



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**DESARROLLO DE APLICACIÓN WEB CON DISEÑO *RESPONSIVE* ORIENTADA A  
OFRECER A LOS USUARIOS UNA CARTELERA DE PROMOCIONES DE DISTINTOS  
ESTABLECIMIENTOS: “FINDFUN”**

**Francisco Antonio Noj Avendaño**

**Lester René Vásquez Dardón**

Asesorado por el Ing. José Alfredo González Díaz

Guatemala, mayo de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DESARROLLO DE APLICACIÓN WEB CON DISEÑO *RESPONSIVE* ORIENTADA A  
OFRECER A LOS USUARIOS UNA CARTELERA DE PROMOCIONES DE DISTINTOS  
ESTABLECIMIENTOS: “FINDFUN”**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**FRANCISCO ANTONIO NOJ AVENDAÑO  
LESTER RENÉ VÁSQUEZ DARDÓN**  
ASESORADO POR EL ING. JOSÉ ALFREDO GONZÁLEZ DÍAZ

AL CONFERÍRSELES EL TÍTULO DE  
**INGENIEROS EN CIENCIAS Y SISTEMAS**

GUATEMALA, MAYO DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Sergio Arnoldo Méndez Aguilar
EXAMINADOR	Ing. José Alfredo González Díaz
EXAMINADOR	Ing. William Samuel Guevara Orellana
SECRETARIA	Inga. Lesbia Magalí Herrera López

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presentamos a su consideración nuestro trabajo de graduación titulado:

**DESARROLLO DE APLICACIÓN WEB CON DISEÑO *RESPONSIVE* ORIENTADA A OFRECER A LOS USUARIOS UNA CARTELERA DE PROMOCIONES DE DISTINTOS ESTABLECIMIENTOS: “FINDFUN”**

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 11 de octubre de 2016.

**Lester René Vásquez Dardón**

**Francisco Antonio Noj Avendaño**



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería

**A QUIEN INTERESE:**

Por este medio hago constar que **Francisco Antonio Noj Avendaño** quien se identifica con número de carnet **201114432** y **Lester René Vásquez Dardón** quien se identifica con carnet **201020320**, ambos de la carrera de ingeniería en ciencias y sistemas, me han presentado la totalidad de su trabajo de graduación, ante el cual doy el visto bueno al trabajo de investigación titulado **DESARROLLO DE APLICACIÓN WEB CON DISEÑO RESPONSIVE ORIENTADA A OFRECER A LOS USUARIOS UNA CARTELERA DE PROMOCIONES DE DISTINTOS ESTABLECIMIENTOS: "FIND FUN"**.

Y para los usos que los interesados convengan, extendiendo la presente constancia a los 21 días del mes de marzo de 2017

**Atentamente,**

José Alfredo González Díaz  
Ingeniero en Ciencias y Sistemas  
Colegiado 6757

**Ing. José Alfredo González Díaz**  
**Asesor del Trabajo de Investigación**



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 26 de Abril del 2017

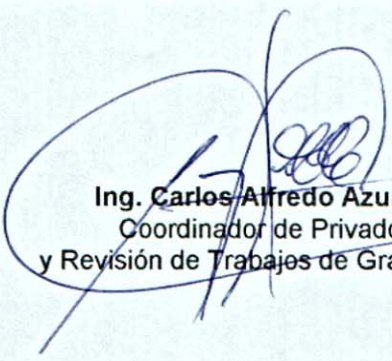
Ingeniero  
**Marlon Antonio Pérez Türk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **FRANCISCO ANTONIO NOJ AVENDAÑO** con carné **201114432** y **CUI 2225 84092 0101**, y **LESTER RENÉ VÁSQUEZ DARDÓN** con carné **201020320** y **CUI 2200 48584 0101**, titulado: **"DESARROLLO DE APLICACIÓN WEB CON DISEÑO RESPONSIVE ORIENTADA A OFRECER A LOS USUARIOS UNA CARTELERA DE PROMOCIONES DE DISTINTOS ESTABLECIMIENTOS: FINDFUN"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
CIENCIAS Y SISTEMAS  
TEL: 24188000 Ext. 1534

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **DESARROLLO DE APLICACIÓN WEB CON DISEÑO RESPONSIVE ORIENTADA A OFRECER A LOS USUARIOS UNA CARTELERA DE PROMOCIONES DE DISTINTOS ESTABLECIMIENTOS: "FINDFUN"**, realizado por los estudiantes, FRANCISCO ANTONIO NOJ AVENDAÑO y LESTER RENÉ VÁSQUEZ DARDÓN, aprueba el presente trabajo y solicita la autorización del mismo.*

**"ID Y ENSEÑADA A TODOS"**

  
Ing. Marcos Antonio Pérez Turck  
Director

*Escuela de Ingeniería en Ciencias y Sistemas*



Guatemala, 16 de mayo de 2016

Universidad de San Carlos  
de Guatemala

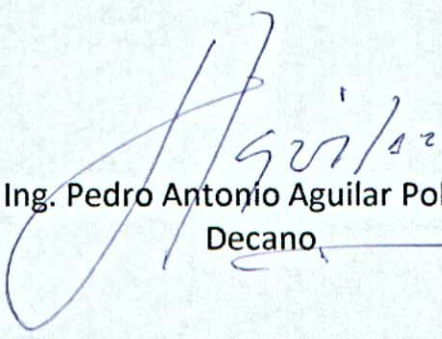


Facultad de Ingeniería  
Decanato

DTG. 227.2017

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DESARROLLO DE APLICACIÓN WEB CON DISEÑO RESPONSIVE ORIENTADA A OFRECER A LOS USUARIOS UNA CARTELERA DE PROMOCIONES DE DISTINTOS ESTABLECIMIENTOS: "FINDFUN"**, presentado por los estudiantes universitarios: **Francisco Antonio Noj Avendaño y Lester René Vásquez Dardón**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. Pedro Antonio Aguilar Polanco  
Decano



Guatemala, mayo de 2017

/gdech



## **ACTO QUE DEDICO A:**

- Dios** Por darme el regalo de la vida, la fuerza, la sabiduría a lo largo de la carrera.
- Mis padres** Edgar Noj y Johanna Avendaño, por todo su apoyo a lo largo de la carrera, sus consejos y confianza para brindarme educación.
- Mis hermanos** Luis Noj y Dania Noj, por su apoyo, compañía y cariño durante todos estos años.
- Mi novia** Abigail Alvizures, por tu amor incondicional, y todo tu apoyo desde el inicio de la carrera hasta este momento.

**Francisco Antonio Noj Avendaño**

## **ACTO QUE DEDICO A:**

- Dios** Por darme el regalo de la vida, la fuerza, la sabiduría a lo largo de la carrera.
- Mis padres** Hugo Vásquez y Milvia Dardón, por todo su apoyo a lo largo de la carrera, sus consejos y confianza para brindarme educación.
- Mis hermanos** David Vásquez y Ligia Vásquez, por su apoyo, compañía y cariño durante todos estos años.
- Mi novia** Jennifer Rios, por tu amor incondicional, y todo tu apoyo desde el inicio de la carrera hasta este momento.

**Lester René Vásquez Dardón**

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Por ser mi casa de estudio y segunda casa.

**Facultad de Ingeniería**

Por brindarme los conocimientos adquiridos.

**Mis amigos de la Facultad**

Por todos los momentos compartidos, metas y este gran sueño

**Mi compañero en este  
trabajo de graduación**

Lester Vásquez, por todas las noches de desvelos invertidas y la dedicación que nos lleva hoy a concluir este proyecto.

**Mis padres**

Porque por ellos me encuentro hoy a punto de concluir esta gran meta.

**Francisco Antonio Noj Avendaño**

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Por ser mi casa de estudio y segunda casa.

**Facultad de Ingeniería**

Por brindarme los conocimientos adquiridos.

**Mis amigos de la Facultad**

Por todos los momentos compartidos, metas y este gran sueño.

**Mi compañero en este  
trabajo de graduación**

Francisco Noj, por todas las noches de desvelos invertidas y la dedicación que nos lleva hoy a concluir este proyecto.

**Mis padres**

Porque por ellos me encuentro hoy a punto de concluir esta gran meta.

**Lester René Vásquez Dardón**

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
GLOSARIO .....	IX
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN .....	XV
1. IDENTIFICACIÓN DEL PROBLEMA ANTECEDENTES.....	1
1.1. Definición y aspectos generales del <i>marketing</i> .....	3
1.1.1. ¿Qué es <i>marketing</i> ? .....	3
1.2. Las P de la mercadotecnia .....	3
1.2.1. Producto .....	4
1.2.2. Precio .....	4
1.2.3. Plaza.....	5
1.2.4. Promoción.....	5
1.2.5. Personal .....	5
1.2.6. Presentación (evidencia física) .....	6
1.2.7. Procesos.....	6
1.3. El <i>marketing</i> digital .....	7
1.4. Estrategias actuales del <i>marketing</i> a nivel nacional .....	8
1.4.1. Poseer un <i>website</i> .....	8
1.4.2. Hacer <i>marketing</i> por email.....	8
1.4.3. Utilizar redes sociales.....	9
1.4.4. Crear eventos .....	9
1.4.4.1. 2 x 1 .....	9
1.4.4.2. <i>All you can eat and drink</i> .....	9

	1.4.4.3.	<i>Ladies night</i> .....	10
	1.4.4.4.	El segundo a mitad de precio .....	10
	1.4.4.5.	Combos .....	10
1.5.		Comunidades virtuales y su relación con las empresas.....	10
	1.5.1.	Comunidad virtual .....	10
	1.5.2.	Estrategia indiferenciada .....	11
	1.5.3.	Uso de las comunidades virtuales.....	11
1.6.		Mercado objetivo.....	12
	1.6.1.	<i>Benchmarking</i> de la aplicación.....	12
2.		INVESTIGACIÓN DEL PROBLEMA Y SOLUCIONES	
		PROPUESTAS ANTERIORMENTE.....	15
2.1.		Descripción del problema.....	15
2.2.		Soluciones propuestas en otros países.....	16
	2.2.1.	Yelp .....	16
	2.2.2.	App móvil Eltenedor .....	17
	2.2.3.	TripAdvisor .....	18
2.3.		Soluciones propuestas en Guatemala.....	19
	2.3.1.	Mister Menú.....	20
2.4.		Debilidades identificadas.....	21
3.		DISEÑO DE LA APLICACIÓN SEGÚN LA NECESIDAD	
		IDENTIFICADA.....	23
3.1.		Necesidades específicas identificadas.....	23
	3.1.1.	Centralización de promociones .....	23
	3.1.2.	Publicación de promociones durante su periodo de vigencia .....	23
3.2.		Prototipos.....	23
	3.2.1.	<i>Login</i> .....	24

3.2.2.	Carga de promociones .....	24
3.2.3.	Promociones públicas.....	25
3.2.4.	Expandir publicación.....	26
3.3.	Definición de casos de uso.....	27
3.4.	Servidor de base de datos y conexiones .....	30
3.5.	Vistas y validaciones de la arquitectura de software .....	31
3.5.1.	Vista funcional .....	31
3.5.1.1.	Justificación .....	31
3.5.1.2.	Concerns .....	31
3.5.1.2.1.	Capacidades funcionales .....	31
3.5.1.2.2.	Interfaces externas .....	32
3.5.1.3.	Modelo.....	32
3.5.1.3.1.	Elementos funcionales .....	32
3.5.1.3.2.	Interfaces.....	32
3.5.1.3.3.	Conectores .....	32
3.5.2.	Vista de información .....	33
3.5.2.1.	Justificación .....	33
3.5.2.2.	<i>Concerns</i> .....	33
3.5.2.3.	Modelo.....	34
3.5.2.3.1.	Modelo de almacenamiento .....	34
3.5.2.3.2.	Flujo de información .....	34
3.5.2.3.3.	¿Dónde se crea y se destruye la información? .....	34
3.5.2.3.4.	¿Dónde son consultados,	

	modificados o enriquecidos los datos? .....	34
3.5.3.	Vista de deployment .....	35
3.5.3.1.	Justificación .....	35
3.5.3.2.	<i>Concerns</i> .....	35
3.5.3.2.1.	Plataforma de ejecución requerida .....	35
3.5.3.2.2.	Requerimientos de software de terceros .....	36
3.5.3.3.	Modelo .....	36
3.5.3.3.1.	Elementos .....	36
3.5.3.4.	Validaciones .....	37
3.5.4.	Diseño intuitivo y usabilidad .....	38
4.	DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN .....	41
4.1.	Requerimientos de hardware .....	41
4.2.	Herramientas .....	41
4.2.1.	Repositorio de control de versiones de github .....	42
4.2.2.	Cliente github desktop .....	42
4.2.3.	Editor de texto Atom .....	43
4.2.4.	JavaScript .....	44
4.2.5.	NodeJs .....	45
4.2.6.	MySQL .....	45
4.2.7.	AngularJs .....	46
4.2.7.1.	<i>Scopes</i> .....	46
4.2.7.2.	<i>Controllers</i> .....	46
4.3.	Tutorial de desarrollo y referencias .....	46



4.3.1.	Implementación del modelo de datos .....	47
4.3.1.1.	Desarrollo de procedimientos almacenados .....	48
4.3.2.	Instalación de node Js y angular Js.....	49
4.3.3.	Desarrollo de capa lógica y presentación de la aplicación.....	52
4.3.3.1.	Capa lógica.....	53
4.3.3.1.1.	Conexión con la base de datos.....	54
4.3.3.1.2.	Consumo de procedimientos almacenados desde la aplicación .....	54
4.3.3.1.3.	Envío de correos por suscripción.....	55
4.3.3.2.	Capa de presentación.....	56
4.3.3.2.1.	Vista de <i>login</i> .....	57
4.3.3.2.2.	Vista crear promoción...	59
4.3.3.2.3.	Vista de promociones activas .....	61
4.3.3.2.4.	Vista de ampliación de promoción.....	63
4.3.3.2.5.	Vista suscripción por email.....	64
4.3.3.2.6.	Vista para edición de perfil.....	66
4.4.	Consideraciones de implementación .....	68

CONCLUSIONES.....69  
RECOMENDACIONES .....71  
BIBLIOGRAFÍA.....73

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Las P de la mercadotecnia.....	7
2.	Captura de Mister Menú.....	13
3.	Captura de aplicación Yelp .....	17
4.	Captura de aplicación Eltenedor .....	18
5.	Captura de aplicación TripAdvisor .....	19
6.	Captura de aplicación Mister Menú.....	20
7.	Prototipo de <i>login</i> .....	24
8.	Prototipo de carga de promociones .....	25
9.	Prototipo de promociones públicas .....	26
10.	Prototipo de expandir publicación .....	27
11.	Modelo entidad relación de la base de datos .....	30
12.	Configuración de github desktop.....	43
13.	Captura de atom .....	44
14.	Ejemplo de código SQL para la creación de tablas de la base de datos .....	48
15.	Ejemplo de procedimiento almacenado con consulta a través de clausula <i>select</i> .....	49
16.	Instalación de node js.....	50
17.	Instalación express.....	50
18.	Creación de cuerpo de la aplicación .....	51
19.	Instalación de dependencias de angular .....	52
20.	Árbol de archivos de la aplicación.....	53
21.	Conexión con la base de datos .....	54

22.	Consumo de procedimiento almacenado desde el código de la aplicación.....	55
23.	Envío de correo con nodemailer .....	56
24.	Fragmento de HTML con angular del <i>login</i> .....	57
25.	Vista de <i>login</i> .....	58
26.	Fragmento de función javascript para <i>login</i> .....	58
27.	Fragmento de HTML con angular de la vista creación de promociones .....	60
28.	Vista de creación de promociones .....	61
29.	Fragmento de código HTML con angular de la vista de promociones activas .....	62
30.	Vista de promociones activas .....	62
31.	Fragmento de código HTML con angular de la vista de promoción ampliada .....	63
32.	Vista ampliación de promoción .....	64
33.	Fragmento de código HTML con angular de la vista de suscripción por <i>email</i> .....	65
34.	Vista de suscripción por <i>email</i> .....	66
35.	Fragmento de código HTML con angular para la vista para edición de perfil .....	67
36.	Vista para edición de perfil.....	68

## TABLAS

I.	Caso de uso, registrar usuario .....	28
II.	Caso de uso, <i>Login</i> .....	28
III.	Caso de uso, carga de promociones .....	29
IV.	Caso de uso, visualizar promociones .....	29

## GLOSARIO

<b>Aplicación <i>responsive</i></b>	Aplicación web que se adapta a cualquier tamaño de pantalla de cualquier navegador web, ya sea este un ordenador, una tableta, un móvil, o cualquier otro dispositivo.
<b>Base de datos</b>	Colección de datos que están debidamente organizados y estructurados.
<b><i>Data warehouse</i></b>	Almacén de datos orientado a determinado ámbito de una empresa, no volátil y variable con el tiempo.
<b>DBMS</b>	Sistema de gestión de base de datos.
<b><i>Framework</i></b>	Es la estructura o armazón conceptual de soporte de cualquier aplicación, con artefactos concretos de software, que sirven de base para el desarrollo de software.
<b>JavaScript</b>	Lenguaje de programación interpretado.
<b>JSON</b>	Javascript <i>object notation</i> .
<b>Localhost</b>	Dirección <i>ip</i> que corresponde al mismo equipo.

***Marketing***

Disciplina que se dedica al estudio y análisis del comportamiento de los mercados y de los consumidores.

**Redes sociales**

Ambiente donde personas se relacionan entre sí.

**Software**

Soporte lógico de todo sistema informático.

## RESUMEN

En su mayoría, los desarrollos web solo se adaptan a un tamaño fijo de pantalla, debido a la gran variedad de dispositivos que en la actualidad existen, y por las técnicas y herramientas de cada uno. En el presente trabajo se plantea una solución a estas dificultades; a través de su desarrollo se ofrece a la población nacional una alternativa para visualizar las diferentes promociones que las distintas empresas de hoy en día.

El primer y segundo capítulos presentan la investigación sobre el desarrollo *responsive*, los antecedentes, aspectos del *marketing*. Se presenta el mercado objetivo de la aplicación; posteriormente, se describen soluciones similares en otros países y en Guatemala para concluir con el *benchmarking*; sí también, las debilidades encontradas en el desarrollo de la aplicación.

En los capítulos posteriores la orientación del trabajo se enfoca en el desarrollo de una aplicación *responsive*; se utiliza las herramientas y los resultados de los capítulos iniciales, de tal manera que la aplicación resultante sea beneficiosa para toda la población guatemalteca: ofrece facilidades en la comparación de promociones de distintas empresas.





# OBJETIVOS

## General

Desarrollar e implementar una aplicación con despliegue de promociones para distintos mercados en Guatemala de fácil uso en cualquier dispositivo.

## Específicos

1. Facilitar la consulta de promociones de distintas empresas desde cualquier dispositivo con conexión a internet.
2. Permitir el almacenamiento de promociones de las distintas empresas que utilicen el servicio.
3. Desarrollar una solución de software que envíe por medio de correos electrónicos las distintas promociones del día a los usuarios suscritos.



## INTRODUCCIÓN

El presente trabajo da a conocer una solución de software basada en el desarrollo de aplicaciones web *responsive* que tiene como objetivo incrementar el impacto de mercado y publicidad de las empresas que distribuyen servicios y productos de uso común; también, se abordarán las tecnologías específicas y todos los requisitos de software y de hardware para el desarrollo de la aplicación.

Para lo cual se tomaron en cuenta varios aspectos importantes para hacer pública una promoción de una empresa: precio, ubicación y afluencia.

La aplicación está destinada a todo usuario con un dispositivo con acceso a internet. Con el desarrollo de esta aplicación se pretende mejorar la cantidad de ventas de las distintas empresas; así mismo, aumentar el interés de diversos usuarios en territorio nacional para utilizar herramientas tecnológicas.



## 1. IDENTIFICACIÓN DEL PROBLEMA ANTECEDENTES

La utilización de nuevas estrategias en el ámbito del *marketing* en Guatemala ha llevado a la tecnología a estar en constante actualización para proveer al mercado de las herramientas necesarias para optimizar cualquier proceso, en este caso las estrategias de *marketing*.

El *marketing* en Guatemala ha tenido un gran impacto gracias a las empresas que promocionan su producto y a la tecnología; hoy en día, las redes sociales son una herramienta muy grande para estas empresas ya que llegan a más clientes de forma inmediata.

Una de las estrategias utilizados actualmente es la de las promociones en determinados días y horarios; este tipo de información es publicada por las empresas constantemente en sus redes sociales; sin embargo, estas promociones únicamente llegan a los consumidores del producto de dicha empresa y no busca ampliar su mercado: si las personas siguen a estas empresas por las redes sociales se deduce que ya han consumido sus productos, por lo tanto, se identifica la necesidad de centralizar este tipo de promociones para que las empresas puedan ampliar su mercado y los consumidores puedan tomar una mejor decisión a la hora de consumir un producto: comparando las diferentes promociones de las empresas que en la herramienta se anuncien.

Actualmente, existen diferentes herramientas: aplicaciones web y móviles que pretenden guiar a los consumidores con una lista de las empresas con servicio de restaurante o bar; sin embargo, estas herramientas sirven para que

la empresa se da a conocer y sea valorada por los usuarios de la aplicación pero no se ofrecen las promociones de la empresa; el enfoque principal es dar a conocer el restaurante y emitir su valoración.

La aplicación Mister Menu es una aplicación móvil que actualmente está disponible para los sistemas operativos móviles Android; esta aplicación posee una interfaz intuitiva que recomienda a los usuarios los distintos restaurantes en Guatemala; permite valorar dichos restaurantes; también, administra la parte del menú; además, se incluye información relevante del restaurante: número telefónico, su ubicación, servicios, horarios y fotos que los usuarios deseen agregar del establecimiento. También, posee un motor de búsqueda para realizar una búsqueda filtrada por ciudad, zona, tipo de comida, servicios, ofertas y precio. Para acceder a la aplicación es necesario tener una cuenta que otorga al usuario un perfil editable según su preferencia. Esta aplicación, actualmente, es una de las más importantes en lo que se refiere a una guía de restaurantes; sin embargo, al consumidor solo se le brinda una valoración del restaurante mas no las promociones con las que cuenta, con lo cual el consumidor podría también obtener un beneficio extra si obtuviera esta información.

La aplicación FindFun es una alternativa a este tipo de propuestas ya que se enfoca exclusivamente a mostrar las promociones de estas empresas según el horario establecido previamente por la empresa para brindar a los usuarios un panorama más amplio para tomar la mejor decisión en cuanto al consumo.

## **1.1. Definición y aspectos generales del *marketing***

### **1.1.1. ¿Qué es *marketing*?**

El *marketing* (mercadotecnia); es la disciplina cuyo principal objetivo es el análisis del comportamiento del mercado y de los consumidores activos, busca satisfacer las necesidades del consumidor y no las necesidades de la empresa.

Para que el *marketing* se dé es necesario estimular ciertos intercambios para lo cual se deben tener 4 condiciones:

- Se requiere la participación de un mínimo de 2 personas: comprador y vendedor.
- Cada parte debe poner algo de valor que la otra parte desea poseer: servicio, bien o dinero.
- Cada parte debe estar dispuesta a ceder su parte de valor.
- Las partes tienen que tener la posibilidad de comunicarse entre sí: oral o escrito.

Si alguna de estas condiciones no se cumplen una de las partes puede quedar insatisfecho haciendo que este tipo de intercambio no vuelve a ocurrir entre las mismas personas.

## **1.2. Las P de la mercadotecnia**

Las P del *marketing* son herramientas utilizadas por toda empresa para implementar ciertas estrategias de mercado y alcanzar los objetivos planteados.

Los autores no llegan a un número fijo de las variables o elementos de las P del *marketing*; unos mencionan solamente 4 elementos (figura 1, círculos de color verde); nuevos estudios indica 7 elementos: los 4 ya definidos por los otros autores más 3 nuevos elementos (figura 1, círculos de color azul), cada una presenta sus propias características.

### **1.2.1. Producto**

Es cualquier bien o servicio que la empresa ofrece a un cliente. Sus características principales son:

- Calidad
- Imagen
- Marca
- Características
- Variantes
- Soporte
- Disponibilidad

### **1.2.2. Precio**

Es el valor económico con el cual se crea el intercambio con el producto. Sus características principales son:

- Posicionamiento
- Lista
- Crédito
- Métodos de pago
- Descuentos



### **1.2.3. Plaza**

Lugar donde ocurre la venta. Sus características principales son:

- Canales de comercio
- Soporte de ventas
- Número de canales
- Canales segmentos

### **1.2.4. Promoción**

La publicidad que se le da a un determinado producto. Sus características principales son:

- La publicidad
- Las relaciones públicas
- La venta personal
- La promoción de ventas

### **1.2.5. Personal**

El personal necesario de la empresa para atender a los clientes. Sus características principales son:

- Reclutamiento
- Remuneración
- Cultura
- Contacto con los clientes
- Actividades de mercadotecnia

### **1.2.6. Presentación (evidencia física)**

La presentación del entorno físico. Sus características principales son:

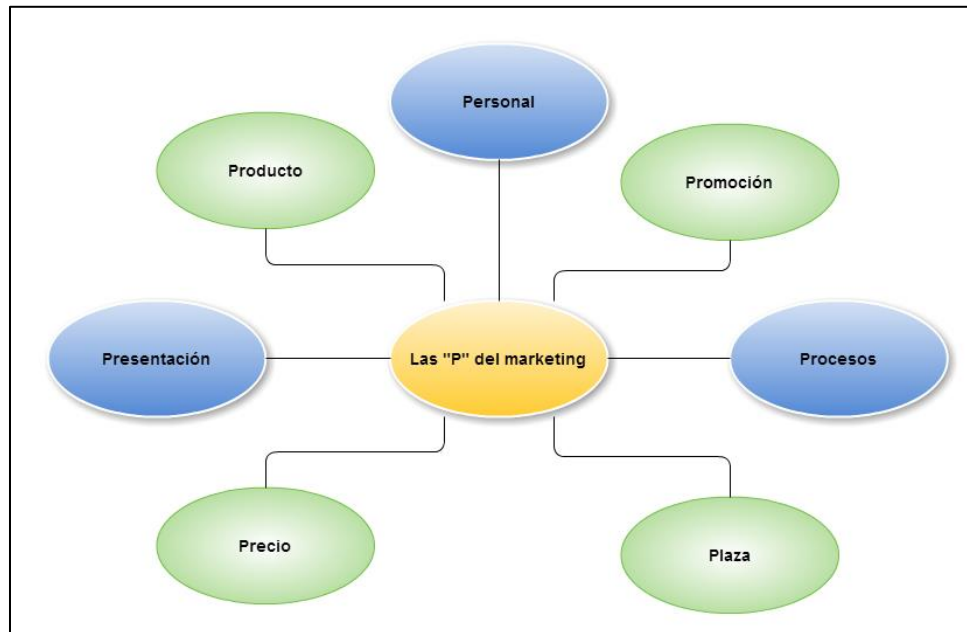
- Ventas
- Experiencia de contacto del personal de la marca
- Embalaje del producto
- Experiencia en línea

### **1.2.7. Procesos**

Los procesos de la empresa deben ser correctamente estructurados. Esto depende de la buena logística de la empresa para optimizar ganancias y costos. Sus características principales son:

- Enfoque del cliente
- Diseño de características
- Investigación y desarrollo

Figura 1. Las P de la mercadotecnia



Fuente: elaboración propia.

### 1.3. El *marketing* digital

El *marketing* digital también llamado *marketing* 2.0 (mercadotecnia en internet) se basa en la aplicación de estrategias y técnicas de *marketing* tradicional en entornos digitales. Todas estas técnicas son convertidas y traducidas a un nuevo mundo con nuevas herramientas como las redes sociales con una entrega inmediata con los clientes de la empresa.

Para este nuevo mundo de *marketing* toda la información publicada por una empresa, según su audiencia, puede llegar a todo el mundo en cuestión de segundos y obtener retroalimentación según se vaya alcanzando nueva audiencia.

## **1.4. Estrategias actuales del *marketing* a nivel nacional**

En Guatemala, el *marketing* ha crecido de forma exponencial por lo cual día a día se crean nuevas estrategias para llegar a más clientes potenciales del producto de empresa.

Para el éxito de cualquier empresa en el *marketing* antes que nada se debe de contar con un buen producto que ofrecer al público, un ambiente agradable para consumir el producto adquirido y, por último, un equipo capacitado, en el caso de restaurantes contar con cocineros y meseros con las aptitudes para el trabajo. Con estos elementos se pueden definir las siguientes estrategias que actualmente se utilizan en Guatemala.

### **1.4.1. Poseer un *website***

Muchas empresas actualmente no alcanzan la audiencia necesaria por lo cual utilizan esta estrategia; logran que sus potenciales clientes conozcan su negocio antes de visitarlos.

### **1.4.2. Hacer *marketing* por email**

Con ofertas especiales, descuentos o promociones a través de correo electrónico se tiene un alto porcentaje de posibilidad de que el cliente potencial se acerque a la empresa para adquirir alguno de los productos promocionados. Esto se puede hacer con un formulario de suscripción en su *website* o *fanpage* en redes sociales.

### **1.4.3. Utilizar redes sociales**

Con el desarrollo de la web 2.0 y el incremento del uso de las redes sociales, toda empresa debe poseer su *fanpage* en las grandes redes sociales: *Facebook, Twitter, Instagram*, entre otras. Contar con el personal adecuado para esta labor es imprescindible: al *community manager*.

### **1.4.4. Crear eventos**

Difundir los eventos o promociones en ocasiones especiales; si el producto de una empresa no se vende como es debido se crean ciertas promociones para mover el producto y que esa materia prima no se pierda. Actualmente, existen promociones para casi todo, se pueden encontrar las siguientes:

#### **1.4.4.1. 2 x 1**

Ciertos restaurantes aplican esta estrategia durante días sin mucha afluencia de personas para lograr mantener un nivel de ventas estable en estos días, por ejemplo, se aplica 2 x 1 los días martes y jueves.

#### **1.4.4.2. *All you can eat and drink***

Este tipo de estrategia es utilizada para una sola persona; por lo cual al llegar un grupo de personas, cada una tiene que comprar su vale de consumo; esto logra optimizar las ganancias.

#### **1.4.4.3. Ladies night**

Los bares para lograr que más clientes los visiten utilizan esta estrategia: las mujeres entran gratis a las discotecas lo que atrae a más hombres; entonces ganan todos: empresa y cliente.

#### **1.4.4.4. El segundo a mitad de precio**

En la compra de uno se puede adquirir el segundo a mitad de precio; esto es muy tentador para el cliente, llaman su atención y lo lleva a comprar las dos.

#### **1.4.4.5. Combos**

En restaurantes es más común esta estrategia; en la compra de un producto, por cierta cantidad de dinero extra, se puede adquirir más productos; por ejemplo, en la compra de dos piezas de pollo se añade una porción de papas por solo Q. 5,00.

### **1.5. Comunidades virtuales y su relación con las empresas**

Ya se ha mencionado que las empresas utilizan las redes sociales para promocionar sus productos; se definirán algunos términos importantes para establecer la relación que existe entre una comunidad virtual y las empresas.

#### **1.5.1. Comunidad virtual**

Designa a las personas unidas a través de Internet por valores o intereses comunes, por ejemplo, gustos, pasatiempos o profesiones. La meta de la

comunidad es crear valores a través del intercambio entre los miembros al compartir sugerencias o consejos o simplemente al debatir un tema.

Establecer una comunidad virtual puede resultar beneficioso para una empresa ya que una comunidad crea en sus miembros un sentimiento de pertenencia y permite que la empresa se desarrolle a través de un esfuerzo colectivo.

Más aún, una gran comunidad de usuarios puede tornarse algo positivo para la imagen de la empresa porque presenta un factor de atracción y otorga al usuario una sensación de confianza.

### **1.5.2. Estrategia indiferenciada**

A pesar de haber identificado diferentes segmentos con diferentes necesidades, la empresa opta por dirigirse a ellos con la misma oferta para intentar conseguir el número máximo de posibles clientes.

### **1.5.3. Uso de las comunidades virtuales**

El interés de las empresas en las comunidades virtuales comienza a activarse seriamente al constatar su valor comercial, además, permite el establecimiento de estrategias de *marketing* altamente segmentadas. Permiten obtener datos sobre los clientes; utilizarlos apropiadamente para obtener valor comercial y utilizar la información que se consigue para proporcionar más valor a los mismos clientes.

Son agentes de desarrollo organizacional y creadoras de valor para la empresa y para sus clientes y permiten diferenciarse de otras empresas. Son

un poderoso factor de fidelización hacia una empresa y sus productos; son generadoras de beneficios. Además, permiten responder a la necesidad de incorporar a proveedores y consumidores a los procesos de una empresa y hacerlos participar en la concepción, producción y distribución de sus productos y servicios.

## **1.6. Mercado objetivo**

La aplicación FindFun será una aplicación web con diseño *responsive* para que los usuarios puedan acceder desde el navegador de una computadora o desde sus dispositivos móviles; el mercado objetivo son las personas comprendidas entre los 15 y 50 años, con acceso a internet y que visiten asiduamente restaurantes o bares. También se busca que utilicen la aplicación las personas que visitan un restaurante, para conocerlo o por sus promociones; en ese sentido, el consumidor busca obtener un beneficio económico: gastar menos de lo que regular.

La aplicación también contará con un espacio para las empresas que deseen agregar sus promociones y obtener un perfil para llegar a más clientes y aumentar sus ventas respecto a sus promociones; además, podrán obtener información útil para la toma de decisiones ya que la herramienta les brindará estadísticas de consumo.

### **1.6.1. Benchmarking de la aplicación**

Al buscar aplicaciones similares a FindFun se pueden encontrar diferentes aplicaciones que muestran diferentes restaurantes a nivel nacional con opciones de búsqueda y filtro, con el caso de Mister Menú.



Como se visualiza en la figura 2, se poseen los filtros necesarios para que el usuario pueda encontrar lo que desea.

Figura 2. Captura de Mister Menú



Fuente: *Mister Menú*. <http://mister-menu.com/>. Consulta: 11 de octubre de 2016.

FindFun busca realizar esta temática de desarrollo más bien solo mostrando los resultados del día actual de las ofertas y promociones que cada empresa ofrezca; además, clasificará las promociones por categoría: restaurantes, bares y discotecas. Logrará de esta manera la atención de más audiencia: usuarios buscando promociones y una nueva estrategia de *marketing* para las empresas registradas en la plataforma.



## **2. INVESTIGACIÓN DEL PROBLEMA Y SOLUCIONES PROPUESTAS ANTERIORMENTE**

### **2.1. Descripción del problema**

Actualmente la sociedad está constantemente bombardeada por publicidad de diferentes empresas que pretenden convencer al cliente de consumir sus productos; este tipo de publicidad generalmente incluye la promoción de sus productos en determinados horarios que son seleccionados estratégicamente para atraer clientes y hacerlos consumir en horarios cuando regularmente su beneficio es bajo. Además, se han identificado muchos mercados como de productos alimenticios y establecimientos de entretenimiento, que publican este tipo de publicidad por diversos medios: televisión, vallas publicitarias, redes sociales, entre muchos otros; el problema para el consumidor es que toda esta información dispersa en las redes sociales de las empresas o en anuncios de televisión que posiblemente no vea.

Para el consumidor es muy importante este tipo de publicidad ya que le permitirá aprovechar los beneficios que generalmente estas empresas ofrecen en sus promociones: precios especiales, dos por uno, etc. Sin embargo, esta información se encuentra dispersa y no poseen un medio para comparar las promociones y escoger la que deseen según criterios como precio, establecimiento, cantidad, etc.

## **2.2. Soluciones propuestas en otros países**

La cantidad de aplicaciones tanto web como para dispositivos móviles que se encuentra actualmente en el mercado es muy grande; realmente, existen muchas aplicaciones en muchos países que ofrecen a sus usuarios una guía muy completa de restaurantes a lo largo de las ciudades más importantes de cada país.

La mayoría de estas aplicaciones ofrece características del restaurante: dirección, menú, ubicación en mapa, horario, imágenes del lugar, reseñas del lugar y de los platillos, etc. Sin embargo, FindFun busca otro tipo de mercado: no será una aplicación enfocada en ofrecer al usuario un catálogo de restaurantes, sino más bien, las promociones de dichos restaurantes.

En el mercado internacional se encuentran aplicaciones que ofrecen catálogos de restaurantes; a continuación, se muestra una descripción breve de algunas.

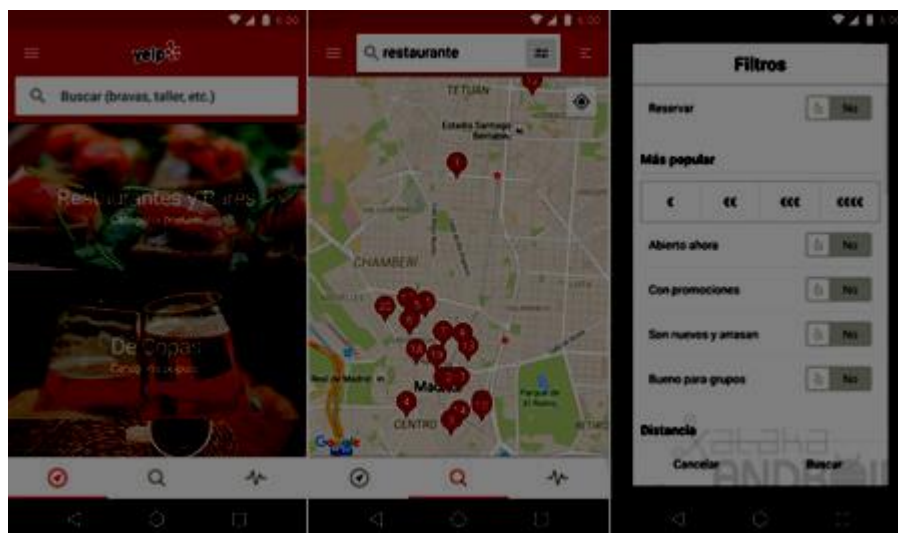
### **2.2.1. Yelp**

Es una gran alternativa para encontrar cualquier tipo de negocio, pero sobre todo para restaurantes y bares. Su buscador permite filtrar los resultados por precio, si están abiertos en ese momento, si aceptan reservas, si tienen alguna promoción vigente y por características. Además, permite ordenar los resultados por valoración o distancia y ver los resultados en el mapa. Cuenta con una gran cantidad de valoraciones de sus usuarios.

Yelp fue fundada por dos ex empleados de Paypal. yelp creció rápidamente, para finales de 2010 ya tenía alrededor de \$30 000 000,00 de ingresos con más de 4,5 millones de publicaciones.

El 78 % de empresas registradas en la plataforma de Yelp tiene entre 3 y 5 estrellas de calificación, dándole a cada una de estas empresas reciben una gran publicidad gracias a Yelp.

Figura 3. **Captura de aplicación Yelp**



Fuente: *Yelp*. <https://play.google.com/store/apps/details?id=com.yelp.android&hl=es>.

Consulta: 11 de octubre de 2016.

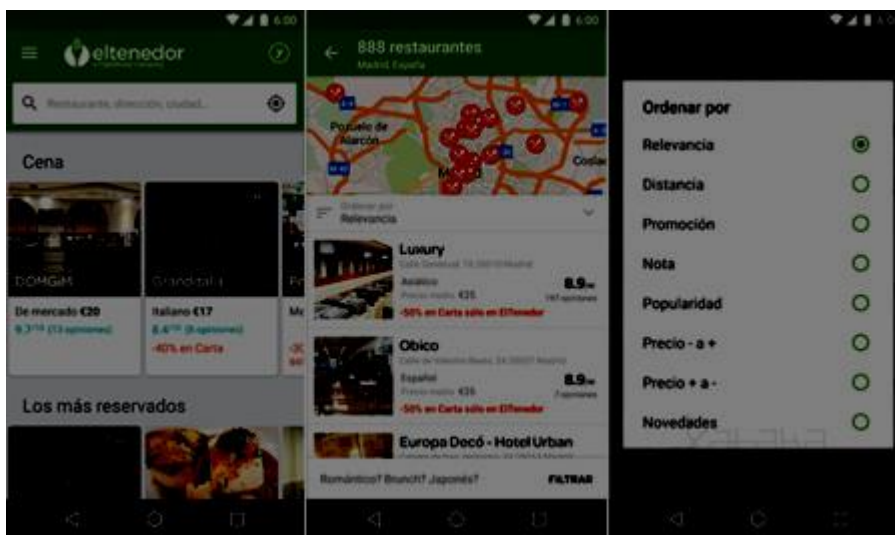
### 2.2.2. **App móvil Eltenedor**

Es una aplicación móvil que se encuentra disponible para los sistemas operativos IOS para iPhone y android; busca y ubica los mejores restaurantes según la ubicación del usuario, cuenta con secciones, la carta de precios y un

potente buscador con distintos filtros. Permite ver las valoraciones de los usuarios de TripAdvisor. Eltenedor muestra las ofertas de cada restaurante.

Eltenedor cuenta con más de 40 000 restaurantes registrados donde puede buscar el usuario la mejor opción según su preferencia. Eltenedor forma parte de TripAdvisor desde mayo de 2014; opera en más de 12 países a nivel mundial.

Figura 4. Captura de aplicación Eltenedor



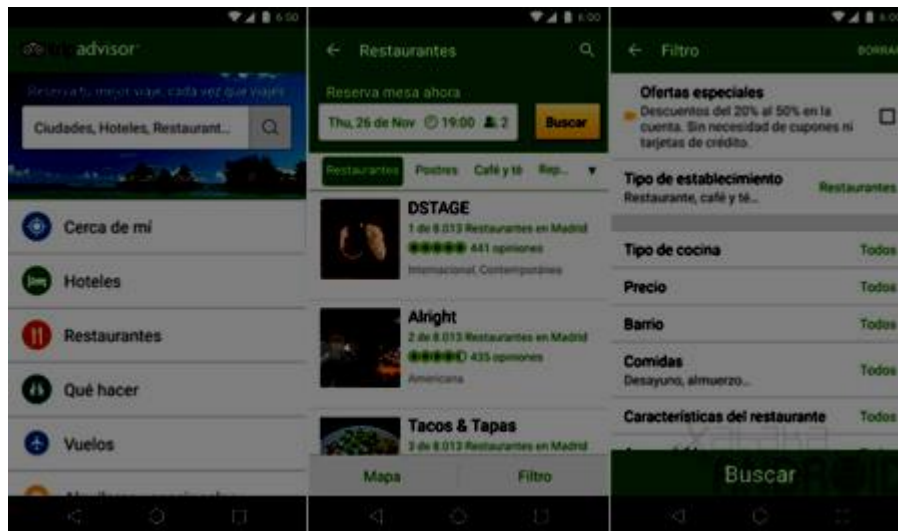
Fuente: *Eltenedor reserva restaurantes*. <https://play.google.com/store/apps/details?id=com.lafourchette.lafourchette&hl=es>. Consulta: 11 de octubre de 2016.

### 2.2.3. TripAdvisor

TripAdvisor es una aplicación bastante versátil ya que además de tener presencia en los sistemas operativos móviles, también, cuenta con una página web donde se puede encontrar el mismo contenido; es una de las guías más populares del mundo. Esta aplicación permite seleccionar el tipo de

establecimiento que se desea visitar y aplicar numerosos filtros para localizar el restaurante más cercano y que más se ajuste a los gustos del usuario; también, ofrece valoraciones de los establecimientos dadas por los usuarios.

Figura 5. Captura de aplicación TripAdvisor



Fuente: *TripAdvisor: hoteles, vuelos*. <https://play.google.com/store/apps/details?id=com.tripadvisor.tripadvisor&hl=es>. Consulta: 11 de octubre de 2016.

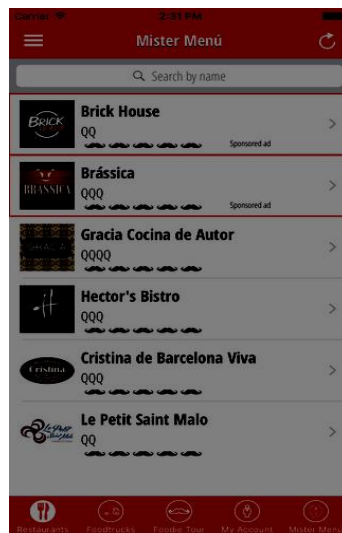
### 2.3. Soluciones propuestas en Guatemala

En Guatemala, debido al nivel de desarrollo del país, este tipo de aplicaciones son muy pocas debido al bajo beneficio que representa mantener una aplicación para la cantidad de personas que la utilizarían; sin embargo, se encuentran algunas aplicaciones especializadas para Guatemala que cuentan con una serie de características que ayudan a los usuarios a elegir el establecimiento que desean visitar.

### 2.3.1. Mister Menú

Mister Menú es una guía digital de restaurantes de Guatemala. Su misión es recolectar todos estos locales en una plataforma. La aplicación funciona con varios filtros según localidad, rango de precio, tipo de comida o tipo de servicio.

Figura 6. Captura de aplicación Mister Menú



Fuente: *Mister menú - Aplicaciones de Android en Google Play*. <https://play.google.com/store/apps/details?id=gt.com.cs.mistermenu&hl=es>. Consulta: 7 de noviembre de 2016.

Cuenta con un gran catálogo de restaurantes, que se clasifican de la siguiente manera:

- Gama alta: restaurantes donde los precios son mayores a cien quetzales.
- Gama media: restaurantes donde los precios son mayores a cincuenta quetzales.
- Gama baja: restaurantes donde los precios son menores de cincuenta quetzales.



Los usuarios tienen la opción de dar reseñas sobre los establecimientos, además, de proporcionar en algunos casos, el menú del establecimiento y su precio. Una de sus principales características es identificar a los establecimientos que permiten el ingreso de mascotas.

#### **2.4. Debilidades identificadas**

FindFun es una aplicación web *responsive* que podrá ser consultada desde cualquier dispositivo con acceso a internet; además, su enfoque únicamente está en las promociones de los establecimientos.

Desde este punto de vista, las aplicaciones anteriores se enfocan en ofrecer establecimientos a los usuarios y las características de los establecimientos, mas no establecen el parámetro específico de las promociones.

La gran debilidad de este tipo de aplicaciones es que no resaltan lo más interesante de los establecimientos hoy en día: las ofertas; los usuarios se ven atraídos por gastar menos u obtener más a precio regular, que por las reseñas de otros usuarios.

Además, los establecimientos que utilizan actualmente esta estrategia de negocios son establecimientos ya conocidos con una reputación muy buena por lo que no necesitan las reseñas; por otra parte, se hace tedioso para el usuario explorar un menú con muchas opciones para decidir a qué establecimiento ir.

Por eso FindFun se concentrará únicamente en las promociones de cada establecimiento sin establecer un límite de promociones. Esto ayudará a los

usuarios a tomar una decisión inmediata sobre el establecimiento que desean visitar, guiándose únicamente por lo que el establecimiento ofrece.

Las aplicaciones como El tenedor y Yelp, funcionan únicamente en dispositivos móviles, en sistemas operativos ios y android, limitaciones a tener en cuenta ya que no llegan a un gran segmento de usuarios que no tienen un dispositivo móvil o simplemente no gustan de utilizarlo para este tipo de consultas. FindFun al ser una aplicación web *responsive* si puede llegar a este segmento de usuarios y desplegar en un dispositivo móvil y en una computadora las promociones disponibles al usuario de forma agradable y explícita.

### **3. DISEÑO DE LA APLICACIÓN SEGÚN LA NECESIDAD IDENTIFICADA**

#### **3.1. Necesidades específicas identificadas**

Las siguientes necesidades fueron identificadas a través de la observación de los consumidores que visitan ciertos comercios de consumo.

##### **3.1.1. Centralización de promociones**

Los consumidores se ven obligados a revisar cada una de las páginas o *fan page* de los establecimientos de los que pudieran visitar; por lo tanto, surge la necesidad de centralizar las promociones para que el consumidor pueda tener acceso a todas las promociones propuestas por las compañías.

##### **3.1.2. Publicación de promociones durante su periodo de vigencia**

Muchas publicaciones en las *fan pages* de los establecimientos continúan visibles al público aun cuando ya vencieron; con FindFun eso cambiaría ya que las promociones desaparecen al momento de que ya no están en su periodo de vigencia; reduciría así las búsquedas de los usuarios.

#### **3.2. Prototipos**

Los prototipos son una representación limitada de lo que será el sistema ya terminado; en el presente trabajo se mostrarán los prototipos de las vistas que el usuario final visualizará.

### 3.2.1. *Login*

Prototipo realizado en <https://balsamiq.com/>. Muestra cómo debe iniciar sesión el usuario para publicar sus promociones.

Figura 7. **Prototipo de *login***



Fuente: elaboración propia, empleando la herramienta online balsamiq.com.

### 3.2.2. **Carga de promociones**

Prototipo realizado en <https://balsamiq.com/>. Solicitud de datos mínimos para la publicación de la promoción cargando la imagen que se mostrará al público.

Figura 8. **Prototipo de carga de promociones**

El prototipo muestra una interfaz de usuario en un navegador con el título "FindFun Carga". La interfaz tiene un fondo negro y los elementos de texto y botones en blanco. En la parte superior, hay una barra de direcciones con iconos de navegación (atrás, adelante, cerrar, inicio) y un campo de búsqueda. El formulario principal contiene los siguientes campos:

- Día de Promoción:** Un menú desplegable con "Lunes" seleccionado.
- Nombre de Promoción:** Un campo de texto vacío.
- Banner:** Un campo de texto con un botón "Buscar..." a la izquierda. Debajo del campo se muestra un banner de promoción para "PAN PIZZA 2x1" de "Pizza Fruit" con el texto "DE LUNES A MIÉRCOLES".
- Descripción:** Un campo de texto grande y vacío.

En la parte inferior del formulario, hay un botón "Aceptar".

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

### 3.2.3. Promociones públicas

Prototipo realizado en <https://balsamiq.com/>. Se muestran las promociones ya publicadas que tienen su tiempo de vigencia activo según la hora del servidor donde serán publicadas.

Figura 9. **Prototipo de promociones públicas**



Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

### 3.2.4. Expandir publicación

Prototipo realizado en <https://balsamiq.com/>. Al expandir la publicación aparece el botón “me gusta” que se registrará en el servidor, las promociones que le gustan al usuario, para cuando esta empresa, publique más promociones se notifique a través de correo electrónico de estas promociones.

Figura 10. **Prototipo de expandir publicación**



Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

### 3.3. Definición de casos de uso

Describen paso a paso las actividades que realizará cada proceso dentro de la aplicación. Define los actores, el flujo normal a tomar y su flujo alterno. Los principales casos de uso se definen a continuación:

Tabla I. **Caso de uso, registrar usuario**

<b>Nombre</b>	Registrar usuario
<b>Actores</b>	Usuario final y comercial
<b>Descripción</b>	El usuario crea su cuenta para poder interactuar con la aplicación web.
<b>Precondiciones</b>	El usuario debe cargar la página.
<b>Flujo normal</b>	<ul style="list-style-type: none"> <li>• Llenar los campos solicitados.</li> <li>• El sistema valida los datos.</li> <li>• El sistema devuelve un mensaje de aviso que su cuenta ha sido creada exitosamente.</li> </ul>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>• Los campos no son llenados completamente, el sistema notifica al usuario de llenar la totalidad de los campos solicitados.</li> <li>• El sistema comprueba que los datos ingresados no son válidos, notifica al actor que verifique los datos ingresados y los pueda corregir.</li> </ul>
<b>Poscondiciones</b>	Los datos del usuario son almacenados en la base de datos.

Fuente: elaboración propia.

Tabla II. **Caso de uso, Login**

<b>Nombre</b>	<i>Login</i>
<b>Actores</b>	Usuario final y comercial
<b>Descripción</b>	El usuario inicia sesión en la aplicación.
<b>Precondiciones</b>	Haberse registrado en la aplicación.
<b>Flujo normal</b>	<ul style="list-style-type: none"> <li>• El usuario ingresa su nombre de usuario y contraseña.</li> <li>• El sistema valida que los datos sean válidos.</li> <li>• El usuario ingresa a la aplicación.</li> </ul>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>• Usuario o contraseña incorrectos, se notifica al usuario que vuelva a ingresar sus datos para poder acceder a la aplicación.</li> </ul>
<b>Poscondiciones</b>	Entra a la aplicación.

Fuente: elaboración propia.



Tabla III. **Caso de uso, carga de promociones**

<b>Nombre</b>	Carga de promociones
<b>Actores</b>	Usuario comercial
<b>Descripción</b>	El usuario crea sus promociones en la aplicación.
<b>Precondiciones:</b>	Haberse logeado en la aplicación previamente.
<b>Flujo normal</b>	<ul style="list-style-type: none"> <li>• El usuario llena los campos solicitados.</li> <li>• El sistema valida que los datos ingresados sean válidos.</li> <li>• Se notifica al usuario que ha sido publicado su promoción durante el periodo asignado.</li> </ul>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>• Algún campo no fue llenado, se notifica al usuario que llene completamente los campos.</li> <li>• Que el tipo de imagen a cargar no sea válido para el sistema, se notificará al usuario los tipos permitidos por el sistema.</li> </ul>
<b>Poscondiciones</b>	Los datos de la promoción son almacenados en la base de datos.

Fuente: elaboración propia.

Tabla IV. **Caso de uso, visualizar promociones**

<b>Nombre</b>	Visualización de promociones
<b>Actores</b>	Usuario final
<b>Descripción</b>	El usuario visualiza las promociones de los comerciantes por el tiempo que la promoción este vigente.
<b>Precondiciones</b>	Haberse iniciado sesión en la aplicación previamente.
<b>Flujo normal</b>	<ul style="list-style-type: none"> <li>• Durante el periodo de vigencia, se visualizará la promoción.</li> <li>• El sistema valida el tiempo asignado y quita la promoción cuando culmine.</li> </ul>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>• Por falta de buena conexión a internet no se puedan visualizar las imágenes.</li> </ul>
<b>Poscondiciones</b>	Volver a cargar página, para visualizar nuevas promociones.

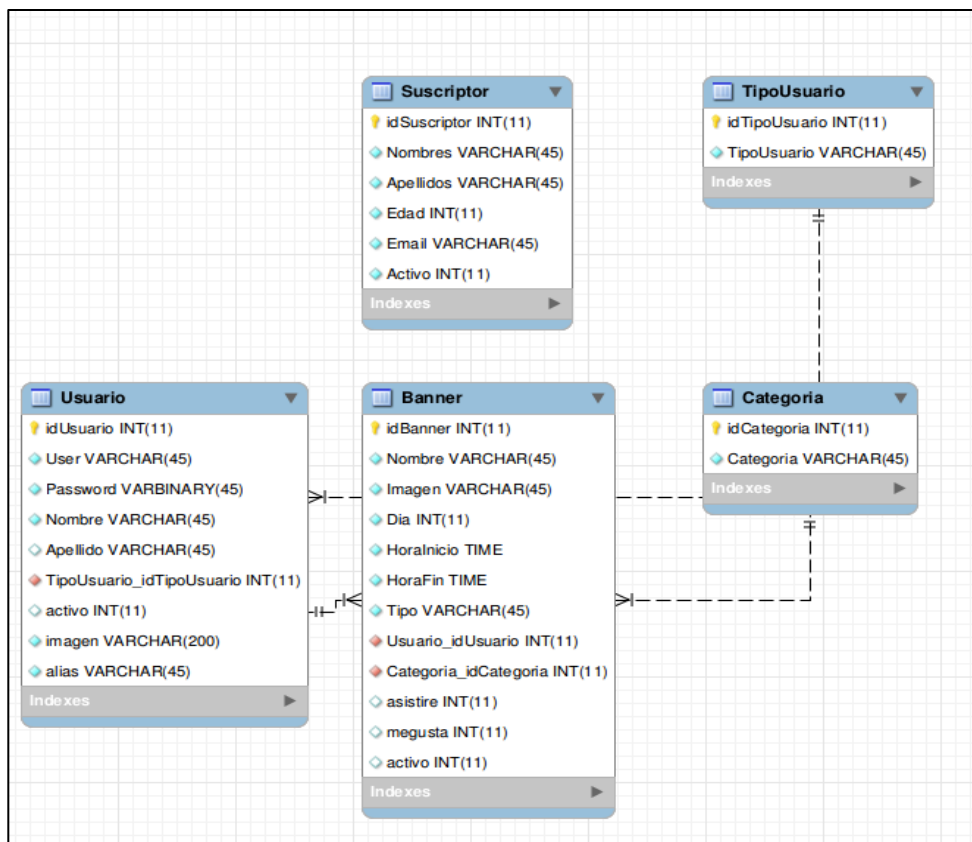
Fuente: elaboración propia.

### 3.4. Servidor de base de datos y conexiones

EL servidor de base de datos es MySQL que está publicado en un servidor en la nube de Digital Ocean – hospedaje de servidores virtuales que se encuentra en la siguiente dirección <https://www.digitalocean.com/>.

Las conexiones a la base de datos se dan a través de procedimientos almacenados; por estar en el mismo servidor la aplicación desarrollada, las conexiones se hacen a través de *localhost*.

Figura 11. Modelo entidad relación de la base de datos



Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

### **3.5. Vistas y validaciones de la arquitectura de software**

La arquitectura de software describe un sistema en sus diferentes componentes y cómo son interconectados y se comunican entre sí. Para documentar la arquitectura, se plantea a través de un conjunto de vistas. Cada vista representa un comportamiento de la aplicación; son las siguientes:

#### **3.5.1. Vista funcional**

Muestra los elementos que aportan funcionalidad a la aplicación y sus principales responsabilidades; es la vista que los interesados del proyecto utilizan más.

##### **3.5.1.1. Justificación**

Es importante porque describe los elementos funcionales en tiempo de corrida, sus responsabilidades, interfaces e interacciones primarias; debido a estas razones es una de las vistas más importantes que se implementan a lo largo del desarrollo de todo sistema.

##### **3.5.1.2. Concerns**

Los *concerns* (preocupaciones del sistema) se separan en dos secciones: funcional e interfaces externas.

###### **3.5.1.2.1. Capacidades funcionales**

Definen que es lo que se requiere que el sistema haga; en este caso, el objetivo del sistema que se presenta (FindFun), es la experiencia de visualizar todas las promociones del día centralizadas en un solo lugar tener fácil acceso.

### **3.5.1.2.2. Interfaces externas**

En este caso, el sistema FindFun solo contará con la interacción e intercambio de datos, con el sistema de bases de datos, el cual es MySQL.

### **3.5.1.3. Modelo**

El modelo es el encargado de separar los datos y la lógica del negocio; dentro de sus responsabilidades está la gestión de eventos y la comunicación entre los elementos de la arquitectura.

#### **3.5.1.3.1. Elementos funcionales**

Es una parte bien definida de tiempo de ejecución del sistema que tiene responsabilidades particulares y expone una interfaz para conectarse a otros elementos: *frontend*, *backend* y usuario.

#### **3.5.1.3.2. Interfaces**

Es un mecanismo bien definido por el cual las funciones de un elemento pueden ser accedidas por otros elementos; en este caso se cuenta con la interfaz en *nodejs* con angular la cual es un *frontend* encargada en conectarse con el *backend* realizado en MySQL.

#### **3.5.1.3.3. Conectores**

Son las piezas de la arquitectura que unen elementos para que puedan interactuar, para el sistema FindFun el único conector que se requiere es tener acceso a un celular móvil o computadora que tenga acceso a internet.

### **3.5.2. Vista de información**

La vista de información permite la observación de la estructura de la información del negocio durante las transacciones de FindFun para los procesos de planificación, control y toma de decisiones.

#### **3.5.2.1. Justificación**

Se incluye esta vista porque se describe la forma en que el sistema almacena, manipula, administra y distribuye información.

Uno de los propósitos de casi cualquier sistema informático es manipular información de alguna manera y este punto de vista desarrolla una vista de alto nivel de estructuras estáticas de datos y flujos de información.

El objetivo de este análisis es responder a las preguntas sobre contenido, estructura, propiedad, latencia, referencia y migración de datos. Generalmente, es una vista que se vuelve obligatoria ya que como se mencionó anteriormente, la mayor parte de sistemas manipulan la información.

#### **3.5.2.2. Concerns**

- Un pequeño conjunto de artículos de datos asociados con las responsabilidades principales del sistema.
- La mayoría de los sistemas tienen en su corazón un almacenamiento de transacciones o base de datos relacional.
- La información histórica puede manejarse en un *datawarehouse*.
- Para el sistema actual, FindFun utiliza una única base de datos relacional para todas las operaciones necesarias, en este caso se utilizó MySQL.

### **3.5.2.3. Modelo**

Un modelo de base de datos determina la estructura lógica de la información almacenada que se puede organizar y manipular los datos de una manera versátil.

#### **3.5.2.3.1. Modelo de almacenamiento**

Se utilizó una base de datos relacional, todos los datos son almacenados bajo relaciones. De esta manera se tiene un acceso simple a la información almacenada.

#### **3.5.2.3.2. Flujo de información**

Determina el camino y la estructura de la información en todos los procesos del negocio; debe responder a las siguientes preguntas:

#### **3.5.2.3.3. ¿Dónde se crea y se destruye la información?**

Lógicamente es creada en el sitio web por los usuarios encargados de crear promociones, físicamente, en el DMBS utilizado, en este caso MySQL. La información no se destruye, solo se oculta.

#### **3.5.2.3.4. ¿Dónde son consultados, modificados o enriquecidos los datos?**

Desde el *frontend*, realizado en *nodejs* y *angular*; en otras palabras, desde el navegador web que utilice el usuario final.

### **3.5.3. Vista de deployment**

La vista de deployment muestra la configuración física del sistema, revelando las piezas de software que se ejecutan en cada parte del hardware. Donde se puede observar la separación de las capas de la aplicación.

#### **3.5.3.1. Justificación**

Se incluye esta vista porque describe el ambiente donde el sistema será desplegado y las dependencias que el sistema tiene.

Esta vista captura el ambiente de hardware que el sistema necesita (nodos de procesamiento, interconexión de redes y facilidades de almacenamiento de disco requeridos), los requerimientos técnicos del ambiente para cada elemento y el mapeo de elementos de software al ambiente de tiempo de corrida que los ejecutará.

#### **3.5.3.2. Concerns**

Los *concerns* de esta vista se centran en los posibles problemas que pueda mostrar el hardware en tiempo de ejecución, haciendo esto que falle el sistema completo.

##### **3.5.3.2.1. Plataforma de ejecución requerida**

Se debe identificar claramente el tipo de plataforma de ejecución que el sistema necesita y el rol de cada parte; el ambiente utilizado para el despliegue del sistema (módulo de administración) se describe a continuación:

- Sistema operativo AMI de Amazon Linux de 64 bits
- 1GB de Memoria RAM
- Procesador Intel Xeon 2.40
- 8GB de disco duro
- Base de datos MySQL

#### **3.5.3.2.2. Requerimientos de software de terceros**

Solo se necesita instalar y configurar *nodejs* y *angular*; para realizar la conexión con la base de datos, se puede utilizar la librería *socket.io* o bien *socket*.

#### **3.5.3.3. Modelo**

El modelo del hardware muestra que parte de software estará en cada node de hardware y como se relacionan entre sí para que el sistema tenga un desempeño óptimo.

##### **3.5.3.3.1. Elementos**

- Nodos de procesamiento
- Nodos cliente
- Contenedores de ejecución
- Hardware de almacenamiento en línea
- Hardware de almacenamiento fuera de línea
- Enlaces de red
- Otros componentes de hardware
- Mapeo de elementos de ejecución a nodos



#### **3.5.3.4. Validaciones**

El usuario deberá registrar sus datos antes de poder ingresar a la aplicación ya que no será posible poder acceder a la aplicación por que no se cuenta con la información necesaria para su acceso.

Se deberá ingresar una contraseña alfa-numérica con un mínimo de 8 caracteres; deberá contener letras en el rango de la A-Z (mayúsculas y minúsculas) y números en el rango de 0-9.

Para que la aplicación pueda funcionar correctamente es necesario contar con conexión a internet, alámbrica o inalámbrica.

El sistema validará los datos existentes en la base de datos a través de procedimientos almacenado, para que no haya SQL injection.

El sistema validará cada uno de los campos que se encuentran listos para ser procesados antes de realizar una transacción para evitar datos nulos en la base de datos, se logra así una consistencia estable en la base de datos.

El usuario podrá contar con una única cuenta en esta aplicación, no se pueden registrar dos usuarios con la misma información.

Al usuario se le notificará por correo electrónico las notificaciones siempre y cuando el usuario ingrese su correo electrónico en el módulo de suscripción.

### 3.5.4. Diseño intuitivo y usabilidad

Para un diseño intuitivo se tiene que ir a la definición que da la Real Academia Española, “facultad de comprender las cosas instantáneamente, sin necesidad de razonamiento”<sup>1</sup>.

Para el desarrollo de FindFun atribuirá al usuario final de las siguientes usabilidades:

- Encontrar lo que busca sin dar tantas vueltas; esto evitará al usuario dar muchos *clicks* para encontrar lo que desea, en este caso, poder encontrar las promociones que le interesan.
- Navegar sin pensar, dando al usuario una navegación rápida y fluida.
- Realizar procesos con varios pasos, esto logrará que el usuario final no tenga que pensar mucho durante la toma de decisiones de que hacer a continuación en cada proceso.

La usabilidad es toda aquella cualidad que posee un sistema que permite a sus usuarios sus objetivos deseados con efectividad; es decir, cuando una aplicación permite realizar sus funciones de la mejor manera mayor es su usabilidad. Si el sistema ayuda a que todo usuario cometa el menor número de errores y que se recupere fácilmente de ellos, si permite hacer la tarea lo más rápidamente posible con eficacia y además el usuario queda satisfecho con las tareas realizadas, el sistema tiene una muy buena usabilidad.

---

<sup>1</sup> Real Academia Española. *Definiciones*. <http://dle.rae.es/srv/search?m=30&w=dise%C3%B1o>. Consulta: 7 de noviembre de 2016.

FindFun posee todas estas cualidades, las que proporcionan una gran usabilidad por parte de los usuarios: finales y comerciales (empresas que publicarán sus promociones).



## 4. DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN

En los capítulos anteriores, se ha presentado una perspectiva muy teórica, se ha dado un panorama actual del desarrollo *responsive* y el mercado objetivo de la aplicación.

En el presente capítulo se implementa y ejemplifica la creación de una aplicación *responsive* multiplataforma; se brindan los requerimientos, las herramientas y el respectivo tutorial de programación que ayudará como referencia para desarrollos futuros de la aplicación.

### 4.1. Requerimientos de hardware

- Computadora portátil
- 6 GB de memoria RAM
- Procesador Intel Core i5 de 2.59 GHz
- Windows 7 Premium de 64 Bits

### 4.2. Herramientas

Para el desarrollo de la aplicación se utilizan diferentes herramientas, las básicas para cualquier desarrollo de software: repositorio, entorno integrado de desarrollo, entornos de desarrollo, entre otros.

#### **4.2.1. Repositorio de control de versiones de github**

Github es una plataforma colaborativa para el control de versiones de proyectos, github utiliza el *framework* Ruby on Rails. Cuenta con un amplio catálogo de herramientas para un óptimo control de versiones.

Este proveedor de servicio fue utilizado durante el desarrollo de la aplicación, creando un repositorio privado para almacenar las fuentes de la aplicación.

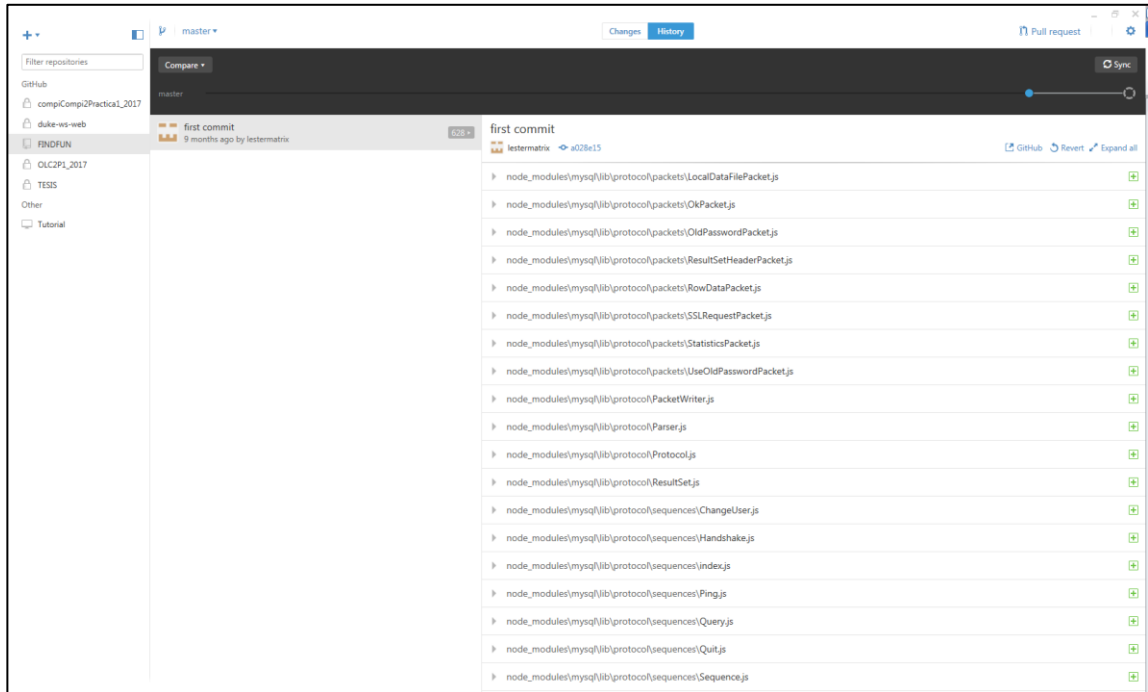
#### **4.2.2. Cliente github desktop**

El servidor de hospedaje de proyectos de github utiliza un repositorio central que permite un ambiente de desarrollo multiusuario para el control de versiones, este utiliza una herramienta de versionamiento para conectarse al repositorio central. Se utiliza la herramienta de github desktop para este fin que se integra rápidamente al desarrollo.

Los pasos necesarios para utilizar el control de versiones son los siguientes:

- Descargar github desktop (<https://desktop.github.com>)
- Instalar el cliente de github en cada una de las máquinas de desarrollo
- Clonar el repositorio
- Crear la rama de trabajo

Figura 12. Configuración de github desktop

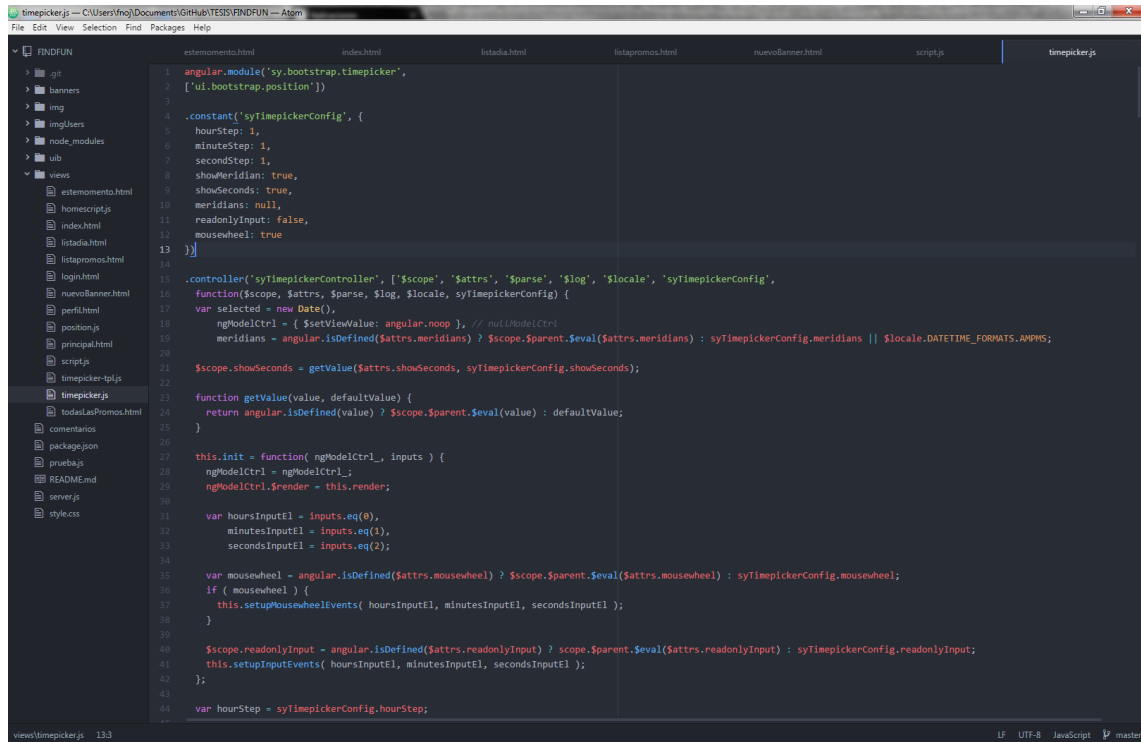


Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

### 4.2.3. Editor de texto Atom

Atom es una herramienta; su función principal es ser un editor de texto que permite la creación y modificación de archivos digitales dando orden y formato a las fuentes de la aplicación. Atom es software libre personalizable a las necesidades que se le demanden.

Figura 13. Captura de atom



```
angular.module('sy.bootstrap.timepicker',
  ['ui.bootstrap.position'])
)
.constant('syTimepickerConfig', {
  hourStep: 1,
  minuteStep: 1,
  secondStep: 1,
  showMeridian: true,
  showSeconds: true,
  meridians: null,
  readonlyInput: false,
  mousewheel: true
})
.controller('syTimepickerController', ['$scope', '$attrs', '$parse', '$log', '$locale', 'syTimepickerConfig',
function($scope, $attrs, $parse, $log, $locale, syTimepickerConfig) {
  var selected = new Date(),
      ngModelCtrl = { $setViewValue: angular.noop }, // nullModelCtrl
      meridians = angular.isDefined($attrs.meridians) ? $scope.$parent.$eval($attrs.meridians) : syTimepickerConfig.meridians || $locale.DATETIME_FORMATS.AMPMS;

  $scope.showSeconds = getValue($attrs.showSeconds, syTimepickerConfig.showSeconds);

  function getValue(value, defaultValue) {
    return angular.isDefined(value) ? $scope.$parent.$eval(value) : defaultValue;
  }

  this.init = function(ngModelCtrl, inputs) {
    ngModelCtrl = ngModelCtrl;
    ngModelCtrl.$render = this.render;

    var hoursInputEl = inputs.eq(0),
        minutesInputEl = inputs.eq(1),
        secondsInputEl = inputs.eq(2);

    var mousewheel = angular.isDefined($attrs.mousewheel) ? $scope.$parent.$eval($attrs.mousewheel) : syTimepickerConfig.mousewheel;
    if ( mousewheel ) {
      this.setupMousewheelEvents( hoursInputEl, minutesInputEl, secondsInputEl );
    }

    $scope.readonlyInput = angular.isDefined($attrs.readonlyInput) ? scope.$parent.$eval($attrs.readonlyInput) : syTimepickerConfig.readonlyInput;
    this.setupInputEvents( hoursInputEl, minutesInputEl, secondsInputEl );
  };

  var hourStep = syTimepickerConfig.hourStep;
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.2.4. JavaScript

“Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico”<sup>2</sup> Una de las ventajas de JavaScript es que es soportado por todos los navegadores, dentro de sus principales características se encuentran:

- Simplicidad
- Manejo de objetos

<sup>2</sup> *JavaScript*. <https://es.wikipedia.org/wiki/JavaScript>. Consulta: 3 de marzo de 2017.



- Respuesta en tiempo real

#### 4.2.5. NodeJs

“Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Nodejs usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo”<sup>3</sup>. Herramienta utilizada como *framework* para el desarrollo de la aplicación.

Aprovechando el motor de chrome le permite a NodeJs proporcionar un entorno de ejecución del lado del servidor que compila y ejecuta javascript a velocidades increíbles.

NodeJs lleva acabo sus transacciones de forma asíncrona. NodeJs mantiene un *event loop* que gestiona todas las operaciones de las peticiones de todos los clientes conectados.

#### 4.2.6. MySql

“Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos *open source* más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web”<sup>4</sup>. Base de datos relacional utilizada en el desarrollo de la aplicación.

---

<sup>3</sup> *Node Js*. <https://nodejs.org/es/>. Consulta: 3 de marzo de 2017.

<sup>4</sup> *MySql*. <https://es.wikipedia.org/wiki/MySQL/>. Consulta: 3 de marzo de 2017.

MySQL es muy utilizado en aplicaciones web por su fácil comunicación con sus interfaces a través de un driver de conexión.

#### **4.2.7. AngularJs**

AngularJs se integra a HTML, logrando transformar páginas estáticas en vistas dinámicas en un entorno web. Angular permite ampliar el vocabulario nativo de HTML, como resultante se obtiene un entorno expresivo, legible y rápido de desarrollar.

Entre los elementos principales de angularjs se encuentran los *controllers* y los *scopes*

##### **4.2.7.1. Scopes**

Son los distintos contextos de ejecución donde trabaja angularjs, por ejemplo: al utilizar la directiva *ng-model*, se está apuntando a un atributo contenido en el *scope*, sobre el que se está trabajando. Los *scopes* guardan información de los modelos que están representados en la vista.

##### **4.2.7.2. Controllers**

Son los encargados de inicializar y modificar la información que se maneja dentro de los *scopes*. Según se le demande desde la aplicación.

#### **4.3. Tutorial de desarrollo y referencias**

El desarrollo de la aplicación fue realizado utilizando una arquitectura cliente servidor, utilizando únicamente procedimientos almacenados para la

manipulación de información en la base de datos, los cuales son consumidos desde el mismo servidor. Posteriormente esta la capa del cliente, las vistas de la aplicación *responsive*.

#### **4.3.1. Implementación del modelo de datos**

Tal como se muestra en la sección 3.4., se utiliza un modelo relacional para satisfacer las necesidades requeridas para la aplicación. Este modelo tiene como responsabilidad principal almacenar la información que las empresas compartirán para que los usuarios puedan visualizar sus promociones. Esta información puede ser consultada desde cualquier dispositivo con un navegador web, las tablas principales del modelo son las siguientes:

- Suscriptor
- TipoUsuario
- Usuario
- Categoría
- Banner (promociones almacenadas por empresa)

El modelo relacional que fue expuesto en la sección 3.4., fue diseñado con notación entidad-relación con la herramienta MySQL workbench. Esta herramienta brinda la creación del código SQL de las tablas del modelo. El script generado se ejecutó en el DBMS MySQL server en el servidor.

Figura 14. **Ejemplo de código SQL para la creación de tablas de la base de datos**

```
DROP TABLE IF EXISTS `Usuario`;
CREATE TABLE `Usuario` (
  `idUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `User` varchar(45) NOT NULL,
  `Password` varbinary(45) NOT NULL,
  `Nombre` varchar(45) NOT NULL,
  `Apellido` varchar(45) DEFAULT NULL,
  `TipoUsuario_idTipoUsuario` int(11) NOT NULL,
  `activo` int(11) DEFAULT NULL,
  `imagen` varchar(200) NOT NULL,
  `alias` varchar(45) NOT NULL,
  PRIMARY KEY (`idUsuario`),
  KEY `fk_Usuario_TipoUsuario_idx` (`TipoUsuario_idTipoUsuario`),
  CONSTRAINT `fk_Usuario_TipoUsuario` FOREIGN KEY (`TipoUsuario_idTipoUsuario`) REFERENCES `TipoUsuario` (`idTipoUsuario`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
--
-- Dumping data for table `Usuario`
--
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.1.1. **Desarrollo de procedimientos almacenados**

Los estándares utilizados para el desarrollo de la aplicación utilizan el acceso a datos a través de procedimientos almacenados que brindan múltiples ventajas con respecto a seguridad, rendimiento, atomicidad y facilita su mantenimiento.

Para el desarrollo de los procedimientos almacenados se utilizó la siguiente notación:

Sp\_nombreDelProcedimiento

Figura 15. **Ejemplo de procedimiento almacenado con consulta a través de clausula select**

```
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_banner_empresa`(id_empresa int(11))
BEGIN
    SELECT b.idBanner, b.Nombre,u.Nombre as NombreU,u.Apellido,b.imagen
    FROM Banner b inner join Usuario u on b.usuario_idUsuario = u.idUsuario
    WHERE b.Usuario_idUsuario = id_empresa;
end ;;
DELIMITER ;
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.2. **Instalación de node Js y angular Js**

Antes de comenzar la instalación se debe tener los repositorios actualizados mediante el siguiente comando:

```
$ sudo apt-get update
```

Así se procede con la instalación de node js.

```
$ sudo apt-get install node js
```

Como se puede observar en la siguiente captura el proceso de instalación.

Figura 16. Instalación de node js

```
root@francisco-Inspiron-5420:/home/francisco# sudo apt-get install nodejs
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  ax25-node libax25 openssh-inetd
Use 'apt-get autoremove' to remove them.
Se actualizarán los siguientes paquetes:
  nodejs
1 actualizados, 0 se instalarán, 0 para eliminar y 652 no actualizados.
Necesito descargar 8,426 kB de archivos.
Se utilizarán 98.3 kB de espacio de disco adicional después de esta operación.
Des:1 https://deb.nodesource.com/node_4.x/ trusty/main nodejs amd64 4.8.0-inodesource1-trusty1 [8,426 kB]
0% [1 nodejs 15.9 kB]
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

Con la instalación se procede con la instalación de express para crear el cuerpo del proyecto.

```
$ sudo npm install express-generator -g
```

Figura 17. Instalación express

```
root@francisco-Inspiron-5420:/home/francisco# npm install express-generator -g
├── depd@1.1.0
├── qs@6.2.0
├── on-finished@2.3.0 (ee-first@1.1.1)
├── finalhandler@0.5.1 (unpipe@1.0.0, statuses@1.3.1)
├── proxy-addr@1.1.3 (forwarded@0.1.0, ipaddr.js@1.2.0)
├── debug@2.2.0 (ms@0.7.1)
├── accepts@1.3.3 (negotiator@0.6.1, mime-types@2.1.14)
├── type-is@1.6.14 (media-typer@0.3.0, mime-types@2.1.14)
├── send@0.14.2 (destroy@1.0.4, ms@0.7.2, statuses@1.3.1, mime@1.3.4, http-errors@1.5.1)
├── pug@2.0.0-beta11 node_modules/pug
├── pug-runtime@2.0.3
├── pug-load@2.0.5 (object-assign@4.1.1, pug-walk@1.1.1)
├── pug-strip-comments@1.0.2 (pug-error@1.3.2)
├── pug-linker@2.0.2 (pug-error@1.3.2, pug-walk@1.1.1)
├── pug-parser@2.0.2 (pug-error@1.3.2, token-stream@0.0.1)
├── pug-lexer@3.0.0 (pug-error@1.3.2, character-parser@2.2.0, is-expression@3.0.0)
├── pug-code-gen@1.1.1 (pug-error@1.3.2, pug-attrs@2.0.2, js-stringify@1.0.2, doctypes@1.1.0, void-elements@2.0.1, constantinople@3.1.0, with@5.1.1)
├── pug-filters@2.1.1 (pug-error@1.3.2, pug-walk@1.1.1, resolve@1.3.2, jstransformer@1.0.0, clean-css@3.4.25, constantinople@3.1.0, uglify-js@2.8.7)
root@francisco-Inspiron-5420:/home/francisco/Escritorio/FINDFUN#
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

Con las fuentes instaladas se crea el cuerpo del proyecto y se instalan las dependencias para su buen funcionamiento. A través de los siguientes comandos.

Comando para crear el cuerpo del proyecto que deja lista la aplicación para su uso.

```
$ sudo express --view=pug FINFUND
```

Se entra a la carpeta *root* de la aplicación creada a través del siguiente comando.

```
$ cd FINDFUN
```

Por último se instalan las dependencias de npm, para poder ejecutar de forma correcta la aplicación con nodejs

```
$ npm install
```

Figura 18. **Creación de cuerpo de la aplicación**

```
root@francisco-Inspiron-5420:/home/francisco/Escritorio# express --view= pug FINDFUN
create : FINDFUN
create : FINDFUN/package.json
create : FINDFUN/app.js
create : FINDFUN/public
create : FINDFUN/public/javascripts
create : FINDFUN/public/images
create : FINDFUN/routes
create : FINDFUN/routes/index.js
create : FINDFUN/routes/users.js
create : FINDFUN/public/stylesheets
create : FINDFUN/public/stylesheets/style.css
create : FINDFUN/bin
create : FINDFUN/bin/www
create : FINDFUN/views
create : FINDFUN/views/index.pug
create : FINDFUN/views/layout.pug
create : FINDFUN/views/error.pug

install dependencies:
$ cd FINDFUN && npm install
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

Por último, se instalan las dependencias de angular para el manejo dinámico de la aplicación.

```
$ sudo npm install angular
```

Figura 19. **Instalación de dependencias de angular**

```
root@francisco-Inspirion-5420:/home/francisco-inspirion-5420/Es  
citorio/FINDFUN# npm install angular  
npm http GET https://registry.npmjs.org/angular  
npm http 200 https://registry.npmjs.org/angular  
npm http GET https://registry.npmjs.org/angular/-/angular-1.6.2  
.tgz  
npm http 200 https://registry.npmjs.org/angular/-/angular-1.6.2  
.tgz  
angular@1.6.2 node_modules/angular  
root@francisco-Inspirion-5420:/home/francisco-inspirion-5420/Es  
citorio/FINDFUN#
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

### **4.3.3. Desarrollo de capa lógica y presentación de la aplicación**

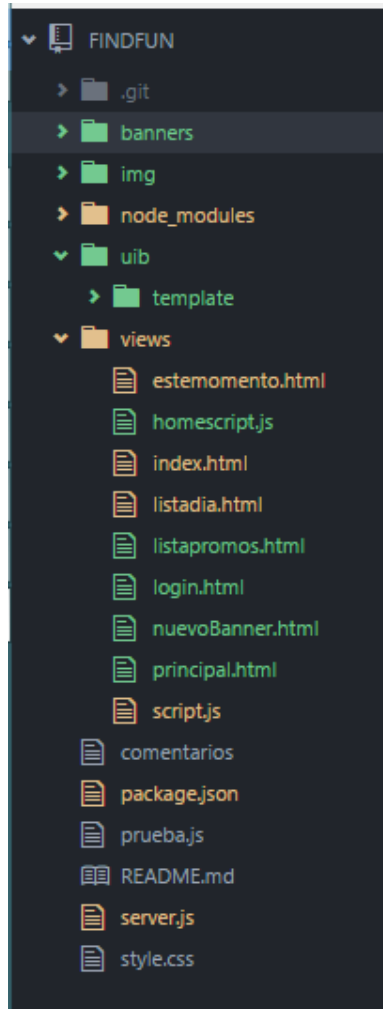
Se separan estas dos capas (lógica y presentación) en la arquitectura seleccionada, siendo el servidor y el cliente.

Siendo la lógica la base de datos en la cual todas las transacciones se realizaron a través de procedimientos almacenados como se describe en la sección 4.3.1.1., de este trabajo.

Y la capa de presentación que está cercana al cliente con las vistas necesarias para el buen funcionamiento de la aplicación; la comunicación con los procedimientos almacenados es directa, desde el servidor de nodejs para una rápida y efectiva respuesta de la base de datos.



Figura 20. **Árbol de archivos de la aplicación**



Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### **4.3.3.1. Capa lógica**

La capa lógica abarca la manipulación de información de la base de datos y la interfaz de usuario; aplican la lógica definida para la aplicación. En esta capa se aplican las reglas del negocio, las validaciones y los flujos de los procesos que se implementan en la aplicación.

Esta capa se encuentra en el archivo server.js, donde se tienen las funciones javascript, para la comunicación de la base de datos y la vista.

#### 4.3.3.1.1. Conexión con la base de datos

La conexión hacia la base de datos se hace a través de la sentencia “mysql.createpool” que devuelve la referencia de donde está alojada la base de datos; de esta manera si es necesaria una migración de servidor o división de capas de la aplicación solo es necesario hacer el cambio de la conexión en este fragmento de código.

Figura 21. Conexión con la base de datos

```
var pool = mysql.createPool({
  connectionLimit : 100, //important
  host      : 'localhost',
  user      : 'root',
  password  : 'vasquez',
  database  : 'FINDFUN',
  debug     : false
});
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.3.1.2. Consumo de procedimientos almacenados desde la aplicación

La integración de los procedimientos almacenados y el código de la aplicación (desarrollado en lenguaje javascript) se realiza en el archivo server.js; a continuación, se ejemplifica el consumo de un procedimiento y su transformación a JSON para su envío a la capa del cliente.

Figura 22. **Consumo de procedimiento almacenado desde el código de la aplicación**

```
app.post('/lista/all',function(req,res){
  console.log("haciendo promociones");

  var id = req.body.id;

  pool.getConnection(function(err,connection){
    if(err){
      connection.release();
    }else {
      console.log("buscando promos..");
      console.log(id);
      var query = "select * from Banner where Usuario_idUsuario= "+ id;
      pool.query(String(query),function(err,rows){
        connection.release();
        if(!err){
          console.log(rows);
          if(rows.length >0) {
            res.write(JSON.stringify(rows));
            //res.write(rows);
            console.log(rows);
            res.send();
          }else {
            rows=[];
            res.write(JSON.stringify(rows));
            res.send();
          }
        }else{
          console.log("query all fail");
        }
      });
    }
  });
});
})
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.3.1.3. **Envío de correos por suscripción**

Al momento de que un usuario se suscribe a las publicaciones, a este usuario se le envía correos periódicamente, para que esté enterado de las promociones de las empresas. Esto se hace a través de nodemailer que es un módulo de envío de correo de node js.

Nodemailer es ejecutado a través de otro plugin de linux que ejecuta periódicamente scripts y comandos de consola, se llama *cron*, el cual se ejecutará todos los días a partir de las 7:00 am.

Figura 23. **Envió de correo con nodemailer**



Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.3.2. **Capa de presentación**

La capa de presentación corresponde a la interfaz (vistas) de usuario de la aplicación; esta capa es desarrollada de manera centralizada, gráficamente y dinámica a través de HTML que es la parte estática de la aplicación y angular para la parte dinámica que enlaza el valor de los campos de la vista con las variables en el controlador.

Para el manejo de variables con angular se utiliza “ng-model='campo'” para enlazar las variables con los campos en la vista del usuario.

#### 4.3.3.2.1. Vista de *login*

Es la primera pantalla que se observa al ingresar a la aplicación; se solicitan los campos necesarios para lograse dentro de la aplicación.

El código HTML es fundamental para mostrar la parte estática de la vista.

Figura 24. Fragmento de HTML con angular del *login*

```
<div class="panel panel-primary" ng-controller="controlador_lista">
  <div class="panel-heading"> Sing in </div>
  <div class="panel-body">
    <form role="form">
      <div class="form-group">
        <label for="email">Email address:</label>
        <input type="text" class="form-control" ng-model="username">
      </div>
      <div class="form-group">
        <label for="pwd">Password:</label>
        <input type="password" class="form-control" ng-model="password" >
      </div>
      <!-- <div class="checkbox">
        <label><input type="checkbox"> Remember me</label>
      </div -->
      <button type="submit" class="btn btn-primary" ng-click="signIn()">Sing in</button>
    </form>
  </div>
</div>
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

Quedando de la siguiente forma a la vista del usuario final.

Figura 25. Vista de *login*



The screenshot shows a web interface for logging in. At the top, there is a navigation bar with links: "Promociones en este momento", "Promociones del día", "Todas las promociones", "Promociones por establecimiento", and a "Sing in" button. Below this is a blue header with the text "Sing in". The main form area contains two input fields: "Email address:" and "Password:". Below the password field is a blue "Sing in" button.

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

La vista del *login* utiliza la función de `signIn()` que se encuentra en el archivo `script.js` que es la que se comunica con el servidor para realizar el *login* del usuario.

Figura 26. Fragmento de función javascript para *login*

```
$scope.signIn = function() {
  console.log("iniciando signIn");
  var data = {
    username:$scope.username,
    password:$scope.password
  }
  console.log(data.username);
  console.log(data.password);
  $http.post('/login/signin',data).success(function(data, status, headers, config) {
    if(data.is_logged){
      $window.location.href= "/home#?ui="+data.ui+"&na="+data.name;
    }else{
      alert("Error, usuarios invalidos");
    }
  }).error(function(data,status){
    alert("ERROR DE CONEXION");
  });
}
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### **4.3.3.2.2. Vista crear promoción**

Al estar dentro de la aplicación como usuario se registra el tipo de empresa, se pueden subir promociones y programar que día y hora están desplegadas y así los usuarios podrán visualizar.

El código HTML es fundamental para mostrar la parte estática de la vista y angular para la parte dinámica y enlazar las variables que serán enviadas al servidor con los campos de la vista.

Figura 27. Fragmento de HTML con angular de la vista creación de promociones

```

<div class="panel panel-primary" >
  <div class="panel-heading"> Sing in </div>
  <div class="panel-body">
    <form ng-controller="MyCtrl as up" name="up.upload_form">
      <div class="form-group">
        <label for="email">Nombre:</label>
        <input type="text" class="form-control" ng-model="up.name">
      </div>
      <div class="form-group">
        <label for="pwd">Dia:</label>
        <div class="form-group">
          <select class="form-control" ng-model="up.dia" id="seldia">
            <option value="2"> Lunes </option>
            <option value="3"> Martes </option>
            <option value="4"> Miercoles </option>
            <option value="5"> Jueves </option>
            <option value="6"> Viernes </option>
            <option value="7"> Sabado </option>
            <option value="1"> Domingo </option>
          </select>
        </div>
      </div>
      <div class="form-group">
        <label for="horai"> Hora de inicio: </label>
        <input id="timepicker1" ng-model="up.horaInicio" type="text" class="form-control input-small"> </input>
      </div>
      <div class="form-group">
        <label for="horaf"> Hora en que finaliza: </label>
        <input id="timepicker2" ng-model="up.horaFin" type="text" class="form-control input-small"> </input>
      </div>
      <div class="form-group">
        <label class="control-label">Seleccionar Banner: </label>
        <input
          type="file"
          ngf-select
          ng-model="up.file"
          name="file"
          ngf-pattern="'image/*'"
          accept="image/*"
          ngf-max-size="20MB"
        />
        Image thumbnail: <img style="width:100px;" ngf-thumbnail="up.file || '/thumb.jpg'"/>
        <i ng-show="up.upload_form.file.$error.required">*required</i><br>
        <i ng-show="up.upload_form.file.$error.maxSize">File too large
        {{up.file.size / 1000000|number:1}}MB: max 20M</i>
        <!--<button type="submit" ng-click="up.submit()">submit</button> -->
        <p>{{up.progress}}</p>
      </div>
      <button type="submit"
        class="btn btn-primary" ng-click="up.submit(controlador.id)">Cargar Banner</button>
    </div>
    <!-- <div class="checkbox">
      <label><input type="checkbox"> Remember me</label>
    </div> -->
  </form>
</div>
</div>

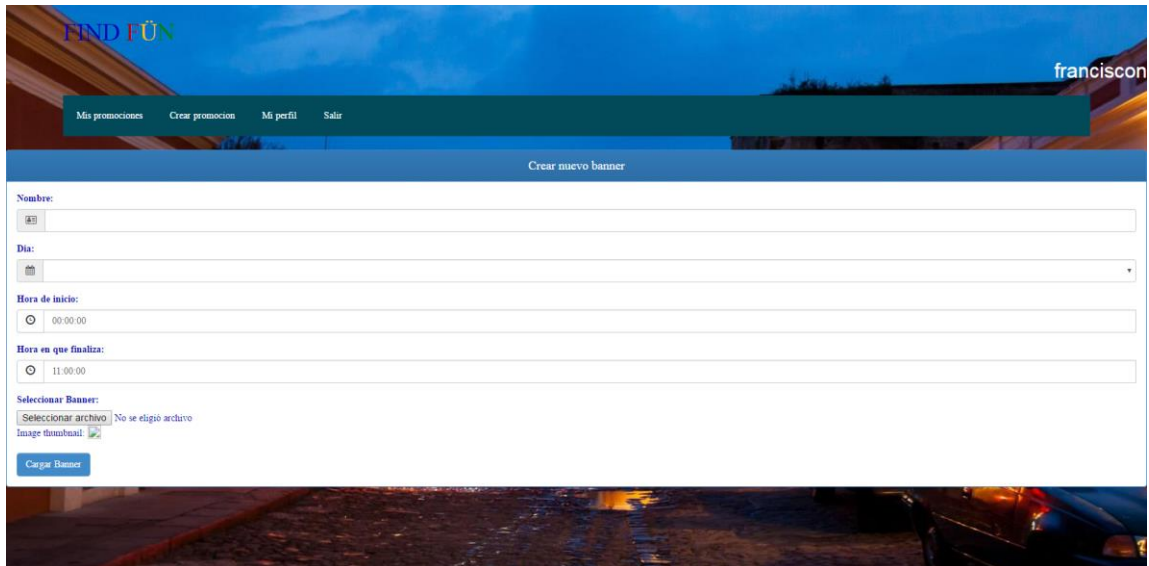
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.



Se tiene como resultado la siguiente vista:

Figura 28. **Vista de creación de promociones**



The screenshot shows a web interface for creating a new banner. At the top, there is a navigation bar with the logo 'FIND FÚN' on the left and the user name 'francisco' on the right. Below the navigation bar, there is a dark green menu with the following items: 'Mis promociones', 'Crear promocion', 'Mi perfil', and 'Salir'. The main content area is titled 'Crear nuevo banner' and contains the following form fields:

- Nombre:** A text input field with a small icon on the left.
- Día:** A dropdown menu with a calendar icon on the left.
- Hora de inicio:** A time selection field with a radio button and the value '00:00:00'.
- Hora en que finaliza:** A time selection field with a radio button and the value '11:00:00'.
- Seleccionar Banner:** A section with a 'Seleccionar archivo' button and the text 'No se eligió archivo'. Below this is an 'Image thumbnail' field with a small image icon.
- Cargar Banner:** A blue button at the bottom of the form.

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.3.2.3. **Vista de promociones activas**

Las promociones según su configuración podrán ser visualizadas por los usuarios.

Figura 29. Fragmento de código HTML con angular de la vista de promociones activas

```
<div data-ng-init="getListDay()">
  <div class="panel panel-default" height="20%" ng-repeat="data in lista">
    <div class="panel-heading"> {{data.Nombre}} </div>
    <div class="panel-body" back-img="{{data.imagen}}">
      <div class="btn btn-default btn-circle" >
         </img>
      </div>
      <div class="btn btn-default btn-circle" >
         </img>
      </div>
    </div>
  </div>
</div>
```

Fuente: elaboración propia.

Con lo que se obtiene la siguiente vista como resultado:

Figura 30. Vista de promociones activas



Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.3.2.4. Vista de ampliación de promoción

Para ver más detalles de la promoción publicada se da un *click*. A continuación, se muestra el código HTML y angular de esta vista:

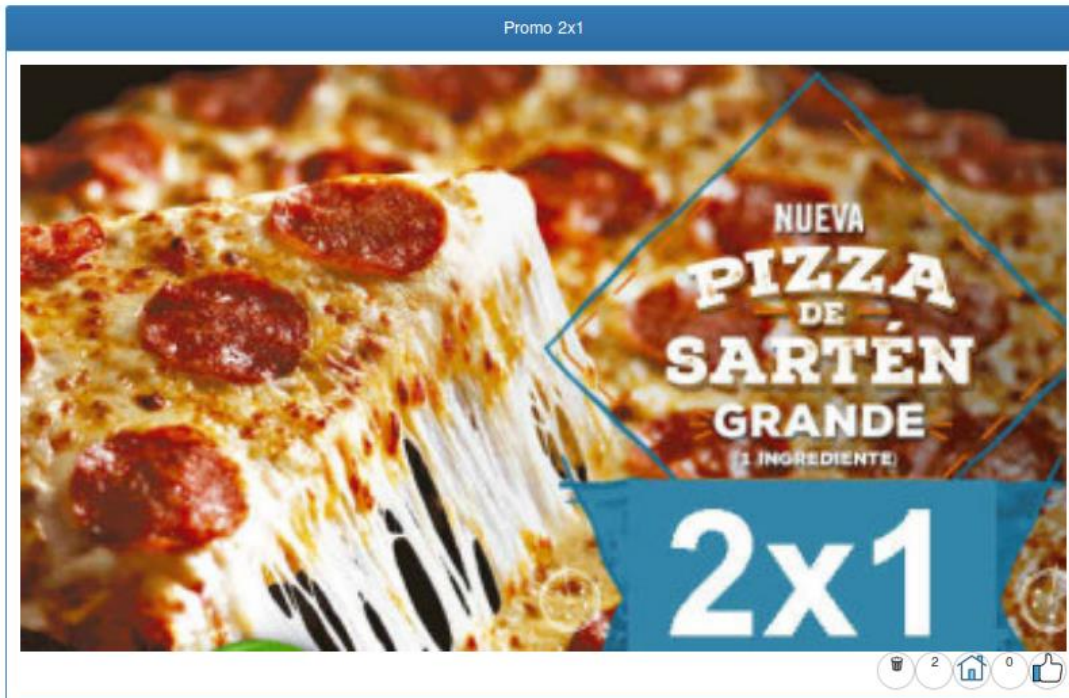
Figura 31. Fragmento de código HTML con angular de la vista de promoción ampliada

```
<div data-ng-init="getList()">
  <div class="panel panel-default" height="20%" ng-repeat="data in lista">
    <div class="panel-heading"> {{data.Nombre}} </div>
    <div class="panel-body" back-img="{{data.Imagen}}">
      <div class="btn btn-default btn-circle" >
        | {{data.megusta}}
      </div>
      <div class="btn btn-default btn-circle" >
        | {{data.asistire}}
      </div>
      <div class="btn btn-default btn-circle" ng-click="eliminar(data.idBanner)">
         </img>
      </div>
    </div>
  </div>
</div>
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

De esta forma se puede observar el resultado: una imagen de alta calidad, con la información más relevante de la promoción, por ejemplo, cantidad de me gusta de la promoción, cantidad de personas que asistirán al evento y el nombre de la promoción. La cual se puede observar de la siguiente manera:

Figura 32. Vista ampliación de promoción



Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.3.3.2.5. Vista suscripción por email

Hoy en día, muchos usuarios quieren tener a su disposición la información actual de las promociones de los distintos establecimientos de su preferencia. Los usuarios lo lograrán suscribiéndose por correo electrónico: llegará desde el servidor a su correo registrado la información.

La información recolectada del cliente es almacenada en la base de datos, utilizada periódicamente para el envío de correos en las promociones de las cuales está suscrito.

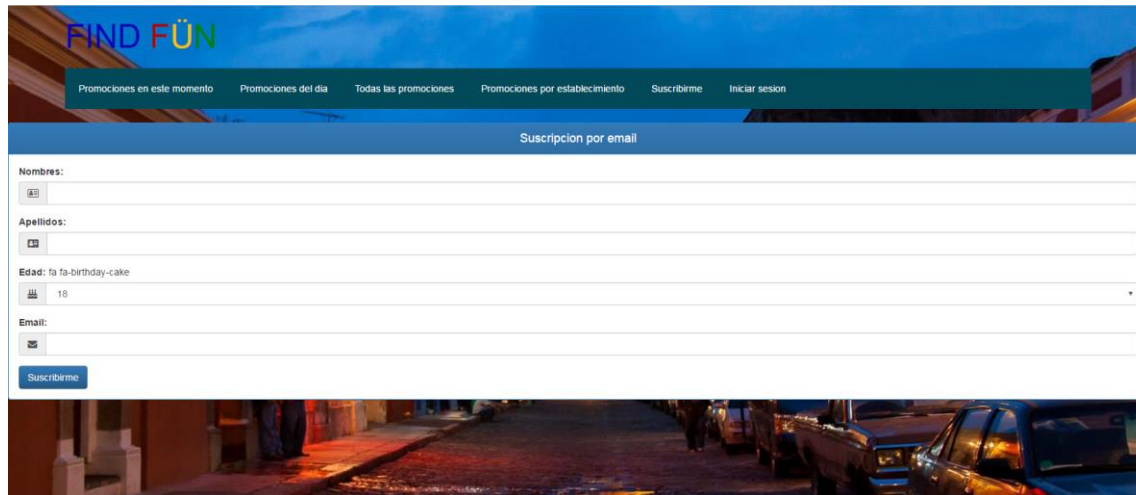
Figura 33. Fragmento de código HTML con angular de la vista de suscripción por *email*

```
<div class="panel panel-primary" ng-controller="controlador_lista">
  <div class="panel-heading"> Suscripcion por email </div>
  <div class="panel-body">
    <form role="form">
      <div class="form-group">
        <label for="email">Nombres:</label>
        <input type="text" class="form-control" ng-model="usernames">
      </div>
      <div class="form-group">
        <label for="email">Apellidos:</label>
        <input type="text" class="form-control" ng-model="userlastnames">
      </div>
      <div class="form-group">
        <label for="email">Edad:</label>
        <select class="form-control" ng-model="userage" ng-options="item for item in edades" id="selage">
        </select>
      </div>
      <div class="form-group">
        <label for="pwd">Email:</label>
        <input type="text" class="form-control" ng-model="usermail" >
      </div>
      <!-- <div class="checkbox">
        <label><input type="checkbox"> Remember me</label>
      </div -->
      <button type="submit" class="btn btn-primary" ng-click="suscribirse()">Suscribirse</button>
    </form>
  </div>
</div>
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

Con lo que se obtiene la siguiente vista como resultado:

Figura 34. **Vista de suscripción por email**



The screenshot displays the FindFun website's email subscription interface. At the top, the 'FIND FÜN' logo is visible on the left, and a navigation menu on the right includes links for 'Promociones en este momento', 'Promociones del día', 'Todas las promociones', 'Promociones por establecimiento', 'Suscríbeme', and 'Iniciar sesión'. Below the navigation bar, a blue header contains the text 'Suscripción por email'. The main form area includes several input fields: 'Nombres:' with a text box and a small icon; 'Apellidos:' with a text box and a small icon; 'Edad: fa-birthday-cake' with a dropdown menu showing '18'; and 'Email:' with a text box and an email icon. A blue 'Suscríbeme' button is positioned below the email field. The bottom of the page features a background image of a street scene at night with cars and buildings.

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### **4.3.3.2.6. Vista para edición de perfil**

Como usuario registrado de la gran familia de FindFun, cada empresa tiene la opción de personalizar su perfil registrado: nombre, apellido, alias y su respectiva imagen pública. De esta manera los usuarios (consumidores), podrán identificar de mejor manera cada una de las empresas registradas.

Figura 35. Fragmento de código HTML con angular para la vista para edición de perfil

```
<div class="panel panel-primary">
  <div class="panel-heading"> Modificar perfil </div>
  <form ng-controller="profileController as pe" name="pe.upload_form">
  <div class="panel-body" data-ng-init="pe.getProfileInfo()">
    <div class="form-group">
      <label for="email">Nombre:</label>
      <input type="text" class="form-control" ng-model="pe.name">
    </div>
    <div class="form-group">
      <label for="apellido">Apellido: </label>
      <input type="text" class="form-control" ng-model="pe.lastname">
    </div>
    <div class="form-group">
      <label for="alias">Alias: </label>
      <input type="text" class="form-control" ng-model="pe.alias">
    </div>
    <div class="form-group">
      <label class="control-label">Seleccionar Imagen: </label>
      <input
        type="file"
        ngf-select
        ng-model="pe.file"
        name="file"
        ngf-pattern="'image/*'"
        accept="image/*"
        ngf-max-size="20MB"
      />
      Image thumbnail: <img style="width:100px;" ngf-thumbnail="pe.file || '/thumb.jpg'"/>
      <i ng-show="pe.upload_form.file.$error.required">*required</i><br>
      <i ng-show="pe.upload_form.file.$error.maxSize">File too large
      {{pe.file.size / 1000000|number:1}}MB: max 20M</i>
      <p>{{pe.progress}}</p>
    </div>
    <button type="submit"
      class="btn btn-primary" ng-click="pe.submit(controlador.id)">Guardar</button>
  </div>
</form>
</div>
</div>
```

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

Figura 36. Vista para edición de perfil

The screenshot shows a web interface for editing a user profile. At the top, there is a dark blue header with the 'FIND FUN' logo on the left and the name 'francisco' on the right. Below the header is a navigation bar with three links: 'Crear promoción', 'Mi perfil', and 'Salir'. The main content area is titled 'Modificar perfil' and contains a form with the following fields: 'Nombre:' with the value 'francisco', 'Apellido:' with the value 'noj', and 'Alias:'. Below these fields is a section for 'Seleccionar imagen:' with a file selection button and a message 'No se eligió archivo'. A 'Guardar' button is located at the bottom of the form.

Fuente: elaboración propia, empleando la herramienta online balsimiq.com.

#### 4.4. Consideraciones de implementación

Existen varias consideraciones a tomar en cuenta para el desarrollo de la aplicación o para mejoras y expansión de futuras funcionalidades, las cuales se mencionan a continuación:

- El manejo de datos y su correspondiente DML se desarrolló completamente en procedimientos almacenados dentro del DBMS de MySQL para no colocar sentencias SQL dentro de la aplicación para dar seguridad a la integridad de la base de datos. Por otra parte, al tener un problema o mantenimiento de la aplicación se logra identificar rápidamente el procedimiento almacenado y se hace su actualización correspondiente sin necesidad de volver a desplegar la aplicación.
- El intercambio entre los procedimientos almacenados y la aplicación es a través de la notación JSON (*javascript object notation*) por lo que toda manipulación de datos se hace a través de esta notación.



## CONCLUSIONES

1. Se desarrolló una aplicación que centraliza las promociones de diversas empresas que puede ser accedida por cualquier usuario desde cualquier dispositivo con conexión a internet.
2. En esta aplicación todo usuario registrado como empresa puede publicar sus promociones para que sus clientes puedan visualizar, comparar y elegir las promociones que más se adapten a sus necesidades.
3. La aplicación desarrollada incluye el manejo de suscripción de clientes a través del envío de correos electrónicos en los cuales se incluyen las promociones del día.
4. La aplicación facilita el proceso de creación e implementación de una herramienta para la publicidad y el *marketing* de las diversas promociones en las empresas; representa un beneficio para los usuarios y para las mismas empresas.



## RECOMENDACIONES

1. Dar seguimiento al funcionamiento de la aplicación para mejorarlo continuamente hasta que el programa funcione tomando en cuenta las necesidades de los usuarios.
2. Verificar las transacciones en la base de datos para evitar cuellos de botella y bloqueos.
3. Monitorear el almacenamiento en el servidor para evitar que el envío de correos se detenga y se pierdan clientes debido a que no se informa de las promociones del día.



## BIBLIOGRAFÍA

1. Angular Js. *Documentación*. [en línea]. <<https://angularjs.org>>. [Consulta: 9 de septiembre de 2016].
2. Bootstrap. *Documentación*. [en línea]. <<https://www.w3schools.com/bootstrap/>>. [Consulta: 11 de octubre de 2016].
3. HTML5. *Documentación*. [en línea]. <[https://www.w3schools.com/html/html5\\_intro.asp](https://www.w3schools.com/html/html5_intro.asp)>. [Consulta: 9 de septiembre de 2016].
4. Node js. *Documentación*. [en línea]. <<https://nodejs.org/es/>>. [Consulta: 21 de septiembre de 2016].
5. Nodemailer. *Documentación*. [en línea]. <<https://www.nodemailer.com/>>. [Consulta: 25 de febrero de 2017].
6. MySQL. *Documentación*. [en línea]. <<https://www.mysql.com/>>. [Consulta: 29 de agosto de 2016].

