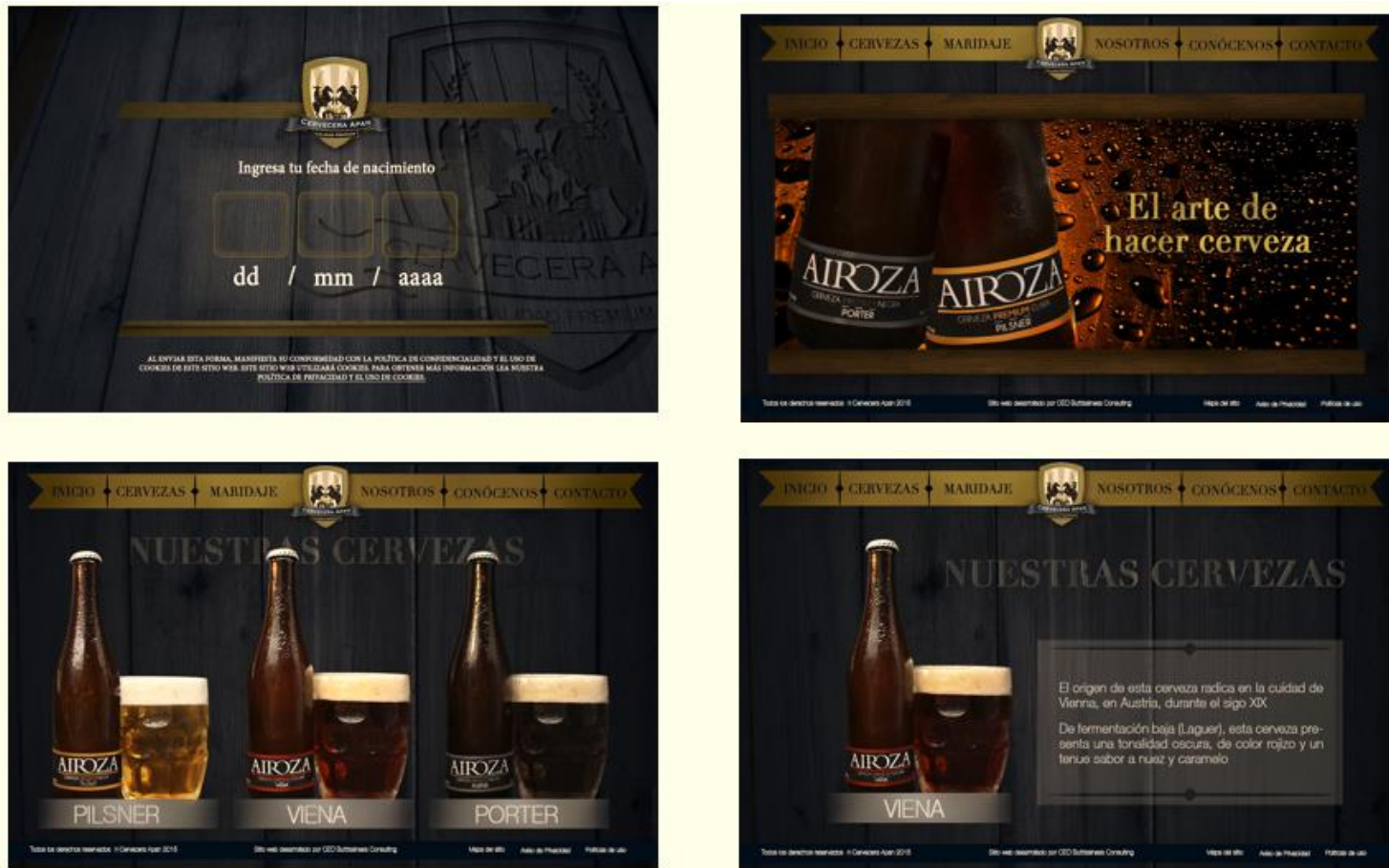


Figura 16. Bosquejo de Web Apan



Figura 17. Segundo bosquejo de las pantallas de la Web Apan

El diseño estético o diseño gráfico es un trabajo artístico que complementara toda la parte técnica de nuestro trabajo. Sin él la WebApp puede ser funcional pero no atractiva. En las cuestiones estéticas no hay una pauta marcada de desarrollo sin embargo hay ciertos lineamientos generales que se toman para obtener una plantilla estética como:

- ✓ No rellenar de información en una ventana.
- ✓ Resaltar el contenido.
- ✓ Organizar los elementos de arriba a la izquierda hacia abajo a la derecha.
- ✓ Agrupar navegación, contenido y función geográficamente.
- ✓ No extender tanto la barra de desplazamiento.
- ✓ Considerar la resolución y el tamaño de la ventana del navegador.

4.3. Diseño arquitectónico.

Identificamos dos clases de arquitectura de una web que son la arquitectura de contenido que se centra en la forma en que los objetos de contenido se estructuran para su presentación y navegación y la arquitectura de la WebApp que aborda la forma en la que la aplicación se estructura para gestionar sus procesos internos y el manejo de datos para la interacción con el usuario.

La arquitectura de contenido se puede manejar con bocetos en papel o usando distintas herramientas de diseño sencillas para podernos dar una idea de cómo nuestros objetos de contenido estará distribuido.

Arquitectura de contenido.

✓ CEO CONSULTING

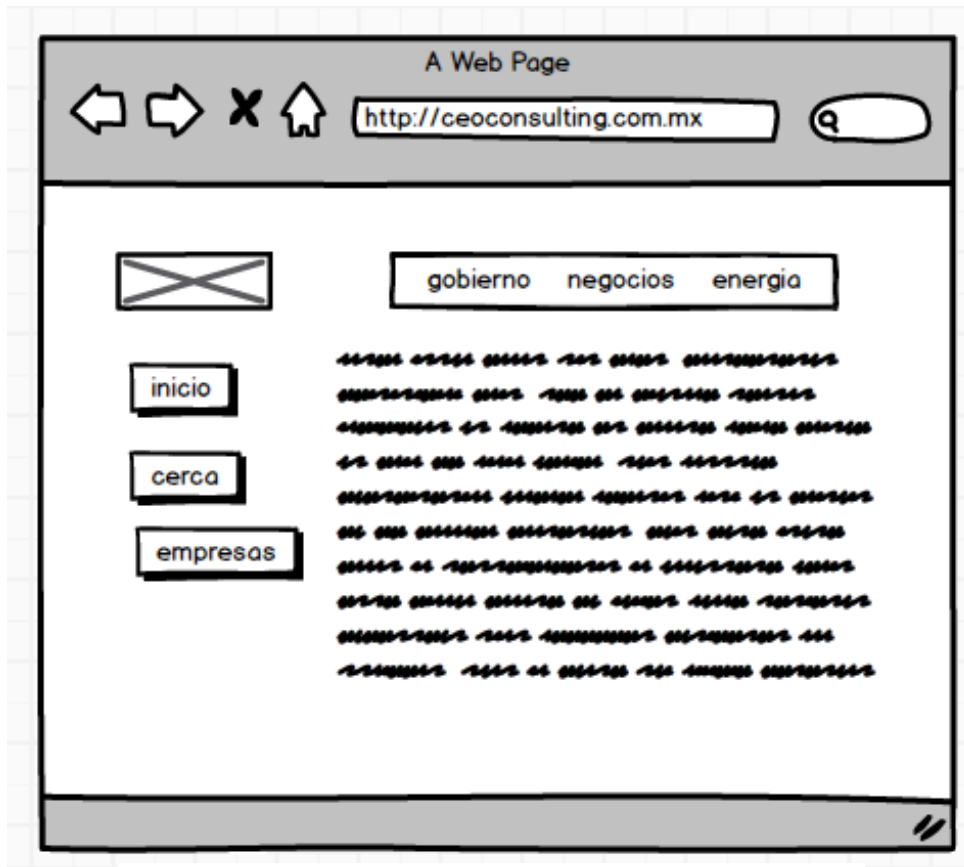


Figura 18. Boceto de CEO

La *figura 18* podemos ver el boceto final de cómo se mostraría el contenido con la plantilla de Bootstrap. Su arquitectura tiene dos tipos de menú, uno lateral y uno superior y el contenido se mostrará en la parte central de estas barras de menú. Se manejó además diferentes imágenes de fondo para cada sección.

✓ GENIOS

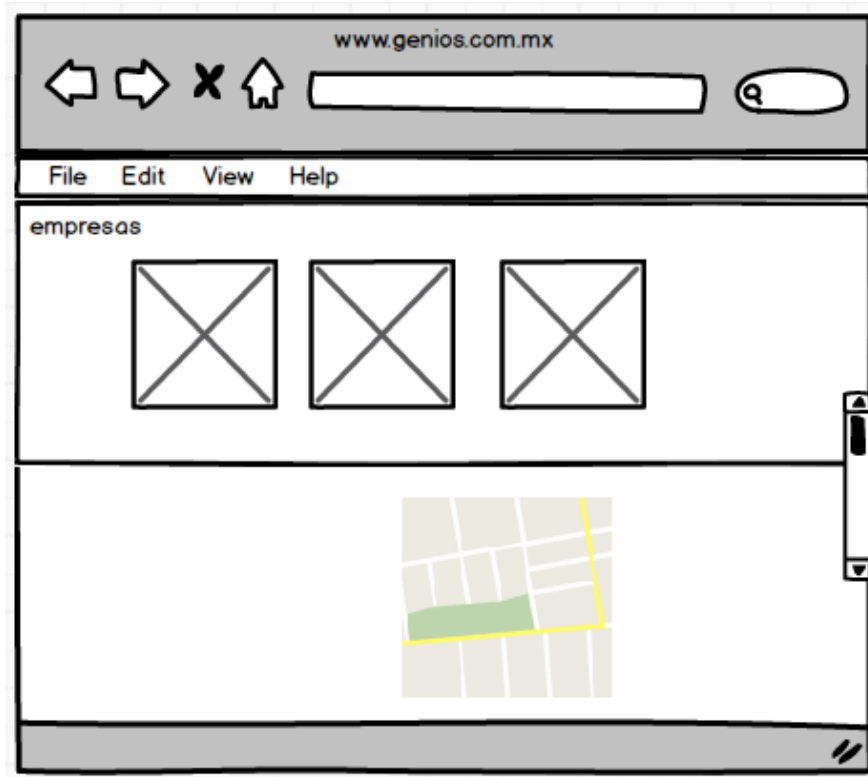


Figura 19. Boceto de GENIOS

En la *figura 19* la manera de mostrar los elementos en la página será en secciones de pantalla que alojarán el contenido en párrafos centrados y las imágenes en cuadrícula, con un menú fijo superior fijo que no tendrá hipervínculo sino solo enviará hasta la sección deseada.

✓ Señor Tlacuache

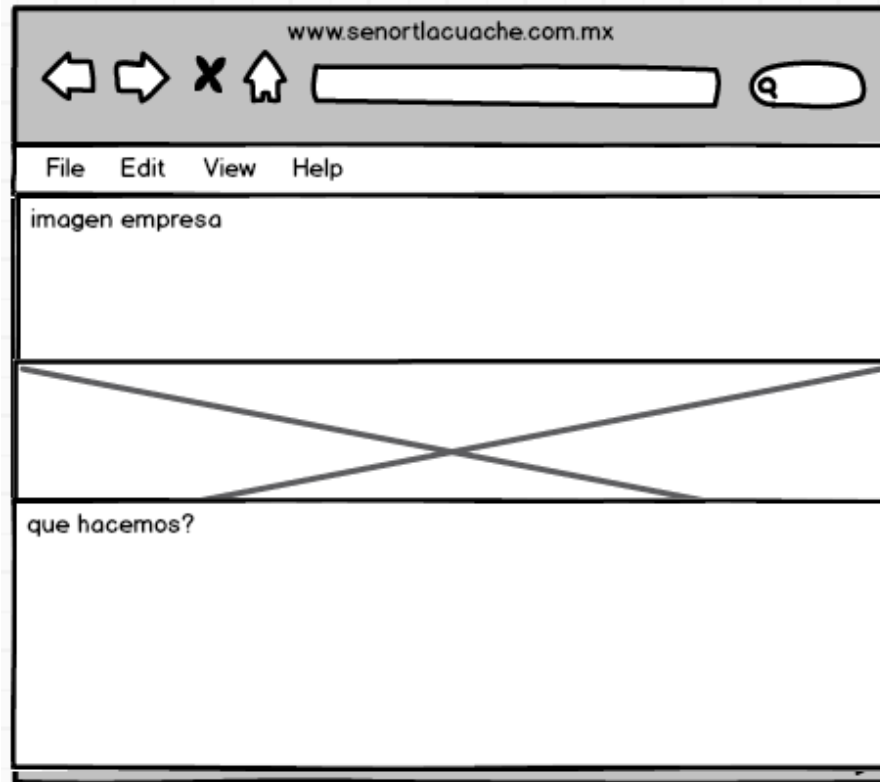


Figura 20. Boceto primario de Sr. Tlacuache

El boceto del señor tlacuache muestra similitudes con el boceto de genios a diferencia de una imagen que tiene en medio de cada sección y que tendrá la función de crear el efecto llamado parallax que se explicó en el capítulo anterior. Este boceto de visualiza en la *figura 20*.

✓ Cervecera Apan

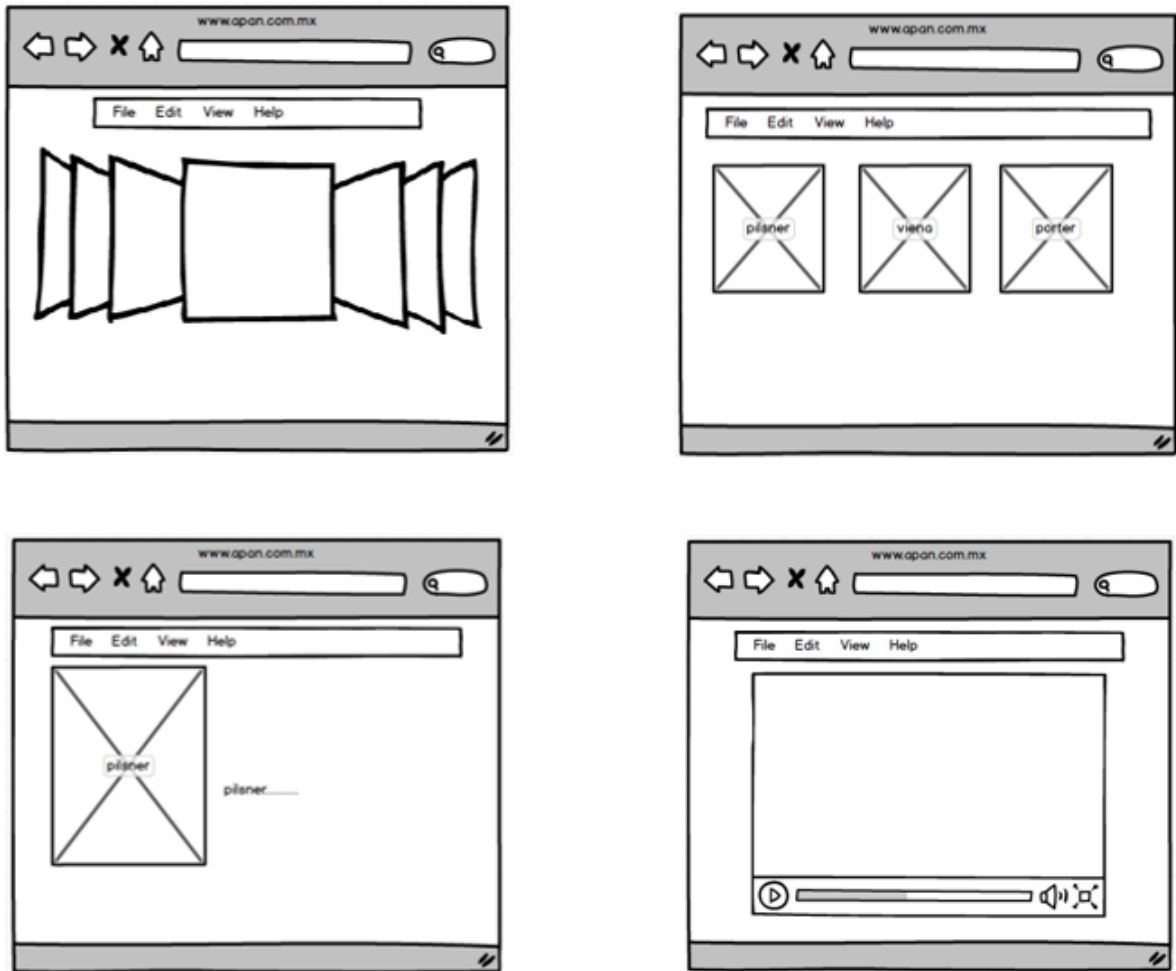


Figura 21. Boceto de Cervecera Apan

En la *figura 21* tenemos cuatro bocetos de la página porque su contenido es más variado a diferencia de las otras Web App. En primera tendremos un slider con efecto carrusel en la parte central de la página, luego en la parte superior derecha tenemos la presentación del producto en tres imágenes centrales y en la parte inferior izquierda tenemos la presentación individual del producto seleccionado, por ultimo una página con la parte multimedia que igual se mostrara centrada.

Arquitectura de la aplicación.

Lo expresado en la arquitectura de contenido también debe reflejarse en la arquitectura interna de la aplicación, es decir, en su código mostrando el funcionamiento y manejo de clases. Aquí mostramos de manera interna el cambio que se hizo en el código fuente de las plantillas de Bootstrap dentro de sus archivos HTML, CSS y JavaScript.

Primero veamos cual es la arquitectura que maneja Bootstrap ya que sobre esta arquitectura trabajamos para cumplir el objetivo, en la *figura 22* podemos ver la versión compilada del Framework, nosotros utilizaremos la versión “min”.

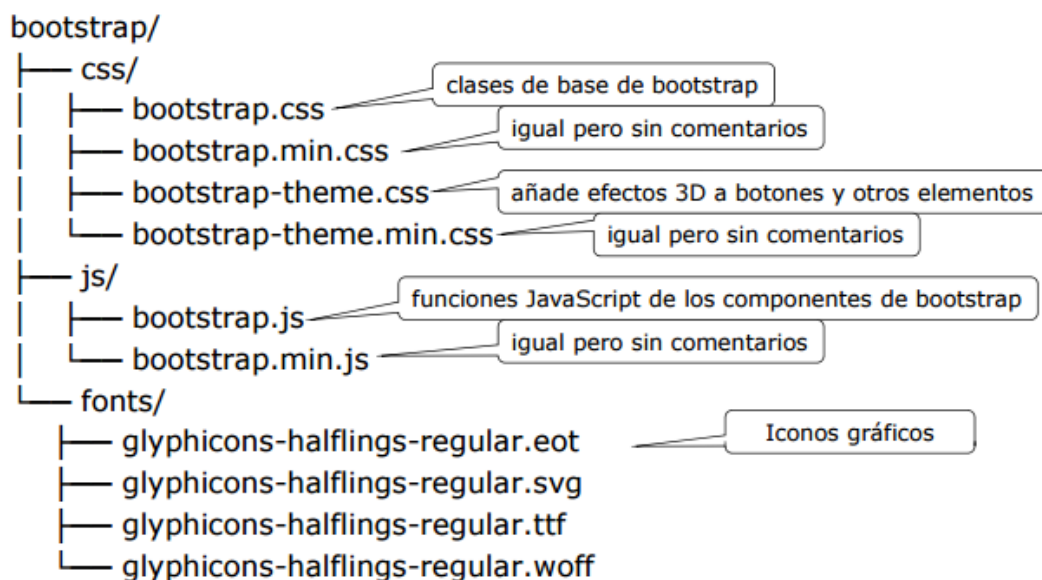


Figura 22. Archivos de código Bootstrap

Para poder empezar a trabajar con el Framework siempre tenemos que respetar la plantilla básica para poder llamar a todos los archivos compilados necesarios y por eso debemos conservar este código. En la *figura 23* podemos ver esta plantilla.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Plantilla básica de Bootstrap en español</title>
    <!-- CSS de Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </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 -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

The diagram shows the HTML code for a Bootstrap template. Callouts point to specific parts of the code: 'Bootstrap funciona con documentos HTML 5' points to the DOCTYPE declaration; 'El viewport es un atributo que permite definir la anchura de la ventana del navegador para adaptar la página automáticamente' points to the viewport meta tag; 'Inclusión de la librería CSS de bootstrap' points to the CSS link; 'Necesita jquery' points to the jQuery script tag; and 'Inclusión de la librería JS de bootstrap' points to the Bootstrap JavaScript script tag.

Figura 23. Plantilla primaria Bootstrap

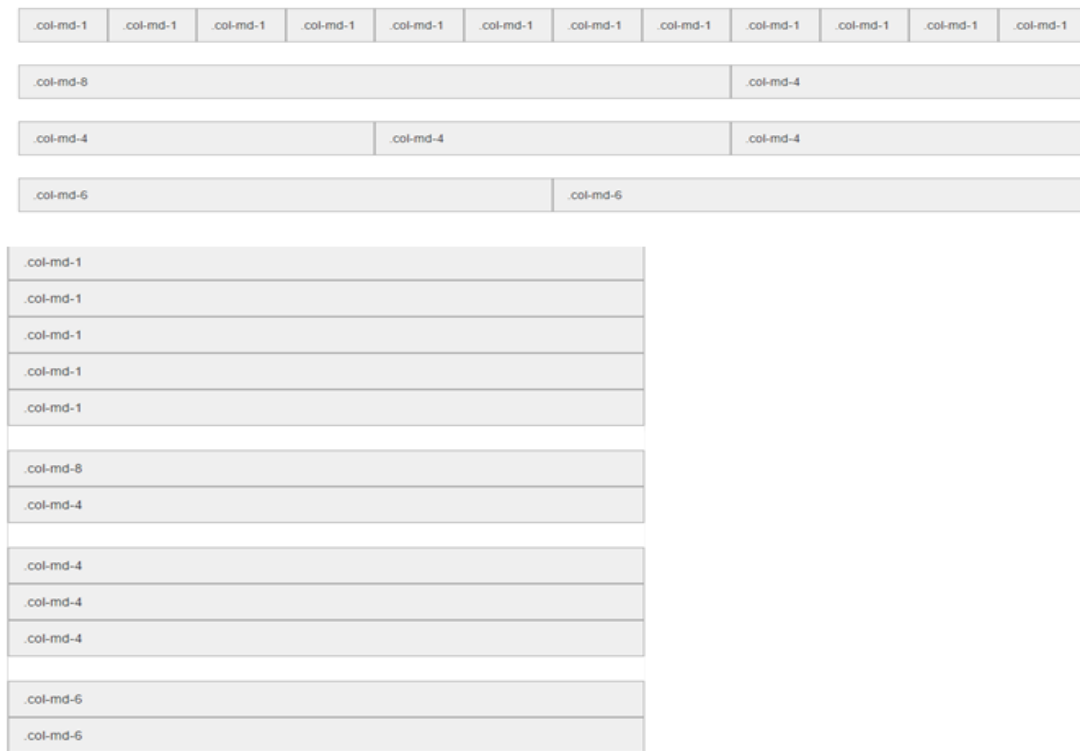
Bootstrap incluye una rejilla o retícula fluida pensada para móviles y que cumple con el diseño web responsivo. Esta retícula crece hasta 12 columnas a medida que crece el tamaño de la pantalla del dispositivo.

El diseño de páginas basado en rejilla se realiza mediante filas y columnas donde se colocan los contenidos. Así funciona la rejilla de Bootstrap:

- ✓ Las filas siempre se definen dentro de un contenedor de tipo *.container* (anchura fija) o de tipo *.container-fluid* (anchura variable). De esta forma las filas se alinean bien y muestran el *padding* correcto.

- ✓ Las filas se utilizan para agrupar horizontalmente a varias columnas.
- ✓ El contenido siempre se coloca dentro de las columnas, ya que las filas sólo deberían contener como hijos elementos de tipo columna.
- ✓ Bootstrap define muchas clases CSS (como por ejemplo `.row` y `.col-xs-4`) para crear rejillas rápidamente. También, existen *mixins* de *Less* para crear diseños más semánticos.
- ✓ La separación entre columnas se realiza aplicando padding. Para contrarrestar sus efectos en la primera y última columnas, las filas (elementos `.row`) aplican márgenes negativos.
- ✓ Las columnas de la rejilla definen su anchura especificando cuántas de las 12 columnas de la fila ocupan. Si por ejemplo quieres dividir una fila en tres columnas iguales, utilizarías la clase `.col-xs-4` (el 4 indica que cada columna ocupa 4 de las 12 columnas en las que se divide cada fila).

Algunos ejemplos del manejo de la rejilla son los que se pueden observar en la *figura 20*, un ejemplo que muestra cómo se maneja el modelo de rejillas con las clases `.col-md-*`.



En los dispositivos móviles (*extra pequeño o pequeño*) esta rejilla se muestra verticalmente, pero en un ordenador (*medio o grande*) se ve horizontalmente.

Ya conociendo el principio de manejo de clases de rejillas para poder ajustar automáticamente nuestro contenido en cualquier pantalla de dispositivos y el código interno de las plantillas básicas de Bootstrap, ahora veremos la arquitectura de las aplicaciones.

La arquitectura de la aplicación la podemos ver en las siguientes imágenes que muestran el código que se desarrolló para alcanzar la arquitectura de contenido.

En la *figura 25* observamos las clases principales que generan el funcionamiento de la Web App de Ceo Consulting. Estas clases están marcadas y explicadas en la misma imagen.

En la *figura 26* y *figura 27* se muestra el código de las clases que manejan el funcionamiento de la Web App Genios. Dentro de la figura se explica brevemente cual es la tarea de la clase.

La *figura 28* nos enseña las clases que utiliza la Web App del señor tlacuache para poder generar la interfaz de navegación.

En la Web App de la Cervecera Apan usamos clases básicas de Bootstrap para poder hacer responsiva nuestra aplicación esto se marca en la *figura 29* que muestra el fragmento de código de sus pantallas.

```

<div class="container">
  <div class="row">
    <div class="box">
      <div class="row">
        <div class="col-sm-3 text-center">
          <a href='indexceo.html'></a>
        </div>
        <div class="col-sm-9">
          <div id="navigation_container">
            <!-- the navigation links -->
            <ul id="navigation">
              <li><a class="g" href='gobierno.html'><p id="tituloP1">Soluciones para <br> <p id="tituloG1"> Gobierno </p>
              <li><a class="n" href='negocios.html'><p id="tituloP2">Soluciones para <br><p id="tituloG2"> Negocios </p>
              <li><a class="e" href='energia.html'><p id="tituloP3">Eficiencia y Generación de <br><p id="tituloG3"> Ene;
            </ul>
          </div>
        </div>
      </div>
    </div>
    <div class="col-sm-3">
      <br><br><br>
      <ul class="nav nav-pills nav-stacked">
        <li class="active"><a href='indexceo.html'>Inicio</a></li>
        <li><a href='acercade.html'>Acerca de CEO</a></li>
        <li><a href='equipoceo.html'>Equipo CEO</a></li>
        <li><a href='clientes.html'>Clientes</a></li>
        <li><a href='contacto.html'>Contacto</a></li>
        <li><a href='practicas.html'>Prácticas</a></li>
        <li><a href='revista.html'>Presentación</a></li>
      </ul>
    </div>
  </div>

```

La clase box genera la parte central de la pantalla donde se metera toda la informacion de la Web

"navigation_container se encarga del estilo y la funcionalidad del menu superior.

El menu lateral maneja 3 clases difernetes que dan estilo y funcionalidad a esta seccion de la Web.

Figura 25. Código de la WebApp CEO

```

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav navbar-right">
    <li class="hidden">
      <a href="#page-top"></a>
    </li>
    <li class="page-scroll">
      <a href="#about">¿Quienes Somos?</a>
    </li>
    <li class="page-scroll">
      <a href="#portfolio">Empresas</a>
    </li>
    <li class="page-scroll">
      <a href="#contact">Contacto</a>
    </li>
    </li>
    <li class="page-scroll">
      <a href="#mapa">Mapa</a>
    </li>
  </ul>
</div>

```

La clase page-scroll se omodifico para que utilizará el plug-in Scrollspy, este se utiliza para actualizar automáticamente los enlaces en una lista de navegación basada en la posición de desplazamiento. Forma parte del código que ofrece Bootstrap.

Figura 26. Código de la WebApp Genios (page-scroll)

```

<section id="portfolio">
  <div class="container">
    <div class="row">
      <div class="col-lg-12 text-center">
        <h2>Nuestras Empresas</h2>
      </div>
    </div>
    <div class="row">
      <div class="col-sm-4 portfolio-item"> <!-- Servicio 1 -->
        <a href="#portfolioModal1" class="portfolio-link" data-toggle="modal">
          <div class="caption">
            <div class="caption-content">
              <i class="fa fa-search-plus fa-3x"></i>
            </div>
          </div>
           <!-- j
        </a>
      </div>
      <div class="col-sm-4 portfolio-item">
        <a href="#portfolioModal2" class="portfolio-link" data-toggle="modal">
          <div class="caption">
            <div class="caption-content">
              <i class="fa fa-search-plus fa-3x"></i>
            </div>
          </div>
        </a>
      </div>
    </div>
  </div>
</section>

```

La clase portfolio-item nos permite enlistar las figuras que se pueden seleccionar de la sección de portafolio.

La clase caption nos da el efecto para poder visualizar ese item y la información que contiene.

La referencia #portafolioModal no situa el evento en la figura seccionada.

Figura 27. Código de la WebApp Genios (portafolio-item)

```

. ***** -->
<div id="tf-mexico">
  <div class="container">
    <h1>MÉXICO</h1>
  </div>
</div>

<!-- que hacemos?
===== -->
<div id="tf-services" class="text-center">
  <div class="container">
    <div class="col-md-12 text-center">
      <br> <br>
      <img src='img/que_hacemos_info.png' height="10%"> <
    </div>
  </div>
</div>

<!-- ***** -->
<div id="tf-vintage">
  <div class="container">
    <h1>VINTAGE</h1>
    <br><br>
  </div>
</div>

```

Los id tf-variable es la funcion que llama a la operaci3n dentro de CSS para lograr el efecto de navegaci3n.

Figura 28. C3digo Bootstrap del Se1or Tlacuache

```

<div class="carousel-inner">
  <div class="item active">
    
  </div>
  <div class="item">
    
  </div>
  <div class="item">
    
  </div>
</div>

```

La primera sección muestra la clase que hace imágenes responsivas.

La segunda sección muestra un formulario básico que evalúa la edad del usuario que debe ser mayor de edad.

```

<form name="formulario" action='index.html' method="POST"><!-- "001-month-input-html5.php"-->
  <input type="number" placeholder="Día" name="dia" min="1" max="31" required pattern="[0-9]">
  <input type="number" placeholder="Mes" name="mes" min="1" max="12" required pattern="[0-9]">
  <input type="number" placeholder="Año" name="año" min="1940" max="1997" required pattern="[0-9]">
  <br><br>
  <input type="submit" value="Entrar" />
</form>

```

Figura 29. Código Bootstrap de la Web App de Cervecería Apan

4.3.1. Arquitectura de contenido.

El diseño de la arquitectura de contenido lo podemos tomar de cuatro tipos de estructuras que son:

1. Estructuras lineales (con flujo opcional, con derivaciones).
2. Estructuras de retícula.
3. Estructuras jerárquicas.
4. Estructuras de estructura en red.

Estas pueden ser combinadas dentro del proyecto desarrollado pero siempre se debe buscar la estructura más clara para el usuario evitando el manejo de muchos hipervínculos para llegar a un objetivo.

Para nuestro trabajo la estructura en jerarquía fue la más factible para poder tener una navegación clara en nuestras WebApp.

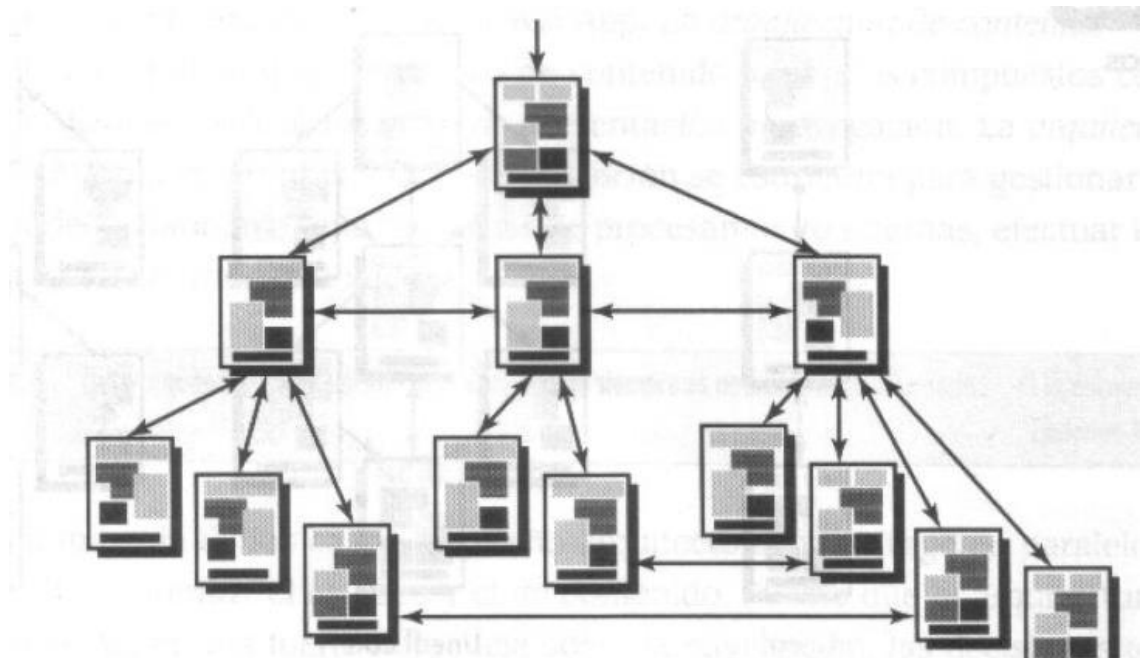


Figura 30. Arquitectura de navegación (Pressman, 2004)

La *figura 30* es la representación de la estructura en forma jerárquica de una Web App este modelo nos muestra que de una página principal se puede ir accediendo a otras subpáginas y así sucesivamente creando varios niveles. Como recomendación se debe tener máximo 4 niveles en la jerarquía para evitar que el usuario se pierda en la navegación.

4.3.2. Arquitectura de Web App.

La arquitectura de una Web App describe una infraestructura que permite que un sistema o aplicación basados en web alcance sus objetivos. Se describen las características básicas de esta infraestructura del modo siguiente:

Las aplicaciones deben construirse con el empleo de capas en las que se tomen en cuenta distintas preocupaciones; en particular, deben separarse los datos de la aplicación de los contenidos de ésta (nodos de navegación), y éstos, a su vez, deben separarse con toda claridad del aspecto y la sensación de la interfaz (páginas).

La arquitectura de modelo de vista controlador es uno de varios modelos sugeridos para la infraestructura de WebApps que desacoplan la interfaz de usuario de sus funciones y contenido informativo.

El modelo (a veces denominado “objeto de modelo”) contiene todo el contenido y la lógica de procesamiento específicos de la aplicación, incluso todos los objetos de contenido, acceso a fuentes de datos o información externos y todas las funciones de procesamiento que son específicas de la aplicación.

La vista contiene todas las funciones específicas de la interfaz y permite la presentación de contenido y lógica de procesamiento, incluidos todos los objetos de contenido, el acceso a fuentes de datos o información del exterior y todas las funciones de procesamiento que requiere el usuario final.

El controlador administra el acceso al modelo y la vista, y coordina el flujo de datos entre ellos. En una Web App, “la vista es actualizada por el controlador con datos del modelo, basándose en las entradas que da el usuario”. En la figura 31 se muestra una representación de la arquitectura MVC.

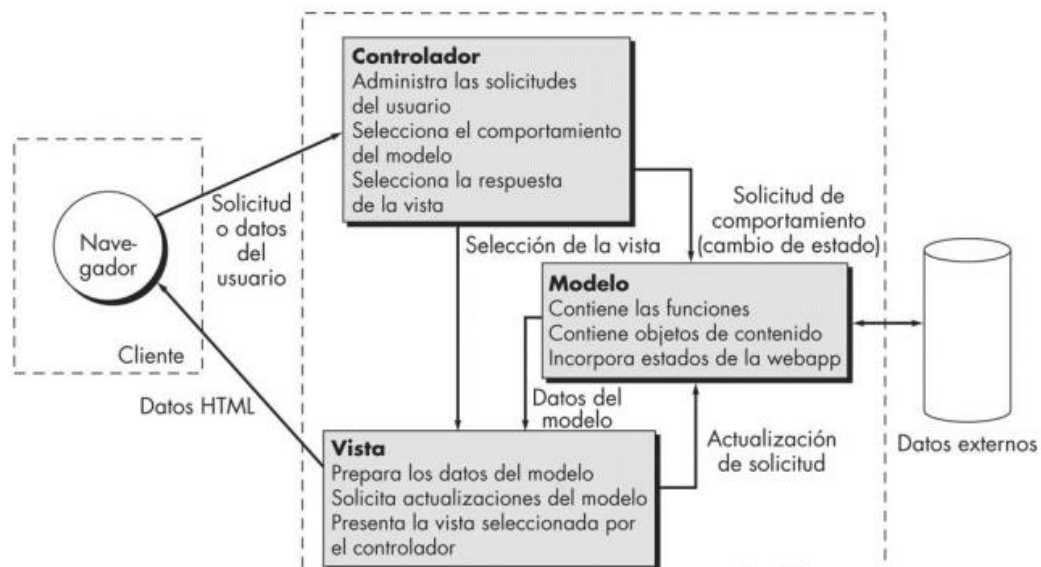


Figura 31. Modelo Vista controlador MVC

4.4. Diseño de navegación.

Una vez que la arquitectura de la Web App ha sido establecida y se han identificado sus componentes (páginas, textos, subprogramas y otras funciones de procesamiento), deben habilitarse las rutas de navegación que permitan a los usuarios acceder al contenido y a sus funciones. Para lograr esto, debe hacerse lo siguiente:

1) Identificar la semántica de navegación:

La semántica se aplica para cada categoría de usuario que se encontró en el análisis del proyecto, para cada uno se elaboró un caso de uso y estos casos servirán para que conforme el usuario interactúe en la Web App encuentre una serie de Unidades Semánticas de Navegación (USN). (Cachero, 2002)

Gnaho y Larcher (Gnaho, 1999) describen las USN de la siguiente forma:

Una estructura que representa la mejor forma o ruta de navegación para los usuarios con ciertos perfiles para lograr sus metas.

2) Definir la sintaxis de navegación.

La sintaxis en la forma de presentar los vínculos de navegación que se habiliten en la semántica de navegación, estos pueden ser:

- Vinculo de navegación individual: Vínculos basados en texto, iconos, botones y metáforas gráficas.
- Barra de navegación horizontal: Lista de principales categorías de contenido o funcionales en una barra con vínculos adecuados.

- Columna de navegación vertical: Se muestra la misma lista de la barra de navegación horizontal.
- Pestañas: Una metáfora que no es más que la variación de la barra o columna de navegación
- Mapas de sitio: Una tabla de contenido incluyente para la navegación.

En nuestro proyecto seleccionamos como sintaxis de navegación los vínculos individuales por ser una manera amigable y muy sencilla de manejar con el Framework Bootstrap.

Cabe mencionar que sea la sintaxis que sea, nosotros como desarrolladores debemos encontrar auxiliares de navegación para todo tipo de usuario, estos auxiliares se aplican en los estilos de CSS3 que los maneja el evento de *.hover* para que se pueda establecer por ejemplo, cambio de colores al toque, sombreados, animaciones y hasta sonido.

Con estos cambios no solo nos enfocamos a la categoría de usuarios encontrada en el trabajo, sino que permitimos crear aplicaciones web con accesibilidad para toda clase de usuario con alguna discapacidad.

4.5. Diseño de interfaz y generación de páginas.

Nuestro diseño de interfaz prácticamente está concluido hasta este paso, lo que resta es unir todas las clases y planillas que se realizaron por separado y fuera del código de Bootstrap, como los efectos de cada WebApp que se realizaron con códigos JavaScript.

En este capítulo mostramos los resultados del diseño de interfaz y la generación de páginas que se realizó antes de hacer la entrega final. En este momento ya se han realizado pequeñas pruebas en búsqueda de errores principalmente en la ortografía de contenido.

La generación de páginas se observa en las figuras 32, 33, 34 y 35 que muestran gráficamente el resultado final de las Web App con formato para monitores medianos a grandes previo a la evaluación.



Figura 32. Vista final Web App CEO Consulting

GENiOS

BUSINESS FACTORY

GENIOS BUSINESS MAKER S.A. DE C.V.

Nace con la finalidad de comercializar diferentes servicios y ser el componente coordinador de diversas empresas integrantes del grupo.

NUESTRAS EMPRESAS



Figura 33. Vista final Web App Genios

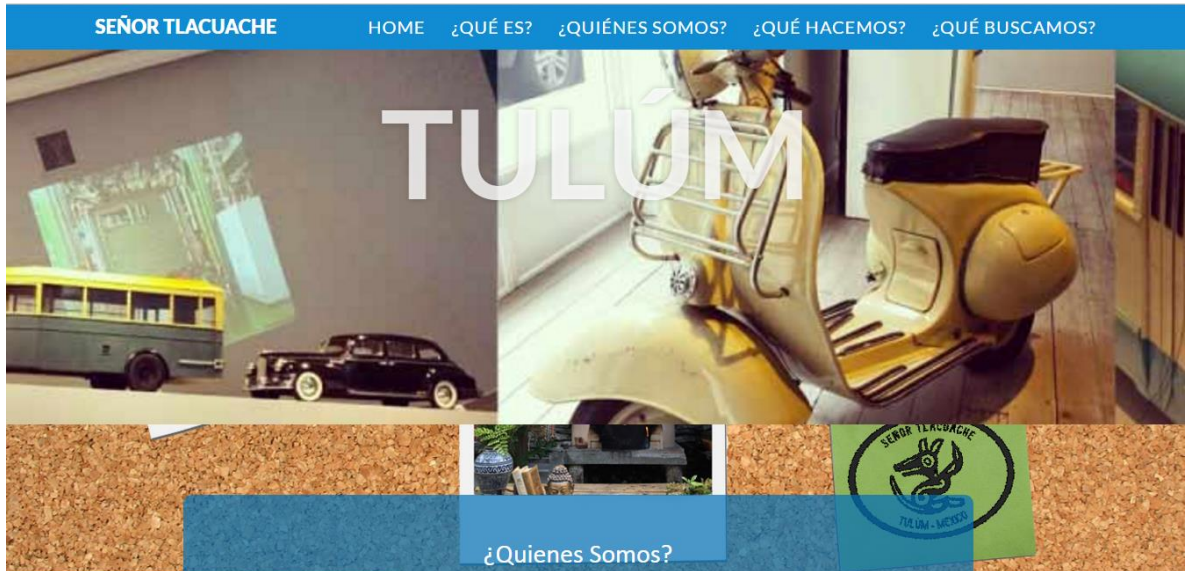


Figura 34. Vista final Web App Señor Tlacuache



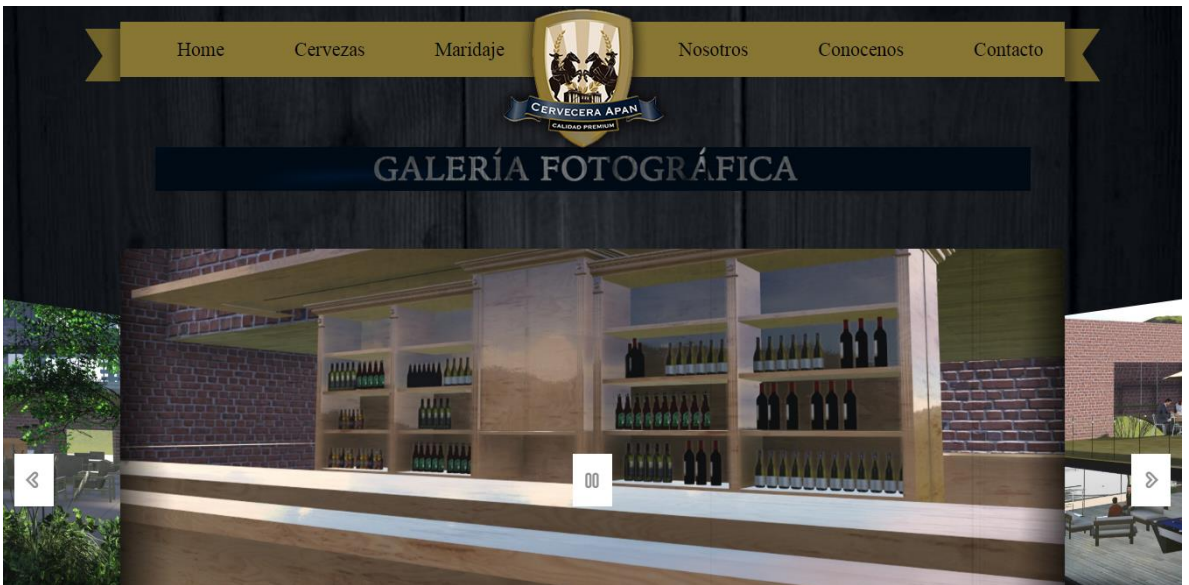


Figura 35. Vista final Web App Cervecería Apan

Las figuras 36 y 37 muestran capturas de pantalla del proyecto finalizado en el formato para pantallas para cualquier dispositivo móvil (Smartphone). En este proyecto se mencionó que contaría con aplicaciones responsivas.



Figura 36. Vista responsiva de Genios



Figura 37. Vista responsiva de Cervecería Apan

Test y Evaluación.

El proceso de someter a prueba nuestras WebApps es el conjunto de actividades que se realiza sobre ellas con la meta de detectar errores ya sean de contenido, navegación, desempeño y seguridad. Empezamos enfocándonos en el aspecto visible de la web para luego probar la arquitectura e infraestructura interna, en este caso el código de Bootstrap. Estas pruebas constan de siete etapas:

- ✓ El contenido: se evaluará tanto el ámbito sintáctico como semántico. En la parte sintáctica, la ortografía, la puntuación y la gramática se valoran para la plantilla. En el ámbito semántico se evaluará la exactitud de la información la consistencia y la falta de ambigüedad.
- ✓ La función: evalúa la exactitud, inestabilidad y concordancia de los estándares de implementación, es decir, estándares de lenguaje.
- ✓ La estructura: valora que la entrega de contenido sea adecuada
- ✓ La facilidad de uso.
- ✓ La navegabilidad: se pone a prueba para garantizar que toda la sintaxis y semántica de navegación no tenga errores como vínculos rotos, inadecuados o erróneos.
- ✓ El desempeño: se prueba en una diversidad de sistemas con diferentes configuraciones.

- ✓ La compatibilidad: se prueban las configuraciones tanto de lado del servidor como del cliente.
- ✓ La interoperabilidad: asegura que el Framework realiza interfaces adecuadas en otras aplicaciones o bases de datos.
- ✓ La seguridad: prueba las vulnerabilidades potenciales intenta pasar por cada una de ellas.

Además de verificar los puntos antes mencionados, debemos comprobar que nuestra Web App tenga usabilidad. La usabilidad en la ingeniería Web se puede entender como el que tan fácil es para el usuario final poder navegar en nuestra aplicación.

5.1. Métodos de Prueba.

Respetando la formalidad de nuestro proceso de creación de aplicaciones web, nuestras Web App fueron sometidas a testeos utilizando métodos formales de pruebas de software.

Los métodos de prueba los utilizamos para intentar mostrar todos los errores que puede tener nuestra aplicación web. Existen varias formas de comprobar los errores en nuestros sistemas, en este caso explicaremos dos, que son:

1. Método de caja negra: Realiza pruebas que comprueban la funcionalidad operativa de nuestra Web App.
2. Método de caja blanca: Desarrolla pruebas para comprobar que el funcionamiento interno de la aplicación se ajusta a los requerimientos.

En nuestro caso utilizamos el método de caja negra que se centra más en la evaluación sobre la interfaz, obviando el comportamiento interno. Este método pretende comprobar:

- Que las funciones son operativas
- La entrada de datos se acepta de forma correcta
- Se produce una salida correcta
- La integridad de la información externa se mantiene (J., 2004)

Cada uno de estos puntos de calidad se pone a prueba durante el proceso de desarrollo de nuestra plantilla, pero una vez generadas las páginas se debe repetir para poder garantizar que nuestro Framework opera de la mejor manera en la mayor gama posible de navegadores y dispositivos móviles.

La evaluación que hizo el supervisor de las plantillas de Bootstrap fueron con buenos resultados solo se hicieron modificaciones mínimas en ortografía, colores de interfaz y se extendió la planeación a largo plazo de nuestras aplicaciones donde se contempló en un futuro realizar nuevos estudios para mejorar la interacción humano-computadora.

Evaluando cada uno de los puntos en una escala del 1 al 5 siendo 1 una mala evaluación y 5 el puntaje más alto las Web App tuvieron en la detección de errores las siguientes calificaciones que se muestran en la tabla 7. Esta tabla sirve de ejemplo para ejemplificar nuestro último capítulo.

	Contenido	Facilidad	Navegación	Desempeño	Compatibilidad	Interpolaridad	Seguridad
CEO	9	10	9	9	10	9	9
GENIOS	7	10	10	10	9	9	9
S. Tlacuache	9	10	10	10	9	9	9
Apan	8	9	7	10	10	9	9

Tabla 7. Evaluación de pruebas de Web App

Conclusiones.

Con el proceso correcto de la ingeniería web se logró cumplir los objetivos del proyecto de tesis y poder satisfacer las necesidades requeridas por la consultoría, que como meta principal fue la modificación de un OpenSource a bajo nivel para la mejora de su interfaz humano-computadora, así también se logró una usabilidad adecuada para la intercomunicación de dispositivos móviles, de esta forma se modeló e implementó mejoras en las plantillas del Framework Bootstrap.

Después del análisis llevado a cabo en este trabajo se puede concluir que una correcta aplicación de la ingeniería web y un buen manejo de herramientas Framework como Bootstrap puede crear contenido dinámico y llamativo a la vista de cualquier tipo de usuario que visite la WebApp, además de que estas herramientas han probado que es necesario ya desarrollar aplicaciones que no se limiten a computadoras personales, sino también deben poder visualizarse en cualquier nuevo dispositivo móvil con acceso a internet.

El buen desarrollo de la ingeniería web nos permitió tener aplicaciones web con una interfaz llamativa con los efectos deseados por la consultora que los solicitó. Si bien, no se logró que todas las aplicaciones fueran completamente responsivas, se planteó hacer modificaciones para poder alcanzar este objetivo. Además, se espera que a futuro se pueda implementar mejoras en el contenido de la información e implementar funcionalidades como la venta de productos.

Bibliografía.

- Cachero, C. e. (2002). *Conceptual Navigation Analysis: a Device and Platform Independent*. Workshop on Web Oriented Tecnology .
- Casario, M. E. (2011). *HTML% Solutions: Essential Techniques for HTML5 developers*. Springer.
- F., V. G. (2010). *OOWS 2.0 Un metodo de ingeniería web dirigido a modelos para la producción de aplicaciones web*. Valencia: Tesis Doctoral.
- Frain, B. (2012). *Responsive Web Design with*. Packt Publishing .
- Gaedke, P. D. (n.d.). *webengineering*. Retrieved from www.webengineering.org/Home/About
- Gauchat, J. D. (2012). HTML5. In *El gran libro de HTML5, CSS3 y JavaScript* (pp. 1-27). Marcombo.
- Gnaho, C. y. (1999). *A User-Centered Methology for Complex and Customizable Web Engeenering*. Los Angeles : ACM.
- Gutiérrez., J. J. (2014, febrero 27). *Isi*. Retrieved from http://www.isi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- J., G. (2004). *The Art of Sostware Testing*. Myers, Jonh Wiley y Sons.
- Jesús Lorés, T. G. (2001). *La ingenieria de la usabilidad y de la accesibilidad aplicada al diseño y desarrollo de sitios web*. Lleida: Universidad de Lleida.
- Kaiser, J. (2002). *Elements of Effective Web Design*.
- Mark Otto, J. T. (2014). *Bootstrap 3, el manual oficial*.
- Miller, E. (2000). *WebSite Testing* .
- Nguyen, H. (2001). *Testing Applications on the Web* . Wiley.
- Olsina, L. (1999). *Specifyng Quality Characteristic and attributes for web sites*. Los Angeles: ACM.
- Powell, T. (1998). *Web Site Engineering*. Prentice Hall.

- Pressman, R. (2004). *Software Engineering: A practitioner's perspective*. McGraw-Hill.
- Rakitin, S. R. (2001). *Software Verification and Validation for Practitioners and Managers*. Artech House.
- Schulz, R. G. (2009). *Diseño Web con CSS*. Marcombo.
- Técnicas Avanzadas de Diseño Web*. (2010). Vertice.