



**UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN**



**DESARROLLO DE SOLUCIONES WEB Y
MÓVIL PARA LA INTEGRACIÓN DE
MARKETING DE PROXIMIDAD Y
GESTIÓN PUBLICITARIA CON
DISPOSITIVOS BEACON COMO
TECNOLOGÍA BASE**

**Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela**

Por los bachilleres:

Br. Alvarado, Edwin.

Br. Díaz, Pedro.

**para optar al título de
Licenciado en Computación**

Tutor:

Prof. Sandoval, Franklin

Caracas, Octubre de 2016



Universidad Central de Venezuela.
Facultad de Ciencias
Escuela de Computación

DESARROLLO DE SOLUCIONES WEB Y MÓVIL PARA LA INTEGRACIÓN DE MARKETING DE PROXIMIDAD Y GESTIÓN PUBLICITARIA CON DISPOSITIVOS BEACON COMO TECNOLOGÍA BASE

Autores:

Edwin Elías Alvarado Morantes.

Pedro Alejandro Díaz Meza.

Fecha: 14 de Octubre de 2016.

RESUMEN

Actualmente nos encontramos en una era donde la tecnología tiene un crecimiento acelerado y exponencial, donde existen infinitudes de temas y áreas desarrolladas con el mismo o similar fin; para tener una mejora evolutiva en una sociedad o país, como la visión de administrar mejor el consumo de tiempo y energía, es necesario hacer la integración de nuevas tecnologías que ayuden a la automatización y eficiencia de procesos con el fin promover de una manera distinta un negocio. En este trabajo de grado se toma como propósito el desarrollo de una solución orientada a mejorar la manera en que se implementa el marketing digital en la actualidad, el objetivo general fue desarrollar una solución de marketing de proximidad y gestión publicitaria con dispositivos *Beacon* como tecnología base que permita la visualización y gestión de promociones a través del desarrollo de dos aplicaciones. Así mismo, esta solución utilizara tecnología *Bluetooth Low Energy* como base, esta tecnología viene dando de qué hablar gracias a la estandarización de este enfoque por *iBeacon* en el 2013 en la Conferencia Mundial de Desarrolladores de Apple, abriendo así un mundo de posibilidades en el entorno de las aplicaciones móviles. Durante la investigación se recogió la información de desarrollo del proyecto, en el cual se aplicaron fundamentos y herramientas aptas para el desarrollo adecuado de una solución que integrara una aplicación móvil para los consumidores o clientes de los comercios afiliados y una aplicación web para la administración de promociones por parte de estos comercios..

Palabras Claves: Aplicaciones móviles, tecnologías, metodologías, propuesta, prototipo funcional, *Beacon*, *Bluetooth*, protocolo, Servicios REST.



Universidad Central de Venezuela.
Facultad de Ciencias
Escuela de Computación

ACTA

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado con el título **“DESARROLLO DE SOLUCIONES WEB Y MÓVIL PARA LA INTEGRACIÓN DE MARKETING DE PROXIMIDAD Y GESTIÓN PUBLICITARIA CON DISPOSITIVOS BEACON COMO TECNOLOGÍA BASE”**, el cual es presentado por el Br. Edwin Alvarado, de Cédula de Identidad 25.304.762 y el Br Pedro Díaz, de Cédula de Identidad 22.037.492 , a los fines de optar al título de Licenciado en Computación, dejamos en constancia lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 21 de Octubre de 2016, a la 1:00 pm, para que sus autores lo defendieran en forma pública en la Sala PB-3 de la Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, lo cual se realizó mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con una nota de 20 puntos.

En fe de lo cual se levanta la presente acta, en Caracas a los 21 días del mes de Octubre del año 2016.

Prof. Franklin Sandoval
Tutor

Profa. Ospina Mercy.
Jurado

Profa. Franky Uzcátegui
Jurado

DEDICATORIAS

Quiero dedicar este trabajo especial de grado en primer lugar a mi familia ya que han estado allí en todo momento, me han visto crecer, triunfar y gracias a ellos es que estoy donde estoy y seguiré siendo no solo mejor persona sino mejor profesional, gracias a mi madre Mariela, mi padre Jose Elías y mi hermana Verónica por ser parte de esta y todas mis metas, sin ustedes esto no sería posible, gracias. Así como también especialmente a mi abuelo paterno Fermín ya que ha sido desde el inicio de mi carrera uno de los motivos más grandes que me ha ayudado a seguir, no te tengo en vida pero te llevo siempre en mi corazón. No puede faltar a los que hoy considero mi segunda familia, Massi Lombardo ya que fueron y espero sigan siendo personas especiales que me apoyaron en esta fase de culminación, Gracias a Viviana también por ser esa persona que me da ese impulso y apoyo para seguir con todo lo que me propongo, por acompañarme y guiarme en este camino.

Dedico este trabajo también a Pedro Diaz, ya que más que mi compañero de tesis se ha convertido en mi gran socio y hermano. Tu gran motivación y tus ganas de seguir fueron factores que nos ayudaron a cumplir esto que hoy en día estamos logrando que es el último paso de nuestra carrera, fue un orgullo haber compartido contigo todos estos momentos de traspaso pero ambos sabemos que valió la pena y la seguiré valiendo, ya que somos más que soñadores, somos ProMotion Team.

Edwin E. Alvarado M.

Primeramente gracias a Dios por haberme puesto en el camino las oportunidades que me puso, quiero agradecer a mi familia y especialmente a mi padre y a mi madre, que lo son todo para mí, sin ellos no habría cabida para esta meta y gracias a ellos mismos hoy tengo la oportunidad de alguna forma retribuirles todo lo que han hecho por mí. A mi eterna luchadora y ejemplo de vida, mi abuela Rafaela, sé que en donde se encuentre estará orgullosa de su nieto preferido. A mis amigos por siempre darme esas palabras de aliento en los momentos más difíciles, y no me puedo olvidar de Adriana, gracias por todo ese apoyo que me has brindado en esta etapa de mi vida, fuiste pieza fundamental de este logro que hoy alcanzo y disfruto contigo.

Edwin Alvarado, mi compañero de un simple proyecto que se terminó convirtiendo en mi gran hermano y socio. Te dedico esta victoria, no permitas que nadie te quite esas ganas de hacer todo lo que sueñas, y menos aún que compartimos los mismos sueños. Gracias por tu esmero, tus ganas de no rendirte y por sobre todo por reclamar ese último esfuerzo cuando parece que ya no podemos seguir. Aunque parezca un fin, esto apenas comienza y yo sé que valdrá la pena.

Pedro A. Díaz M.

AGRADECIMIENTOS

Para la culminación de este Trabajo Especial de Grado, muchas personas sirvieron de apoyo y ayuda para que el mismo se realizara con total éxito, por tal motivo queremos agradecerles el hecho de habernos ayudado, así sea en un mínimo instante o durante toda la carrera.

Le agradecemos principalmente a dios que nos da sus bendiciones y fuerza diariamente así como también a todos nuestros familiares por su preocupación, apoyo y motivación durante la realización de este trabajo especial de grado. En especial a nuestras madres y padres, ya que con la educación y los principios que nos impartieron pudimos crecer y convertirnos en los profesionales que somos actualmente; de igual manera agradecemos su guía, amor y apoyo incondicional que forman una parte vital del éxito de esta investigación.

Gracias a nuestros amigos por acompañarnos durante esta aventura, por estar ahí en los momentos más difíciles y cuando eran más necesarios. El impulso que nos dieron fue absolutamente vital para culminar nuestra carrera exitosamente.

Gracias a nuestro tutor, Franklin Sandoval, por ser no solo nuestro profesor y amigo durante nuestra carrera y proyecto de grado, sino también excelente persona y profesional, futuro colega si dios nos da licencia. Gracias por estar con nosotros en todo momento y por impulsarnos en este emprendimiento. Por abrirnos puertas que nosotros no pensábamos tocar aun y por darnos la fuerza y la motivación de saber que siempre después del final de un camino, existe el inicio de otro. De verdad gracias.

Agradecemos a los profesores de la Facultad de Ciencias de la Universidad Central de Venezuela y a la Universidad, por brindarnos los conocimientos, las metodologías y, sobre todo, la estructura de pensamiento y resolución de problemas que son esenciales para ser un profesional exitoso en el área de la computación, además de impulsarnos con cada enseñanza a ser cada día mejores.

ÍNDICE

ÍNDICE	6
ÍNDICE DE TABLAS.....	8
ÍNDICE DE FIGURAS.....	9
INTRODUCCIÓN.....	11
CAPÍTULO 1 PLANTEAMIENTO DEL PROBLEMA	13
1.1. OBJETIVOS GENERALES.....	16
1.2. OBJETIVOS ESPECÍFICOS DEL TEG	16
1.3. PROPUESTA DE SOLUCIÓN.....	16
1.4. ARQUITECTURA DE LA SOLUCIÓN	18
CAPÍTULO 2 MARCO CONCEPTUAL	20
2.1. SISTEMAS DE INFORMACIÓN	20
2.2. DISPOSITIVOS MÓVILES	21
2.3. SISTEMAS OPERATIVOS MÓVILES.....	22
2.4. ARQUITECTURA CLIENTE/SERVIDOR.....	23
2.5. APLICACIONES WEB.....	24
2.6. APLICACIONES MÓVILES.....	26
2.7. TECNOLOGÍAS DE DESARROLLO.....	27
2.8. ANTECEDENTES DE DESARROLLO	50
CAPÍTULO 3 MARCO METODOLÓGICO	52
3.1. METODOLOGÍAS TRADICIONALES	52
3.2. METODOLOGÍAS ÁGILES	53
CAPÍTULO 4 MARCO APLICATIVO.....	64
4.1. METODOLOGIA IMPLEMENTADA.....	64
4.2. HERRAMIENTAS DE SOLUCIÓN.....	65
4.3. DISEÑO DEL SISTEMA	72

4.4. DISEÑO DE LA SOLUCIÓN	79
4.5. DISEÑO DEL API.....	100
4.6. IMPLEMENTACION Y DISEÑO DE PROTOTIPO DE APLICACIÓN WEB.....	104
4.7. IMPLEMENTACION Y DISEÑO DE PROTOTIPO DE APLICACIÓN MÓVIL	107
4.8. PRUEBAS FUNCIONALES Y RESULTADOS	114
CONCLUSIONES Y RECOMENDACIONES.....	138
REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES.....	141
ANEXOS	146
1. MANUAL DE APLICACIÓN MÓVIL APP.....	147
2. MANUAL DE APLICACIÓN WEB APP.....	171

Índice de tablas

Tabla 1 - Sistemas operativos más usados en Venezuela	23
Tabla 2 - Tipos de aplicaciones móviles	26
Tabla 3 - Diferencias entre GPS, NFC, Beacon.....	30
Tabla 4 - Ventajas y desventajas de bases de datos	45
Tabla 5 - Comparación entre MySQL y PostgreSQL	49
Tabla 6 - Diferencias entre metodologías ágiles y tradicionales.....	53
Tabla 7 - Historia de usuario #1 - Sprint 0.....	73
Tabla 8 - Historia de usuario #2 - Sprint 0.....	74
Tabla 9 - Historia de usuario #3 - Sprint 1.....	75
Tabla 10 - Historia de usuario #4 - Sprint 1.....	76
Tabla 11 - Historia de usuario #5 - Sprint 2.....	77
Tabla 12 - Historia de usuario #6 - Sprint 2.....	78
Tabla 13 - Documentación API.....	103

Índice de figuras

Ilustración 1 - Arquitectura de la solución.	18
Ilustración 2 - Modelo de Cliente/Servidor	24
Ilustración 3- Como funcionan los Beacon	33
Ilustración 4 - Registro XML	35
Ilustración 5 - Asignación de color a elemento DOM	37
Ilustración 6 - Función Suma	38
Ilustración 7 - Ejemplo básico JQuery	38
Ilustración 8 - Hola mundo Java.....	39
Ilustración 9 - Ejemplo Hola Mundo en PHP.....	40
Ilustración 10 - Ejemplo método sumar Ruby	41
Ilustración 11 - Arquitectura Modelo Vista Controlador.....	42
Ilustración 12 - Diseño Arquitectónico Framework	43
Ilustración 13 - Clico de vida XP.	56
Ilustración 14 - Ciclo de vida RUP.	58
Ilustración 15 - Ciclo de vida de eventos dentro de un Sprint.	62
Ilustración 16 - Plantilla de documento de trazabilidad.....	67
Ilustración 17 - Plantilla de Definición de Completado.....	68
Ilustración 18 - Plantilla de Control de Cambios.....	68
Ilustración 19 - Tablero de ciclo de vida de una actividad.	69
Ilustración 20 - Git en Android Studio.	70
Ilustración 21 - Nube Estimote.....	71
Ilustración 22 - Heroku - Administración y publicación de aplicación web.	71
Ilustración 23 - Consola de desarrollador de google.	72
Ilustración 24 - Caso de uso #1, modelado móvil.....	80
Ilustración 25 - Caso de uso #2. Rol cliente	85
Ilustración 26 - Caso de uso #3, rol Administrador	86
Ilustración 27 - Caso de uso #4. Rol propietario	87
Ilustración 28 - Diagrama de clases de la solución.....	99
Ilustración 29 - Diagrama de Base de Datos	100
Ilustración 30 - Definición de ámbito API.	101
Ilustración 31 - Directorio de controladores de API.	101
Ilustración 32 - Directorio de vistas de API.	102
Ilustración 33 - Estructura de API Restful, tiendas.	103
Ilustración 34 - Integración Bootstrap.	105
Ilustración 35 - Ejemplo Bootstrap.....	105
Ilustración 36 - Configuración de manejador PostgreSQL.	106
Ilustración 37 - Dependencia Retrofit Square.	108
Ilustración 38 - Definición de clase interfaz para detalle de promociones.....	109

Ilustración 39 - Clase Objeto (Promoción).....	109
Ilustración 40 - Asignación de host de servicio.....	109
Ilustración 41 - Asignación de URL de servicio.....	110
Ilustración 42 - Llamar Servicio con Retrofit.....	110
Ilustración 43 - Librería Estimote.	111
Ilustración 44 - Servicios en segundo plano.	112
Ilustración 45 - Inicialización e inicio de monitoreo dispositivo Beacon.	112
Ilustración 46 - Construcción y envío de notificación.	113
Ilustración 47 - Despliegue de notificación en barra superior del dispositivo.	113
Ilustración 48 - Configuración de proxy.....	115
Ilustración 49 - Captura de peticiones Blazemeter.	116
Ilustración 50 - Definición de plan de prueba.	117
Ilustración 51 - Tabla resultado, tráfico de red (50 usuarios).	119
Ilustración 52 - Carga de CPU - Bluetooth.....	120
Ilustración 53 - Carga de CPU - Prototipo ProMotion (Usuario).	121
Ilustración 54 - Carga de CPU - Prototipo ProMotion (kernel).	121
Ilustración 55 - Encuesta - Prueba de Aceptación App Móvil.....	122
Ilustración 56 - Encuesta - Prueba de Aceptación App Web.....	126
Ilustración 57 - Comentario de Usuarios App Movil.....	130
Ilustración 58 - Comentario de Usuarios App Web.....	131
Ilustración 59 - Imagen de promoción. App Movil	131
Ilustración 60 – Compartir App, modulo más cerca. App Movil.....	132
Ilustración 61 - Actualización de aplicación. App Movil	132
Ilustración 62 - Registro de teléfono celular. App web	133
Ilustración 63 - Estadísticas de inicio. App web.....	133
Ilustración 64 - Actualización de imágenes de promociones. App web	134

INTRODUCCIÓN

Los consumidores se han vuelto más exigentes, las comunicaciones unidireccionales ya no tienen el mismo efecto que solían tener, con la llegada de la web 2.0 ahora todo gira en torno a la interacción, a la capacidad que tiene el consumidor de poder recibir directamente la información y generar una retroalimentación inmediata.

Así mismo la alta competitividad que hay entre las empresas, marcas y productos, además de la necesidad de buscar vías alternas para darse a conocer, ha traído como resultado el nacimiento de nuevas formas de hacer publicidad, conocidas como publicidad no convencional. Para los anunciantes en la actualidad no es suficiente ofrecer los productos que puedan satisfacer las necesidades de los consumidores. Deben apostar no sólo a aumentar sus ventas, sino a generar recuerdo, posicionamiento y a estar dentro del *top of mind* pero ¿cómo logran esto? Claro, haciendo publicidad, y para ello deben hacer uso de los nuevos recursos que la tecnología pone en sus manos.

El marketing de proximidad existe desde hace muchos años y es utilizado en todo momento por las diferentes compañías un ejemplo tangible y constante de este tipo de publicidad son los volantes. Sin embargo, el marketing de proximidad se ha adecuado a las necesidades que presenta la sociedad actual, una sociedad sumergida en un mundo tecnológico. Es esta nueva sociedad, con sus necesidades de interacción, ha dado pie a que surjan nuevos medios de publicidad, como lo son el *Bluetooth* y el *Wi-Fi*, que más que redes inalámbricas que permiten las conexiones entre diversos dispositivos, ahora son parte de las herramientas de publicidad utilizadas por los anunciantes para comunicarse tú a tú con los consumidores. Estas redes inalámbricas utilizadas dentro del mundo publicitario se consideran parte del mencionado marketing de proximidad. Sin embargo, para que las herramientas de publicidad

tengan un alto alcance, impacto y generen beneficios positivos para quien anuncia, deben ser aprobadas por los consumidores que son quienes van a recibir la información y decidir qué harán con ella. Aunque para muchos puede que el escuchar la palabra Bluetooth le suene al pasado, a hace unos años cuando los dispositivos se ceñían a este tipo de tecnología o a los tan sufridos infrarrojos, esta conexión sigue vigente y es parte de la vida diaria, que puede aportar una nueva funcionalidad, como esta, en cualquier campaña de gestión de presencia en internet. En este contexto entran los *Beacons*, faros en inglés, es un modo de conexión entre la empresa, negocio o ubicación, con el usuario de cualquier tipo de dispositivo móvil. Basándose en la tecnología *Bluetooth* de baja energía estos dispositivos, también denominados balizas informativas, son un paso más allá en lo relativo a la comunicación entre cliente y usuario.

Capítulo 1. En este capítulo se plantea el contexto del trabajo de grado y la descripción del planteamiento del problema. También se describe la justificación de la investigación, objetivo general, objetivos específicos, principales requerimientos, arquitectura de la solución y propuesta de la solución.

Capítulo 2. En este capítulo se define el marco conceptual de esta investigación donde se pueden encontrar los conceptos bases referentes a dispositivos móviles, aplicaciones móviles y web, tecnología *Beacon*, tecnología BLE como tecnología base y tecnologías de desarrollo.

Capítulo 3. En este capítulo se despliega la información que llevan de la mano las metodologías para el desarrollo. En el cual se definirá esta metodología como propia de esta investigación y su implementación en el proyecto final.

Capítulo 4. En este capítulo de despliega principalmente el marco aplicativo para el trabajo especial de grado, así como las herramientas utilizadas y procesos realizados para la culminación del mismo.

Finalmente se plantearan las conclusiones y recomendaciones, anexos y referencias bibliográficas.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

Todos los tejidos de la sociedad han venido evolucionando, unos más que otros, a pasos acelerados como es el caso de la tecnología. La sociedad se ha convertido en un constante dar y recibir información, y el proceso de la comunicación, dentro del cual se encuentra la publicidad, ha tenido que evolucionar a la par para poder adaptarse. Así mismo la alta competitividad que hay entre las empresas, marcas y productos y la necesidad de buscar vías alternas para darse a conocer, ha traído como resultado el nacimiento de nuevas formas de hacer publicidad usando estrategias similares a otras ya existentes, como por ejemplo publicidad detallista (local) en la cual los comerciantes venden directamente al consumidor y en donde generalmente existe una alta competitividad entre comerciantes (Russell,2005).

Los consumidores se han vuelto más exigentes, las comunicaciones unidireccionales ya no tienen el mismo efecto que solían tener, con la llegada de la web 2.0 ahora todo gira en torno a la interacción, a la capacidad que tiene el consumidor de poder recibir directamente la información y generar una retroalimentación inmediata.

Para los anunciantes hoy en día no es suficiente ofrecer los productos que puedan satisfacer las necesidades de los consumidores. Deben apostar no sólo a aumentar sus ventas, sino a generar recuerdo y posicionamiento, y esto se logra a través de publicidad, y para esto dichos comercios deben hacer uso de los nuevos recursos que la tecnología pone en sus manos.

El marketing de proximidad existe desde hace muchos años, es utilizado en todo momento por diferentes compañías, un ejemplo tangible y constante de este tipo de publicidad es la entrega de volantes. Sin embargo este término, se ha adecuado a las necesidades que presenta la sociedad actual, una sociedad sumergida en un

ambiente tecnológico. Una primera definición de marketing de proximidad, la brinda Altiria (2016), compañía especializada en marketing móvil. De acuerdo a esta firma, el marketing de proximidad “engloba las acciones de marketing que tienen en cuenta la ubicación actual del target o cliente potencial”. La gran ventaja es que este tipo de marketing permite “segmentar según su posición a los destinatarios, consiguiendo una mayor efectividad por impactar al target en el entorno de interés de la campaña”.

En el pasado las técnicas de publicidad se consideraban como: repartir folletos en mano, en donde la principal ventajas de este tipo de comunicación era la personalización del mensaje, la cercanía de tu público objetivo y la rapidez de la comunicación. Pero esta vertiente del marketing también se ha adaptado a las nuevas tecnologías.

Actualmente Venezuela, es uno de los países que posee mayor adopción de teléfonos o *Smartphones*, para cierto rango de edad, por eso esto se convierte en una gran oportunidad de establecer la comunicación con los clientes a través de la comodidad de su dispositivo móvil, siendo la mejor oportunidad el momento en que los consumidores próximos a la zona de interés (local o establecimiento). A partir de esto se puede incluir el termino *beacons*, dada que esta nueva tecnología permite comunicarte con los clientes de forma muy eficaz. A través de la conexión que se establece mediante el *bluetooth* se podrán enviar notificaciones al *smartphone* del consumidor.

Esta tecnología consiste en nodos *Bluetooth*, que envían información directamente a los dispositivos móviles, sin la necesidad de interactuar con las operadoras, dado esto se desarrolla una nueva vía de comunicación llamada “Marketing de Proximidad”, estos nodos envían una invitación a descargar el contenido al dispositivo. Se posee un radio de envío de data y sólo envía notificaciones a las personas que estén realizando alguna actividad en particular y por ende poseen algo en común.

El alcance se puede regular y se podrán tener diferentes *beacons* repartidos en una determinada zona para distintos locales comerciales, dando así la facilidad de enviar notificaciones de promociones a los usuarios que estén en los alrededores del

comercio. Las aplicaciones de estos nuevos dispositivos son infinitas. Se puede no sólo estudiar el comportamiento de los usuarios, sino también comunicarse con ellos con campañas específicas y personalizadas.

En este contexto, surge este emprendimiento que usa como tecnología base *Beacons* para captar la atención de clientes que se encuentren en los alrededores de un local comercial en un determinado momento, específicamente ofreciendo mercado de proximidad. En donde el modelo de negocio consiste en posicionar la tecnología como una nueva vía de comunicación, no sólo entregando la herramienta tecnológica, sino también al consumidor final, quien descarga los contenidos a través de un api.

Hay que resaltar que las pequeñas y grandes empresas invierten recursos monetarios y talento humano para la realización de publicidad y marketing, aplicando diversas estrategias de promoción o comunicación como; entrega de volantes, invitaciones a locales, anuncios promocionales, muestras de productos, entre otros. Con la finalidad de adquirir la atención de la mayor cantidad de consumidores posibles.

Las empresas que operan actualmente en Venezuela, por ejemplo Arturo's o Ferretotal, no ofrecen este servicio de publicidad o marketing de proximidad en tiempo real en sus locales comerciales y esto requiere que la información de productos y promociones sea solicitada o visualizada directamente en una tienda física o publicada en alguna red social. Desde la perspectiva del consumidor, se dificulta la obtención de información oportuna sobre productos y promociones que le permitan generar un ahorro monetariamente importante.

La consolidación de los dispositivos móviles en esta última década como instrumento de comunicación, ha facilitado la creación de un nuevo canal de marketing, concretamente el marketing de proximidad, una actual herramienta de comunicación empresarial para la difusión de contenidos e información. Aunque en estos últimos años, este modelo de *marketing* se ha consolidado más en un producto, que un mero instrumento de comunicación, gracias a su multifuncionalidad que aporta un valor añadido a los contenidos de servicios que se ofrecen, a la vez, que

proporciona mayores niveles de satisfacción en la mejora de las necesidades de los clientes.

En este contexto el presente trabajo especial de grado plantea la necesidad de establecer una solución para que consumidores que se encuentren en los alrededores de los locales comerciales accedan a información de los productos y promociones a través de la tecnología *Beacon*. Este prototipo funcional de solución estará compuesto por dos aplicaciones; una móvil orientada a los clientes o consumidores de los locales comerciales y una aplicación *web* orientada a los dueños de los comercios ya que esta se encargara de la administración de comercios y promociones.

1.1. Objetivos Generales

Desarrollar una solución de marketing de proximidad y gestión publicitaria con dispositivos *beacon* como tecnología base que permita la visualización y gestión de promociones a través del desarrollo de dos aplicaciones.

1.2. Objetivos Específicos del TEG

- Definir los requerimientos funcionales y no funcionales en el desarrollo de una aplicación web y móvil que controle el envío de alertas de publicidades.
- Desarrollar las interfaces y módulos de la propuesta *Web* y de la Aplicación Móvil.
- Implementar una forma de comunicación que permita la centralización de bases de datos que sustentara el acceso desde la aplicación web y a su vez desde la aplicación móvil.
- Desarrollar un conjunto de pruebas de carga, rendimiento, usabilidad, aceptación correspondientes a la evaluación de los componentes de las aplicaciones.

1.3. Propuesta de solución

El emprendimiento propuesto como trabajo especial de grado será una solución que integra principalmente, una aplicación móvil para la visualización de

promociones a través de una notificación perteneciente a locales comerciales a través de la tecnología *Beacon*, las cuales estarán administradas por dichos comercios a través de una aplicación web incluida como parte de la solución. También estará destinada a comercios que deseen ofrecer sus artículos o servicios a un público más amplio. En esta nueva era del marketing digital, es necesario que grandes y pequeñas empresas se mantengan a la vanguardia con la tecnología de punta, explotando la misma al máximo para obtener todos los beneficios que esta ofrece, tales como incremento de ventas, aumento de clientes potenciales, reducción de gastos de personal, entre otros.

Con esta solución se introduce una nueva forma de hacer marketing, la cual se encuentra dentro de la categoría de marketing por proximidad, debido a que los usuarios finales de la solución podrán acceder a promociones y servicios relacionados a locales comerciales, sin necesidad de estar físicamente dentro del mismo.

1.3.1. Principales requerimientos

- Acceso a Internet.
- Servidor para el alojamiento de la aplicación.
- Servidor de bases de datos para el acceso a la información necesaria.
- Permisologías y librerías necesarias para la conexión con la tecnología *Beacon*.

1.3.2. Principales funcionalidades

- Registro de usuarios.
- Registro de comercios.
- Registro de promociones.
- Manejo de listas de promociones.
- Notificaciones *push*.
- Detalle de promociones.

1.4. Arquitectura de la solución

La arquitectura de la propuesta de solución a desarrollar en la aplicación será como se muestra en la Ilustración 1.

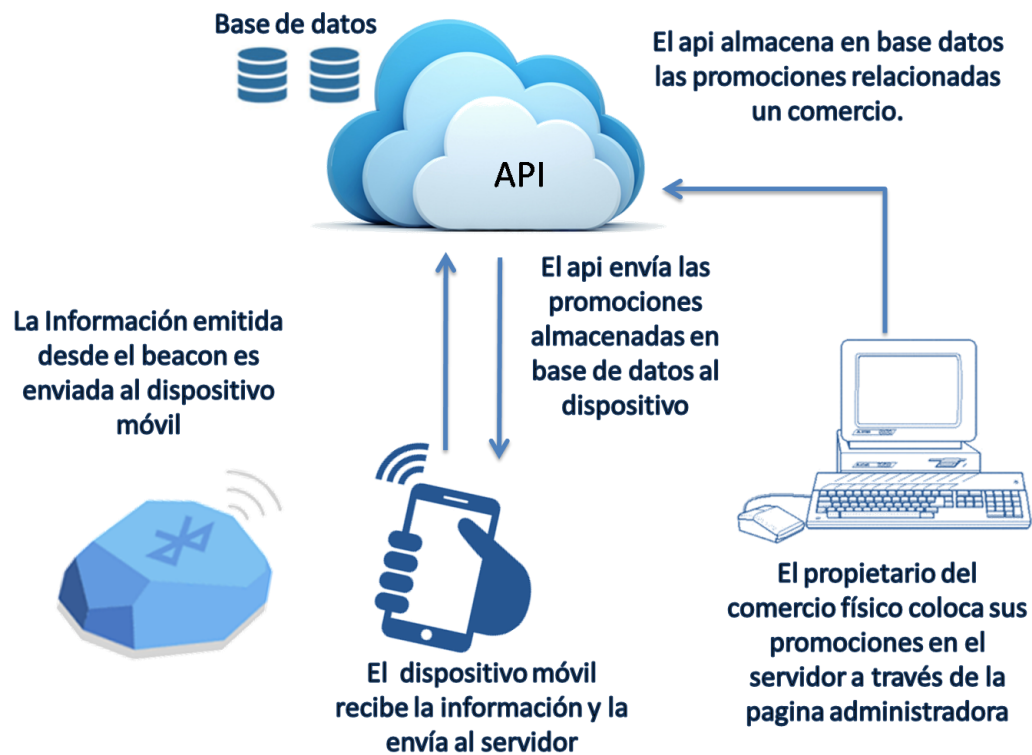


Ilustración 1 - Arquitectura de la solución.
Fuente: Los Autores

La solución estará conformada por:

- Modulo central, Ruby on Rails *framework* que servirá de API para proveer los servicios Web de la solución.
- Modulo cliente usuario, estará conformada por la aplicación móvil desarrollada en Android Studio.
- Modulo cliente administrador, estará conformada por la aplicación web desarrollada en Ruby on Rails.

Para esto se proyecta usar las siguientes herramientas y tecnologías en la implementación de la solución:

- **HTML5** para la estructura del documento generado. Es la versión más reciente y ampliamente soportada en los navegadores actuales.

- **JavaScript** para modificar dinámicamente la aplicación de acuerdo a la interacción del usuario. Esto incluye la biblioteca **jQuery** por las facilidades que ofrece en la manipulación del documento y por su capacidad de envío y recepción de información con **AJAX**. Este, a su vez, permitirá el envío y recepción de información con el servidor sin recargar la página actual.

- **CSS3** para utilizar las últimas características de estilos que incluyen la utilización de gradientes, sombras y transiciones que permiten tener un diseño más atractivo.

- **Ruby** como lenguaje de programación para la implementación de la aplicación *Web*, por su simplicidad en sintaxis, su fácil utilización y las ventajas de rapidez en el desarrollo que aporta.

- **PostgreSQL 9.3** al tener el mejor soporte de base de datos por parte del *framework* Ruby on Rails.

- **Ruby on Rails** como *framework* a través del cual se implementarán las funcionalidades de la aplicación *Web*.

- **Heroku** como servidor web especializado para servir rápidamente aplicaciones de Ruby on Rails.

- **Java para Android** como lenguaje de programación para la implementación de la aplicación móvil nativa para ese sistema operativo.

CAPÍTULO 2

MARCO CONCEPTUAL

En este capítulo se describen teorías, conceptos, procesos y plataforma tecnológica relacionados con el problema a resolver y con la solución planteada. También se indica cual es el estado del arte, es decir, si ya existen soluciones similares se indican o describen, destacando cuáles son sus debilidades y que es lo que la aplicación a desarrollar mejora.

2.1. Sistemas de información

Según Montilva (1986),

“Un sistema de información es un sistemas caracterizado porque: (1) es abierto, interactúa con su ambiente mediante el intercambio de información y se adapta a las necesidades del ambiente que lo contiene; (2) es un sistema hombre-máquina, permite el uso de máquinas (computadoras) que automatizan los procesos rutinarios de transformación, los cuales son controlados y dirigidos por el hombre; (3) su entrada está constituida por datos y su salida por información, el proceso de transformación de datos en información constituyen la función esencial del sistema de información.”

Los sistemas de información generalmente pertenecen o forman parte de un subsistema padre, en la actualidad a este tipo de subsistemas padre es denominado empresa u organización. Para estos subsistemas, la información es un recurso esencial en el proceso de toma de decisiones y en la resolución de problemas. En este contexto los sistemas de información se convierten en componentes fundamentales para el buen desempeño de una organización.

A fin de que el sistema de información pueda producir información, han de ocurrir en la organización hechos o eventos cuyas características puedan ser representadas simbólicamente mediante lo que se denomina datos. Los datos son, por lo tanto, capturados y procesados por el sistema con el objeto de producir información (Montilva,1986).

Existen distintos tipos de sistemas de información, esta clasificación viene dada principalmente por las diferentes necesidades que pueden existir dentro de una organización.

Así mismo el contexto de esta investigación entraría dentro de la clasificación de sistemas de información como un sistema de información estratégico debido a que estos persiguen la mejora constante de la relación de la organización con su entorno (clientes, proveedores, distribuidores, competidores), con la finalidad de conseguir mejoras cualitativas, intangibles, como por ejemplo, la obtención de una posición de ventaja con respecto a competidores mediante reducción de tiempos de respuesta en los cambios del mercado, aumento de la calidad los productos o la obtención de nuevos mercados (Gomez y de Abajo,1997). Y esto es precisamente lo que busca la solución propuesta.

2.2. Dispositivos móviles

Según Porto *et al* (2011), “Del latín *dispositus* (“dispuesto”), es un aparato o mecanismo que desarrolla determinadas acciones. Su nombre está vinculado a que dicho artefacto está dispuesto para cumplir con su objetivo”. En otras palabras un dispositivo no es más que una pieza o conjunto de piezas o elementos preparados para realizar una función determinada y que generalmente forman parte de un conjunto más complejo. Ahora bien, cuando se habla de “Dispositivos Móviles” se hace referencia a dispositivos que poseen ciertas características como; pequeño tamaño, algunas capacidades de procesamiento, conexión permanente o intermitente a una red, memoria limitada, diseñados para una determinada función pero que pudiesen llevar a cabo funciones más complejas. Estas son unas de las características más comunes, sin

embargo estos dispositivos pueden poseer otras características como la conexión a través de *Bluetooth*, la cual se explicará más adelante.

Esta investigación estará orientada principalmente a los teléfonos celulares, los cuales entran dentro del gran conjunto definido por dispositivos móviles. Es por ello que el término “dispositivo móvil” se empleará haciendo referencia a aquellos dispositivos cuyas funciones principales es permitir recibir y emitir comunicaciones a larga distancia a través de la conexión a una red telefónica. Además dichos dispositivos suelen ser de pequeño tamaño, no poseen hilos ni cables externos y permite la comunicación desde cualquier lugar, siempre que sea dentro del área de cobertura del servicio que lo facilita.

El rasgo característico principal de este tipo de dispositivos es que se trata de un aparato portable e inalámbrico, esto es, que la realización de llamadas no es dependiente de ningún terminal fijo y que no requiere de ningún tipo de cableado para llevar a cabo la conexión a la red telefónica. Además de ser capaz de realizar llamadas como cualquier otro teléfono convencional, un celular más o menos moderno suele incorporar un conjunto de funciones adicionales, tales como mensajería instantánea (SMS), agenda, juegos, así como también todo tipo de aplicaciones desarrolladas para movilidad, que aumentan la potencialidad de utilización de estos dispositivos (Martínez,2007).

2.3. Sistemas operativos móviles

Un sistema operativo móvil, al igual que un sistema operativo común y corriente, es un programa o conjunto de programas encargado de gestionar recursos de hardware y proveer servicios a los programas de aplicación de software con la diferencias de que estos son mucho más simples y están orientados a ciertas características como: la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de interacción entre el usuario y el dispositivo móvil. Android es el sistema operativo más usado en Venezuela. De acuerdo con

datos descritos por Emen acerca de Conatel en el diario El Mundo se construye la tabla 1.

Tabla 1 - Sistemas operativos más usados en Venezuela

Sistema Operativo	Empresa Propietaria	Última Versión Estable	Porcentaje de usuarios en Venezuela
IOS	Apple INC	9.3.3	20,99%
Android	Google INC	6.0.1	45,25%
Blackberry OS	RIM	7.1.0.2994	26,08%
Windows Phone	Microsoft	8.1.2	4,04%

Fuente: <http://www.elmundo.com.ve/noticias/tecnologia/dispositivos/android-domina-el-mercado-venezolano-de-telefonos-.aspx>

2.4. Arquitectura Cliente/Servidor

La arquitectura denominada Cliente/Servidor es usada en todas las aplicaciones que se basan en internet/intranet, un servidor es un ordenador remoto en algún lugar de la red que proporciona información según la petición a través del protocolo HTTP (*Hypertext Transfer Protocol*) véase ilustración 2. Un cliente normalmente es ejecutado en un cómputo local (Llámesese computador escritorio, laptop, dispositivo móvil, entre otros.) y se comunica con este servidor remoto pidiendo a este información para ser procesada y desplegada al usuario, el cliente es el que inicia el proceso de comunicación haciendo solicitudes o peticiones al servidor. Un servidor típicamente sirve a una multitud de clientes, es el encargado de recibir dichas solicitudes, procesarlas y enviar la respuesta al cliente, ahorrando así a cada uno de ellos el problema de tener la información instalada y almacenada localmente, desplazando de esta manera a las aplicaciones de escritorio que no poseen dicha arquitectura. Valle (2005).

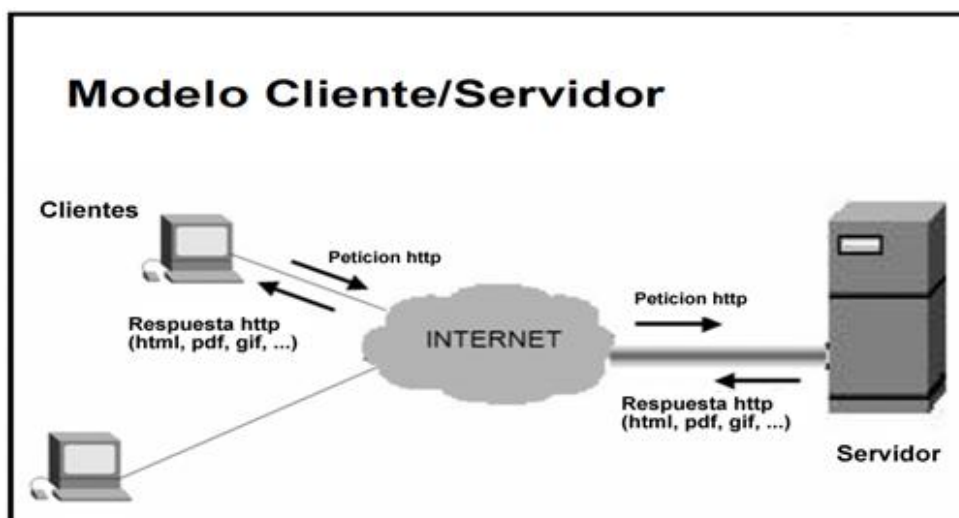


Ilustración 2 - Modelo de Cliente/Servidor

Fuente: <http://redespomactividad.weebly.com/modelo-cliente-servidor.html>

Dependiendo de la aplicación que el servidor pone a disposición de los clientes, la arquitectura Cliente/Servidor pueden ser de distintos tipos, entre ellos se tienen:

- **Servidor de Impresión:** mediante el cual los usuarios imprimen remotamente.
- **Servidor de Archivos:** con el cual los clientes comparten y/o almacenan archivos, a este servicio se le conoce como Servidor FTP.
- **Servidor de Base de Datos:** en este existen uno o varios sistemas de Base de Datos y acceden distintos tipos de clientes. Normalmente administrado por un Administrador de Base de Datos (DBA).
- **Servidor de Nombres:** este convierte las direcciones IP (Protocolo internet) en nombres y viceversa también conocido como Servidor DNS.
- **Servidor de Correo:** este permite enviar y/o recibir correos electrónicos mediante un cliente de correo electrónico.

2.5. Aplicaciones web

Una Aplicación Web incluye un contenido que se muestra del mismo modo cada vez que se solicita desde un navegador. Un ejemplo de una página Web estática es una página de servicio al cliente que contiene información de contacto como por

ejemplo: los números de teléfono, los números de fax y las direcciones de correo electrónico que no suelen cambiar con frecuencia (Castillo, 2004).

Una página Web estática se crea utilizando sólo HTML lenguaje que interpretan los navegadores Web. Una página Web estática contiene además del código HTML, texto, así como otros elementos apropiados para la página como imágenes y animación, pero no utilizan la información almacenada en Base de Datos. Según Kon (1997) el contenido de una página Web dinámica en cambio se genera cuando el usuario solicita la página. Generalmente el contenido se extrae de una Base de Datos, lo que permite presentar la información más actual sin volver a codificar la página Web. Una página Web dinámica actúa como una plantilla: contiene código para recuperar la información solicitada y dar formato a la salida.

Este tipo de aplicaciones son características de poseer la practicidad de permitir una comunicación más fluida y dinámica en la computación ya que su arquitectura es basada en Cliente/Servidor. El alto grado de desarrollo actual permite la actualización y el mantenimiento (vía Internet) de dichas aplicaciones, sin que se deba distribuir e instalar software específico o versiones individuales a un usuario cada vez. Estas están desarrolladas en un formato estándar como lo es HTML o XHTML, que son soportados por los navegadores web más comunes. También se usan lenguajes del lado del cliente como JavaScript dando así dinamismo a la interfaz de usuario.

Se puede acceder a estas aplicaciones desde cualquier tipo de plataforma, usando cualquier navegador y desde cualquier parte del mundo, así como también toda su información está distribuida en miles de ordenadores que ofrecen su espacio para almacenarla. Haciendo así que esta se encuentre centralizada dándole la comodidad al usuario de no actualizar el cliente. (García, 2010).

2.6. Aplicaciones móviles

Según Cuello y Vittone (2015)

“Las aplicaciones móviles -también llamadas apps- son programas software. Para entender un poco este concepto, se puede decir que las aplicaciones son lo que los programas son para los ordenadores de escritorio. Muchas de ellas ya se encontraban entre otros; de hecho, ya estaba incluidas en los sistemas operativos de *Nokia* o *BlackBerry* años atrás. Los móviles de esa época son llamados feature phone, en contraposición a los más actuales smartphones.”
(p. 14.)

De acuerdo a lo anterior, una aplicación móvil, es un programa de software que usa como arquitectura base Cliente/Servidor y es utilizado para facilitar al usuario la realización de diversos trabajos, ejecutada desde una plataforma móvil, pudiendo encontrarse en cualquier lugar o zona geográfica por medio de la utilización de tecnología de conexión de datos (H, 3G, EDGE, entre otros), el uso de esta señal permite establecer una conexión, consulta de información y almacenamiento de la misma a una base de datos externa.

Actualmente se conocen tres tipos principales de aplicaciones móviles, en la tabla 2 se pueden ver tanto concepto como las ventajas y desventajas de cada una de ellas.

Tabla 2 - Tipos de aplicaciones móviles

Tipos de aplicaciones	Concepto	Ventajas	Desventajas
App Nativa	Desarrollada específica para un sistema operativo.	- Acceso completo al dispositivo. - Mejor experiencia de usuario. - Actualización de app constante	- Costosas. - Código no reutilizable entre plataformas. - Diferentes habilidades/idiomas/herramientas entre plataformas.
App Web	Desarrollada con	- Código base	- Requiere conexión a

	lenguajes conocidos como HTML, CSS, JavaScript. Programación independiente de sistemas operativos.	reutilizable. - Proceso de desarrollo sencillo y económico. - Reutilización de sitios responsivos. - Fácil versionamiento.	internet. - Acceso muy limitado a hardware del dispositivo. - Tiempo de respuesta. - Requiere mayor esfuerzo en promoción y visibilidad.
App Híbrida	Combinación de App Nativa y Web, esta recoge lo mejor de ellas. Desarrolladas en HTML, CSS, Java y JavaScript pero provee la facilidad de acceder a hardware del dispositivo.	- Multiplataforma. - Instalación nativa pero construida en HTML, JavaScript y CSS. - Acceso a parte del hardware del dispositivo.	- Experiencia del usuario más propia de la aplicación web que de la app nativa. - Diseño visual no siempre relacionado con el sistema operativo en el que se muestre.

Fuente: Los Autores

2.7. Tecnologías de desarrollo

En la siguiente sección se definen las tecnologías de desarrollo que fueron investigadas y usadas en la construcción de la solución que implementa dos prototipos funcionales uno móvil y uno web.

2.7.1. Tecnología Bluetooth Low Energy (BLE)

Según *Bluetooth* (2016), *Bluetooth Low Energy* (BLE) también llamado *Bluetooth* Versión Inteligente 4.0+ es la versión mejor desarrollada y amigable de *Bluetooth* que fue construida y comercializada por *Bluetooth Special Interest Group, Inc* para la internet de los objetos (“Internet of Things” IoT) esta tecnología de red de área personal inalámbrica es usada para transmitir datos bajo cortas distancias, la

potencia-eficiencia de BLE lo hace perfecto para dispositivos que están en constante uso durante largos periodos con fuentes de energías como pilas de botón o dispositivos de capacitación de energía.

La parte inteligente de esta nueva versión es el soporte nativo que esto posee para la tecnología *Bluetooth* en los principales sistemas operativos, desarrollo de aplicaciones móviles más fácil, conectividad para computación en la nube y economía social. Esto permite la mejora de dispositivos como relojes, sitios públicos o juguetes con tecnología inalámbrica *Bluetooth*. También proporciona la capacidad para los desarrolladores incorporar nuevas funcionalidades en las aplicaciones móviles tales como deporte y fitness, cuidado de salud y lo más importante de todo interfaz humana (HID).

Características principales

Dentro de las características de esta tecnología, *Bluetooth* (2016) menciona las siguientes:

- *Ultra-LowPeak*, consumo medio de energía en modo de reposo que le da capacidad de ejecutarse por tiempos prolongados en baterías tipo botón.
- Tamaño reducido y arquitectura de desarrollo de aplicaciones estandarizada que conduce a un bajo desarrollo y costos de operación.
- Protocolo estándar inalámbrico que permite la interoperabilidad de múltiples proveedores.
- Integra una parte de la más estricta seguridad en la industria con el cifrado de datos AES de 128 bits.
- Rango de alcance similar al de *Bluetooth* en su versión 4.0.

Características generales

- **Compatibilidad:** *Bluetooth Smart* 4.0+ no es compatible con versiones anteriores de *Bluetooth*, llamada a menudo “clásico”. La especificación de *Bluetooth* 4.0 permite a dispositivos implementar tanto sistemas clásicos como a la vez *Bluetooth* de energía baja (BLE). *Bluetooth Smart* 4.0+ use la misma radio frecuencia de 2.4 GHz, lo que permite a los dispositivos de modo dual compartir una única antena de radio. LE, sin embargo utiliza un sistema de modulación más simple. (Bluetooth SIG, 2011).

- **Seguridad:** Para asegurarse que la comunicación de *Bluetooth* con característica baja energía (Smart, BLE, LE) es siempre segura y protegida, la especificación principal de *Bluetooth* proporciona varias características para cubrir el cifrado, la confianza la integridad de los datos y la privacidad de los datos del usuario. (Bluetooth SIG, 2011). Estas se nombran a continuación:

- Emparejamiento.
- Generación de claves.
- Cifrado.
- Data firmada.
- Privacidad.

2.7.2. Tecnología Beacon

Los *Beacons* son pequeños dispositivos que trabajan mediante tecnología *Bluetooth* (específicamente a través de *Bluetooth Low Energy*), éstos permiten determinar la distancia y/o ubicación con respecto a otros dispositivos que utilicen la misma tecnología, como por ejemplo teléfonos inteligentes. Cada *Beacon* tiene un alcance entre cuarenta y setenta metros dentro de los cuales se pueden definir

múltiples rangos para que la interacción con los otros dispositivos sea diferente según la cercanía. (Premper, 2015).

Los dispositivos *Beacon* están basados en la tecnología *Bluetooth Low Energy* (BLE) como se mencionó anteriormente, creada con la intención de hacer frente a otras tecnologías como NFC y GPS, sin embargo existen diversas diferencias que en algunos casos se convertirán en ventajas para BLE y en otros casos serán desventajas tal y como lo muestra la tabla 3.

Tabla 3 - Diferencias entre GPS, NFC, Beacon

	GPS	NFC	<i>Beacon</i>
Consumo de energía.	Alto	Medio	Bajo
Rango de alcance	Ilimitado	~20cm	~100m
Precisión	Bajo (Solo al aire libre)	Muy alto	Alto
Costo	Gratis	Alto	30\$ por <i>beacon</i>

Fuente: Los Autores.

Aplicaciones y usos

Es posible agrupar múltiples *beacons* dentro de un determinado lugar y que éstos se encuentren relacionados entre sí, lo cual es posible debido a los componentes que integran la tecnología *beacon*. Esto abre una gran cantidad de posibilidades al momento de implementar esta tecnología. Al establecer una buena distribución de los *beacons* dentro de un espacio (como un comercio por ejemplo), se puede dar a cada *beacon* una función específica, por ejemplo: dar la bienvenida a los clientes (a su vez informar al gerente de la tienda la visita de un cliente VIP), otros para información de productos según el pasillo que se visita, ofertas, sugerencias basadas en gustos del cliente y así muchos ejemplos más como los que comenta Gonzalo (2015), a continuación:

- **Ubicación en interiores:** Debido a que se conoce la ubicación de cada dispositivo *beacon*, es posible detectar dispositivos que se encuentren cerca del mismo. En los museos por ejemplo puede ser una utilidad de interés, ya que, según la

ubicación de los visitantes con respecto a los dispositivos, se podrían ir reproduciendo guías de audio que orienten al visitante sobre el lugar donde se encuentren.

- **Marketing de proximidad:** Se podrían enviar notificaciones a los teléfonos inteligentes de los usuarios con descuentos, novedades, ofertas o invitaciones de cualquier al momento de que se acerquen a algún comercio en particular.

- **Check-in automático:** Se podrá registrar automáticamente que un cliente está en una tienda o local concreto, pudiendo ser premiado por ello con algún tipo de promoción o con la notificación de que hay un amigo cerca, por ejemplo.

- **Pagos sin contacto o Contactless Payments:** Se podría evitar que las personas tengan que utilizar su billetera o las tarjetas de crédito para realizar algún pago. Con la tecnología *Beacon* a través de empresas como *PayPal*, los pagos se podrían hacer automáticamente al estar cerca de la línea de cajas. Esta idea ofrece es que los comerciantes instalen un *PayPal Beacon* en su tienda, conectado a cualquier enchufe. Al mismo tiempo, los clientes deben llevar la aplicación de *PayPal* en su teléfono inteligente. Cuando el dispositivo detecta la señal del móvil, se conecta con la aplicación, estableciéndose una conexión de pago seguro sin necesidad de que el usuario intervenga.

Componentes

Apple ha estandarizado el formato para anuncios tipo BLE. Bajo un formato, y transferencia de paquetes consistentes en 4 piezas principales de la información. *IBeacon insider* (2015), describe los siguientes componentes a través de un ejemplo con la compañía XXX inc.:

- **UUID:** *String* de 16 *byte* usado para diferenciar un largo grupo de *Beacons* relacionados. Por ejemplo, si la compañía XXX, inc. mantiene una red con *beacons* en una cadena de negocios de comida, todos los *beacons* de esta compañía deberían compartir la misma UUID. Esto permite a la aplicación de la empresa XXX, inc. saber que los anuncios provienen de los *Beacons* de esta organización.

- **Major:** *String* de 2 *byte* usado para distinguir un pequeña subconjunto de *beacons* en un grupo grande de ellos. Por ejemplo, si empresa XXX, inc. tiene 4

beacons en una tienda de chucherías en particular, todos los 4 *beacons* deberían tener el mismo *Major*. Esto permite a la compañía conocer exactamente a qué tienda entran los consumidores.

- **Minor:** *String* de 2 *byte* que significa la identificación individual del *beacon*. Siguiendo con el ejemplo anterior, un *beacon* que está en el frente de la tienda debería tener su propio *Minor*. Este permite que la aplicación de la empresa XXX, inc. saber exactamente donde los compradores están dentro de la tienda.

- **Tx Power:** Este es usado para determinar proximidad (distancia) desde el *beacon*. ¿Cómo trabaja esto? *Tx power* es definido como la intensidad de la señal a exactamente un metro desde el dispositivo. Esto debe ser calibrado y programado anteriormente. Dispositivos pueden utilizar esto como una línea de base para dar una estimación de la distancia aproximada.

Ejemplo.

Un *Beacon* que permite el siguiente paquete.

UUID: 12345678910245.

Major: 22.

Minor: 2.

Un dispositivo recibe este paquete debería entender que este proviene de un *Beacon* de la empresa XXX, inc. (UUID) como portada, prosiguiéndole el *Major* y luego el *Minor*.

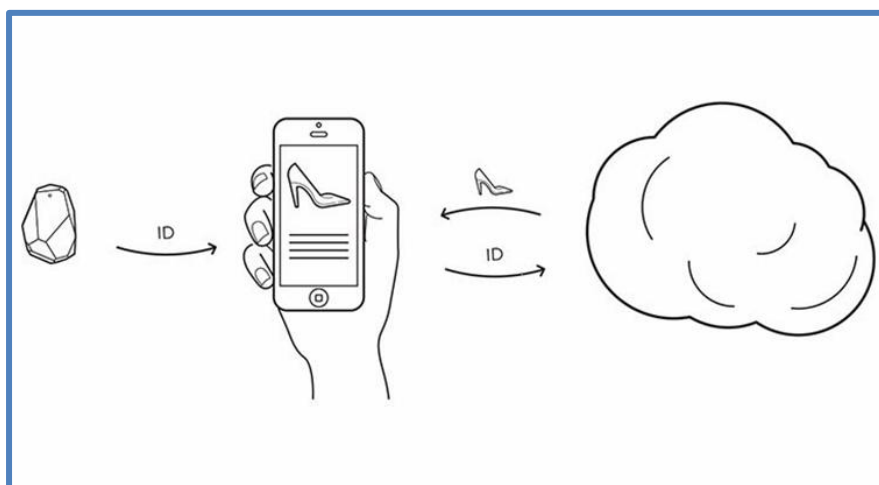


Ilustración 3- Como funcionan los Beacon

Fuente: estimote.com

2.7.3. Interfaz de programación de aplicación (API)

Según Merino (2014),

“Una API (siglas de ‘Application Programming Interface’) es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software. Las API son valiosas, ante todo, porque permiten hacer uso de funciones ya existentes en otro software (o de la infraestructura ya existente en otras plataformas) para no estar reinventando la rueda constantemente, reutilizando así código que se sabe que está probado y que funciona correctamente. En el caso de herramientas propietarias (es decir, que no sean de código abierto), son un modo de hacer saber a los programadores de otras aplicaciones cómo incorporar una funcionalidad concreta sin por ello tener que proporcionar información acerca de cómo se realiza internamente el proceso.”

Según lo anterior, un API servirá para definir ciertas rutinas que a su vez permitirán mantener la persistencia de los datos, tanto para la aplicación web como para la aplicación móvil, permitiendo además, crear y consultar registros en una base de datos a través de invocaciones de servicios REST, lo cual se explicara a continuación.

REST (*Representational State Transfer*)

Es un tipo de arquitectura de desarrollo que se apoya totalmente en el estándar HTTP. A diferencia de otras alternativas que se han implementado en el pasado como SOAP, este enfoque se muestra más simple y más estándar debido a que permite crear servicios y aplicaciones que puedan ser usadas por cualquier dispositivo que entienda HTTP. Márquez (2013). Este tipo de arquitectura define un conjunto de reglas que deben seguirse estrictamente para conseguir su aplicación de forma correcta, cuando esto ocurre se conoce como el nombre de arquitectura de desarrollo *Restful*.

2.7.4. Lenguajes de programación

Front-End

El Front-end es considerado el componente de desarrollo de software que interactúa directamente con el usuario en la capa de presentación y es el encargado de presentar los datos al usuario, este es desarrollado generalmente con los siguientes lenguajes:

- **Lenguaje de Marcado Extensible (XML):** Quin (2015) comenta que el Lenguaje de Marcado Extensible es un simple y muy flexible formato de texto derivado de SGML (ISO 8879). XML juega cada vez más un papel muy importante en el intercambio amplio de data en la Web y en otros medios. XML se convirtió en un estándar el 10 de febrero de 1998 y fue diseñado principalmente para ser auto-descriptivo, guardar y transportar data; el mismo es recomendación de la Word Wide Web Consortium (W3C). En la ilustración 4 se verá un ejemplo de una estructura básica en XML para el almacenamiento de registros en una base de datos en algún sistema.

```
1 <?xml version=" 1.0 " encoding=" UTF-8 " standalone=" yes ">
2 <ficha>
3 <nombre> Franklin </nombre>
4 <apellido> Sandoval </apellido>
5 <direccion> Facultad ciencias, UCV </direccion>
6 </ficha>
7 |
```

Ilustración 4 - Registro XML

Fuente: Los autores

- **Lenguaje de marcado de hipertexto (HTML):** Álvarez (2001) señala que el HTML (*Hypertext Markup Language* por sus siglas en inglés) es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web. El HTML es un lenguaje de marcación de elementos para la creación de documentos hipertexto, muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida, pueda enfrentarse a la tarea de crear una web. HTML es fácil y pronto fácilmente se logra dominar el lenguaje. Más adelante se conseguirán los resultados profesionales gracias a nuestras capacidades para el diseño y nuestra vena artista, así como a la incorporación de otros lenguajes para definir el formato con el que se tienen que presentar las webs, como CSS Álvarez (2001).

Se puede decir que HTML es un lenguaje de marcado fácil de aprender el cual es la base para la realización de casi cualquier desarrollo *online*. Por ser un lenguaje básico normalmente solo se utiliza para darle la estructura inicial a diseño, pero no se pueden crear diseños o estructuras profesionales por así decirlo, y es por ello que se fueron creando nuevas tecnologías que permiten un diseño profesional y que permiten realizar numerosas acciones dinámicas, interactividad y estilo a la página web o sistema que se está realizando, como lo son el lenguaje CSS y JavaScript. A partir del 2009 se comienza a utilizar la expansión de HTML4, por la nueva versión HTML5, el cual tiene muchas mejoras en cuanto a el diseño de la página, también es de una gran ayuda ya que te permite realizar animaciones, con la cual ya no se dependerá de la

herramienta *FLASH* para realizarla, algo que gusto mucho entre los desarrolladores web.

Según Franganillo (2011) a diferencia de HTML, HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas `<div>` y `<span>`, pero tienen un significado semántico, como por ejemplo `<nav>` (bloque de navegación del sitio web) y `<footer>`. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos `<audio>` y `<video>`. Mejora el elemento `<canvas>`, capaz de renderizar elementos 3D en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE). (p.262).

También se puede resaltar que HTML5 está desarrollado con la intención de reflejar mejor la nueva anatomía de los sitios web. Es por eso que hay algunos elementos nuevos, como `<header>` y `<footer>`, que están diseñados específicamente para marcar estas partes de la web. Con este desarrollo en el lenguaje ya no es necesario identificar a estos dos elementos con la etiqueta `<div>` y posee nuevos elementos de `<section>` y `<article>` los cuales permitirá a los desarrolladores marcar estas áreas de la página como tal.

- **Hoja de estilos de cascada (CCS):** W3C.CSS (2010) señala que CSS (siglas en inglés de *cascading style sheets*), es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille.

Diseñar con hojas de estilos es muy sencillo. Ya que solo se necesitan conocimientos básicos de HTML y terminología de autoedición. Por ejemplo, para asignar el color de un texto de un elemento 'H1' en azul. Véase Ilustración 5.

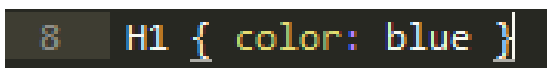
A screenshot of a code editor showing a CSS rule. The text '8 H1 { color: blue }' is displayed in a monospaced font with syntax highlighting. The '8' is in a light blue box, 'H1' is in a light green box, '{' is in a light blue box, 'color:' is in a light green box, 'blue' is in a light blue box, and '}' is in a light blue box.

Ilustración 5 - Asignación de color a elemento DOM
Fuente: Los autores

Este ejemplo es acerca de una simple regla de CSS. Una regla consiste en dos partes principales: selector ('H1') y la declaración ('color: blue'). Aunque esto se considera como una única propiedad ya se considera una hoja de estilo. El selector es el *link* entre el documento HTML y la hoja de estilo, todos los tipos de elementos de HTML son posibles selectores. (W3C.CSS, 2008). Se puede decir que las hojas de estilo en cascada ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web, por tanto tiene una gran importancia a la hora de realizar cualquier tipo de aplicación web. Uno de los grandes avances de esta tecnología es su actualización hacia CSS3, que con respecto al CSS es que el CSS3 con ayuda del HTML5 ya antes mencionado se puede realizar animación, dejando atrás a la tecnología *FLASH* que resultaba muy engorrosa para los desarrolladores y los limitaba ya que no era compatible con dispositivos móviles e IPAD. Otro avance importante con el CSS3 es la facilidad para poder realizar efecto en las capas, de esta manera facilita mucho el trabajo de los desarrolladores al incluir transparencias, bordes, texto, entre otros.

El objetivo inicial de CSS3 según Álvarez (2008) es, separar el contenido de la forma, esto se cumplió ya con las primeras especificaciones del lenguaje. Sin embargo, el objetivo de ofrecer un control total a los diseñadores sobre los elementos de la página ha sido más difícil de cubrir. Las especificaciones anteriores del lenguaje tenían muchas utilidades para aplicar estilos a las webs, pero los desarrolladores aún continúan usando trucos diversos para conseguir efectos tan comunes o tan deseados como los bordes redondeados o el sombreado de elementos en la página.

- JavaScript y JQuery

JavaScript: Según Duckett (2014), JavaScript es un lenguaje de programación utilizado comúnmente para el desarrollo Web, cuya sintaxis es similar a la del lenguaje de programación C a pesar de estar influenciado por el lenguaje Java. Es un lenguaje utilizado del lado del cliente (en la capa de presentación) para agregar elementos dinámicos e interactivos a las páginas Web. Un ejemplo de una función de JavaScript que calcula la suma de dos números puede verse a continuación:

```
1  function sum(a,b)
2  {
3  return a + b;
4  }
5  var total = sum(7,11);
6  alert(total);
7  |
```

Ilustración 6 - Función Suma

Fuente: Los autores

JQuery: Librería de JavaScript muy rica en características, rápido y pequeño. Este hace cosas como recorrido de DOM, manipulación de HTML, manejo de eventos, animaciones y Ajax mucho más simple y fácil de usar a través de múltiples navegadores. Con una combinación de versatilidad y extensibilidad, JQuery ha cambiado la manera de que millones de personas escriban JavaScript. Según Duckett (2014), la sintaxis básica de JQuery y el significado de cada una de sus partes es como se describe en la ilustración 7.

```
10 $(selector).action()
```

Ilustración 7 - Ejemplo básico JQuery

Fuente: Los Autores

- El símbolo \$ se utiliza para definir o acceder el JQuery.
- El (selector) se utiliza para ubicar el elemento HTML a utilizar.

- El `action()` se refiere a la acción que será ejecutada sobre el elemento.

Un ejemplo del uso de JQuery:

`$(this).hide()` – Se utiliza para ocultar el elemento actual.

`$("p").hide()` – Se utiliza para ocultar todos los elementos `<p>`

- **Java:** Arnold y Gosling (2001) definen que Java es un lenguaje de programación de propósito general, marca registrada, y como tal es válido para realizar todo tipo de aplicaciones profesionales. Una de las características más importantes es que los programas ejecutables creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos. (p. 12). Como características principales se tiene:

- Orientado a objeto.
- Aprovecha características de la mayoría de los lenguajes modernos evitando sus inconvenientes. En particular C++.
- Manejo de memoria gestionada por el propio lenguaje y no el programador.
- Incorpora multihilos (multi-threading) para permitir la ejecución de tareas concurrentes dentro de un mismo programa.

Salinas, (2004) comenta, los programas se hacen con clases ya que ellas son el núcleo de la programación orientada a objeto (POO). Los campos son datos de la clase y componen algún estado del objeto. Los métodos son funciones que operan sobre los campos para manipular el estado de un objeto. Se ve cómo sería un “Hola Mundo” en Java: (p. 2).

```

1  class HolaMundo {
2      public static void main(String[] args) {
3          System.out.println("Hola Mundo");
4      }
5  }
```

Ilustración 8 - Hola mundo Java
Fuente: Los Autores

Back End

Dentro del desarrollo de *software*, el *Backend* forma parte de un subsistema que permite crear abstracción al momento de consultar y modificar datos relacionados al sistema que lo contiene. Es fundamental a la hora de desarrollar el *Backend* de cualquier *software*, conocer los diferentes tipos de lenguajes de programación que existen, para seleccionar el que mejor se adapte a los requerimientos del sistema en cuestión. A continuación se mencionan dos de los lenguajes de programación más utilizados al momento de construir el *Backend* de un *software*.

- **PHP** (acrónimo recursivo de PHP: *Hypertext Preprocessor*)

Según PHP (2016)

“Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.”

Según lo anterior se puede decir que lo que se escribe en código PHP es interpretado para luego mostrarse como código HTML. Véase Ilustración 9.

```
2  <!DOCTYPE HTML>
3  <html>
4      <head>
5          <title>Ejemplo</title>
6      </head>
7      <body>
8          <?php
9              echo "¡Hola, soy un script de PHP!";
10         ?>
11      </body>
12 </html>
```

Ilustración 9 - Ejemplo Hola Mundo en PHP
Fuente: Los Autores

- RUBY

Según Ruby, (2016)

“Un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla.”

Desde su liberación pública en 1995, Ruby ha atraído devotos desarrolladores de todo el mundo. En el 2006, Ruby alcanzó reconocimiento masivo, formándose grupos de usuarios activos en las ciudades más importantes del mundo y llenando las capacidades de las conferencias relacionadas a Ruby. Según el índice TIOBE (2016), que mide el crecimiento de los lenguajes de programación, ubica a Ruby en la posición #11 del *ranking* mundial en Julio del año 2016 mientras que en Julio del año 2015 se encontraba en el puesto #15. Ruby es considerado un lenguaje flexible, ya que permite a sus usuarios alterarlo libremente. Las partes esenciales de Ruby pueden ser quitadas o redefinidas. Se puede agregar funcionalidad a partes ya existentes. Ruby intenta no restringir al desarrollador. Por ejemplo, la suma se realiza con el operador suma (+). Pero si prefieres usar la palabra sumar, puedes agregar un método llamado sumar a la clase *Numeric* que viene incorporada. Véase Ilustración 10.

```
3  class Numeric
4      def sumar(x)
5          self.+(x)
6      end
7  end
8
9  y = 5.sumar 6
10 # ahora y vale 11
```

Ilustración 10 - Ejemplo método sumar Ruby
Fuente: Los Autores

2.7.5. Arquitectura MVC (Modelo Vista Controlador)

Murazzo et al. (2010) definen a la arquitectura MVC como un patrón de arquitectura de aplicación que se separa de los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Dicha arquitectura fue introducida como parte de la versión Smalltalk-80 (*Smalltalk* es un lenguaje de programación). Su objetivo era reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples. Se tiene como principal característica, que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; con esto se logra que cualquier cambio que se produzca en el Modelo se refleje automáticamente en cada una de las Vistas. Ver Ilustración 11.

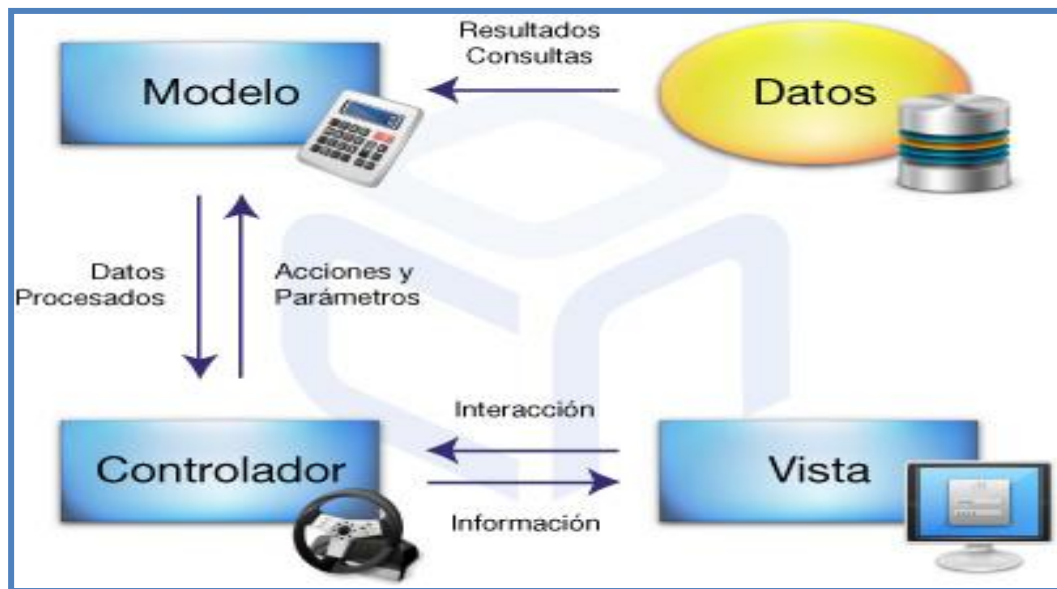


Ilustración 11 - Arquitectura Modelo Vista Controlador

Fuente: (Murazzo, 2010)

El componente Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. Este componente no sabe que hay internamente en los otros componentes, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el

Modelo. Referente al componente Vista es el objeto que maneja la presentación visual de los datos representados por el componente Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Como tercer componente se tiene el Controlador que es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando hay cambios se activa, bien sea por cambios en la información del Modelo o por alteraciones de la Vista.

2.7.6. Frameworks MVC

Frameworks puede definirlo en su traducción al castellano como ventana de trabajo o marco de trabajo, Gutiérrez (2010) comenta que es una abstracción en la que cierto código común provee una funcionalidad genérica que puede ser sobrescrita o especializada de forma selectiva por medio de código con funcionalidad específica. Esto facilita el desarrollo de software permitiendo a los programadores centrarse en requerimientos de software y evitar detalles de bajo nivel. En la ilustración 12 se puede ver el diseño arquitectónico de una solución donde se puede situar el *framework*.

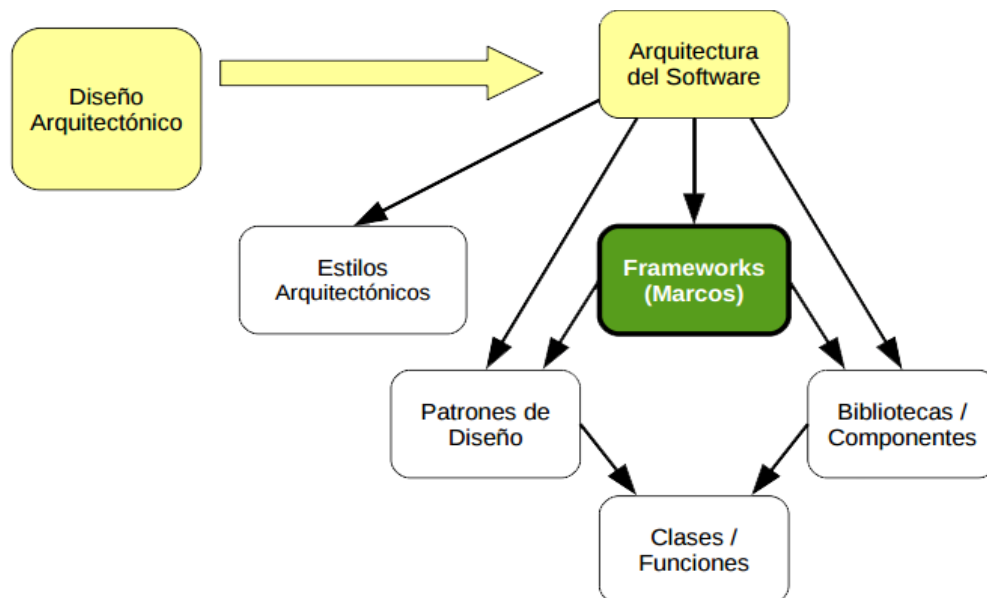


Ilustración 12 - Diseño Arquitectónico Framework
Fuente: Gutiérrez (2010)

Es, en definitiva, un modelo de trabajo que facilita la creación de aplicaciones complejas. Se les conoce también con el nombre de plantillas. A continuación se mencionan algunos de los *frameworks* utilizados para el desarrollo de aplicaciones.

- **Ruby on Rails:** Se trata de un marco de trabajo muy exitoso por la facilidad que tiene de programar y sus buenos resultados visuales. Se puede ejecutar en casi cualquier servidor web, basta con instalar el componente correspondiente.

- **Apache Struts:** El marco de trabajo más famoso para la creación de aplicaciones J2EE. Muy preparado para utilizar con Apache.

- **Spring:** Otro marco para trabajar en Java J2EE que tiene bastante éxito. Tiene incluso una versión para las aplicaciones .NET

- **Django:** Escrita en Python y pensada para utilizar en ese lenguaje que facilita la creación de aplicaciones web.

- **Laravel:** *Framework* PHP con influencia de otros *frameworks* como *RubyonRails*, Sinatra y ASP.NET.

- **Android Studio:** *Framework* Java, IDE oficial de Android, su propósito es acelerar el proceso para el desarrollo y construcción de aplicaciones de máxima calidad para todos los dispositivos Android.

2.7.7. Base de Datos

Una Base de Datos (BD) es un repositorio centralizado de datos que permite almacenar y organizar hechos o eventos y restituirlos a demanda del usuario para producir información. (Date, 2001). Se puede decir entonces que una BD es un repositorio de datos, que se relacionan entre sí, cuyos datos corresponden a un hecho o evento y que gracias a que existen métodos de extracción para ellos, se puede producir información.

Date (2001) caracteriza a las base de datos de la siguiente manera:

- Integridad de los datos.
- Acceso concurrente a los datos.

- Facilidad para el cambio de hardware o software.
- Independencia de los datos.
- Control centralizado de los datos.
- Costo mínimo de almacenamiento y mantenimiento.
- Versatilidad para la representación de relaciones.
- Establecimiento de medidas de seguridad.

Ventajas y desventajas

Date (2001) explica en su libro de Introducción a los sistemas de bases de datos señala diferentes ventajas y desventajas que se despliegan en la tabla 4 a continuación:

Tabla 4 - Ventajas y desventajas de bases de datos

Ventajas	Desventajas
Los datos pueden compartirse, haciendo posible la integración con distintas aplicaciones.	Vulnerabilidad si no se controla adecuadamente.
Redundancia controlada, debido al sistema tradicional de archivos independientes.	Falta de integridad si no se controla la BD.
Se logra consistencia de los datos, gracias al control de la redundancia.	Pérdida de tiempo al recuperar una BD.
Manejo de transacciones, es una unidad de trabajo lógica.	Dependiendo de la cantidad de datos se ve afectado el rendimiento.
Integridad, validación de condiciones al introducir datos	Complejidad en relaciones entre los datos.
Seguridad, el ABD al tener control central de los datos.	
Cumplimiento de estándares, se pueden estandarizar procesos, formas, nombres de datos.	

Fuente: Los Autores

2.7.8. Sistema de Base de Datos

Un Sistema de Base de Datos (SBD) es un sistema diseñado para manejar grandes cantidades de datos y producir información. Un SBD es básicamente un sistema computarizado cuyo propósito general es mantener información y hacer que esté disponible cuando se solicite. La información puede ser cualquier cosa que se considere importante para el individuo o la organización a la cual debe servir el sistema. (Date, 2001).

Sistema Manejador de Base de Datos (SMBD)

Un Sistema Manejador de Base de Datos (SMBD) es colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. (Silberschatz, Korth y Sudarshan, 2002). Un SMBD se apoya en la tecnología para cumplir con el propósito de permitir, a las personas, la manipulación (consultas, actualizaciones, eliminaciones e inserciones) de su información. Se puede decir que un SMBD es un sistema que se encarga de almacenar información y que las personas pueden manipularlas mediante operaciones sobre ellos.

Funciones de SMBD

Date (2001) establece las siguientes funciones:

- Definición de los datos: Definición de datos: debe ser capaz de aceptar definiciones de datos en versión fuente y convertirlas en la versión objeto apropiado. Para hacer esta conversión el SMBD debe incluir un procesador DDL (Lenguajes de definición de datos) como uno de sus componentes.
- Manipulación de datos: debe estar capacitado para atender las solicitudes del usuario para extraer, actualizar, o ingresar nuevos datos. Para atender estas solicitudes debe tener un procesador DML como otro de sus componentes.
- Optimización y ejecución: las peticiones del DML, deben ser procesadas por un componente optimizador, cuya finalidad es determinar una forma eficiente de implementar una petición.

- Seguridad e integridad de los datos: debe controlar las solicitudes de los usuarios para rechazar aquellas solicitudes no permitidas para que los datos estén seguros e íntegros.
- Recuperación y concurrencia de los datos: el SGBD debe tener un componente que sea un monitor de procesamiento de transacciones, que debe cuidar del cumplimiento de ciertos controles de recuperación y concurrencia.
- Diccionario de Datos: debe incluir una función de diccionario de datos, en este caso existen los metadatos que son datos acerca de los datos.
- Rendimiento: debe ejecutar todas las funciones recién identificadas en la forma más eficiente posible.
- Existen diferentes manejadores de base de datos que se encuentran en el mercado y dentro de los cuales existen algunos que se inclinan por la filosofía de software libre. A continuación se describen dos de ellos:

PostgreSQL: De acuerdo con PostgreSQL. (2010) es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente y robusto del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Como características principales de este SGBD se tienen:

- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, entre otros).
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos se tienen disponibles, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, entre otros.

- Copias de seguridad en caliente (*Online/hotbackups*)
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

MYSQL: De acuerdo con MySQL (2013), es el Sistema Manejador de Base de Datos más popular del mundo, es desarrollado por MySQL AB. MySQL AB es una compañía comercial que construye su negocio brindando servicios para la base de datos MySQL. Esta es un sistema de gestión de bases de datos relacionales. Es decir que almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. SQL se refiere a "*Structured Query Language*". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. MySQL es muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SMBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento, precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo. (Gilbert, Pérez y Castilla, 2008. p. 5). También mencionan las siguientes características:

- Interioridades, portabilidad y multiplataforma
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de *multi-threaded* mediante *threads* del kernel.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Relativamente sencillo de añadir otro sistema de almacenamiento.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor.

En la tabla 5 se muestra una comparativa entre los manejadores de base de datos MySQL y PostgreSQL.

Tabla 5 - Comparación entre MySQL y PostgreSQL

Característica	MySQL	PostgreSQL
Definición de Esquema	Relacional	Objeto / Relacional
SQL para Consultas	SQL92	SQL92 y SQL99
Herramientas de Administración Grafica	PhpMyAdmin, perlDBDmysql y herramientas que soporten java o jdbc	PgAccess y herramientas que soporten java o jdbc
Tipos de Datos Soportados	En la versión 4 soporta aproximadamente 27 tipos de datos entre carácter, numéricos y fecha.	En la versión 7.4 soporta 62 tipos de datos entre carácter, numéricos, fecha y otros de tipo administrativo como aclitem, oid, xid.
Procedimientos almacenados	No tiene soporte de procedimientos almacenados	Disparadores y Funciones
Elementos adicionales (Vistas, Sinónimos, Secuencias, entre otros.)	No soporta Vistas, no soporta sinónimos, pero si soporta algunas secuencias.	soporta vistas, dominios, tipos de objetos nuevos, secuencias, reglas, no soporta sinónimos
Integridad de Datos	No tiene soporte de restricciones	Buen manejo de restricciones
Joins	No tiene soporte	Soporta, natural join, innerjoin, crossjoin sin ningún inconveniente, además permite manejar subconsultas con unión, intersección y excepciones.
Recursos	Bajo consumo de recursos como memoria RAM y hardware	el consumo de memoria aumenta aproximadamente en 25% más que MySQL
PHP	buena integración con php	Buena integración con php, en las últimas versiones ha aumentado el soporte de funciones.
Rendimiento	alta velocidad	Buena velocidad pero menor que la ofrecida por mysql

Fuente: <http://docplayer.es/469041-Bases-de-datos-con-software-libre.html>

Una vez descrita la tecnología se explicaran los trabajos previos que sirven de antecedentes a esta investigación.

2.8. Antecedentes de desarrollo

En términos generales, es en Estados Unidos y en Europa donde la evolución de las aplicaciones, y en consecuencia, la consolidación y madurez de mercados electrónicos, se encuentra en una fase de mayor desarrollo. Según el estudio realizado se determina que actualmente en Venezuela no existen aplicaciones que se basen en esta tecnología, y esto hace que nuestra investigación se vuelva un emprendimiento innovador tanto para nuestra casa de estudio (Universidad Central de Venezuela), como para nuestro país. Tomando en cuenta se puede encontrar una gran cantidad de investigaciones e implementaciones que soportan la tecnología base usada en esta investigación, *Beacon*, a través de *BluetoothLowEnergy*. En la siguiente sección se habla de aplicaciones desarrolladas para dos ramas de mercado diferentes y entre ellas se tienen:

Museos y arte:

- ASK app: Los visitantes del museo quieren una conversación: aplicaciones que traen el diseño UX en una exposición. Los museos están cambiando. Muchos de ellos, con instalaciones interactivas, aplicaciones de compañía, y los contenidos digitales, han pasado por la transformación completa en los últimos años. El Museo de Brooklyn de Nueva York está a la vanguardia de este proceso. Su último proyecto, la aplicación ASK, utiliza *Beacon's* para que los visitantes preguntan al personal acerca de exposiciones cercanas. (Estimote, ASK app, 2016).

Puedes acceder a su descarga en google play:

<https://play.google.com/store/apps/details?id=ask.brooklynmuseum.org>

Compras y marketing:

- Frisco.pl App: De compras en el camino con Frisco.pl, esta empresa ha colgado carteles en las paredes de las estaciones del metro de Varsovia – Polonia, cada cartel es la tienda temática de marca (que ofrece productos seleccionador de marcas específicas) que está equipado con un *Beacon*. Dándoles así a los pasajeros

del metro la comodidad de hacer compras mientras esperan su tren interactuando directamente con el cartel. Los compradores reciben un 10% de descuento si compran a través de la aplicación y el *Beacon*. (Estimote, Frisco app, 2016)

Puedes acceder a su descarga en googleplay:

<https://play.google.com/store/apps/details?id=pl.frisco.app&hl=en>

Visto ya lo anterior se puede decir que una aplicación web es un software que no necesita instalación y que se accede a través de un navegador por medio de internet o una intranet. Por otro lado una aplicación móvil es un software que necesita instalación y es accedida a través de un dispositivo móvil usando, así mismo, acceso a la internet/intranet. Ambas se basan en la arquitectura cliente/servidor, por un lado el cliente es el navegador a través del cual se visualiza la aplicación y por otra parte el servidor es el programa que está constantemente esperando solicitudes de conexión por parte del cliente y cada uno de estos entes es manejado por diversas tecnologías web y móviles como por ejemplo HTML, CSS, JavaScript, PHP, Ruby, Java, entre otros.

CAPÍTULO 3

MARCO METODOLÓGICO

Vasilis (2011) señala que una metodología de software es un proceso de software detallado y completo, y ésta puede basarse en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, espiral, entre otros). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas) actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis y/o diseño.

Por otra parte Blanco (2008) dice que una metodología de desarrollo es una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software. Es importante desatacar que existen metodologías tradicionales y ágiles las cuales se describen a continuación.

3.1. Metodologías tradicionales

Abrahamsson (2003) define a las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales. Aunque en el caso particular de RUP, por el especial énfasis que presenta en cuanto a su adaptación a las condiciones del proyecto (mediante su configuración previa a aplicarse), realizando una configuración adecuada, podría considerarse Ágil (Abrahamsson, 2003).

3.2. Metodologías Ágiles

Sommerville (2002) relata que un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento). En la tabla 6 describe algunas diferencias entre las metodologías Ágiles y las metodologías Tradicionales:

Tabla 6 - Diferencias entre metodologías ágiles y tradicionales

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Más roles.	La arquitectura del software es esencial y se expresa mediante modelos.

Fuente: Sommerville (2002)

Canós (2003) sugiere que cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación se mencionan algunas metodologías ágiles. La mayoría de ellas ya estaban siendo utilizadas con éxito en proyectos reales pero les faltaba una mayor difusión y reconocimiento:

- *Extreme Programming.*
- *SCRUM.*
- *Rational Unified Process (RUP).*
- *Crystal Methodologies.*
- *Dynamic Systems Development Method (DSDM).*
- *Adaptive Software Development (ASD).*
- *Feature -Driven Development (FDD).*
- *Lean Development (LD).*

A continuación se presentarán en detalle tres de estas metodologías para tener una idea de cómo funcionan las metodologías ágiles y se destacarán aspectos como características, ciclo de vida, roles, ventajas, desventajas, entre otros.

3.2.1. Programación Extrema (XP)

Según Beck (1999) XP (*Extreme Programming*, por sus siglas en inglés) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Ciclo de vida

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (*Release*), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. Estas fases se presentan de acuerdo con Beck (1999):

En **la fase de exploración**, los clientes escriben las historias de usuario que desean para ser incluidas en la primera versión. Cada historia describe una característica que se añade en el programa.

La fase de planificación establece el orden de prioridad para las historias y se acuerda el contenido de la primera versión. Los programadores primero estiman cuánto esfuerzo requiere cada historia y se acuerda el calendario.

La fase de iteraciones incluye varias iteraciones de los sistemas antes la primera versión. El cliente decide que historias serán seleccionarán para cada iteración. Las pruebas funcionales creadas por el cliente se ejecutan al final de cada iteración. Al final de la última iteración el sistema está listo para la producción.

La fase de producción requiere pruebas adicionales y el control del rendimiento del sistema antes de que el sistema pueda ser entregado al cliente. Después de que se produce la primera versión para uso del cliente, el proyecto XP debe mantener el sistema en funcionamiento mientras que se producen nuevas iteraciones. Con el fin de hacer esto.

La Fase de mantenimiento requiere un mayor esfuerzo también para satisfacer las tareas cliente. Por lo tanto, la velocidad de desarrollo puede desacelerarse después de que sistema está en producción.

La fase de muerte está cerca cuando el cliente ya no tiene más historias para implementar. Esto requiere que el sistema satisfaga las necesidades cliente también en otros aspectos (por ejemplo, en relación con el rendimiento y fiabilidad). En la ilustración 13 se muestra el ciclo de vida de XP.

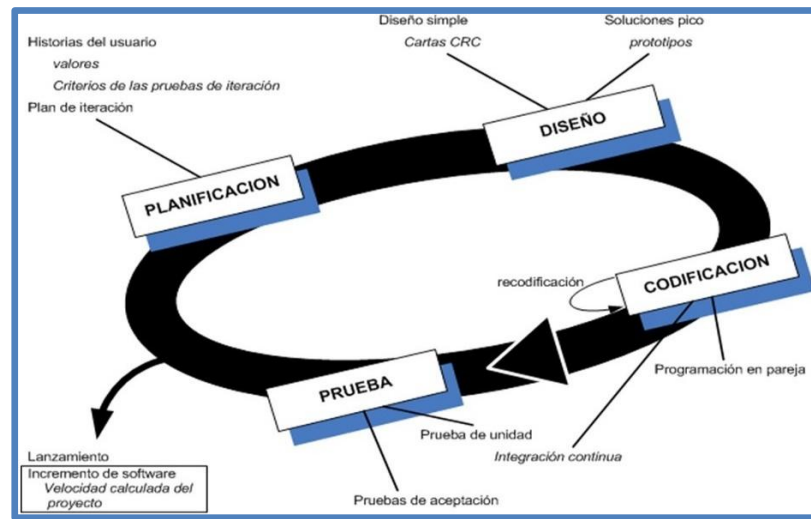


Ilustración 13 - Clico de vida XP.
Fuente: Espinoza (2007). (p.55).

Roles y responsabilidades

Hay diferentes roles en XP para diferentes tareas y propósitos durante el proceso y sus prácticas. A continuación, estos papeles se presentan según Beck (1999).

- **Programador:** es el encargado del desarrollo de la solución, arquitecto y responsable de la codificación.

- **Cliente:** El cliente escribe las historias y las pruebas funcionales, y decide cuando cada requisito se cumple. El cliente establece la prioridad de ejecución de los requisitos.

- **Encargado de pruebas (Tester):** corren las pruebas funcionales regularmente, transmiten los resultados de la prueba y mantienen las herramientas de prueba.

- **Encargado de seguimiento (Tracker):** encargado de seguimiento da retroalimentación en XP.

- **Entrenador (Coach):** responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

- **Consultor:** El consultor es un miembro externo que posee los conocimientos técnicos específicos necesarios. El consultor guía al equipo en la solución de sus problemas específicos.

- **Gestor (*Big Boss*):** El gestor toma las decisiones. Con el fin de ser capaz de distinguir cualquier dificultad o deficiencia en el proceso.

3.2.2. Metodología RUP

Díaz (2009) señala que RUP es un proceso formal, el cual provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de *software* de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por *Rational Software*, y está integrado con toda la *suite Rational* de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte (Customización). Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

Ciclo de vida

Para Díaz (2009) las cuatro fases del ciclo de vida son:

Inicio: El objetivo general de esta fase es establecer un acuerdo entre todos los interesados acerca de los objetivos del proyecto, se identifican los riesgos relacionados con el negocio y requerimientos.

Elaboración: El objetivo en esta fase es establecer la arquitectura base del sistema para proveer bases estables para el esfuerzo de diseño e implementación en la siguiente fase.

Construcción: El objetivo de la fase de construcción es clarificar los requerimientos faltantes y completar el desarrollo del sistema basados en la arquitectura base.

Transición: Esta fase se enfoca en asegurar que el software esté disponible para sus usuarios, además incluye pruebas del producto para poder hacer el entregable del mismo.

En la ilustración 14 se muestra el ciclo de vida de RUP.

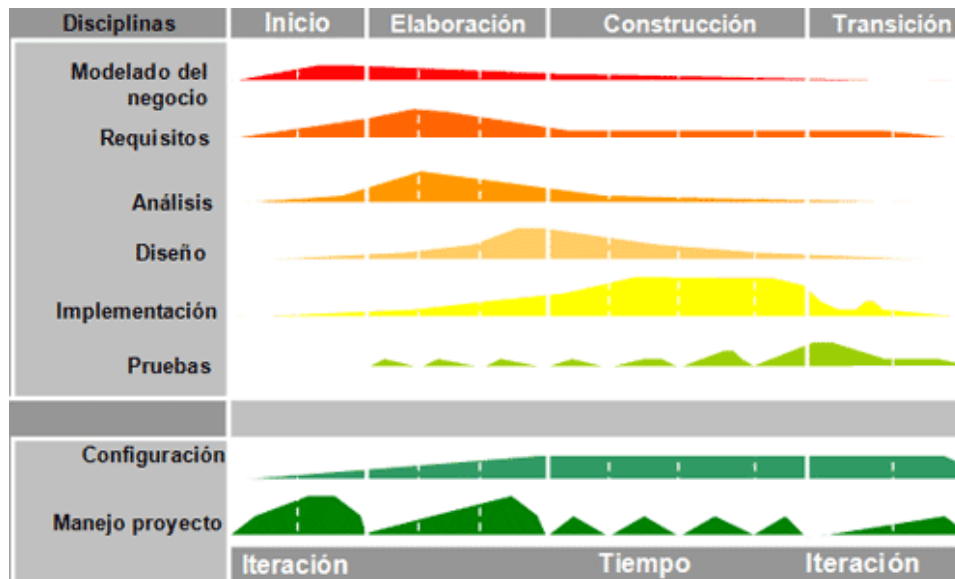


Ilustración 14 - Ciclo de vida RUP.

Fuente: Rincón (2012).

Ventajas

- Evaluación en cada fase que permite cambios de objetivos
- Funciona bien en proyectos de innovación.
- Sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

Desventajas

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos
- Nuestro cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

3.2.3. Metodología Ágil: Scrum

Marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum es:

- Ligerero.
- Fácil de entender.
- Extremadamente difícil de llegar a dominar.

Según Schwaber y Sutherland (2014), Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que se pueda mejorar. (p.4).El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos.

Schwaber y Sutherland (2014) dicen,

“Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.” (p. 4).

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación, los cuales son:

- **Transparencia:** Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que

dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo. Por ejemplo: lenguaje común, definición de terminado.

- **Inspección:** Los usuarios Scrum deben inspeccionar los artefactos y progreso frecuentemente, para detectar variaciones. Esto no debe interferir en el flujo de trabajo.

- **Adaptación:** Si se detecta una variación con respecto a las desviaciones de límites aceptables de un proceso, y que el producto resultante no será aceptable, el proceso o material debe ser ajustado. Dicho ajuste debe realizarse cuando antes para minimizar desviaciones mayores y re trabajos.

Scrum prescribe cuatro eventos formales para estas inspecciones y adaptaciones:

- Reunión de planificación del sprint (Sprint Planning Meeting).
- Scrum diario (Daily Scrum).
- Revision del sprint (Sprint Review).
- Retrospectiva del sprint (Sprint Retrospective).

El equipo Scrum

El equipo Scrum consiste en un Dueño del producto (*Product Owner*), el Equipo de Desarrollo (*Development Team*) y *Scrum Master*. Los equipos Scrum son auto organizados y multifuncionales; esto quiere decir que estos eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas eternas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo de Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad (Schwaber y Sutherland, 2013). (p .6).Estos equipos entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

- **El Dueño del Producto (*Product Owner*):** El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (*Product Backlog*). El Dueño de Producto es una única persona, no un comité. El Dueño de Producto podría representar los deseos de un comité en la Lista del Producto, pero aquellos que quieran cambiar la prioridad de un elemento de la Lista deben hacerlo a través del Dueño de Producto. Para que el Dueño de Producto pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Dueño de Producto se reflejan en el contenido y en la priorización de la Lista del Producto.

- **El Equipo de Desarrollo (*Development Team*):** El Equipo de Desarrollo consiste en profesionales que desempeñan el trabajo de entregar un incremento de producto “Terminado”. Son los participantes y responsables en la creación del producto, así como también, son empoderados para organizar y gestionar su propio trabajo. El tamaño de estos equipos debe ser lo suficientemente pequeños como para mantener la agilidad pero a su vez lo suficientemente grandes como para completar una cantidad de trabajo significativa.

- **El Scrum Master:** Recurso responsable de asegurar que Scrum es entendido, aplicado y adoptado. Líder que está al servicio del Equipo Scrum, encargado de modificar las iteraciones para maximizar el valor creado por el Equipo Scrum. Las actividades principales del Scrum Master hacia la metodología y demás roles los se puede definir como entender la planificación del producto, guiar al equipo de desarrollo para ser auto organizados y multifuncionales, eliminar impedimentos para el progreso del equipo, liderar y guiar la adopción de Scrum.

Eventos de Scrum

Para conllevar una mejor organización y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (time-boxes), de modo que tienen una duración máxima. Una vez que se planifica y se comienza un Sprint, su duración no debe ser modificada así como tampoco debe ser modificado su

alcance, esto último referente a nuevos requerimientos en el transcurso del Sprint o modificación de las historias de usuario definidas. A continuación se verá los eventos que se definen en el proceso de implementación de dicha metodología:

- **Sprint:** Es considerado el evento más importante y la base de esta metodología, se debe planificar en periodos de tiempo (time-box) no mayores a un mes y en el cual se crea un incremento de producto “Terminado”. Este contiene diferentes reuniones que deben ser llevadas a cabo para la buena práctica del Scrum, las cuales son: Reunión de planificación de Sprint (*Sprint Planning Meeting*), Scrums Diarios (*Daily Scrum*), Revisión de Sprint (*Sprint Review*), Retrospectiva del Sprint (*Sprint Retrospective*). En la ilustración 15 se puede ver el ciclo de vida de los eventos de Scrum para cada Sprint planificado.

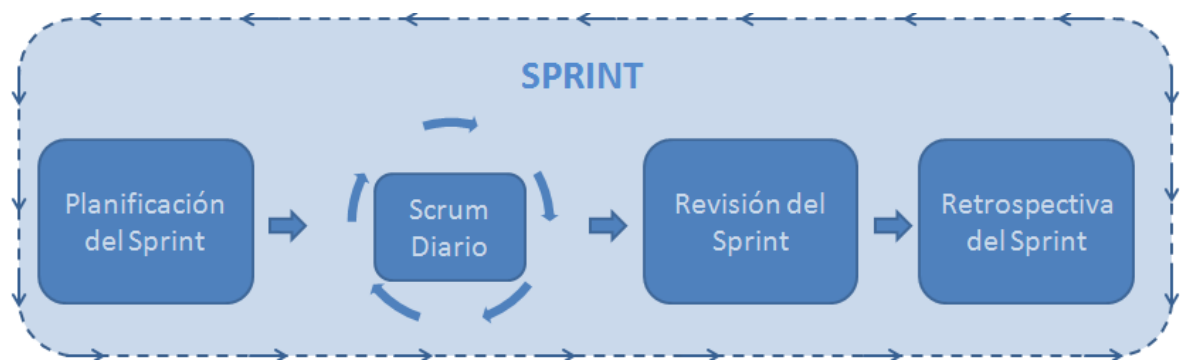


Ilustración 15 - Ciclo de vida de eventos dentro de un Sprint.

Fuente: Vila (2015)

Artefactos de Scrum

- **Lista de producto (*Product Backlog*):** Lista ordenada prioritariamente en la cual se despliegan todas las actividades necesarias para el desarrollo del producto y se convierte en la única fuente de requerimientos. El dueño del producto (*Product Owner*) es el responsable de mantener esta lista de producto. Dicha lista es evolutiva ya que va creciendo a medida que el producto y el entorno también lo hace, cambia constantemente para identificar lo que el producto necesita para ser adecuado. Estas listas pueden ser armadas a raíz de otros artefactos externos como por ejemplo.

Historias de usuario, en las que se define una actividad con valor para el negocio y de ella se despliegan tareas que son tomadas por el Equipo de Desarrollo.

- **Lista de pendientes (*Sprint Backlog*):** Lista que posee un conjunto de elementos de la lista de producto que no se encuentran en estado “Terminado”, esta es generada por el Equipo de Desarrollo y se toma como el incremento para el siguiente Sprint dando así más valor al mismo.

- **Definición de “Terminado” (*Definition of “Done”*):** Es definición para que un elemento de la lista de producto sea “Terminado” debe ser la misma para todo el equipo Scrum, debe cumplir con estándares específicos y ser un incremento utilizable para el Dueño del Producto. A medida que los equipos de Scrum maduran, se espera que su definición de “Terminado” se amplíe para incluir criterios más rigurosos para una mayor calidad en las actividades que se desarrollan.

CAPÍTULO 4

MARCO APLICATIVO

Para el desarrollo de cualquier trabajo de investigación es necesario tener una metodología de trabajo definida con el objetivo de estructurar, planear y controlar todo el proceso. En este capítulo se describe la metodología de desarrollo de software que fue implementado en la elaboración de este trabajo de grado así como el conjunto de herramientas que fueron utilizadas.

4.1. Metodología implementada

Para el proceso de desarrollo de la solución se implementó la metodología de desarrollo ágil Scrum, sin embargo existen aspectos definidos en diversas fuentes sobre éste método de desarrollo de software que no se adaptaron fácilmente a las características de éste proyecto por lo que fueron modificados u omitidos. La metodología se implementó siguiendo las siguientes fases:

- Durante la **fase inicial de planificación** se realizó la captura de requerimientos del *product Owner* (Equipo conformado por: Franklin Sandoval, Edwin Alvarado, Pedro Diaz) y se definió una arquitectura para la solución. Ésta estuvo sujeta a cambios pero fue necesaria para dar comienzo a la siguiente etapa.
- Luego se procedió a la **fase de desarrollo**, en esta el objetivo fue completar el proyecto de manera incremental por medio de una serie de sub fases llamadas sprints, cada una de éstas tiene una corta duración, generalmente de una a dos semanas, en éste intervalo de tiempo se culminan las tareas asignadas para ese sprint. Las tareas consideradas más complejas y que conllevan más tiempo se subdividen en tareas más pequeñas y/o simples para poder cumplir los objetivos de cada sprint.
- Las tareas a cumplir son organizadas por prioridad en una lista llamada product backlog, ésta lista es revisada y se actualizan las prioridades de cada tarea antes de cada sprint. Una parte resaltante de esta metodología es que plantea realizar

reuniones del equipo diariamente, en donde se rinde cuenta de los avances realizados, que obstáculos se presentan en la actividad actual y que cosas se planean cumplir antes de la siguiente reunión, así como también que tarea se asumirá al culminar la que se posee en curso.

- Luego de dar por culminada la fase de desarrollo se procedió a la **fase de pruebas**. Se realizaron pruebas para medir el rendimiento de la aplicación en cuanto a uso de Procesamiento y memoria Ram; el desempeño y comportamiento en la red se midió a través del jMeter y el porcentaje de paquetes perdidos y por último se realizaron las pruebas funcionales.

4.2. Herramientas de solución

En el desarrollo de la solución se utilizaron varias herramientas en conjunto, entre ellas se encuentran lenguajes de programación, herramientas de soporte para la metodología Scrum, herramientas de desarrollo del sistema, y los frameworks usados tanto para el desarrollo del API como de la aplicación. A continuación se explica cada una de estas herramientas y sus características más resaltantes por la cual fueron seleccionadas.

4.2.1. Lenguajes de programación

Para el desarrollo de la aplicación móvil se utilizó el lenguaje de programación Java. Éste es el lenguaje estándar para desarrollar sobre Android actualmente y provee los elementos necesarios para la aplicación. Se utilizó el Software Development Kit (SDK) que provee Android en conjunto con Android Studio.

Según Martínez (2015), Java es un lenguaje de programación de propósito general, concurrente y orientado a objetos desarrollado por Sun Microsystems y ahora a cargo de Oracle, uno de sus objetivos principales fue proporcionar portabilidad en

el software desarrollado. Aunque el lenguaje oficial para desarrollar para el sistema operativo Android es Java, estos dos mantienen importantes diferencias, el API de Java no es igual al API de Android, esta última no usa la máquina virtual Java sino una máquina virtual llamada Dalvik o ADR (Android Runtime).

El API de consumo fue desarrollado bajo el lenguaje Ruby, Ruby es un lenguaje de programación de propósito general popular que es especialmente adecuado para el desarrollo web. Matz (2001) comentó que Ruby es un lenguaje con un balance cuidado. Su creador, Yukihiro “Matz” Matsumoto, mezcló partes de sus lenguajes favoritos (Perl, Smalltalk, Eiffel, Ada y Lisp) para formar un nuevo lenguaje que incorpora tanto la programación funcional como la imperativa. Se considera un lenguaje flexible, ya que permite a sus usuarios alterarlo libremente. Ruby no restringe al desarrollador.

La aplicación Web es desarrollada en Rails ya persigue explotar en gran medida el patrón de arquitectura de software MVC (Modelo Vista Controlador), además trabaja bajo la idea de reutilizar constantemente componentes que se hayan creado con anterioridad. Siguiendo con estas premisas Rails se muestra como un framework flexible que permite desarrollar aplicaciones de una manera versátil y rápida, además de atender distintos requerimientos sin necesidad de repetir la definición de componentes.

4.2.2. Herramientas de soporte para la metodología Scrum

Durante todo el proceso de desarrollo se usaron un conjunto de herramientas que dieron soporte a la metodología ágil Scrum. Estas herramientas facilitaron la organización de tareas en el proyecto y el mantenimiento del código de la solución.

Para mantener un orden, priorizar y establecer fechas límites se creó una cartelera de tareas en Trello. Trello es una página web que provee una interfaz para manejar proyectos mediante tarjetas que representan tareas. Las tareas se crearon con estado “Pendiente”, se pasaban a estado “En progreso” y finalmente al culminarse se

establecía el estado “Culminado”. Las tareas se corresponden con requerimientos, funcionalidades y errores a corregir dentro de la aplicación. También se manejó el progreso de la documentación haciendo uso de esta herramienta, adjuntando así mismo tres documentos generales.

- Trazabilidad de la actividad: que especifica el desarrollo de la tarea.

TRAZABILIDAD	
#IDPUNTO:	Nombre del Punto
DESCRIPCIÓN	
Aquí va la descripción del punto.	
PREMISA	
Aquí va la premisa, sólo si amerita, (Ejemplos: Haber ejecutado los script desplegados con este punto) de no tener premisa se puede dejar en blanco este renglón.	
EJECUCIÓN DEL PROCESO	
<u>Identificación del proceso que se va a exponer</u>	
Aquí va el paso a paso de la verificación.	
OBSERVACIONES	
Aquí va la observación si lo amerita, en caso de no tenerlas, se puede dejar en blanco este renglón.	

Ilustración 16 - Plantilla de documento de trazabilidad.
Fuente: Los Autores

- Definición de completado: que define la culminación de los pasos para el reporte del cambio de estado de la actividad.

REQUERIMIENTO: ID DEL REQUERIMIENTO - DESCRIPCION DEL REQUERIMIENTO

Definición de Completado (Definition of Done)

Actividades	Realizada
Análisis	☐
Diseñar	☐
Desarrollo	☐
Verificación de los íconos - Formatos - Estilos establecidos	☐
Pruebas (<i>Elaboración Trazas, Instructivo, etc</i>)	☐
Guardar cambios (<i>COMMIT - CHECK IN</i>)	☐
Enviar correo informando sobre cambios efectuados	☐
Cierre del caso con los documentos y conclusiones del mismo (<i>TRELLO</i>)	☐

Ilustración 17 - Plantilla de Definición de Completado.
Fuente: Los Autores

- Documento de cambios: que define los archivos y versiones de los commits realizados en la herramienta de sincronización.

CONTROL DE CAMBIOS

Información General					
Elaborado por:			Fecha: Haga clic aquí para escribir una fecha.		
Descripción:					

Control de Código Fuente / Datos Técnicos y Funcionales					
Lenguaje o Herramienta:		Versión:			
Módulo / Proceso	Librería [PB] – Objeto	Elemento [PB] – Widget	Función, Evento o Propiedad	Descripción de la Solución	Versión GitHub

Control de Base de Datos / Datos Técnicos y Funcionales			
Manejador:		Versión:	
Nro.	Nombres de Scripts en Orden de Ejecución	Datos Modificados	Descripción de la Solución
1			
2			
3			
4			

Ilustración 18 - Plantilla de Control de Cambios.
Fuente: Los autores

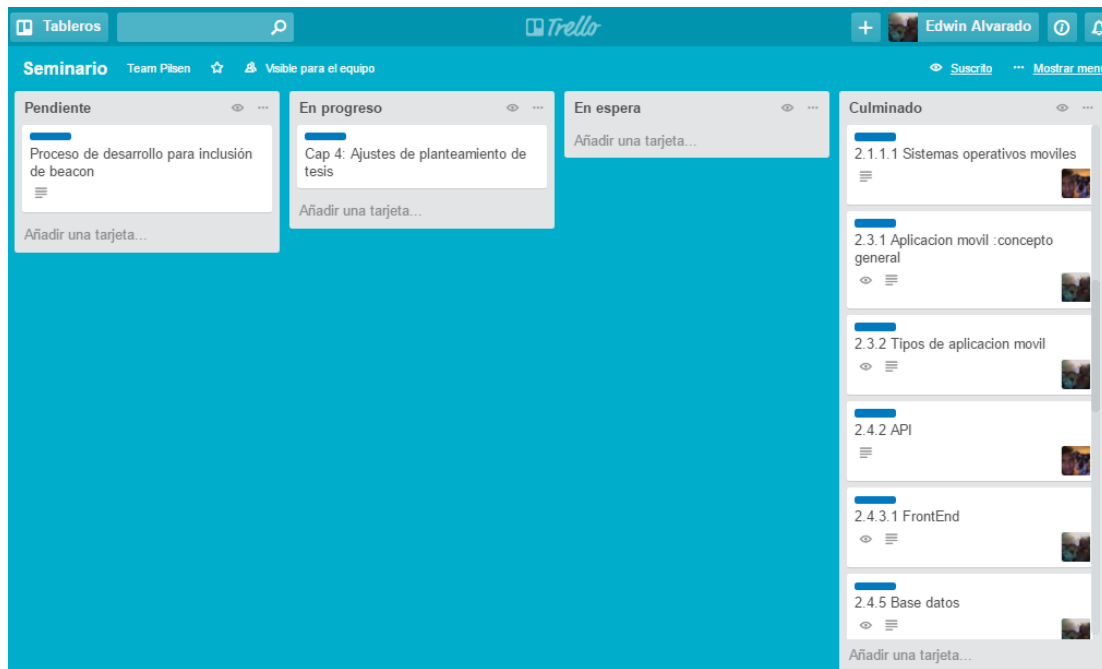


Ilustración 19 - Tablero de ciclo de vida de una actividad.
Fuente: Los autores

Para llevar el control de los cambios sobre el código de la aplicación se utilizó un repositorio de código Git. Ya que permite almacenar y manejar proyectos y repositorios Git en la nube, sin costo para cinco o menos usuarios. Esta herramienta se utilizó en conjunto con el plug-in incluido en Android Studio visualizado en la Ilustración 20, para manejar copias locales del proyecto y desplazar los cambios realizados al repositorio Git en la nube.

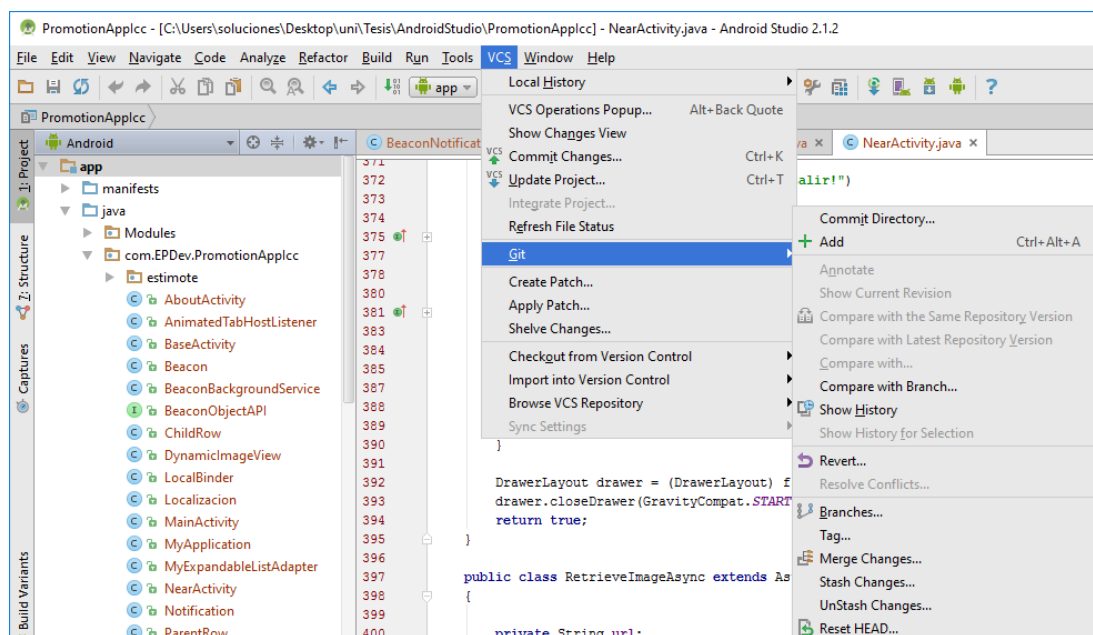


Ilustración 20 - Git en Android Studio.

Fuente: Los Autores

Así mismo las sesiones diarias (*Daily's*) fueron programadas en horas de la mañana alrededor de las 8:30 am, con una duración aproximada de 15-20min vía Skype, herramienta de software que permite comunicaciones de texto, voz y video sobre internet (VoIP). En las cuales se respondían tres preguntas básicas, ¿Que estás haciendo ahora? ¿Qué impedimentos se tienen hasta los momentos? Y ¿Que tarea se atenderá luego de culminar la actual? Para estas sesiones diarias se hizo uso de una de las herramientas de chat desktop más potentes en la actualidad, Skype Desktop.

4.2.3. Herramientas de desarrollo y administración del sistema

Estimote Cloud: Plataforma administrativa de dispositivos *Beacons* en la cual se hicieron trámites de compra y configuración de los mismos. En dicha página se logra visualizar y editar las características de cada *beacon*. Así como también la facilidad de descargar la librería de uso para la conexión con los dispositivos.

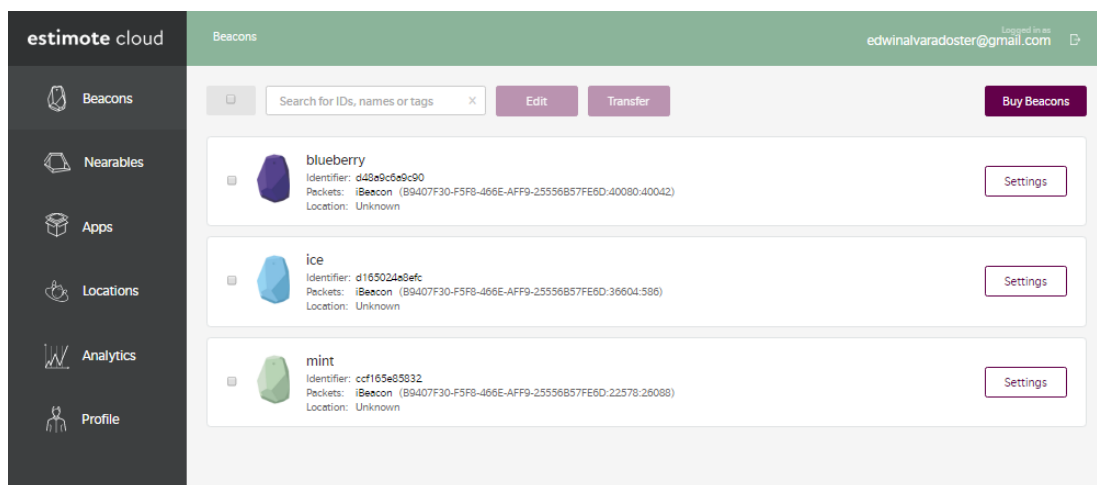


Ilustración 21 - Nube Estimote.
 Fuente: <https://cloud.estimote.com>

Heroku: Heroku es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación y funciona como hosting de aplicación. Heroku es propiedad de Salesforce.com.

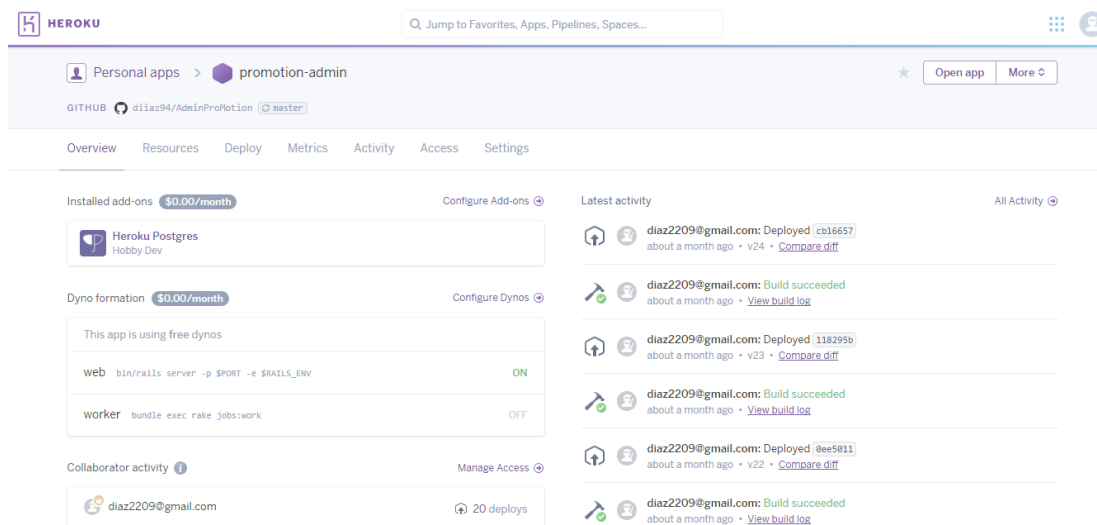


Ilustración 22 - Heroku - Administración y publicación de aplicación web.
 Fuente: Heroku.com

Consola de google: Permite a los programadores publicar aplicaciones, habilitar API's y mucho más. Para el prototipo funcional de solución se usó el API de google Maps.

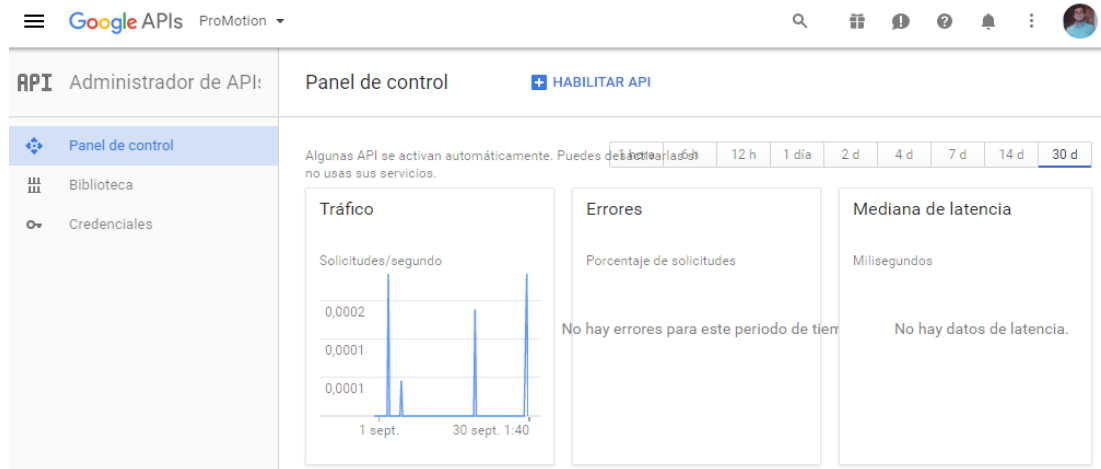


Ilustración 23 - Consola de desarrollador de google.

Fuente: <https://console.developers.google.com>

4.3. Diseño del sistema

Luego que se definen los requerimientos funcionales del sistema se procede al diseño de cada uno de los componentes que se desarrollaron y su integración entre la aplicación Web administrativa y la aplicación móvil enfocada al usuario final. Para esto se hace uso del artefacto historias de usuario para el desarrollo del sistema.

Se definieron seis (6) historias de usuario las cuales se dividen en dos (2) historias por sprint. El *sprint* 0 o llamado sprint inicial involucra reuniones de planificación y preparación de ambientes de desarrollo. Los *sprints* 1 y 2 involucran a su vez las historias correspondientes al desarrollo de los prototipos funcionales de aplicaciones. Todas estas historias de usuario fueron definidas con el equipo que representa el *product owner* del proyecto (Franklin Sandoval, Edwin Alvarado, Pedro Diaz).

Sprint 0 – Planificación y preparación de ambientes

En este *Sprint* se definieron una serie de historias de usuario para la preparación de los ambientes de desarrollo tanto móvil como *web* y de esta manera lograr la realización de los *sprints* (1 y 2) de requerimientos funcionales siguientes sin inconvenientes. Así como también se define un tiempo de 40 horas para dicho Sprint.

Tabla 7 - Historia de usuario #1 - Sprint 0

Historia de Usuario – Móvil App		
Número: 1	Usuario: Desarrollador	
Nombre historia: Construcción y configuración de ambiente de desarrollo móvil		
Prioridad en negocio: Alta		Riesgo en desarrollo: Alta
Puntos estimados: 4	Estimación: 40 horas	Iteración asignada: 0
Programador responsable: Edwin Alvarado		
Descripción: Yo como desarrollador quisiera tener en mi ordenador todas las herramientas y configuraciones necesarias para iniciar el desarrollo y construcción de la aplicación móvil que forma parte de la solución propuesta.		
Criterios de Aceptación: - Dado el desarrollador cuando inicie la implementación de la aplicación entonces podrá tener la herramienta de desarrollo. - Dado el desarrollador cuando necesite hacer pruebas de la aplicación desarrollada entonces podrá tener un emulador para probar la aplicación. - Dado varios desarrolladores cuando necesiten modificar el código en paralelo entonces puedan trabajar en conjunto a través de un manejador de versiones. - Dado un desarrollador cuando necesite debuggear la aplicación desarrollada entonces podrá tener una consola para debuggear la misma. - Dado un desarrollador cuando necesite conectarse con los dispositivos bluetooth debe poseer la cuenta del distribuidor del producto para la configuración de los equipos.		
Observaciones:		

Fuente: Los Autores

Tabla 8 - Historia de usuario #2 - Sprint 0

Historia de Usuario – Web Admin		
Número: 2	Usuario: Desarrollador	
Nombre historia: Construcción y configuración de ambiente de desarrollo web		
Prioridad en negocio: Alta		Riesgo en desarrollo: Alta
Puntos estimados: 4	Estimación: 40 horas	Iteración asignada: 0
Programador responsable: Pedro Díaz		
Descripción: Yo como desarrollador quisiera tener en mi ordenador todas las herramientas y configuraciones necesarias para iniciar el desarrollo y construcción de la aplicación web que forma parte de la solución propuesta.		
Criterios de Aceptación: • Dado el desarrollador cuando inicie la implementación de la aplicación web entonces deberá tener configurado en el entorno de desarrollo de Ruby on <u>Rails</u>. • Dado el desarrollador cuando inicie la implementación de la aplicación web entonces deberá tener a su disposición el diseño de las pantallas que conformaran la misma. • Dado varios desarrolladores cuando necesiten modificar el código de forma paralela entonces puedan trabajar en conjunto a través de un manejador de versiones.		
Observaciones:		

Fuente: Los Autores

Sprint 1 – Notificaciones y Usuario Administrador

Para el Sprint uno (1) se definieron una serie de módulos y funcionalidades tales como, Notificaciones push y Rol de Usuario Administrador para las aplicaciones tanto móvil como web respectivamente. Así como también se define un tiempo de 80 horas para dicho Sprint.

Tabla 9 - Historia de usuario #3 - Sprint 1

Historia de Usuario – Móvil App – Rol Cliente		
Número: 3	Usuario: Consumidor	
Nombre historia: Notificaciones push y servicios en background		
Prioridad en negocio: Alta		Riesgo en desarrollo: Alta
Puntos estimados: 8	Estimación: 80 horas	Iteración asignada: 1
Programador responsable: Edwin Alvarado		
Descripción: Como usuario final consumidor de la aplicación deseo que se me desplieguen notificaciones push con las promociones y servicios de las tiendas asociadas en la aplicación al momento de estar en cercanía de estos comercios, para de esta manera poder seleccionar, visualizar y tener conocimiento de las promociones que poseen estas tiendas. Así como también deseo que estas notificaciones mi teléfono se encuentre en modo inactivo y poder configurarlas desde un módulo directamente en la aplicación		
Criterios de Aceptación: • Dado un usuario final consumidor del aplicativo móvil cuando ejecute la aplicación se iniciara la búsqueda de tiendas en cercanía por rango de configuración por tienda (beacon). • Dado un usuario final consumidor de la aplicación móvil cuando se aproxime a una tienda con la aplicación activa y que a su vez posea la configuración y dispositivos asociados a nuestra solución se le desplegara a través de una notificación push la promoción asociada a la misma tienda. • Dado un usuario final consumidor de la aplicación móvil cuando se aproxime a una tienda con la aplicación en background y que a su vez posea la configuración y dispositivos asociados a nuestra solución se le desplegara a través de una notificación push la promoción asociada a la misma tienda. • Dado un usuario final consumidor de nuestras tiendas asociadas al momento de hacer clic en la notificación que se le despliega podrá ver la información específica de la misma promoción y a su vez la tienda asociada a la promoción. • Dado un usuario final consumidor podrá configurar para que la frecuencia de envío de notificaciones sea de un minuto. • Dado un usuario final consumidor podrá configurar para que la frecuencia de envío de notificaciones sea de una hora. • Dado un usuario final consumidor podrá configurar para que la frecuencia de envío de notificaciones sea de un día.		
Observaciones:		
Fuente: Los Autores		

Tabla 10 - Historia de usuario #4 - Sprint 1

Historia de Usuario – Web Admin - Rol Administrador		
Número: 4	Usuario: Desarrollador	
Nombre historia: Portal Administrativo		
Prioridad en negocio: Alta		Riesgo en desarrollo: Alta
Puntos estimados: 8	Estimación: 80 horas	Iteración asignada: 1
Programador responsable: Pedro Díaz		
Descripción: Yo como cliente administrador quisiera ingresar a un portal para poder administrar todas las opciones del sistema de promociones.		
Criterios de Aceptación: • Dado un usuario con rol de administrador cuando ingresa al sistema a través de un login de usuario entonces se muestran las opciones relacionadas a ese rol. • Dado un usuario con rol de administrador cuando presione la opción de usuarios entonces pueda ver, crear, editar y eliminar todos los usuarios del sistema. • Dado un usuario con rol de administrador cuando presione la opción de categorías entonces pueda ver, crear, editar y eliminar todas las categorías. • Dado un usuario con rol de administrador cuando presione la opción de comercios entonces pueda ver, crear, editar y eliminar todos los comercios. • Dado un usuario con rol de administrador cuando presione la opción de tiendas entonces pueda ver, crear, editar y eliminar todas las tiendas del sistema además de se debe poder indicar en que parte exactamente se encuentra ubicada la tienda usando un mapa. • Dado un usuario con rol de administrador cuando presione la opción de promociones entonces pueda ver, crear, editar y eliminar todas las promociones del sistema. • Dado un usuario con rol de administrador cuando presione la opción de beacons entonces pueda ver, crear, editar y eliminar la información relacionada con los beacons.		
Observaciones:		

Fuente: Los Autores

Sprint 2 – Construcción Móvil y Rol Propietario

Para el Sprint uno (2) se definieron una serie de módulos y funcionalidades tales como, Construcción de app Móvil y Rol de Usuario Administrador para las aplicaciones tanto móvil como web respectivamente. Así como también se define un tiempo de 80 horas para dicho Sprint.

Tabla 11 - Historia de usuario #5 - Sprint 2

Historia de Usuario – Móvil App – Rol Cliente		
Número: 5	Usuario: Consumidor	
Nombre historia: Construcción y diseño de módulos de aplicación.		
Prioridad en negocio: Alta		Riesgo en desarrollo: Alta
Puntos estimados: 8	Estimación: 80 horas	Iteración Asignada: 2
Programador responsable: Edwin Alvarado		
Descripción: Como usuario final deseo poder acceder a las opciones de Categorías, tiendas más cercanas, las promociones más destacadas, acerca de la aplicación y compartir en redes sociales.		
Criterios de Aceptación: • Dado un usuario final consumidor de la aplicación móvil podrá acceder a un módulo de inicio donde pueda visualizar las promociones más destacadas y categorías como filtro de búsqueda de tiendas y promociones. • Dado un usuario final de la aplicación cuando necesite buscar una tienda podrá buscarlas por el nombre de la misma en el módulo de categorías. • Dado un usuario final cuando hace clic en una tienda vera la descripción y ubicación de cada una así como también sus promociones asociadas. • Dado un usuario final cuando hace clic en una promoción vera la descripción y especificación de la misma. • Dado un usuario final de la aplicación al ingresar al módulo de más cerca podrá ver la ruta de la tienda más cerca a la ubicación actual del mismo. Así como también, la ubicación exacta de cada una de las tiendas asociadas a nuestro sistema. • Dado un usuario final de la aplicación al ingresar al módulo de Nosotros podrá visualizar quienes somos, como contactarnos y como afiliar sus tiendas en nuestro comercio. • Dado un usuario final de la aplicación al seleccionar la opción de compartir, podrá		

enviar referir y dar a conocer la aplicación a todos sus contactos a través de las redes sociales instaladas en su dispositivo. • Dado un usuario final de la aplicación al ingresar al módulo de categorías de la aplicación podrá ver las tiendas separadas por su rubro de comercio, ya sea tiendas deportivas, restaurantes, locales nocturnos, entre otros.
Observaciones:

Fuente: Los Autores

Tabla 12 - Historia de usuario #6 - Sprint 2

Historia de Usuario – Web Admin - Rol Propietario		
Número: 6	Usuario: Desarrollador	
Nombre historia: Portal Administrativo para comercios		
Prioridad en negocio: Alta		Riesgo en desarrollo: Alta
Puntos estimados: 8	Estimación: 80 horas	Iteración asignada: 2
Programador responsable: Pedro Díaz		
Descripción: Yo como cliente propietario de uno o varios comercios quisiera ingresar a un portal para poder administrar la información asociada a mis comercios afiliados tales como tiendas y promociones.		
Criterios de Aceptación: • Dado un usuario con rol de propietario cuando ingresa al sistema a través de un login de usuario entonces se muestra la opción “Mis comercios” en el menú, la cual al presionar desplegara como items todos los comercios asociados a ese usuario además de una opción para visualizarlos todos y otra para agregar uno nuevo. • Dado un usuario con rol de propietario cuando presione el ítem de alguno de sus comercios entonces dentro del menú podrá visualizar un ítem de “Tiendas”, además de una opción para ver el detalle de ese comercio. • Dado un usuario con rol de propietario cuando presione el ítem “Tiendas” de algún comercio seleccionado entonces dentro del menú podrá visualizar como items todas las tiendas relacionadas a ese comercio además de una opción para ver todas las tiendas y otra para agregar una nueva. • Dado un usuario con rol de propietario cuando presione el ítem de alguna tienda perteneciente a un comercio previamente seleccionado entonces dentro del menú podrá visualizar una opción para ver el detalle de la tienda y otra para ver las promociones asociadas a dicha tienda.		

-
- | |
|--|
| <ul style="list-style-type: none"> • Dado un usuario con rol de propietario cuando ingrese al portal entonces a través de las opciones del menú podrá ver, crear, editar y eliminar toda la información relacionada a sus comercios (comercios, tiendas y promociones). |
|--|

Observaciones:**Fuente: Los Autores**

Luego de analizar y definir las historias de usuario que se desarrollaron para la construcción de las funcionalidades del sistema se definio diseño de la solucion para ambos esquemas tanto web como movil.

4.4. Diseño de la solución

La solución propuesta compone el desarrollo de dos aplicaciones (Web administrativa, móvil) el esquema de diseño que se impone se divide en dos secciones.

4.4.1. Casos de uso esquema movil

El Esquema de diseño móvil impone el desarrollo sobre Android específicamente Android Studio basado en Activities que lleva a diseñar la aplicación como un flujo de pantallas, cada una con funciones específicas y comunes a la vez. Se define un modelo sencillo y minimalista en el que el usuario no necesita estar logeado en la aplicación para el uso del prototipo funcional de solución, esto indica que al iniciar la aplicación ya el usuario se ubicara en la pantalla de inicio, dándole así la facilidad de navegar entre los diferentes flujos de la aplicación. En la siguiente sección se encuentran el caso de uso descrito en la Ilustracion 24 que se desarrollo para en el diseño de solución en el aplicativo móvil.

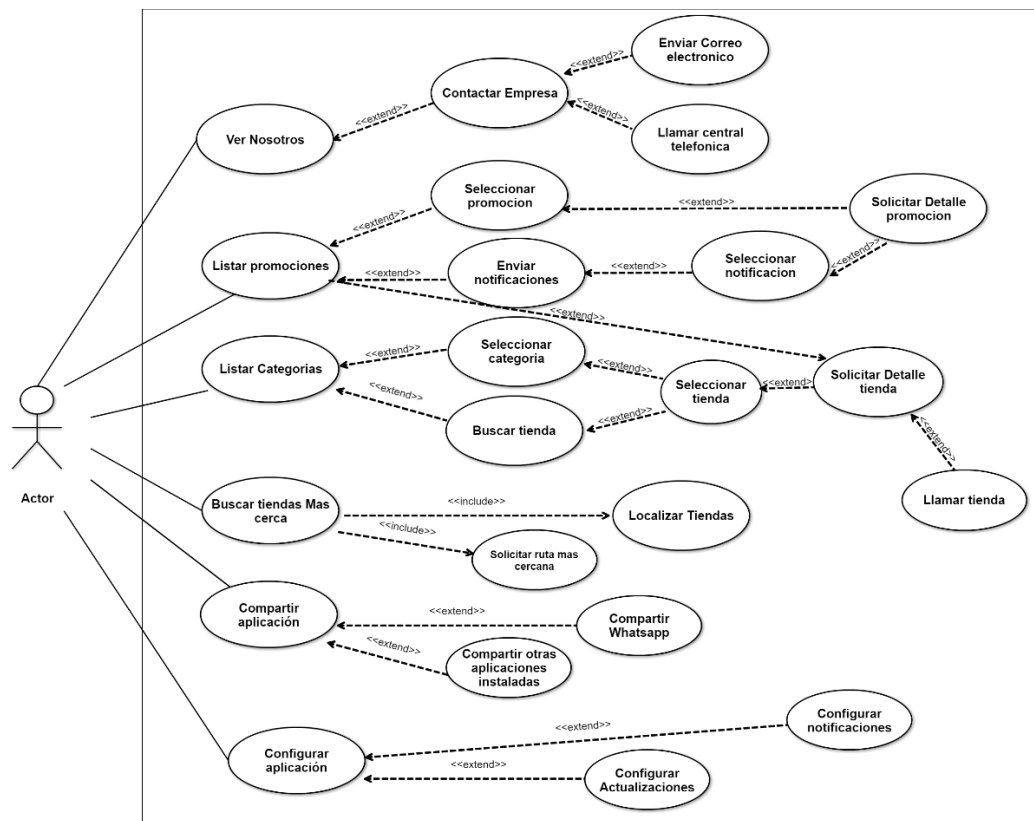


Ilustración 24 - Caso de uso #1, modelado móvil.
Fuente: Los Autores

Especificacion Caso de Uso, modelado movil.

- **Identificador y nombre:** Ver Nosotros.
 - **Breve Descripción:** permite que el consumidor o cliente final logre visualizar la información necesaria de quienes somos, como se pueden afiliar a nuestro sistema, llamar a nuestra central telefónica o enviar correos con sus comentarios o dudas.
 - **Actores:**
 - **Actor Primario:** Consumidor
 - **Actor Secundario:**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El consumidor puede ver información básica de

quienes somos, como se pueden afiliar a nuestro sistema y contactarnos vía telefónica o a través de un correo electrónico.

▪ **Flujos Alternativos:** El consumidor selecciona el botón flotante disponible en la parte inferior de la pantalla y puede llamar o enviar correos a nuestro sistema.

- **Requerimientos Especiales:-.**
- **Pre-condición:** El usuario debe haber iniciado la aplicación y seleccionar el módulo de nosotros.
- **Post-condición:** El consumidor entrara en contacto con el sistema.
- **Puntos de Extensión**
 - **Include:**
 - **Extend:-.** Contactar Empresa.

- **Identificador y nombre:** Listar promociones
 - **Breve Descripción:** Permite que el consumidor logre la visualización de las promociones en el sistema.
 - **Actores:**
 - **Actor Primario:** consumidor
 - **Actor Secundario:**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El consumidor inicia la aplicación y podrá ver en el módulo de inicio la lista de las promociones más destacadas.
 - **Flujos Alternativos:** El consumidor ingresa al detalle de una tienda y logra visualizar la lista de promociones de la misma.
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema. El usuario debe haber seleccionado ver el detalle de una tienda o en su defecto el módulo de inicio.
 - **Post-condición:** El propietario podrá visualizar todas las promociones disponibles.

- **Puntos de Extensión**
 - **Include:**
 - **Extend:** Seleccionar promoción, Enviar notificaciones
- **Identificador y nombre:** Listar categorías
 - **Breve Descripción:** permite que el consumidor de algún comercio pueda visualizar las categorías de las tiendas o rubros de negocio a los que pertenecen.
 - **Actores:**
 - **Actor Primario:** consumidor
 - **Actor Secundario:**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El consumidor ingresa al módulo de inicio y logra visualizar la lista de categorías que se poseen en la aplicación.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El consumidor debe haber iniciado sesión en el sistema.
 - **Post-condición:** El consumidor podrá visualizar las categorías que se poseen en el sistema.
 - **Puntos de Extensión**
 - **Include:**
 - **Extend:** Seleccionar Categoría, Buscar Tienda
- **Identificador y nombre:** Buscar tiendas Más cerca
 - **Breve Descripción:** permite que el consumidor logre encontrar la tienda o comercio más cerca de su ubicación actual, así como también la ubicación exacta de todas las tiendas disponibles.
 - **Actores:**
 - **Actor Primario:** consumidor
 - **Actor Secundario:**

- **Flujo de Eventos:**
 - **Flujo Básico:** El consumidor selecciona el módulo de Más cerca y se le despliega un mapa con la ruta de la tienda más cercana a su ubicación, así como también un marcador que le indica el punto exacto de ubicación de los comercios afiliados al sistema.
 - **Flujos Alternativos:-.**
- **Requerimientos Especiales:-.**
- **Pre-condición:** El consumidor debe haber iniciado sesión en el sistema.
- **Post-condición:** El consumidor podrá visualizar la ruta a la tienda.
- **Puntos de Extensión**
 - **Include:** Localizar tiendas
 - **Extend:** Solicitar ruta más cerca
- **Identificador y nombre:** Compartir aplicación
 - **Breve Descripción:** permite al consumidor compartir la aplicación en redes sociales que tenga instaladas en su dispositivo.
 - **Actores:**
 - **Actor Primario:** consumidor
 - **Actor Secundario:**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El consumidor selecciona el botón flotante para compartir la aplicación. Y se despliega un popup donde se podrá elegir en que red social se desea enviar esta información.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El consumidor debe haber iniciado sesión en el sistema.
 - **Post-condición:** El consumidor habrá compartido la aplicación y el link de descarga de la misma en la red social seleccionada.
 - **Puntos de Extensión**
 - **Include:**

- **Extend:** Compartir en WhatsApp, Compartir en otras aplicaciones instaladas.
- **Identificador y nombre:** Configurar aplicación
 - **Breve Descripción:** permite al consumidor configurar valores de actualización y frecuencia de envío de notificaciones.
 - **Actores:**
 - **Actor Primario:** consumidor
 - **Actor Secundario:**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El consumidor ingresa al módulo de configuración y tiene la posibilidad de configurar la frecuencia de envío de notificaciones y las políticas de actualización de la misma.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El consumidor debe haber iniciado sesión en el sistema.
 - **Post-condición:**
 - **Puntos de Extensión**
 - **Include:**
 - **Extend:** Configurar notificaciones, configurar Actualizaciones

4.4.2. Casos de uso esquema web

El enfoque de solución de ProMotion Apps llevo a desarrollar la aplicación web administrativa basándose en el siguiente Esquema de diseño web que incluye tres roles diferentes: administrador de sistema, administrador de comercios y api Restful para terceros, este desarrollo se llevó a cabo utilizando el Framework Ruby on Rails, el cual a través de su facilidad para la gestión de ámbitos, permitió crear un ámbito correspondiente a cada rol de la solución. De esta manera se logró separar la lógica de

aplicación para aquellos roles que requerían de ciertos componentes como inicio de sesión e interfaces de usuario, y un manejo totalmente diferente para permitir la consulta de datos a través de un api Restful utilizando la librería jbuilder provista por el framework en cuestión. Además se logró definir ciertas restricciones para la consulta de los datos de la aplicación así como también asignar privilegios a los roles del sistema mencionados anteriormente. Definido esto se encuentra a continuación en la siguiente sección los casos de uso aplicados en el diseño de solución para el aplicativo web. Se tienen principalmente tres (3) casos de uso, donde se definen los roles que se poseen desde nuestro sistema web Rol Cliente. Ver Ilustracion 25, Rol Administrador. Ver Ilustracion 26 y Rol Propietario o dueño de local comercial. Ver Ilustracion 27.

CU 1 – Rol Cliente

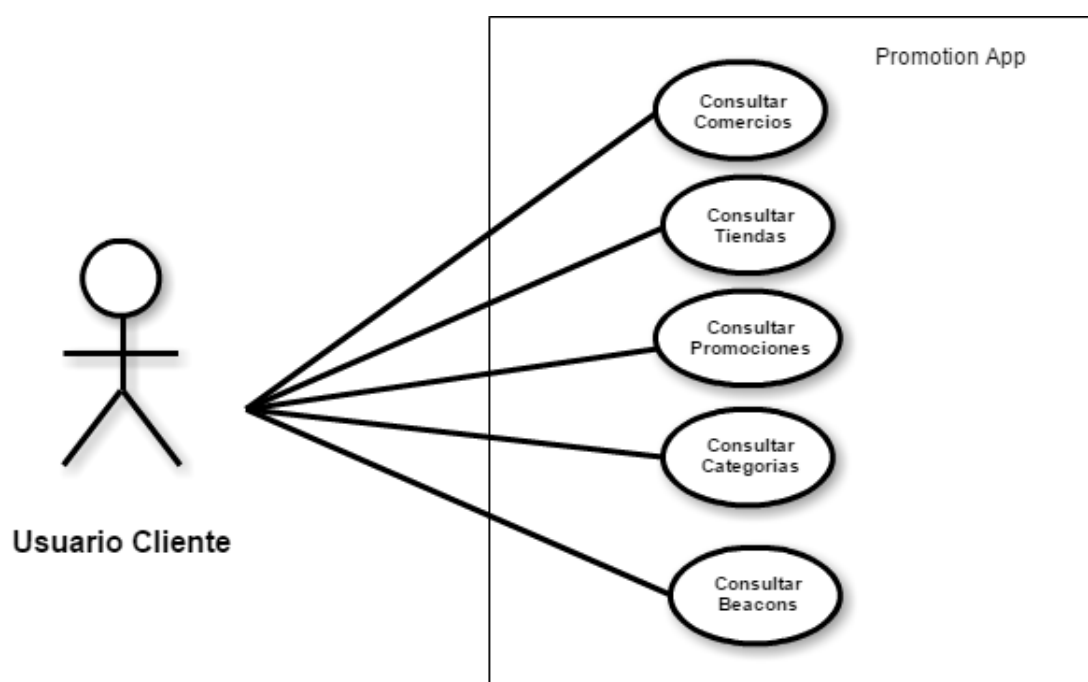


Ilustración 25 - Caso de uso #2. Rol cliente
Fuente: Los Autores

CU 2 – Rol Administrador

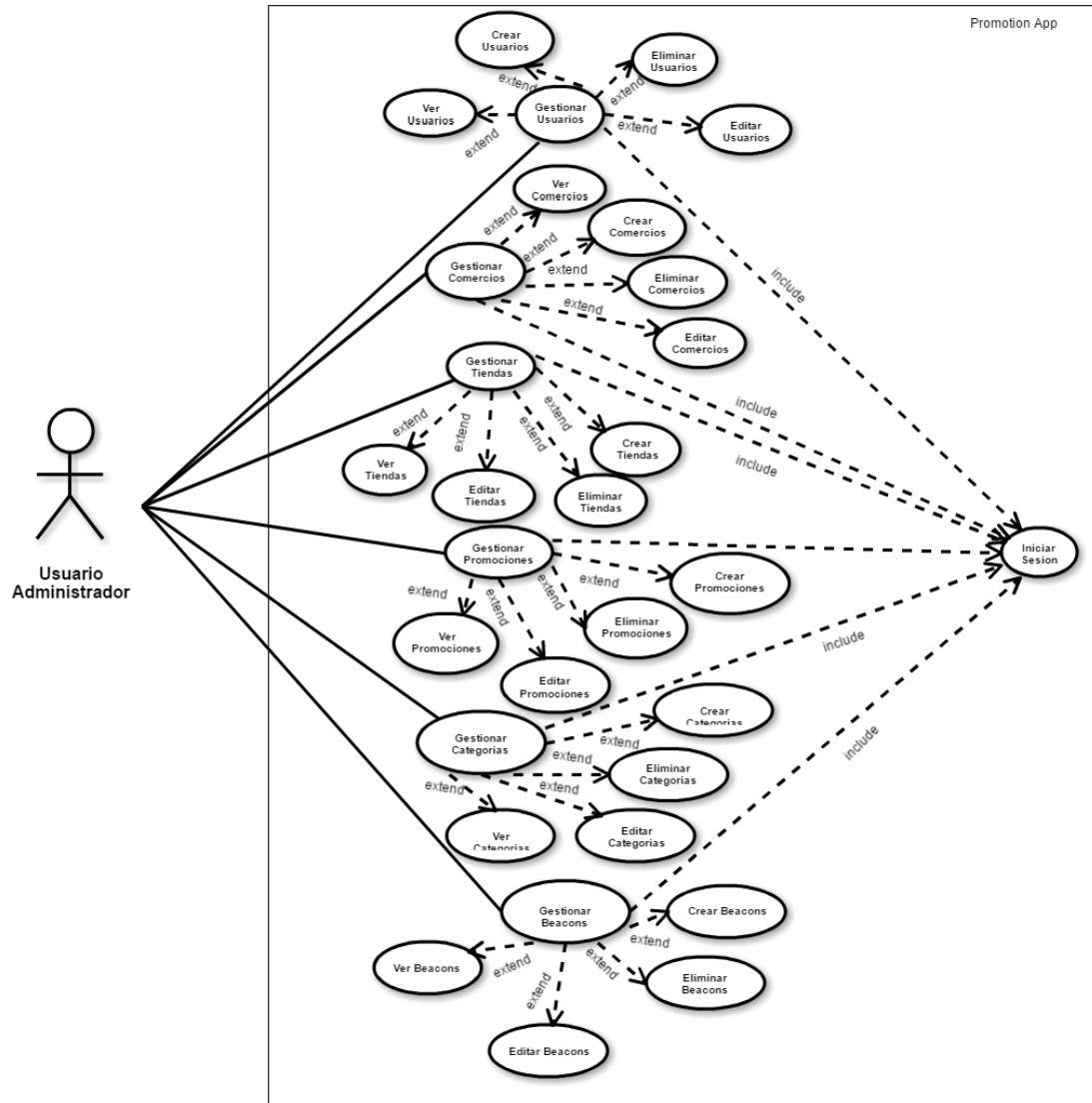


Ilustración 26 - Caso de uso #3, rol Administrador
Fuente: Los Autores

CU 3 – Rol Propietario

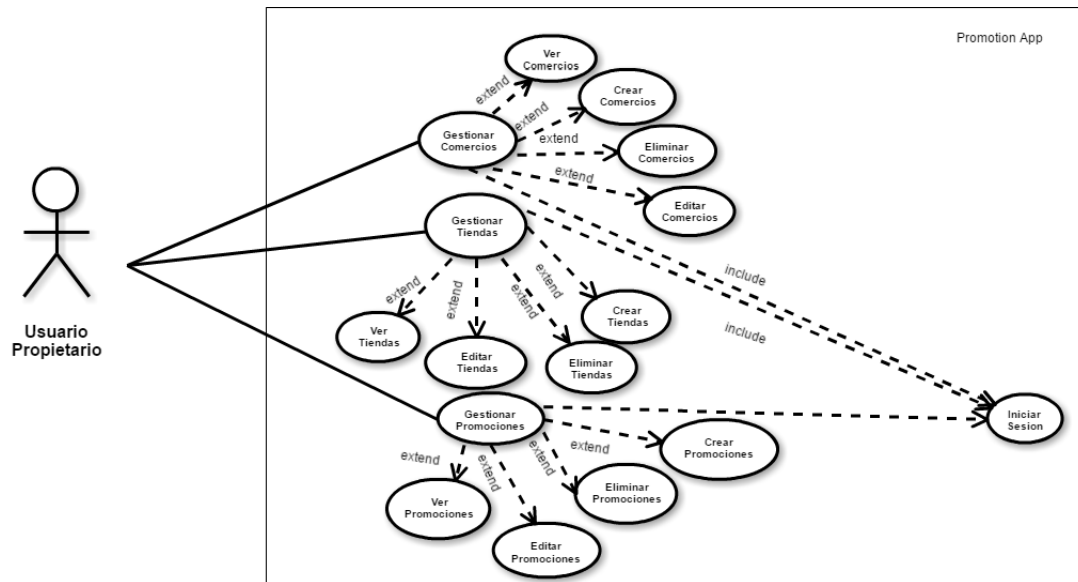


Ilustración 27 - Caso de uso #4. Rol propietario

Fuente: Los Autores

Especificacion Caso de Uso, modelado web, rol propietario.

- **Identificador y nombre:** Gestionar Comercios.
 - **Breve Descripción:** permite que el *propietario* de algún comercio realice toda la gestión necesaria sobre la información de sus comercios.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:** -.
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede ver, editar, crear y eliminar sus comercios, y el sistema solicitará o mostrara datos como nombre y ubicación del comercio.
 - **Flujos Alternativos:** -.
 - **Requerimientos Especiales:** -.
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema.

- **Post-condición:** El propietario pondrá a disposición de terceros, la información relacionada con sus comercios.
- **Puntos de Extensión**
 - **Include:** Iniciar Sesión.
 - **Extend:** Ver Comercios, Crear Comercios, Eliminar Comercios, Editar Comercios.
- **Identificador y nombre:** Gestionar Tiendas.
 - **Breve Descripción:** permite que el *propietario* de algún comercio realice toda la gestión necesaria sobre la información de sus tiendas.
 - **Actores:**
 - **Actor Primario:** *Propietario*
 - **Actor Secundario:-.**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede ver, editar, crear y eliminar sus tiendas, y el sistema solicitará o mostrará datos como, nombre, correo, numero de contacto, ubicación exacta, imagen, descripción y categoría de la tienda.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema.
 - **Post-condición:** El propietario pondrá a disposición de terceros, la información relacionada con sus tiendas.
 - **Puntos de Extensión**
 - **Include:** Iniciar Sesión.
 - **Extend:** Ver Tiendas, Crear Tiendas, Eliminar Tiendas, Editar Tiendas.

- **Identificador y nombre:** Gestionar Promociones.
 - **Breve Descripción:** permite que el *propietario* de algún comercio realice toda la gestión necesaria sobre la información de sus promociones.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede ver, editar, crear y eliminar sus promociones, y el sistema solicitará o mostrará muestra datos como, titulo, fecha de caducidad, imagen y precio de la promoción.
 - **Flujos Alternativos:-**
 - **Requerimientos Especiales:-**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema.
 - **Post-condición:** El propietario pondrá a disposición de terceros, la información relacionada con sus promociones.
 - **Puntos de Extensión**
 - **Include:** Iniciar Sesión.
 - **Extend:-** Ver Promociones, Crear Promociones, Eliminar Promociones, Editar Promociones.

- **Identificador y nombre:** Ver Comercios.
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda visualizar todos los comercios de los cuales él es dueño.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede ver todos sus comercios, y el sistema mostrara datos como nombre y descripción del comercio.

- **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios asociados.
 - **Post-condición:** El propietario podrá visualizar todos sus comercios.
 - **Puntos de Extensión**
 - **Include:-.**
 - **Extend:-.**
 - **Identificador y nombre:** Ver Tiendas.
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda visualizar todas las tiendas asociadas a los comercios de los cuales él es dueño.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-.**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede ver todas sus tiendas por comercio, y el sistema mostrara datos como, nombre, correo, numero de contacto, ubicación exacta, imagen, descripción y categoría de las tiendas.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios y tiendas asociadas.
 - **Post-condición:** El propietario podrá visualizar todas sus tiendas por comercio.
 - **Puntos de Extensión**
 - **Include:-.**
 - **Extend:-.**
-

- **Identificador y nombre:** Ver Promociones.
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda visualizar todas las promociones asociadas a las tiendas que a su vez están relacionadas con algún comercio del cual él es dueño.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:** -.
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede ver todas sus promociones por tienda, y el sistema mostrara datos como, titulo, fecha de caducidad, imagen y precio.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios, tiendas y promociones asociadas.
 - **Post-condición:** El propietario podrá visualizar todas sus promociones por tiendas.
 - **Puntos de Extensión**
 - **Include:-.**
 - **Extend:-.**

- **Identificador y nombre:** Crear Comercios.
 - **Breve Descripción:** permite que el *propietario* pueda crear comercios relacionados a él.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-.**
 - **Flujo de Eventos:**

- **Flujo Básico:** El *propietario* puede crear comercios, y el sistema solicitara datos como nombre y descripción del comercio.
 - **Flujos Alternativos:-**
 - **Requerimientos Especiales:-**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema.
 - **Post-condición:** El propietario podrá visualizar el comercio creado.
 - **Puntos de Extensión**
 - **Include:-**
 - **Extend:-**
- **Identificador y nombre:** Crear Tiendas.
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda crear tiendas asociadas a los comercios de los cuales él es dueño.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede crear tiendas por comercio, y el sistema solicitara datos como, nombre, correo, número de contacto, ubicación exacta, imagen, descripción y categoría de las tiendas.
 - **Flujos Alternativos:-**
 - **Requerimientos Especiales:-**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios asociados.
 - **Post-condición:** El propietario podrá visualizar la tienda creada.
 - **Puntos de Extensión**
 - **Include:-**
 - **Extend:-**

- **Identificador y nombre:** Crear Promociones.
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda crear promociones asociadas a las tiendas que a su vez están relacionadas con algún comercio del cual él es dueño.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-.**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede crear promociones por tienda, y el sistema solicitara datos como, titulo, fecha de caducidad, imagen y precio.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios y tiendas asociadas.
 - **Post-condición:** El propietario podrá visualizar la promoción creada.
 - **Puntos de Extensión**
 - **Include:-.**
 - **Extend:-.**

- **Identificador y nombre:** Editar Comercios.
 - **Breve Descripción:** permite que el *propietario* pueda editar comercios relacionados a él.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-.**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede editar comercios, y el sistema solicitara datos como nombre y descripción del comercio.
 - **Flujos Alternativos:-.**

- **Requerimientos Especiales:-**
- **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios relacionados a él.
- **Post-condición:** El propietario podrá visualizar el comercio con la información actualizada.
- **Puntos de Extensión**
 - **Include:-**
 - **Extend:-**
- **Identificador y nombre:** Editar Tiendas.
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda editar tiendas asociadas a los comercios de los cuales él es dueño.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:** -.
 - **Flujo de Eventos:**
 - **1.3.1 Flujo Básico:** El *propietario* puede editar tiendas por comercio, y el sistema solicitara datos como, nombre, correo, número de contacto, ubicación exacta, imagen, descripción y categoría de las tiendas.
 - **1.3.2 Flujos Alternativos:-**
 - **Requerimientos Especiales:-**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios y tiendas asociadas.
 - **Post-condición:** El propietario podrá visualizar la tienda con la información actualizada.
 - **Puntos de Extensión**
 - **Include:-**
 - **Extend:-**

- **Identificador y nombre:** Editar Promociones.
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda editar promociones asociadas a las tiendas que a su vez están relacionadas con algún comercio del cual él es dueño.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:-.**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede editar promociones por tienda, y el sistema solicitara datos como, titulo, fecha de caducidad, imagen y precio.
 - **Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios, tiendas y promociones asociadas.
 - **Post-condición:** El propietario podrá visualizar la promoción con la información actualizada.
 - **Puntos de Extensión**
 - **Include:-.**
 - **Extend:-.**

- **Identificador y nombre:** Eliminar Comercios.
 - **Breve Descripción:** permite que el *propietario* pueda eliminar comercios relacionados a él.
 - **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario: -.**
 - **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede eliminar alguno de sus

comercios, y el sistema mostrara información relacionada con la operación realizada (exitosa, fallida).

▪ **Flujos Alternativos:-**

- **Requerimientos Especiales:-**
- **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios relacionados a él.
- **Post-condición:** El propietario habrá eliminado el comercio y toda la información de promociones y tiendas relacionadas a dicho comercio.
- **Puntos de Extensión**
 - **Include:-**
 - **Extend:-**

• **Identificador y nombre:** Eliminar Tiendas.

- **Breve Descripción:** permite que el *propietario* de algún comercio pueda eliminar alguna tienda asociada a los comercios de los cuales él es dueño.
- **Actores:**
 - **Actor Primario:** *Propietario*.
 - **Actor Secundario:** -.
- **Flujo de Eventos:**
 - **Flujo Básico:** El *propietario* puede eliminar alguna tienda seleccionada, y el sistema mostrara información relacionada con la operación realizada (exitosa, fallida).
 - **Flujos Alternativos:-**
- **Requerimientos Especiales:-**
- **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios y tiendas asociadas.
- **Post-condición:** El propietario habrá eliminado la tienda y toda la información de promociones relacionadas a dicha tienda.
- **Puntos de Extensión**
 - **Include:-**

▪Extend:-.

- **Identificador y nombre:** Eliminar Promociones
 - **Breve Descripción:** permite que el *propietario* de algún comercio pueda eliminar alguna promoción asociada a una tienda que a su vez está relacionada con algún comercio del cual él es dueño.
 - **Actores:**
 - Actor Primario:** *Propietario*.
 - Actor Secundario:** -.
 - **Flujo de Eventos:**
 - Flujo Básico:** El *propietario* puede eliminar alguna promoción seleccionada, y el sistema mostrara información relacionada con la operación realizada (exitosa, fallida).
 - Flujos Alternativos:-.**
 - **Requerimientos Especiales:-.**
 - **Pre-condición:** El usuario debe haber iniciado sesión en el sistema y debe tener comercios, tiendas y promociones asociadas.
 - **Post-condición:** El propietario habrá toda la información de la promoción seleccionada.
 - **Puntos de Extensión**
 - Include:-.**
 - Extend:-.**
- **Identificador y nombre:** Iniciar Sesión.
 - **Breve Descripción:** permite que un usuario pueda ingresar al portal administrativo.
 - **Actores:**
 - Actor Primario:** *Propietario*.
 - Actor Secundario:-.**
 - **Flujo de Eventos:**

▪ **Flujo Básico:** El *propietario* ingresa correo y contraseña, y el sistema validara si la combinación de datos ingresados es correcta y en ese caso dará ingreso al portal y caso contrario mostrara un error.

▪ **Flujos Alternativos:-.**

- **Requerimientos Especiales:-.**
- **Pre-condición:** El usuario debe estar pre registrado por un usuario administrador.
- **Post-condición:** El propietario estará dentro del portal y podrá tener acceso a las funcionalidades asociadas a su rol.
- **Puntos de Extensión**
 - **Include:-.**
 - **Extend:-.**

4.4.3. Diagrama de clases

En la ilustración 28 se presenta el diagrama de clases de la aplicación, en este se muestran las relaciones entre las diferentes entidades y roles que se definieron dentro del sistema, así como también las relaciones y métodos principales para cada entidad definida. Este diagrama estatico describe la estructuta del sistema incluyendo de esta manera los objetos que percibe el usuario.

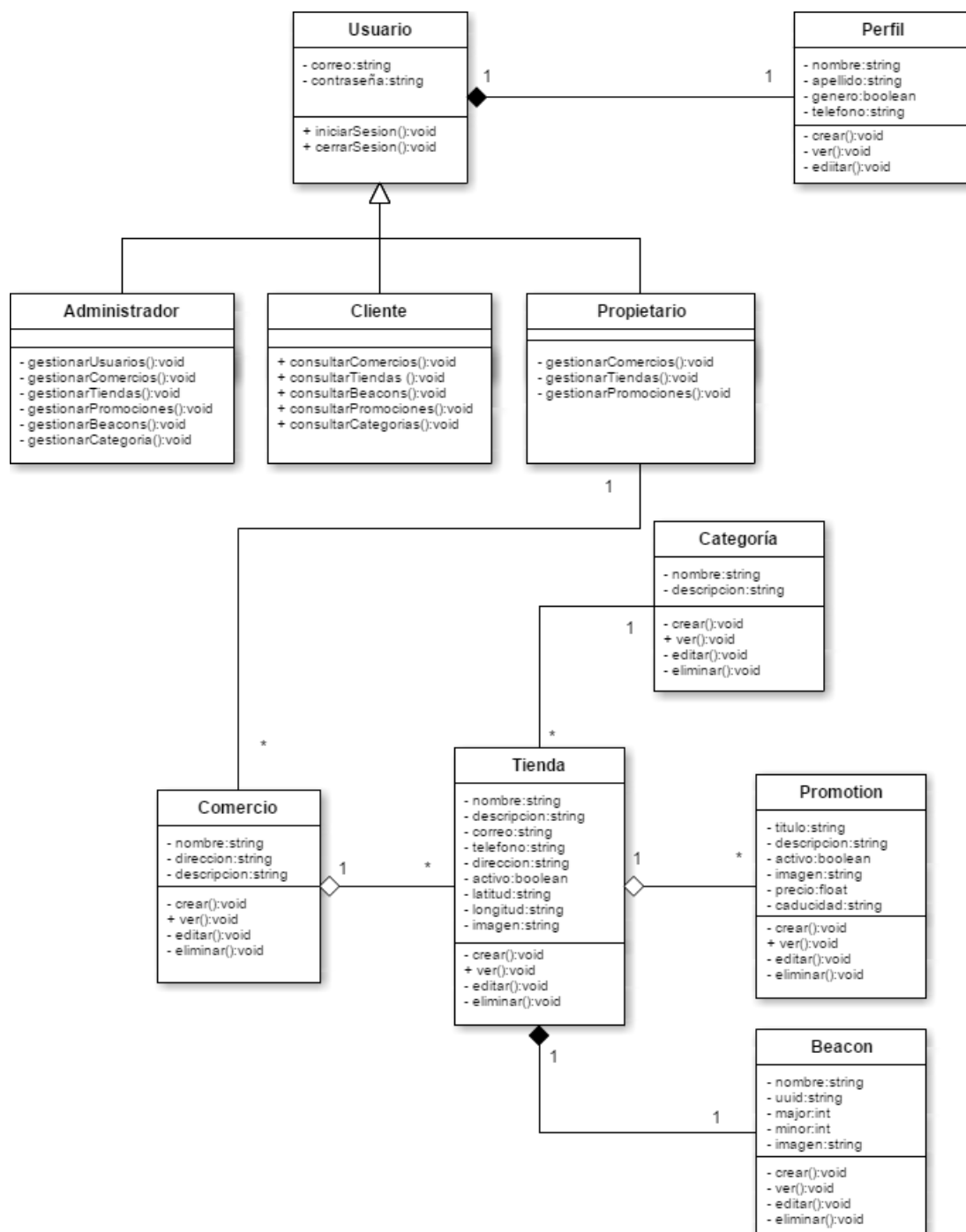


Ilustración 28 - Diagrama de clases de la solución.
Fuente: Los autores

definido en el controlador solo maneja respuestas en formato JSON, las cuales serán generadas a través del componente jbuilder de Rails para establecer la estructura de la respuesta.

4.5.1. Estructura de directorios

Para el correcto funcionamiento del API en Rails es necesario mantener una jerarquía de directorios. Luego de definir el ámbito como se observa en la Ilustración 30, es necesario mantener esta misma jerarquía en la directorio *controllers* (Ilustración 31), que es donde se definen los controladores que a su vez llevaran a cabo la ejecución de los métodos asociados a las peticiones y en el directorio *views* que es donde se definen las plantillas utilizando la sintaxis de jbuilder para retornar la estructura correspondiente al método del controlador ejecutado en formato JSON (Ilustración 32).

```
36 namespace "api" do
37   resources :commerces
38   resources :stores
39   resources :promotions
40   resources :beacons
41   resources :categories
42 end
```

Ilustración 30 - Definición de ámbito API.
Fuente: Los Autores

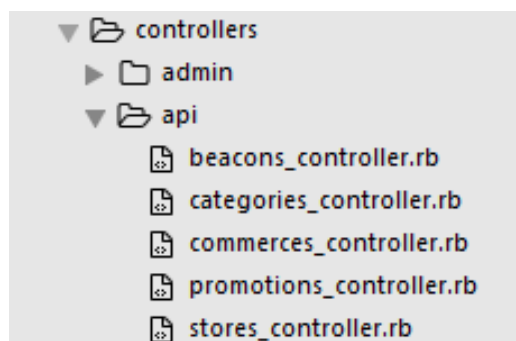


Ilustración 31 - Directorio de controladores de API.
Fuente: Los Autores

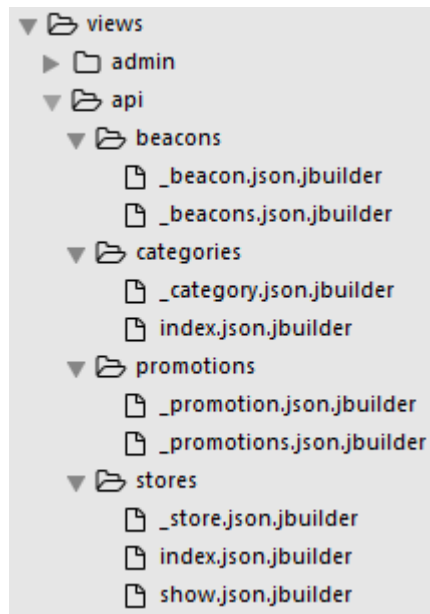


Ilustración 32 - Directorio de vistas de API.
Fuente: Los Autores

4.5.2. Formato de respuesta

JSON (JavaScript Object Notation) es un formato de texto ligero para el intercambio de datos. En la actualidad este formato está siendo utilizado cada vez más, esto debido a la flexibilidad que posee y sobre todo por la facilidad que tiene para interpretarse. Jbuilder es un componente de Rails que permite definir la estructura en forma de plantilla de una respuesta en formato JSON, utilizando una sintaxis específica que permite leer los valores de las variables de Ruby, las cuales contienen datos que se consultan en el método del controlador asociado a la petición. Se puede ver la estructura de este formato de texto ligero JSON en la Ilustración 33.

```

1 [{
2   "shopID": 1,
3   "isActive": true,
4   "picture": "http://sm.uploads.im/t/j08ug.png",
5   "name": "Arturos's SB",
6   "categoryId": 1,
7   "category": "Restaurants",
8   "company": "Arturo's",
9   "email": "arturos@arturos.com",
10  "phone": "+58 (4142) 430-962_",
11  "address": "Sabana Grande, Caracas, Distrito Capital, Venezuela",
12  "about": "Esta es la sucursal mas antigua",
13  "registered": "2016-08-18T00:19:13.000Z",
14  "latitude": "10.494478106391837",
15  "longitude": "-66.8772386533903",
16  "promotions": [{
17    "id": 2,
18    "title": "Chamo Pack por tan solo 2.000 bs",
19    "about": "Ahora puedes llevar tu chamo pack por tan solo 2.000bs",
20    "isActive": true,
21    "picture": "http://sk.uploads.im/t/u04ZR.jpg",
22    "price": 2000.0,

```

Ilustración 33 - Estructura de API Restful, tiendas.**Fuente: Los Autores**

4.5.3. Documentación del API

Para efectos de este prototipo este api solo estará conformado por métodos de consultas, los cuales serán invocados utilizando el método http GET, a través de las rutas ya definidas en la Tabla 13.

Tabla 13 - Documentación API

Método	Ruta	Descripción
GET	/api/commerces.json	obtiene todos los comercios en formato Json
GET	/api/commerces/:id.json	obtiene un comercio especifico en formato json
GET	/api/stores.json	obtiene todos las tiendas en formato json
GET	/api/stores/:id.json	obtiene una tienda

		especifica en formato json
GET	/api/promotions.json	obtiene todos las promociones en formato json
GET	/api/promotions/:id.json	obtiene una promoción especifica en formato json
GET	/api/beacons.json	obtiene todos los beacons en formato json
GET	/api/beacons/:id.json	obtiene un beacon especifico en formato json
GET	/api/categories.json	obtiene todos las categorias en formato json
GET	/api/categories/:id.json	obtiene una categoría especifico en formato json

Fuente: Los Autores

4.6. Implementacion y diseño de prototipo de aplicación web

Rails cuenta con un ORM (Object Relational Mapping) bastante robusto, el cual permite que exista una abstracción entre la aplicación y el acceso a los datos. Al incorporar esta capa de acceso, Rails ofrece la flexibilidad de que se puedan utilizar distintos manejadores de base de datos y que a nivel de aplicación se mantenga utilizando la misma sintaxis para el acceso a los datos.

4.6.1. Uso del framework Ruby on Rails

La motivación de usar Rails para realizar la aplicación web y el api es que al momento de implementar la arquitectura MVC (Modelo Vista Controlador) y definir métodos dentro de controladores de la aplicación se pueden estructurar distintos formatos de respuesta dentro de un mismo método, es decir, se pueden responder con distintos formatos de respuesta para una misma petición, lo cual evita crear bloques de código similares para cada tipo de respuesta que sea solicitada y de esta manera cumplir con la filosofía de reutilización de componentes.

4.6.2. Bootstrap y Ruby on Rails

Bootstrap es un framework que facilita el desarrollo de diseños de una página web o una aplicación web. El mismo esta predefinido con clases CSS para la creación de componentes comunes y de esta manera generar una plantillas. La integración en Ruby on Rails se logra de diferentes maneras, para el proyecto se utilizó a integración por etiqueta link dentro del código HTML haciendo referencia a el archivo css del bootstrap como se observa en la Ilustracion 34.

```
<!-- Bootstrap 3.3.5 -->
<link rel="stylesheet" href="/bootstrap/css/bootstrap.min.css">
<!-- Font Awesome -->
```

Ilustración 34 - Integración Bootstrap.
Fuente: Los Autores

En la Ilustración 35 se puede ver una sección donde se hace uso de componentes de Bootstrap.

```
<div class="row">
  <div class="col-xs-12">

    <div class="box">
      <div class="box-header">
        <h3 class="box-title">Lista de Promociones</h3>
      </div><!-- /.box-header -->
      <div class="box-body">
        <table id="promotions_table" class="display table table-bordered table-striped" cellspacing="0" width="100%">
          <thead>
            <tr>
              <th>Código</th>
              <th>Titulo</th>
              <th>Precio</th>
              <th>Vence</th>
            </tr>
          </thead>
          <tbody>
            <% @promotions.each_with_index do |promotion,index| %>
              <tr class="detail-inside link <%=promotion.active ? 'tr-active' : 'tr-inactive' %>" data-toggle="modal" data-target="#modal-promotion-<%=promotion.id.to_s%>" data-url="#<%=admin_promotion_path(promotion)%>">
                <td><%= promotion.id %></td>
                <td><%= promotion.title %></td>
                <td class="money"><%= promotion.price %></td>
                <td><%= promotion.until %></td>
```

Ilustración 35 - Ejemplo Bootstrap.
Fuente: Los Autores

4.6.3. PostgreSQL y Ruby on Rails

PostgreSQL es una base de datos de código abierto. Rails no restringe el uso de diferentes manejadores, por defecto posee MySQL pero es muy sencilla la adaptación a PostgreSQL. Simplemente se definen las credenciales en el archivo *database.yml* ubicado en la carpeta */config* del proyecto.

```
##
default: &default
  adapter: postgresql
  encoding: unicode
  # For details on connection pooling, see rails configuration guide
  # http://guides.rubyonrails.org/configuring.html#database-pooling
  pool: 5
  username: postgres
  password: prueba$1

development:
  <<: *default
  database: AdminProMotion_development
```

Ilustración 36 - Configuración de manejador PostgreSQL.
Fuente: Los Autores

4.6.4. Desarrollo en Ruby on Rails

El código de la aplicación consiste en un conjunto de vistas, modelos y controladores, haciendo así que las vistas representen cada pantalla en un archivo HTML que define la disposición, la forma y color de los elementos de interfaz de usuario como botones, campos de datos y títulos. Dicho esto, a continuación se explica por pantalla y por funcionalidad el flujo del prototipo funcional de aplicación web desarrollada en un manual de usuario para el uso de la aplicación que se adjunta en los anexos del trabajo de grado. [Manual de usuario de app web.](#)

4.7. Implementacion y diseño de prototipo funcional de aplicación móvil

La aplicación sobre Android se desarrolló en su totalidad usando el lenguaje de programación Java. Las funcionalidades que corresponden al establecimiento de la comunicación con el servidor de llamadas fueron proporcionadas por la librería *RetroFit* basada en el *framework* Android Studio, siendo esta una de las mejores formas de consumo de servicios web de manera asíncrona ya sea a través de un arreglo o un objeto definido previamente. Siguiendo el paradigma de programación de la aplicación de Android se crearon las *Activities* necesarias y el flujo entre estas para cubrir cada funcionalidad, desde la consulta de promociones y el registro de localización de cada tienda hasta la búsqueda definida por nombre de tienda. Para cada *Activity* se programó la lógica necesaria que cubre con las necesidades de los requerimientos y la funcionalidad que la misma expone.

4.7.1. Uso del IDE Android Studio

Entorno de desarrollo integrado que permite la construcción de aplicaciones móviles para dispositivos que trabajen con sistema operativo Android. Basado en el lenguaje de programación Java y en IntelliJ IDEA. Posee un sistema de compilación flexible basado en gradle, emulador rápido con múltiples funciones, entorno unificado en el que se logra desarrollar para todos los dispositivos Android, compatibilidad con C++ y NDK, Soporte de Google Cloud Platform integrado.

4.7.2. Consultas al API rest desde Android

La consulta y extracción de data del API Restful creado en Ruby on Rails a través de la librería Jbuilder, es accedido y mapeado como un servicio web asíncrono definido en Android Studio gracias a la librería *Retrofit*.

Retrofit es una librería de redes (Cliente Rest) desarrollada por *Square*, a través de ella se puede capturar respuestas JSON que son enviadas desde la Web. Gracias a esto no es necesario parsear el JSON para consultar la data. *Retrofit* literalmente compone todas las características requeridas por un web Service. No es necesario un objeto *AsyncTask*, *HttpURLConnection* o *Json Parser*.

Uno de los requisitos indispensables para poder acceder a la librería de *Retrofit* es colocar la dependencia de la librería en el archivo *build.gradle* como se observa en la Ilustración 37.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:23.4.0'
    compile 'com.android.support:design:23.4.0'
    compile 'com.estimote:sdk:0.10.+@aar'
    compile 'com.squareup.retrofit:retrofit:2.0.0-beta2'
    compile 'com.google.code.gson:gson:1.7.2'
    compile 'com.squareup.retrofit:converter-gson:2.0.0-beta2'
    compile 'com.squareup.okhttp:okhttp:2.4.0'
    compile 'com.loopj.android:android-async-http:1.4.9'
    compile 'com.github.snowdream.android:smartimageview:0.0.2'
```

Ilustración 37 - Dependencia Retrofit Square.

Fuente: Los Autores

Para poder obtener la data del servicio web involucrado es necesario definir 2 principales clases, una clase objeto (Ilustración 39) en la cual estará la estructura de respuesta del servicio y una clase interfaz (Ilustración 38) la cual servirá de medio de comunicación entre la clase objeto y el mapeo que realiza la librería *Retrofit* contra el servicio *Rest* y en ella se define si el servicio será llamado síncrono o asíncrono.

```

public interface PromotionsObjectAPI {

    /*
     * Retrofit get annotation with our URL
     * And our method that will return us details of student.
     */
    @GET("api/RetrofitAndroidObjectResponse")
    Call<Promotion> getPromotionDetails();

}

```

Ilustración 38 - Definición de clase interfaz para detalle de promociones.
Fuente: Los Autores

```

/**
 * Created by Edwin Alvarado on 10-08-2016.
 */
public class Promotion {

    private String id;
    private String title;
    private String about;
    private Boolean isActive;
    private String picture;
    private Float price;
    private String endDate;

    public Promotion(String id, String title, String about, .
        this.id = id;

```

Ilustración 39 - Clase Objeto (Promoción).
Fuente: Los Autores

Luego se definió la URL del servicio al que se accederá. Separando a su vez el host del URL como observamos en la Ilustración 40 y el método de este servicio visto en la Ilustración 41.

```

static String urlPost = "http://promotion-admin.herokuapp.com";
//static String urlStores = "http://200.74.223.98:8001/";
static String urlStores = "http://promotion-admin.herokuapp.com/";

```

Ilustración 40 - Asignación de host de servicio.
Fuente: Los Autores

```

    @GET("/api/stores.json")
    Call<List<Store>> getStoresDetails();

```

Ilustración 41 - Asignación de URL de servicio.

Fuente: Los Autores

Y por último hacer el llamado del servicio usando inicialmente la primitiva *Retrofit.Builder* para la construcción del *request* del mismo (Ilustración 42).

```

public static void getRetrofitArray(final MyApplication app) {

    Retrofit retrofit = new Retrofit.Builder()
        .baseUrl(urlPost)
        .addConverterFactory(GsonConverterFactory.create())
        .build();

    RetrofitArrayAPI service = retrofit.create(RetrofitArrayAPI.class);

    Call<List<Post>> call = service.getPostDetails();

    call.enqueue(new Callback<List<Post>>() {
        @Override
        public void onResponse(Response<List<Post>> response, Retrofit retrofit) {

```

Ilustración 42 - Llamar Servicio con Retrofit.

Fuente: Los Autores

4.7.3. Envío de notificaciones – Tecnología Beacon

El envío de notificaciones en el prototipo funcional de aplicación móvil se enfoca al despliegue de las promociones asociadas a los comercios afiliados al sistema que se encuentren dentro del rango configurado para cada dispositivo *Beacon*. Para el envío estas notificaciones y el registro de beacon's, se debe poseer como requisito inicial la cuenta asociada y la librería Estimote descargada de la página oficial del proveedor de los dispositivos Beacon (Estimote. Sitio: <https://cloud.estimote.com>). Esta librería debe ser importada al proyecto Android,

para de esta manera hacer uso de las primitivas y de los métodos de desarrollo de *Estimote Beacon*, esta librería está compuesta por tres clases principales

AppBaseActivity: encargada del envío y administración de notificaciones en segundo plano, se observa el servicio levantado en la Ilustración 44.

BeaconID: clase que define el objeto beacon, con sus características principales y métodos de consulta a nivel funcional.

BeaconNotificationsManager: la cual se enfoca en la inicialización y manejo tanto de beacons asociados, como de notificaciones push. Esta es la encargada de realizar el envío de notificación al momento que se ejecute el evento de ingreso en zona configurada.

En la Ilustración 43 se observan estas tres (3) clases.

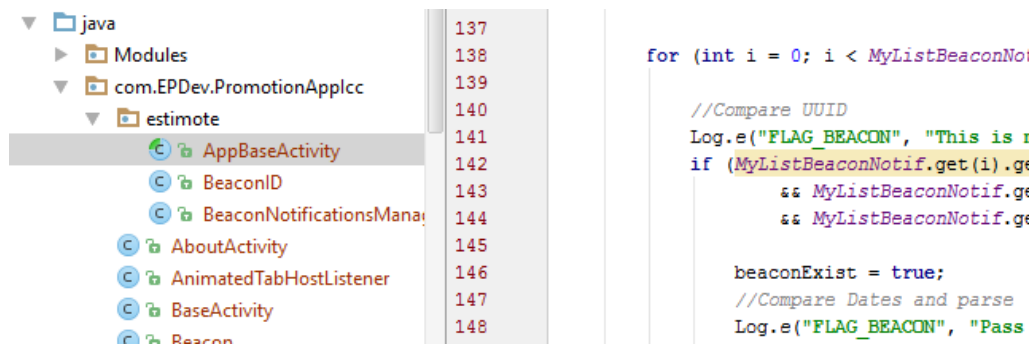


Ilustración 43 - Librería Estimote.
Fuente: Los Autores

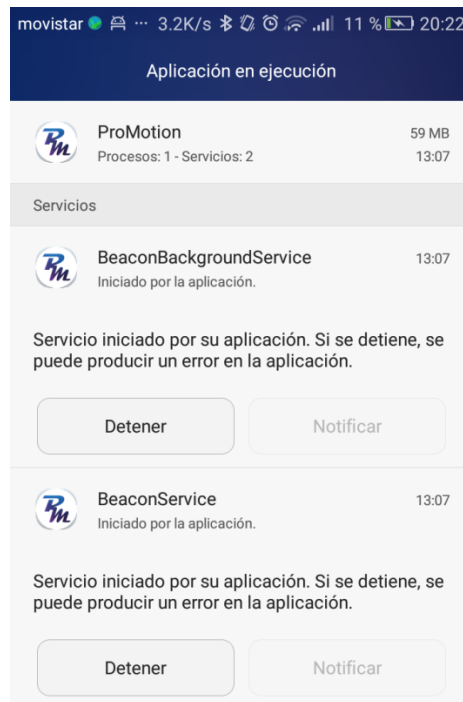


Ilustración 44 - Servicios en segundo plano.
Fuente: Los Autores

El mapeo entre dispositivo beacon (Ubicado físicamente en la locación del comercio) y la promoción asociada (Consumida desde *API Restful*) se hace a través de una serie de métodos propios de la librería *Estimote Beacon* de esta manera se certifica la correcta comunicación con estos dispositivos tecnológicos. Teniendo así no solo el registro del beacon sino su monitoreo y recepción de señal así la aplicación se encuentre en segundo plano. En la Ilustracion 45 se denota la inicialización.

```
Log.d(TAG, "INIT Beacon Initialize, UUID: " + UUID + " , Mayor: " + Mayor + " , Minor: " + Minor);
BeaconNotificationsManager beaconNotificationsManager = new BeaconNotificationsManager(this);
beaconNotificationsManager.addNotification(
    new BeaconID(UUID, Integer.parseInt(Mayor), Integer.parseInt(Minor)),
    StoresData.getStoresData().get(i).getPromotions().get(j).getTitle(), StoresData.getStoresData().get(i).getPromotions().get(j).getId(),
    StoresData.getStoresData().get(i).getShopID());

beaconNotificationsManager.startMonitoring(StoresData.getStoresData());
Log.d(TAG, "END Beacon Initialize");
```

Ilustración 45 - Inicialización e inicio de monitoreo dispositivo Beacon.
Fuente: Los Autores

Gracias a que el mapeo del dispositivo Beacon contra la promoción asociada es ejecutado y definido del lado del cliente, se tiene la libertad de trabajar la lógica de notificaciones dirigida específicamente a los requerimientos y necesidades de clientes en específico, esto quiere decir que dependiendo del usuario se puede filtrar las notificaciones que se le desplegaran en la aplicación, ver Ilustración 47. Una vez que se posee el dispositivo beacon inicializado y la aplicación registra en ingreso en la región configurada de dicho *beacon* se generan la construcción de la misma como se observa en la Ilustración 46.

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(context)
    .setSmallIcon(R.mipmap.ic_launcher)
    .setContentTitle(Tittle)
    .setContentText(message)
    .setDefaults(NotificationCompat.DEFAULT_ALL)
    .setPriority(NotificationCompat.PRIORITY_HIGH)
    .setContentIntent(resultPendingIntent)
    .setExtras(bundle)
    .setWhen(System.currentTimeMillis()).setTicker(Tittle)
    .setAutoCancel(true);

NotificationManager notificationManager =
    (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
//notificationManager.notify(1, builder.build());

//notificationManager.notify(random.nextInt(100), builder.build());
notificationManager.notify(random.nextInt(1000), builder.build());
```

Ilustración 46 - Construcción y envío de notificación.
Fuente: Los Autores

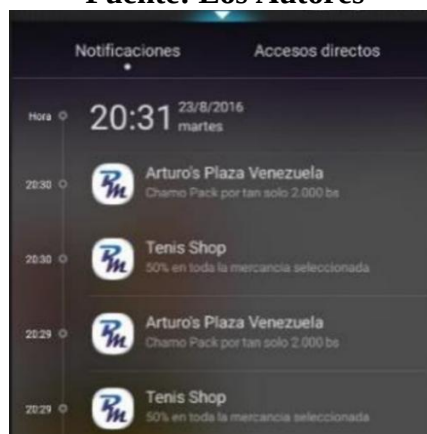


Ilustración 47 - Despliegue de notificación en barra superior del dispositivo.
Fuente: Los Autores

4.7.4. Desarrollo en Java/Android

El código de la aplicación consiste en un conjunto de actividades que representan cada pantalla junto a un archivo XML que define la disposición, de la forma y color de los elementos de interfaz de usuario como botones, campos de datos y títulos. Dicho esto, a continuación se explica por pantalla y por funcionalidad el flujo del prototipo funcional de aplicación móvil desarrollada en un manual de usuario para el uso de la aplicación que se encuentra en los anexos del trabajo de grado. [Manual de usuario móvil.](#)

4.8. Pruebas funcionales y resultados

Según Oré (2009), se denominan pruebas funcionales a las pruebas de software que tienen por objeto probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de calidad con apoyo de algunos usuarios finales de la aplicación, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción. También se les denomina pruebas de comportamiento o pruebas caja negra, ya que los analistas se enfocan en el análisis de los datos de entrada y su correspondencia contra los parámetros de salida.

4.8.1. Pruebas de carga y rendimiento

En las pruebas de carga y rendimiento se miden diferentes factores que implican el funcionamiento del prototipo funcional de aplicación tales como tráfico de red, consumo de CPU, memoria, espacio de almacenamiento (tanto cache, como física), entre otros. Blazermeter es una herramienta propia de Jmeter donde se graba un script de conexión, configurando un proxy en la red del dispositivo.

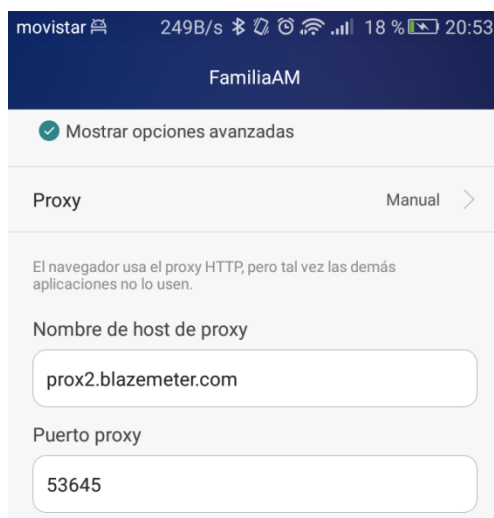


Ilustración 48 - Configuración de proxy.
Fuente: Los Autores

Una vez configurado el proxy visto en la Ilustración 48, se accedió a la aplicación y se ejecutaron las funcionalidades que impliquen accesos a servicios vistas en la Ilustración 49, consulta de tiendas y promociones, obtención de imágenes alojadas en servidores dedicados, entre otros.

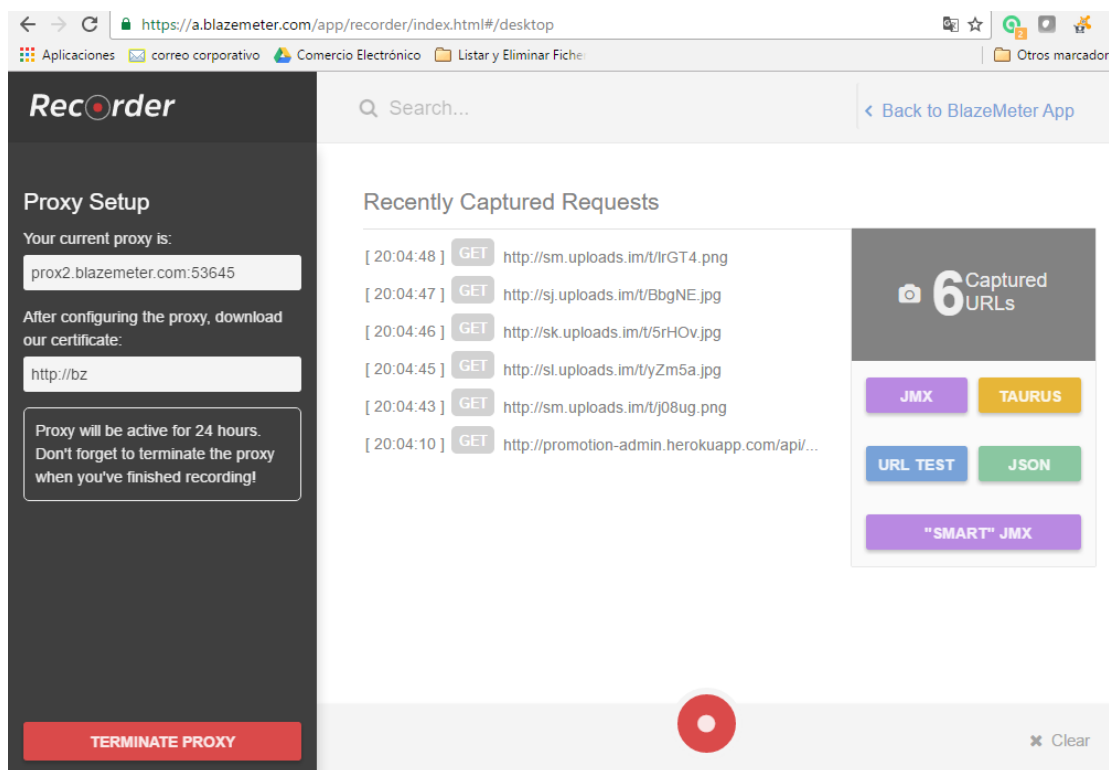


Ilustración 49 - Captura de peticiones Blazemeter.
Fuente: Los Autores

Al culminar el proceso de grabado se procedió a descargar el archivo jmx para su visualización en la herramienta Jmeter, haciendo clic en el botón de color morado, al tener el archivo “Recording_09_26,_2016,_12-08_am.jmx” se importa en Jmeter. Y se extrae la data de tráfico de servicios y programación de usuarios. Principalmente se genera un plan de pruebas mostrado en la Ilustracion 50 y estas pruebas se realizaron con 50 usuarios que acceden al mismo instante en la aplicación y los resultados fueron los siguientes:

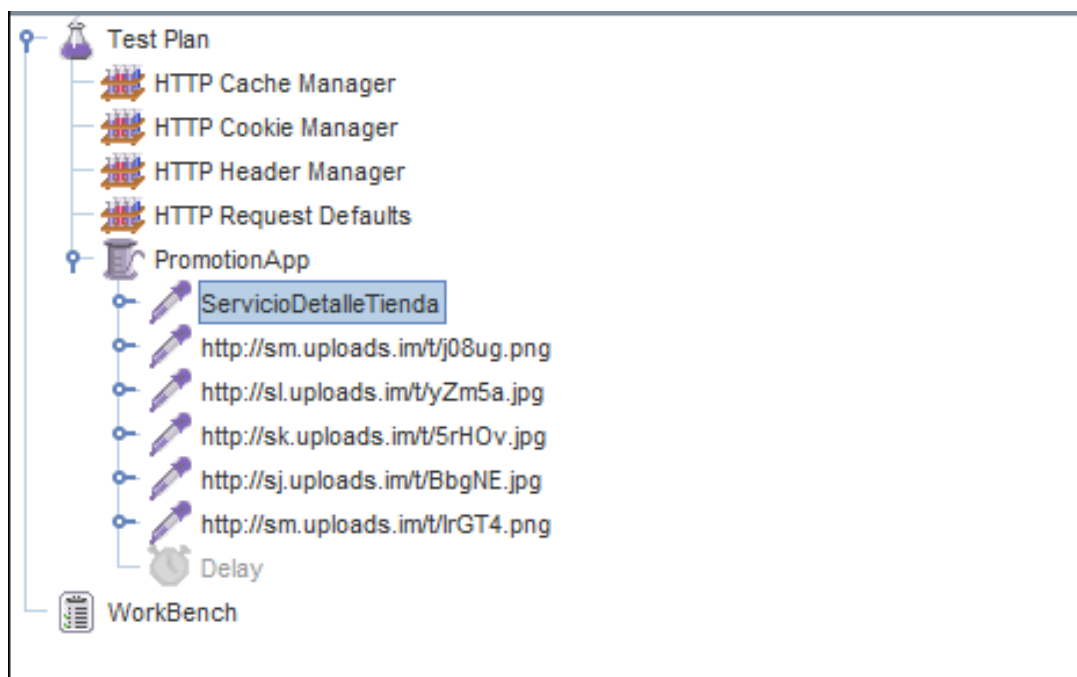


Ilustración 50 - Definición de plan de prueba.

Fuente: Los Autores

Los resultados de datos por servicio que incluye usuarios, promedio, media, porcentaje de error, tiempo de respuesta, entre otros. Esto se visualiza en la Ilustración 51. Estas pruebas fueron realizadas con cincuenta (50) usuarios virtuales que simulaban el acceso o consultas a estos servicios. Teniendo como datos de respuesta lo siguiente:

- **Grupo de Hilos (*Thread Group*):** Representan a los usuarios dentro del ambiente de pruebas JMeter. *Thread Group* le indica JMeter el número de usuarios que desea simular y con qué frecuencia los usuarios deben enviar las solicitudes.

- **Transacción:** Proceso que implica la llamada al servicio web, y la respuesta que se obtiene de dicha llamada.

- **Script:** Consiste en un archivo en el que se registran los procesos, acciones, llamadas e interacciones de un usuario real en un flujo de la aplicación.

- **Listener:** Mecanismo de visualización de resultados almacenados en pruebas.

- **#Sample:** Cantidad de hilos (*thread*) utilizados por los usuarios.
- **Average:** Tiempo promedio en milisegundos de la solicitud para un conjunto de hilos.
- **Median:** Valor en tiempo del percentil 50.
- **90% line:** Máximo tiempo utilizado por el 90% de la muestra.
- **95% line:** Máximo tiempo utilizado por el 95% de la muestra.
- **99% line:** Máximo tiempo utilizado por el 99% de la muestra.
- **%Error:** Porcentaje de requerimientos con errores
- **Min:** Tiempo mínimo que demora un hilo en acceder al objeto prueba.
- **Max:** Tiempo máximo que demora un hilo en acceder al objeto prueba.
- **Throughput:** Rendimiento medido en los requerimiento por seg/min/hr.
- **Kb/sec:** Rendimiento medido en Kbytes por segundo.
- **Windows Resources:** Muestra un resumen de los recursos del sistema Windows que fue usado como servidor. Mostrando indicadores del porcentaje de tiempo de Disco Duro, porcentaje de tiempo de Procesador y memoria disponible (MB).

Aggregate Graph				
Name: Aggregate Graph				
Comments:				
Write results to file / Read from file				
Filename				
Label	# Samples	Average	Median	
ServicioDetalleTienda	50	6019	1724	
http://sm.uploads.im/tj08ug.png	50	18854	16278	
http://sl.uploads.im/tyZm5a.jpg	50	5963	2194	
http://sk.uploads.im/t5rHOv.jpg	50	1838	1258	
http://sj.uploads.im/VBbgNE.jpg	50	1573	1026	
http://sm.uploads.im/MrGT4.png	50	2713	0	
TOTAL	300	6160	1790	
90% Line		95% Line		99% Line
10343		11831		133543
21253		21771		130675
7384		11146		131138
3632		3785		4373
3981		4264		5205
1833		3412		63243
15532		19005		122042
Min	Max	Error %	Throughput	KB/sec
858	133543	2.00%	22.5/min	1.7
1186	130675	10.00%	22.4/min	68.0
571	131138	14.00%	21.9/min	8.2
349	4373	4.00%	22.4/min	4.4
534	5205	0.00%	23.1/min	9.2
0	63243	84.00%	23.9/min	9.1
0	133543	19.00%	2.1/sec	93.1

Ilustración 51 - Tabla resultado, tráfico de red (50 usuarios).

Fuente: Los Autores

De esta manera se concluye que el acceso y consumo a los servicios RestFul se considera un proceso eficaz, que emplea poco tiempo y no aumenta el rendimiento normal del aplicativo. Una de las grandes razones por las cuales el tráfico de red no es alto, es que se realizó un enfoque de desarrollo SPA (*Single Page Application*) en la cual la data utilizada por la aplicación se solicita al inicio de la misma, los servicios definidos se encuentran en formato Json y este es la razón de la rapidez y fluidez de los mismos.

- Uso y carga de CPU

El uso y carga de CPU en el prototipo funcional de aplicación y los componentes que se utilizan en la misma son medidos a través de una herramienta que integra el kit de desarrollo de software (Android SDK), el nombre de dicha herramienta es *Monitor.bat*, en el mismo se registra el log de transacciones del dispositivo así como también la carga tanto de CPU como de memoria, a continuación se ven los porcentajes de uso de los componentes que usa actualmente nuestro prototipo funcional de aplicación:

Paquete: com.android.bluetooth (usuario) (0.3%). Se observa en la Ilustracion 52

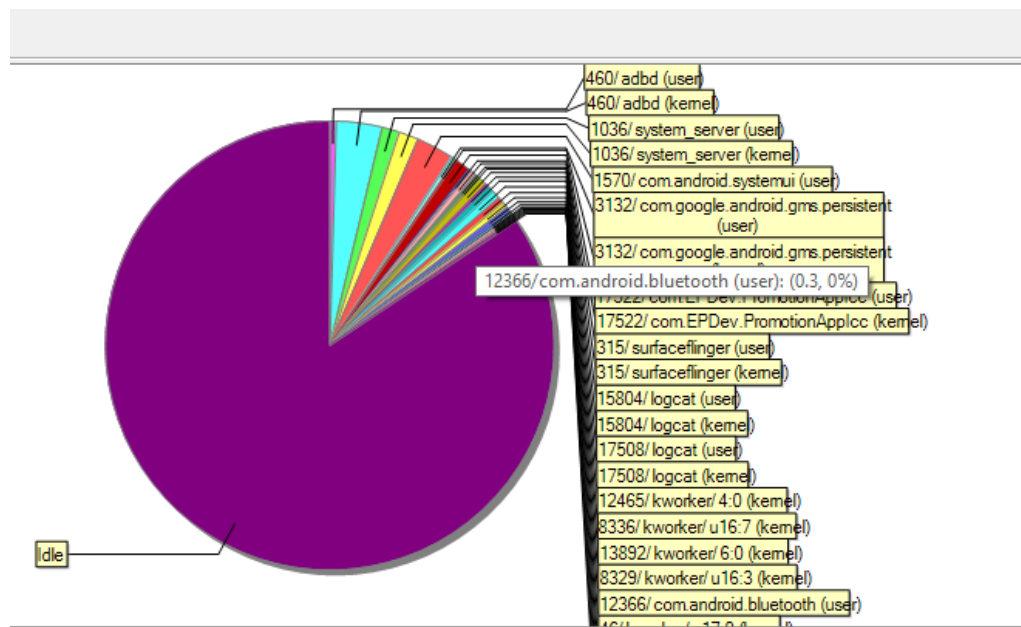


Ilustración 52 - Carga de CPU - Bluetooth.
Fuente: Los Autores

Paquete: com.EPDev.PromotionAppIcc(usuario)(1%). Se observa en la Ilustracion 53

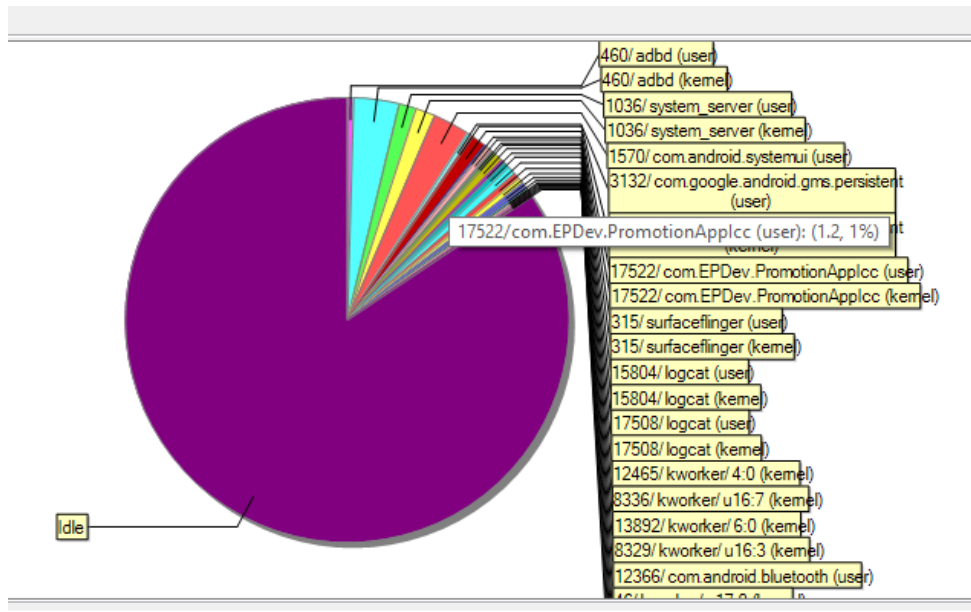


Ilustración 53 - Carga de CPU - Prototipo ProMotion (Usuario).
Fuente: Los Autores.

Paquete: com.EPDev.PromotionAppIcc(kernel)(0,2%). Se observa en la Ilustracion 54

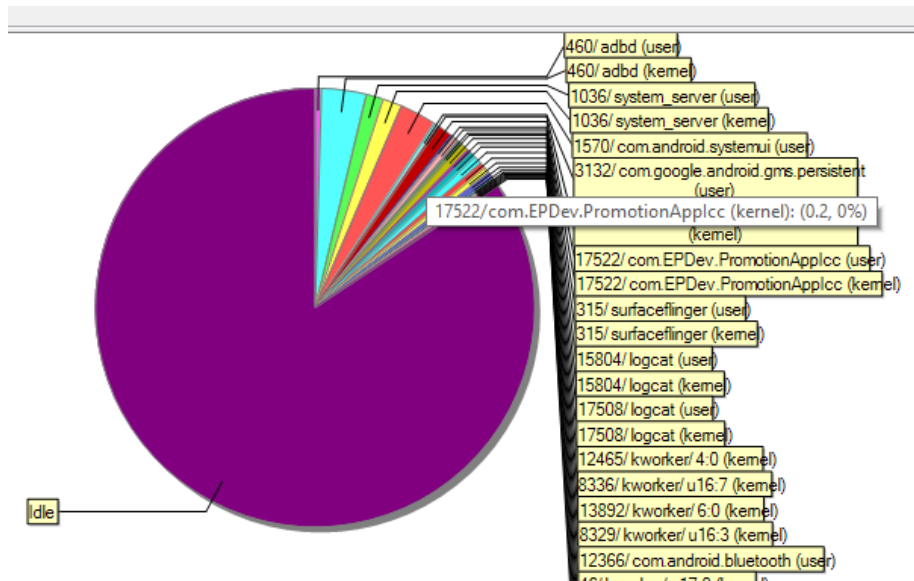


Ilustración 54 - Carga de CPU - Prototipo ProMotion (kernel).
Fuente: Los Autores

Se concluye de las graficas (tortas) anteriores que el porcentaje de uso del procesador de la aplicación es bajo, esto quiere decir que la aplicación va a poseer funcionamiento fluido y sin inconvenientes de cierres inesperados o largos tiempos de respuesta para la misma.

4.8.2. Pruebas de aceptación

Se aplicó una encuesta técnica a usuarios finales con conocimientos de computación para conocer su opinión general sobre la usabilidad y funcionamiento de la solución incluyendo así una encuesta para la aplicación móvil que podemos visualizar en la Ilustración 55 y una encuesta para la aplicación web vista en la Ilustración 56. Las encuestas fueron aplicadas a un grupo de diez (10) personas, estudiantes o graduados de computación.

Móvil APP:

Encuesta - Prueba de Aceptación - ProMotion Móvil App

Descarga ya la aplicación de ProMotion app para conocer tu opinión sobre este desarrollo, que fue pensado para tu comodidad y la de tu bolsillo.

En ProMotion app pensamos en ti es por esto que tomamos muy en cuenta tu opinión, navega en la aplicación y disfruta de esta experiencia excepcional.

Link de descarga de Apk (Android): <https://drive.google.com/file/d/0BzhMM0hY5kjZUg0TUtTVZqQ2s/view?usp=sharing>

A continuación se presentaran ocho (8) preguntas, indique según la siguiente escala su opinión:

- (4) Totalmente de acuerdo
- (3) De acuerdo
- (2) Medianamente de acuerdo
- (1) En desacuerdo

* Required

ProMotion App



Ilustración 55 - Encuesta - Prueba de Aceptación App Móvil.
Fuente: Los Autores. Recopilado de: <https://goo.gl/forms/gR2jbAlK5L41Sbb63>

Las preguntas contestadas por los usuarios fueron:

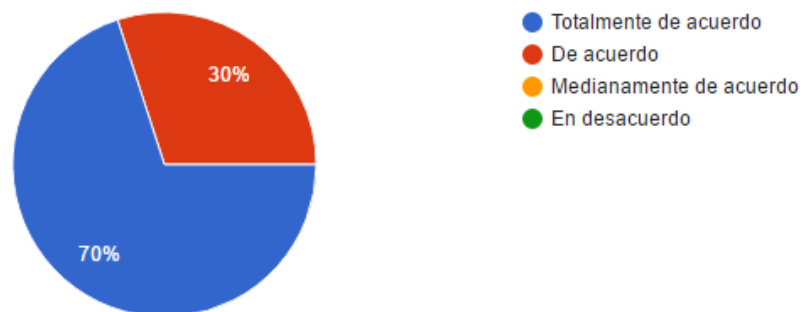
1. ¿Le parece adecuada la estructura de la interfaz de usuario?
2. ¿Se utiliza un buen contraste en la interfaz de usuario?
3. ¿El diseño de la página le parece consistente?
4. ¿De manera general, considera que el sistema es atractivo?
5. ¿Se presenta de manera intuitiva la navegación de la aplicación?
6. ¿Se representa de forma sencilla la funcionalidad para consulta de tiendas y promociones?
7. ¿Se representa de forma sencilla la funcionalidad para consulta de tiendas más cerca, y su ruta?
8. ¿Se representa de forma sencilla la configuración de la aplicación?

Las cuatro opciones para responder a estas preguntas fueron:

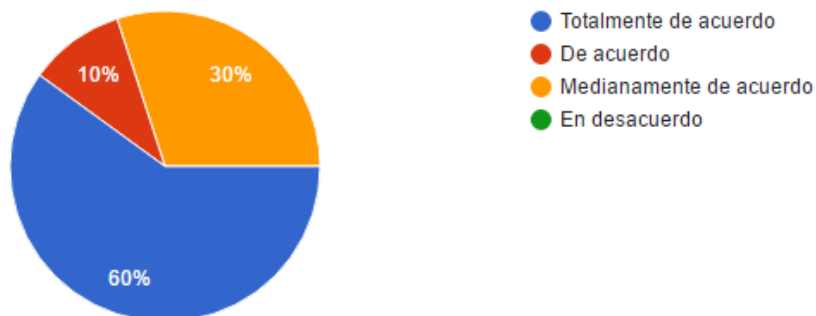
- Totalmente de acuerdo
- De acuerdo
- Medianamente de acuerdo
- En desacuerdo

Los resultados obtenidos fueron los siguientes:

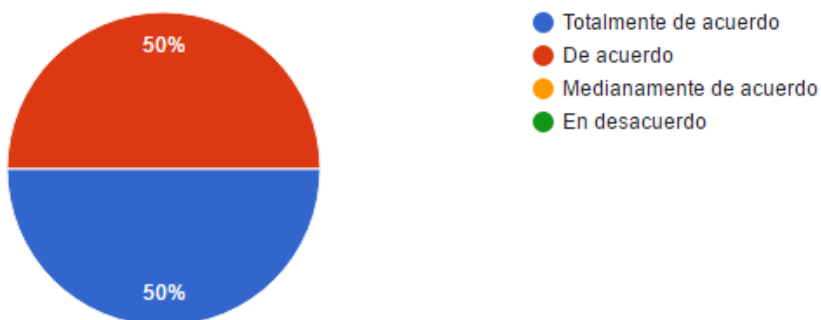
¿Le parece adecuada la estructura de la interfaz de usuario? (10 responses)



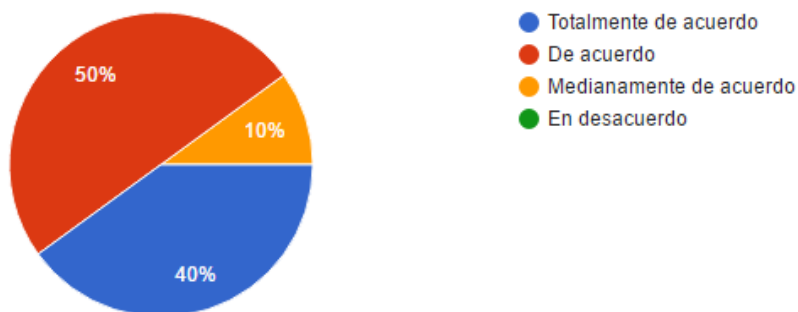
¿Se utiliza un buen contraste en la interfaz de usuario? (10 responses)



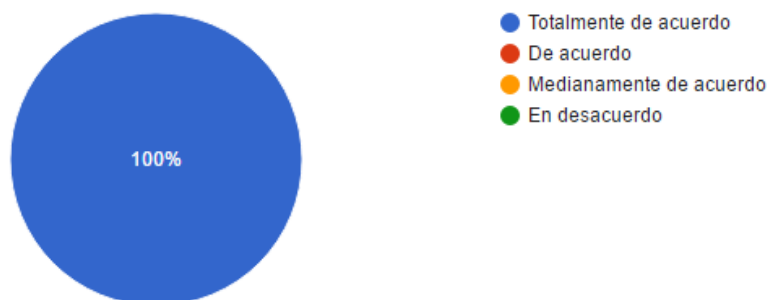
¿El diseño de la aplicación le parece consistente? (10 responses)



¿De manera general, considera que el sistema es interactivo? (10 responses)

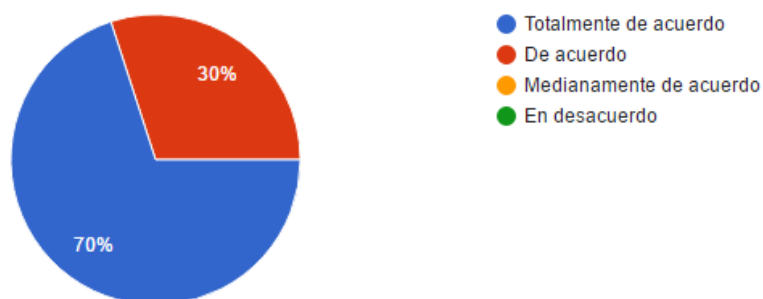


¿Se presenta de manera intuitiva la navegación de la aplicación? (10 responses)



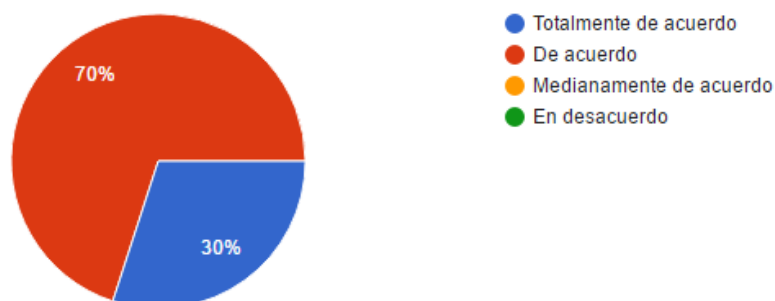
¿Se representa de forma sencilla la funcionalidad para consulta de tiendas y promociones?

(10 responses)

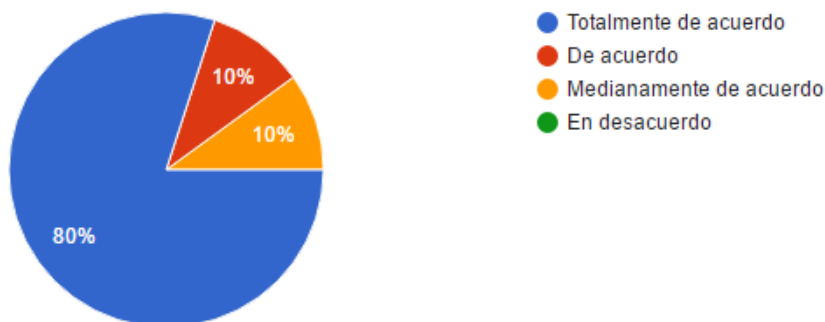


¿Se representa de forma sencilla la funcionalidad para consulta de tiendas mas cerca, y su ruta?

(10 responses)



¿Se representa de forma sencilla la configuración de la aplicación?
(10 responses)



Web APP:

Encuesta - Prueba de Aceptación - ProMotion Web App

Ingresa ya a la aplicación de ProMotion Web app para conocer tu opinión sobre este desarrollo, que fue pensado para tu comodidad y la de tu bolsillo.

En ProMotion app pensamos en ti es por esto que tomamos muy en cuenta tu opinión, navega en el sitio y disfruta de esta experiencia excepcional.

Link de aplicación: <http://promotion-admin.herokuapp.com/>

Credenciales de ingreso:

Correo: edwinalvaradoster@gmail.com

Contraseña: 12345678

A continuación se presentaran ocho (8) preguntas, indique según la siguiente escala su opinión:

- (4) Totalmente de acuerdo
- (3) De acuerdo
- (2) Medianamente de acuerdo
- (1) En desacuerdo

* Required

Ilustración 56 - Encuesta - Prueba de Aceptación App Web.

Fuente: Los Autores. Recopilado de:

<https://goo.gl/forms/g5UWMEdyMq5jckQC3>

Las preguntas contestadas por los usuarios fueron:

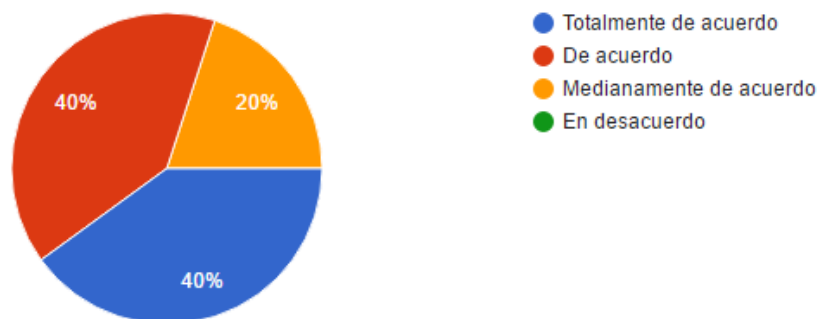
1. ¿Le parece adecuada la estructura de la interfaz de usuario?
2. ¿Se utiliza un buen contraste en la interfaz de usuario?
3. ¿El diseño de la página le parece consistente?
4. ¿De manera general, considera que el sistema es atractivo?
5. ¿Se presenta de manera intuitiva la navegación de la aplicación?
6. ¿Se representa de forma sencilla la funcionalidad para consulta de tiendas y promociones?
7. ¿Se representa de forma sencilla la funcionalidad para el registro de tiendas y promociones?
8. ¿Se representa de forma sencilla los gráficos (Dummy) expresados en la aplicación?

Las cuatro opciones para responder a estas preguntas fueron:

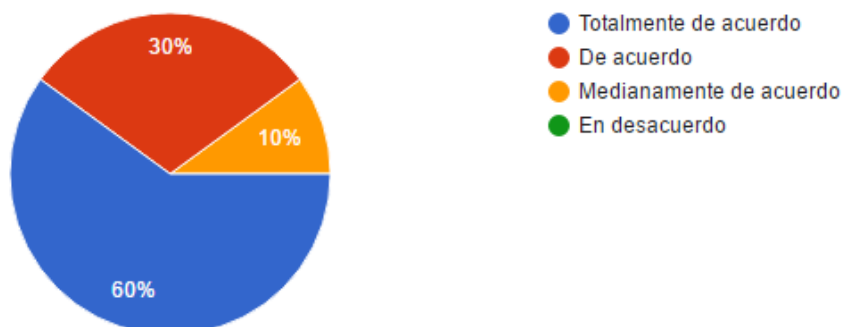
- Totalmente de acuerdo
- De acuerdo
- Medianamente de acuerdo
- En desacuerdo

Los resultados obtenidos fueron los siguientes:

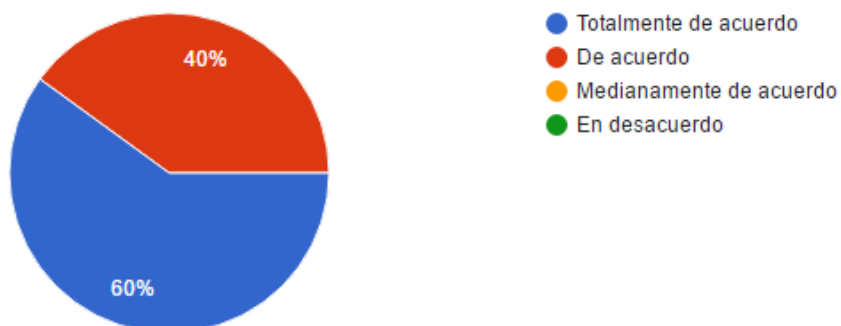
¿Le parece adecuada la estructura de la interfaz de usuario? (10 responses)



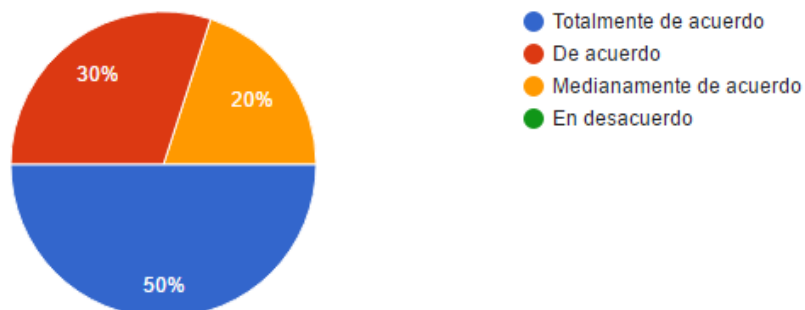
¿Se utiliza un buen contraste en la interfaz de usuario? (10 responses)



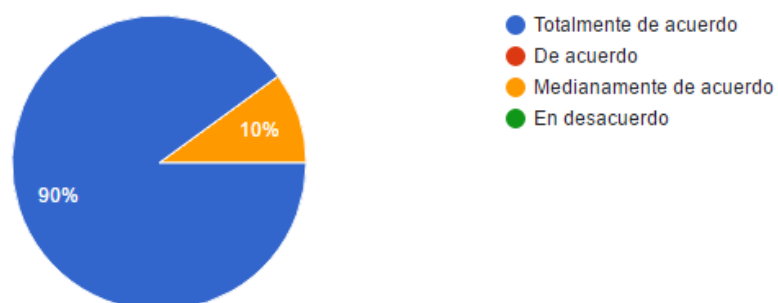
¿El diseño de la aplicación le parece consistente? (10 responses)



¿De manera general, considera que el sistema es interactivo? (10 responses)

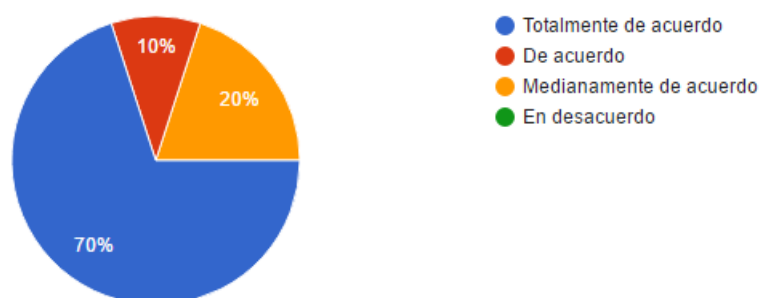


¿Se presenta de manera intuitiva la navegación de la aplicación? (10 responses)



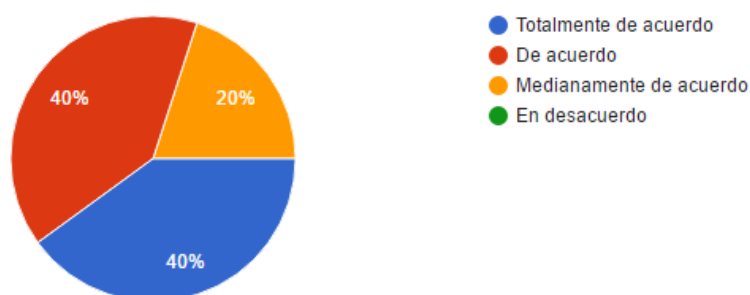
¿Se representa de forma sencilla la funcionalidad para consulta de tiendas y promociones?

(10 responses)



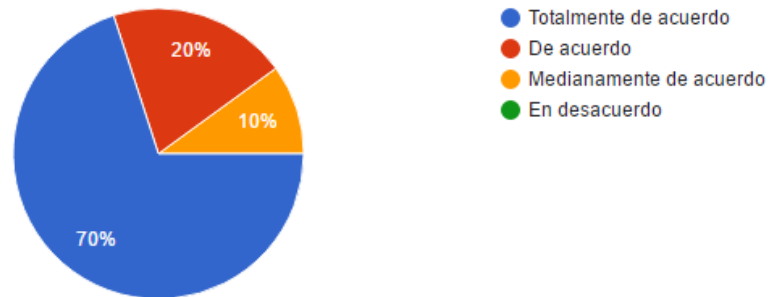
¿Se representa de forma sencilla la funcionalidad para el registro de tiendas y promociones?

(10 responses)



¿Se representa de forma sencilla los gráficos (Dummy) expresados en la aplicación?

(10 responses)



En general, las respuestas obtenidas fueron positivas, las sugerencias hicieron evidentes ciertos problemas de usabilidad y funcionamiento que fueron corregidos antes de la entrega final del producto y las opiniones recibidas mostraron un alto grado de aceptación por parte de los usuarios finales encuestados, por lo que se asume que este emprendimiento tendrá una alta probabilidad de éxito una vez se comience a utilizar.

Opiniones de usuarios encuestados

App Movil: en la Ilustración 57 se observa un buen comentario de una usuario de las encuestas, así como también una sugerencia para el sistema móvil.

Deja tu comentario o sugerencia:

La aplicación esta excelente, muy buen manejo de los elementos de diseño y funcionalidades extraordinarias!

Deberían agregarle en el modulo de configuración la actualización automática directa desde el playstore dependiendo de la selección del usuario, por ejemplo actualizar con datos, actualizar con WiFi o no actualizar.

Ilustración 57 - Comentario de Usuarios App Movil
Fuente: Los Autores

App Web: Se puede observar en la Ilustración 58 uno de los ajustes que fueron realizados gracias a las pruebas de usabilidad y funcionalidad que realizaron estos usuarios.

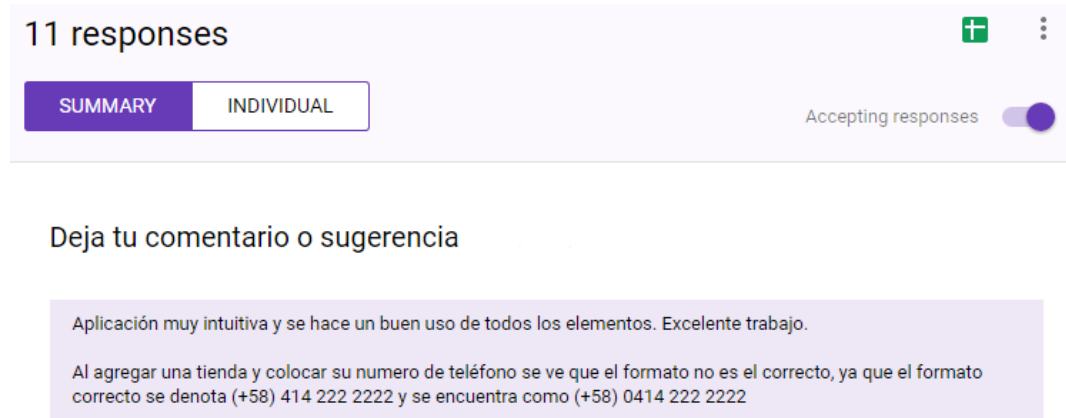


Ilustración 58 - Comentario de Usuarios App Web
Fuente: Los Autores

Problemas de usabilidad y funcionamiento

Aplicativo Movil

- Mejor visualización de imágenes de promociones: se coloca la configuración del elemento de imagen ajustada a la pantalla y esto hace que se visualice mejor.



Ilustración 59 - Imagen de promoción. App Movil
Fuente: Los Autores

- Error de funcionalidad: compartir app en Modulo de Mas Cerca: al hacer clic en el botón flotante que se puede observar en la Ilustracion 60, para compartir la aplicación en el modulo de Mas cerca la funcionalidad no se ejecutaba, se coloco el llamado del Activity que realiza esta funcionalidad y se realizaron las pruebas de regresión a este modulo.

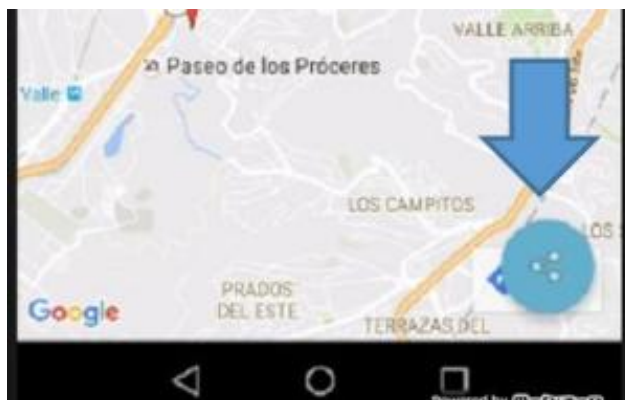


Ilustración 60 – Compartir App, modulo más cerca. App Movil
Fuente: Los Autores

- Actualización de aplicación automática: Se sugirió que se colocara configuración para actualizar la aplicación desde el play store con las tres (3) opciones que podremos ver en la Ilustracion 61 principalmente.

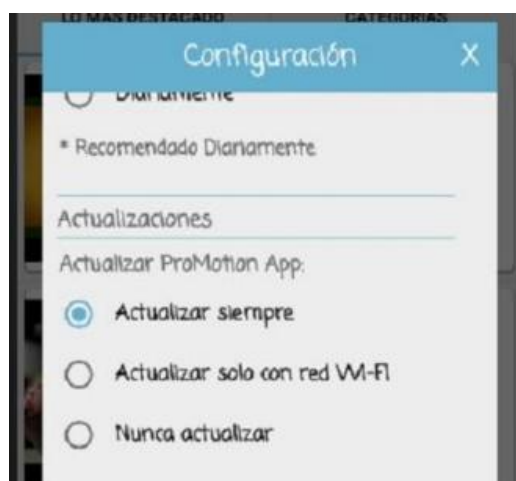
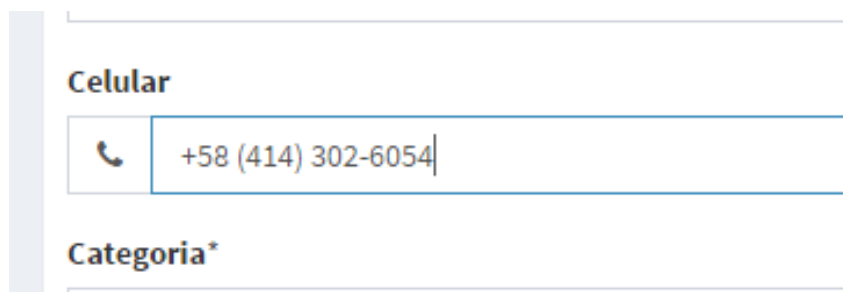


Ilustración 61 - Actualización de aplicación. App Movil
Fuente: Los Autores

Aplicativo Web

- Error en registro de teléfono a tiendas, enmascaramiento: Se modifica el enmascaramiento del campo Celular para que el formato sea valido como se observa en la Ilustracion 62.



The image shows a web form with two fields. The first field is labeled 'Celular' and contains the text '+58 (414) 302-6054'. The second field is labeled 'Categoria*' and is currently empty. The form has a light blue border and a white background.

Ilustración 62 - Registro de teléfono celular. App web
Fuente: Los Autores

- Ajuste de data en gráficos de inicio: se modificaron las graficas para poseer diagramas y graficos coherentes a lo que refieren. “Promociones mas visitadas, Tiendas con mas ventas en el año, entre otros” ver Ilustracion 63.

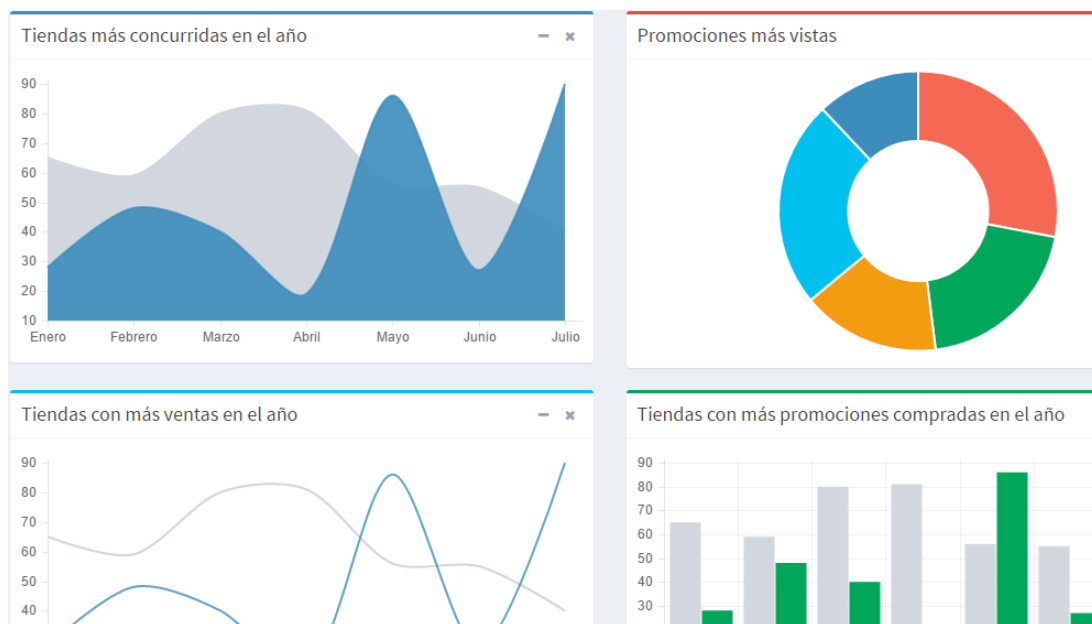


Ilustración 63 - Estadísticas de inicio. App web
Fuente: Los Autores

- Actualización de imágenes de promociones: se actualizaron las imágenes de las promociones ya que no coincidían con el título de la misma. Se puede ver la coherencia en dicha promoción en la Ilustración 64.



Ilustración 64 - Actualización de imágenes de promociones. App web
Fuente: Los Autores

Considerando los problemas de usabilidad y funcionalidad mencionados anteriormente, estos fueron ajustados al producto final y podemos ver estos resultados en las Ilustraciones 59 a 64 mencionando de esta manera cada uno de estos casos, se desarrollaron estos cambios y ajustes en ambas aplicaciones gracias a los comentarios de los encuestados en las pruebas de usabilidad.

4.8.3. Pruebas de funcionalidad

Se realizaron pruebas funcionales sobre las aplicaciones desde el punto de vista del usuario, se probaron las funcionalidades y requerimientos de la aplicación una por una para la determinación de posibles errores y fallas en el sistema luego del proceso de desarrollo. Las pruebas comprenden las operaciones que el usuario puede realizar y se evaluó el resultado esperado contra el resultado obtenido. De esta manera se identificaron los errores de la aplicación y fueron corregidos, realizando de esta manera las pruebas de regresión para la certificación de la aplicación.

Móvil App

- **Nosotros:** para el contacto con el equipo administrador se realiza la llamada y el envío de correo con resultados satisfactorios.
- **Notificaciones push:** Se logra ver la notificación push, no solo mientras la app está en primer plano sino también cuando se encuentra en Segundo plano, esta con respecto a las promociones que se tienen afiliadas por comercio, se selecciona la notificación y se despliega el detalle de la promoción.
- **Lista de promociones:** se logra visualizar las promociones más destacadas en el módulo de inicio, y a su vez la lista de promociones por comercio afiliado a nuestro sistema.
- **Seleccionar promoción:** al seleccionar una promoción ya sea como notificación, en módulo de inicio o en detalle de tienda, se despliega satisfactoriamente el detalle de la promoción.
- **Lista de categorías:** se logra visualizar la lista de categorías tanto en módulo de inicio como *banners* de selección, como en módulo de categorías en listas desplegables.
- **Selección de categoría:** se despliega satisfactoriamente la categoría seleccionada

- **Búsqueda de tiendas:** se logra realizar la búsqueda de tiendas por carácter ingresado.
- **Selección de tiendas:** al seleccionar una tienda se despliega el detalle de la misma.
- **Detalle de tienda:** Se logra visualizar sin inconvenientes el detalle de la tienda, así como también el mapa asociado a la localización de la misma y la funcionalidad de llamar tienda.
- **Buscar ruta más cercana:** al iniciar el módulo de Más cerca, se despliega satisfactoriamente la ruta más cercana a una tienda asociada al sistema.
- **Ubicación en mapa de tiendas asociadas:** se logra visualizar un marcador en la ubicación específica con el nombre de la tienda.
- **Compartir aplicación:** se comparte la aplicación satisfactoriamente en las redes sociales instaladas en el dispositivo.
- **Configuración de aplicación:** se realizan los cambios en la configuración de la aplicación y de manera satisfactoria se reflejan tanto en el tiempo de las notificaciones como en la actualización de la aplicación.

Web App

- **crearUsuarios:** proceso satisfactorio
- **editarUsuarios:** proceso satisfactorio
- **verUsuarios:** proceso satisfactorio
- **eliminarUsuarios:** proceso satisfactorio
- **crearComercios:** proceso satisfactorio
- **editarComercios:** proceso satisfactorio
- **verComercios:** proceso satisfactorio
- **eliminarComercios:** proceso satisfactorio
- **crearTiendas:** proceso satisfactorio
- **editarTiendas:** proceso satisfactorio
- **verTiendas:** proceso satisfactorio
- **eliminarTiendas:** proceso satisfactorio
- **crearPromociones:** proceso satisfactorio
- **editarPromociones:** proceso satisfactorio

- **verPromociones:** proceso satisfactorio
- **eliminarPromociones:** proceso satisfactorio
- **crearBeacons:** proceso satisfactorio
- **editarBeacons:** proceso satisfactorio
- **verBeacons:** proceso satisfactorio
- **eliminarBeacons:** proceso satisfactorio
- **crearCategorias:** proceso satisfactorio
- **editarCategorias:** proceso satisfactorio
- **verCategoria:** proceso satisfactorio
- **eliminarCategorias:** proceso satisfactorio
- **actualizarPerfil:** proceso satisfactorio

Las pruebas de funcionalidad que se realizaron no arrojaron inconvenientes en las funcionalidades probadas, por lo que es posible para el usuario cumplir con todos los flujos requeridos para ambas aplicaciones.

CONCLUSIONES Y RECOMENDACIONES

El producto final de este proceso de investigación y desarrollo fue el prototipo funcional de solución que comprende la aplicación que implementa marketing de proximidad sobre la plataforma Android que puede ser aplicado a cualquier tipo de local comercial. Teniendo a su vez el prototipo funcional de aplicación *web* que administra y es enfocado directamente a los dueños de estos locales comerciales, entendiendo como local comercial cualquier comercio sin importar su rubro como Farmacias, Restaurantes, Tiendas, Franquicias, Bancos, entre otros.

El mismo prototipo funcional de solución desarrollado comprende la conexión con un API Restful, para de esta manera hacer uso de una Base de datos centralizada que futuras soluciones puedan consultar sin inconvenientes. Teniendo así el prototipo funcional Web para la administración de dicha data y el prototipo funcional móvil para la consulta e interacción directa con el usuario final (Consumidores de los comercios afiliados a nuestro sistema). El desarrollo de la solución comprende una aplicación móvil desarrollada en Android Studio, una conexión Bluetooth Low Energy que permite el envío de notificaciones por proximidad al usuario, una aplicación Web y un API Restful (jbuilder) desarrollados en Ruby on Rails. Logrando así con la integración de todos estos componentes la correcta implementación de las funcionalidades con el uso de dichos *frameworks*.

Se implementó la metodología de desarrollo ágil Scrum para el manejo del proyecto, esta permitió no solo organizar los requerimientos y dividirlos en tareas más sencillas de ejecutar sino hacer cumplir con el desarrollo de una solución de marketing de proximidad y gestión publicitaria con dispositivos *beacon* como tecnología base que permitió la visualización y gestión de promociones a través del desarrollo de estas dos aplicaciones. Así como también proporciono flexibilidad en las iteraciones y cambios que se presentaron.

Este prototipo funcional proporciona una solución definitiva al *marketing de proximidad* que no sólo forma parte del presente sino que es el futuro del este tipo de

mercadeo, más aún los sistemas diseñados para dispositivos móviles como *tablets* y teléfonos inteligentes, debido a su amplia utilización en casi todos los países del mundo, a la rápida difusión de la información a través de las redes sociales, a la comodidad que ofrecen estos sistemas para los clientes y las tendencias de mercadeo que cada vez más apuntan a la comercialización apoyada en medios tecnológicos. Todo esto hace sencillo pensar que las soluciones de marketing digital tienen el potencial de ser alternativas de marketing para medianos y pequeños empresarios.

Es por esto que se considera el conjunto de prototipo funcional de solución “*ProMotionApplication*” es un emprendimiento con éxito, en el que además se aplicaron conocimientos de diseño, usabilidad de aplicaciones, sociabilidad e interconexión de plataformas, que muy probablemente contribuirá a mejorar la calidad del servicio de marketing digital ofrecido a la población y generará mejoras económicas para todas las partes involucradas.

Posibles limitaciones

- Dependiendo de la arquitectura del dispositivo, será necesario mantener el bluetooth encendido ya que para IOS la tecnología *Bluetooth Low Energy* ya viene integrado en el hardware del dispositivo y no es necesario interactuar contra el bluetooth, aunque para la plataforma Android la aplicación requiere de la activación del *Bluetooth*, pero esto no se considera un inconveniente ya que gracias a la tecnología de *Bluetooth* de bajo consumo el rendimiento y duración de la carga del dispositivo se mantiene, una vez que el componente entra en un estado de inactividad mientras no se existan emisores Beacon en la proximidad configurada. Al momento que se ingrese en la región de un Beacon esta señal es recibida por el dispositivo y de esta manera se activan las notificaciones push asociadas a la descripción configurada en dicho beacon.

- Validación de información publicada por los comercios afiliados, se considera una limitación aunque la solución a esto es un usuario administrador o un sistema que detecte contenido inapropiado que es cargado como una promoción o en su defecto una tienda y que al consumir el API se muestra al usuario en la aplicación móvil.

Trabajos futuros

- Módulo de autenticación de usuarios para la mejor segmentación de usuarios, de esta manera se podrán tener estadísticas de promociones vistas por usuario, segmentación de promociones por usuario, recomendación por favoritos, entre otros.
- Modulo localización *In-door*, este módulo refiere a la localización de interiores de manera que se posee un mapa de la estructura interna de una construcción, esto desarrollado con la tecnología *Google Maps* (Mapas de interiores) y geolocalización *In-door Beacon Technology (Bluetooth Low Energy)*.
- Botón de pago de cupón de promoción, producto o servicio desde la aplicación Móvil, por el cual el usuario podrá hacer compras de las promociones, productos y servicios que ofrecen nuestros comercios afiliados.
- Control y registro de inventario, ya que se habilitaría la carga de inventario en el sistema desde la Web administrativa para gestionar la compra-venta de productos desde la aplicación móvil.
- Estadísticas de compra y visualización de promociones para la Web administrativa.
- Comentarios y calificación de tiendas para la aplicación móvil.
- Inclusion y desarrollo de Modulo de BI para el sistema.

REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES

- Abrahamsson, J. Warsta, M. T. Siponen, y J. Ronkainen, «*New directions on agile methods: a comparative analysis*», 2003, pp. 244-254. Recuperado de: http://www.uelbosque.edu.co/sites/default/files/publicaciones/revistas/revista_tecnologia/volumen12_numero2/12Articulo_Rev-Tec-Num-2.pdf
- Academia Android, (2015). Aplicaciones cliente-servidor y redes de telefonía móvil. Recuperado de: <http://academiaandroid.com/aplicaciones-cliente-servidor-y-redes-de-telefonía-móvil/>
- Altiria (2016). Empresa experta en marketing movil. Marketing de proximidad. Recuperado de: <https://www.altiria.com/marketing-de-proximidad/>
- Álvarez (2001). Que es HTML. Recuperado de: <http://www.desarrolloweb.com/articulos/que-es-html.html>
- Álvarez (2008). Introducción a CSS3. Recuperado de: <http://www.desarrolloweb.com/articulos/introduccion-css3.html>
- Arnold y Gosling (2001). El Lenguaje de Programación Java TM. Recuperado de: <https://docs.google.com/file/d/0Byy7aUI9u4fBRnJwc1U5Vkdnlk/edit>
- Beck (1999), *Extreme Programming Explained (Second Explained)*. Recopilado de: <http://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf>
- Blanco, S. Metodologías de desarrollo (2008). Recuperado de: <http://www.marblestation.com/?p=644>
- Bluetooth (2016). *Bluetooth Smart (Low Energy) Technology*. Recuperado de: <https://www.developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx>
- Canós, J., Letelier, P., y Penadés, C. (2003) *Metodologías Ágiles en el Desarrollo de Software*. Recuperado de: http://noqualityinside.com/nqi/nqifiles/XP_Agil.pdf
- Castillo P., (2004). “Páginas Web Estáticas vs. Páginas Web Dinámicas”. Recuperado de: <http://www.articulo.org/idx/15/2039/Negocios-en-Internet/article/Paginas-WebEstaticas-vs-Paginas-Web-Dinamicas.html>

- Date, C. (2001). Introducción a los Sistemas de Bases de Datos. Séptima edición. ISBN 968-444-419-2.
- Díaz (2009). *Rup vs Extreme Programming* Recuperado de: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>
- Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development Hardcover*. Indianápolis, IN. Estados Unidos. Wiley.
- Emen. (2015), Android domina el mercado venezolano de teléfonos inteligentes. El Mundo. Recuperado de: <http://www.elmundo.com.ve/noticias/tecnologia/dispositivos/android-domina-el-mercado-venezolano-de-telefonos-.aspx>
- Espinoza E. (2007) Lima, Programación Extrema. Recuperado de: <http://es.slideshare.net/edgarespinoza/programacion-extrema>
- Estimate, ASK app, 2016. Recuperado de: <http://blog.estimate.com/post/141953007415/museum-visitors-want-a-conversation-bringing-apps><https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>
- Estimate, Frisco app, 2016. Recuperado de: <https://community.estimate.com/hc/en-us/articles/204400203-Shopping-on-the-go-with-Frisco-pl>
- Franganillo (2011). Html5: el nuevo estándar básico de la Web. Recuperado de: <http://franganillo.es/html5.pdf>
- García I. (2010). Aplicación web para el conocimiento y conversión de unidades (TEG). Recuperado de: http://meteo.ieec.uned.es/www_Usumeteo2/Memoria/Capitulo3.pdf
- Gilbert, Pérez y Castilla. (2008). Bases de datos en MySQL. Recuperado de: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf
- Gomez y De Abajo (1997), en Los Sistemas de Informacion en la Empresa. Universidad de Oviedo. Capitulo 4 Sistemas de información tácticos y estratégicos. (P. 54).

- Gonzalo M. (2015), ¿Qué es y qué potencial tiene la tecnología *Beacon*? Recuperado de: <http://www.getglocal.com/que-es-y-que-potencial-tiene-la-tecnologia-beacon>
- Gutiérrez (2010). *Frameworks* y Componentes, Universidad de los Andes Venezuela. Recuperado de: http://www.codecompiling.net/files/slides/IS_clase_10_frameworks_componentes.pdf
- iBeacon insider. (2015), What is iBeacon? What are iBeacons? Recuperado de: <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>
- Kon M. (1997). “El Software Libre”. Recuperado de: <http://www.monografias.com/trabajos12/elsoflib/elsoflib.shtml>
- L. Rinsing, N. Janoff, (Agosto, 2000) The Scrum Software Development Process for Small Teams. Software, IEEE. Recuperado de: <http://web.lindarising.info/uploads/IEEEScrum.pdf>
- Luque L. (2009), Sistemas de Informacion. Instituto Universitario Santiago Mariño. Unidad I Sistemas de Información. Recuperado de: http://sistemasdeinformacionpsm.blogspot.com/2009/03/unidad-i-sistemas-de-informacion_29.html
- Márquez A. (2013). Conceptos sobre APIs REST. Recuperado de <http://asiermarques.com/2013/conceptos-sobre-apis-rest/>
- Martínez (2007), Dispositivos Móviles. Recuperado de <http://exa.unne.edu.ar/informatica/SO/tfbasterretche.pdf>
- Martinez (2016), Fundamentos de Programación en Java, Recuperado en: <<http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/fjava.pdf>>.
- Matz (2001), en [An Interview with the Creator of Ruby](http://www.linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html), Recuperado de: <http://www.linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html>
- Merino M. (2014), ¿Qué es una API y para qué sirve? , Recuperado de: <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>
- Montilva (1986), en Desarrollo de Sistemas de Informacion. Administración, Metodología y Técnicas. Seccion 1.2 DATOS E INFORMACIÓN. (P. 1-9).

- Murazzo, M., Millán, F., Rodríguez, N., Segura, D., y Villafañe, D. (2010). Desarrollo de aplicaciones para *Cloud Computing*. Universidad Nacional de San Juan, Facultad de Ciencias Exactas, Físicas y Naturales. CACIC 2010 - XVI Congreso Argentino de Ciencias de la Computación. Trabajos presentados (p. 941-949). ISBN: 978-950-9474-49-9. Recuperado de: http://sedici.unlp.edu.ar/bitstream/handle/10915/19374/Documento_completo.pdf?sequence=1
- MySQL. (2013). Recuperado 14 de Marzo de 2015 de, <http://www.mysql.com/>
- Oré (2009), en Functional Testing – Pruebas Funcionales. Recuperado de: http://www.calidadyssoftware.com/testing/pruebas_funcionales.php
- PHP, (2016), “¿Qué es PHP?”. Recuperado de <http://php.net/manual/es/intro-what-is.php>
- Porto J., Marino M. (2011), Definición de Dispositivo. Recuperado de: <http://definicion.de/dispositivo/>
- PostgreSQL. (2010). Recuperado 12 de Marzo de 2015 Recuperado de <http://www.postgresql.org>
- Premper (2015). Los *Beacons* hablan. Recuperado de: <https://www.premper.com/la-tecnologia-beacon>
- Quin (2015). *Extensible Markup Language* (XML). Recuperado de: <https://www.w3.org/XML/>
- Rincón J. (2012). Ciclo de Vida Rup. Recuperado de: <http://josemiguelrincon.blogspot.com/2012/04/ciclo-de-vida-rup.html>
- RUBY, (2016), “Ruby es...”. Recuperado de <https://www.ruby-lang.org/es/>
- Russell et al (2005), en Publicidad Administración. Primera Parte Publicidad para el consumidor. (P. 60)
- Salinas, E. (2004). El Lenguaje de Programación Java. Recuperado de: <https://www.inf.utfsm.cl/~mcloud/iwi-253/tareas/t2-lp-2004-Ayudantia1.pdf>
- Schwaber y Sutherland (2014). La Guía de Scrum: Las Reglas del juego. Recuperado de: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

- Silberschatz, Korth y Sudarshan. (2002). Fundamentos de bases de datos. Recuperado de: <https://unefazuliasistemas.files.wordpress.com/2011/04/fundamentos-de-bases-de-datos-silberschatz-korth-sudarshan.pdf>
- Sommerville I. (2002). Ingeniería de Software, Pearson Educación, Recuperado de: http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI441/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf
- The University of California, Riverside, ed. (3 de diciembre de 2015). «*When Apps Talk Behind Your Back*» (en inglés). Consultado el 10 de diciembre de 2015. «*We focused on a relatively neglected aspect of security research, which is the potential for good apps to leak personal information through the sites they interact with.* »
- TIOBE (2016), *TIOBE Index for July 2016*, Recuperado de <http://www.tiobe.com/tiobe-index/>
- Trabajo de Adscripción Dispositivos Móviles - Universidad Nacional del Nordeste Facultad de Ciencias Exactas, Naturales y Agrimensura. Juan FelixBasterretche - L.U.: 34039 Prof. Director: Mgter. David Luis la Red Martínez Licenciatura en Sistemas Corrientes - Argentina 2007. Recuperado de: “<http://exa.unne.edu.ar/informatica/SO/tfbasterretche.pdf>”
- Valle J. (2005). "Definición Modelo Cliente Servidor". Recuperado de: <http://ri.bib.udo.edu.ve/bitstream/123456789/1005/1/TESIS.Dise%C3%B1o%20de%20una%20aplicacion%20Web.pdf>
- Vasilis, (2011). Metodologías de desarrollo de *software*. Recuperado de: <http://ingenieriasoftwarecufm.blogspot.com/2012/10/metodologias-para-desarrollo-de-software.html>
- Vila, (2015). Eventos Scrum: El Scrum diario. Recuperado de: <http://managementplaza.es/blog/el-scrum-diario/>
- W3C.CSS (2008). *Cascading Style Sheets, Level 1*. Recuperado de: <https://www.w3.org/TR/CSS1/#references>

Anexos

1. Manual de aplicación móvil app



ProMotion Móvil App

Página | 1



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN



Manual de Usuario

Versión 1.0

Autor:

Br. Edwin Alvarado

C.I. 25.304.762

Caracas, 29 de septiembre de 2016



Tabla de contenido

1	Descripción de aplicación	3
2	Manejo de Notificaciones.....	3
3	Módulo de inicio	4
3.1	Lo más destacado.....	4
3.2	Categorías	6
4	Menú Hamburguesa	7
5	Módulo Categorías	7
6	Modulo Mas Cerca.....	14
7	Modulo Configuración	14
7.1	Notificaciones	17
7.2	Actualizaciones	18
7.3	Versión de aplicación.....	18
8	Modulo Nosotros.....	19
8.1	¿Qué es ProMotion App?.....	20
8.2	¿Cómo puedo afiliarme?.....	21
8.3	Contáctanos.....	21
9	Compartir aplicación	22
10	Salir.....	24

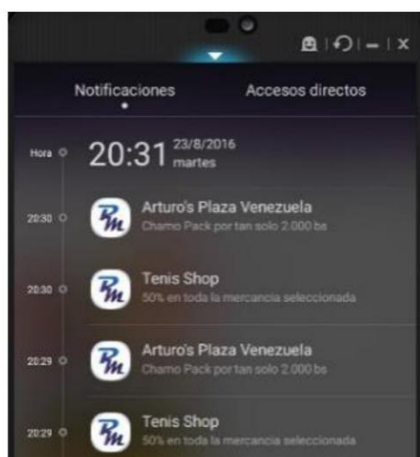


1 Descripción de aplicación

ProMotion App es una aplicación en la que podrás economizar tus ingresos sin tener que dejar de adquirir todo lo que más te gusta, en nuestra aplicación encontraras todas las tiendas y comercios afiliados a nuestro sistema en donde podrás ver las promociones de estos comercios y lo mejor de todo es que usa un sistema de notificación de promociones al momento de estar en la cercanía de los locales afiliados, donde veras una notificación en la barra superior de tu dispositivo móvil al momento de estar en un rango de 0-30mts de nuestros comercios. Y esto no es todo podrás buscar las tiendas por su nombre y ver toda la descripción de la misma, así como sus promociones, ubicación y más, sin necesidad de estar físicamente en la ubicación de la tienda.

2 Manejo de Notificaciones

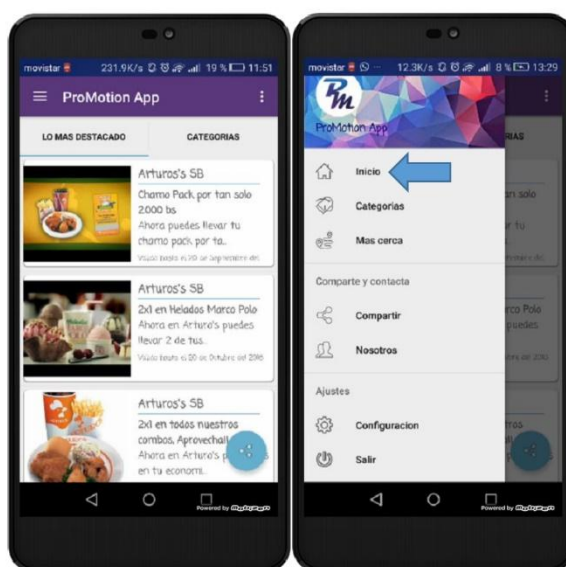
Con el manejo de notificaciones de ProMotion app podrás ver en el panel de notificaciones de tu dispositivo las promociones de los locales comerciales afiliados a nuestro sistema en lo que te encuentres en los alrededores de las tiendas.





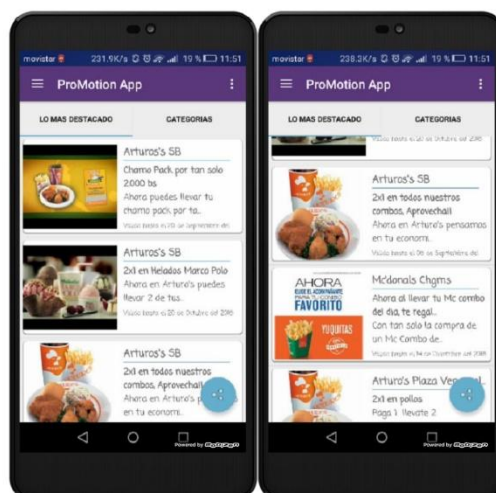
3 Módulo de inicio

La aplicación cuenta con un módulo de inicio que se desplegará al iniciar la aplicación, en este encontraremos en primera plana una banner donde podremos ver las promociones más destacadas y un filtro de tiendas por categorías.

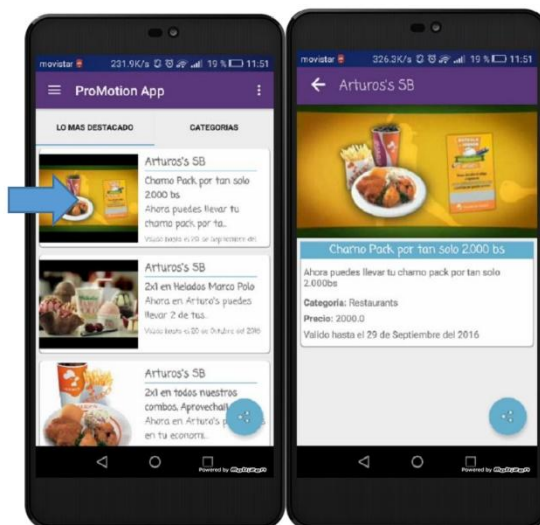


3.1 Lo más destacado

En esta opción podremos encontrar las promociones más destacadas del sistema, por visualización de la promoción



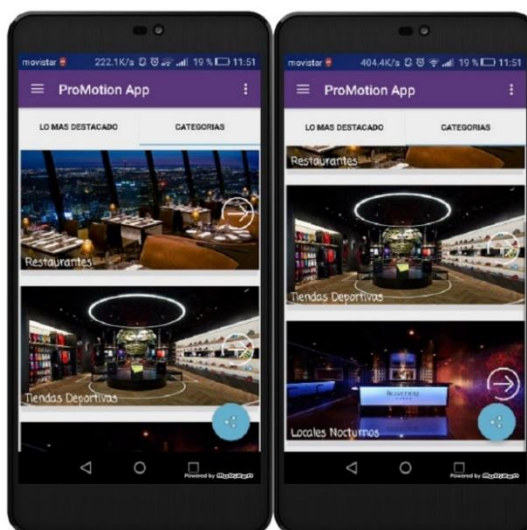
Al hacer clic en una promoción podremos ver la descripción de la misma.



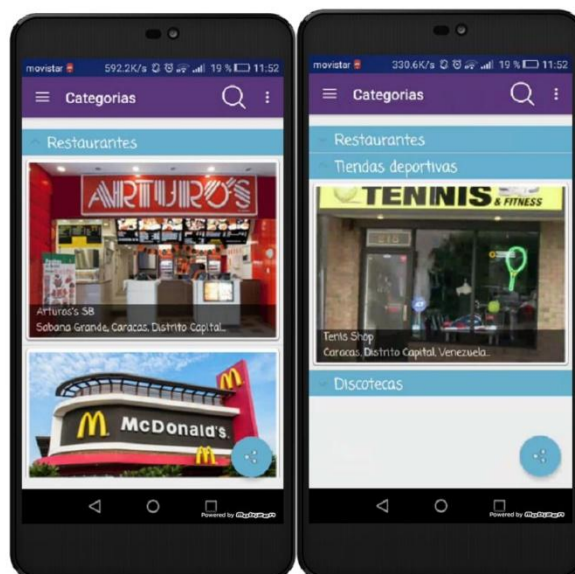


3.2 Categorías

En la siguiente opción tendremos las categorías registradas en el sistema de tal manera que al seleccionar una categoría podremos ver todas las tiendas que se encuentren en este rubro.

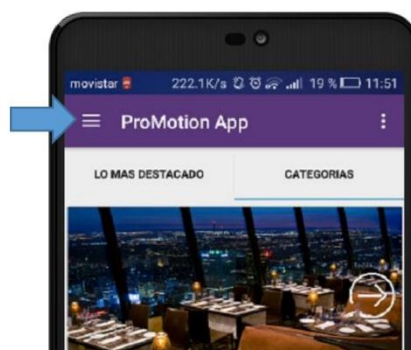


Al hacer clic en una opción de la lista vamos a poder visualizar el módulo de categorías con la lista desplegada del rubro seleccionado **ver sección 4 Categorías**



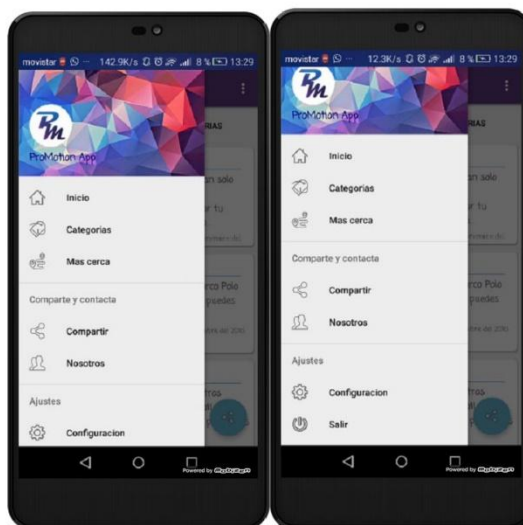
4 Menú Hamburguesa

La aplicación posee en la esquina superior izquierda el menú de la misma, el cual es un menú de tipo hamburguesa.





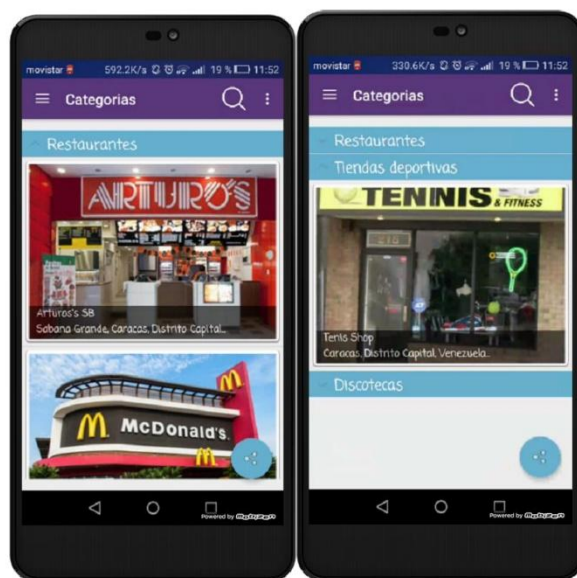
Al momento de abrir dicho menú visualizaremos todos los módulos involucrados en la aplicación así como también el botón de salir para cerrar la aplicación.



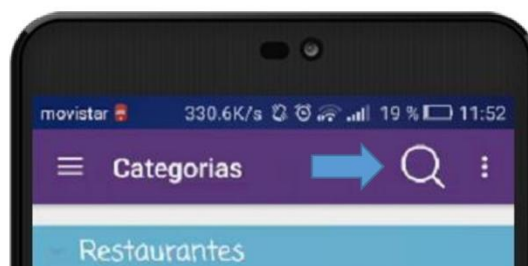
5 Módulo Categorías

En el módulo de categorías vamos a encontrar listas desplegables por cada una de las categorías configuradas en el sistema, de esta manera se considera un filtro de búsqueda por categorías.



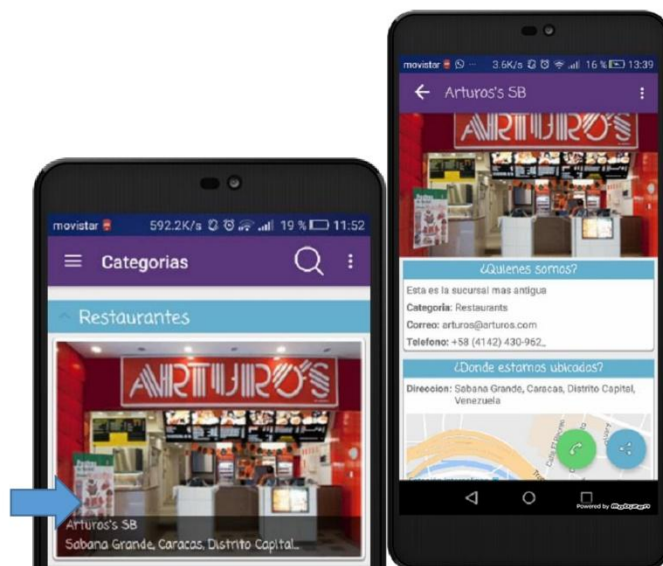


En el banner superior podremos ver un botón de buscar en el cual podremos, buscar la tienda de nuestra preferencia a través de su nombre teniendo una función de auto filtrado al ingresar cada carácter de búsqueda.



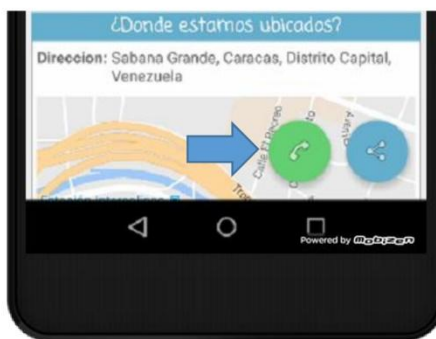


Al seleccionar una tienda podremos ver el detalle específico.

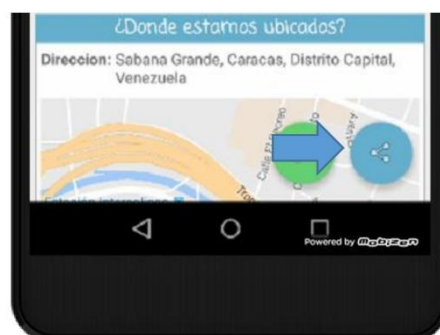


También se poseen dos botones flotantes en la parte inferior de la pantalla donde podremos ver la opción de llamar y la opción de compartir la aplicación.

- Llamar: esta funcionalidad activa una llamada al teléfono configurado para la tienda que se está detallando.



- Compartir aplicación: **Ver apartado 9 Compartir aplicación**



Teniendo a su vez las siguientes secciones

5.1 ¿Quiénes somos?

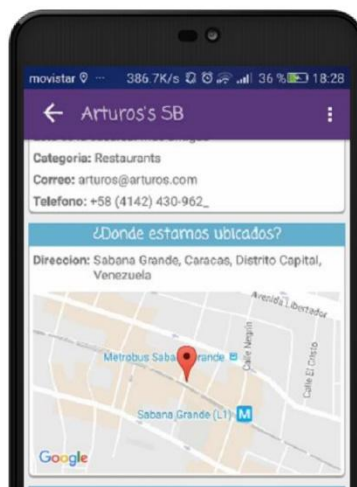
En la sección de quienes somos tendremos la descripción de la tienda, la categoría o rubro en el cual está definida, el correo electrónico y el teléfono de la misma.





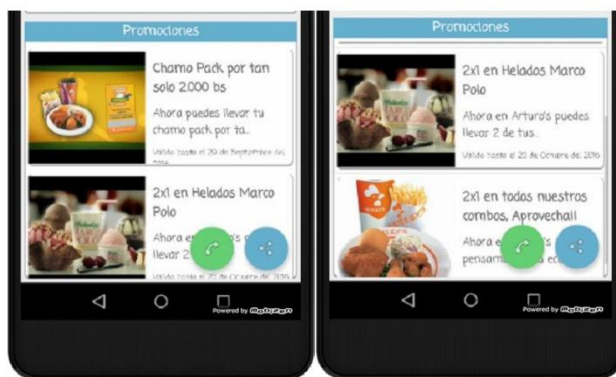
5.2 ¿Dónde estamos ubicados?

En esta podremos ver la dirección exacta de la tienda a la que estamos consultando, tanto en texto como en el mapa de la dirección.

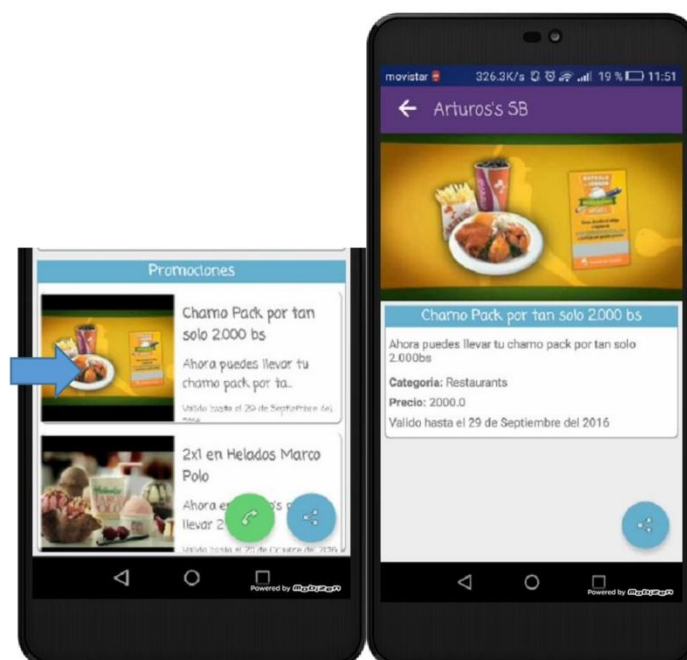


5.3 Promociones

Veremos en esto una lista de promociones de la tienda en las cuales podremos seleccionar la promoción de nuestro interés.



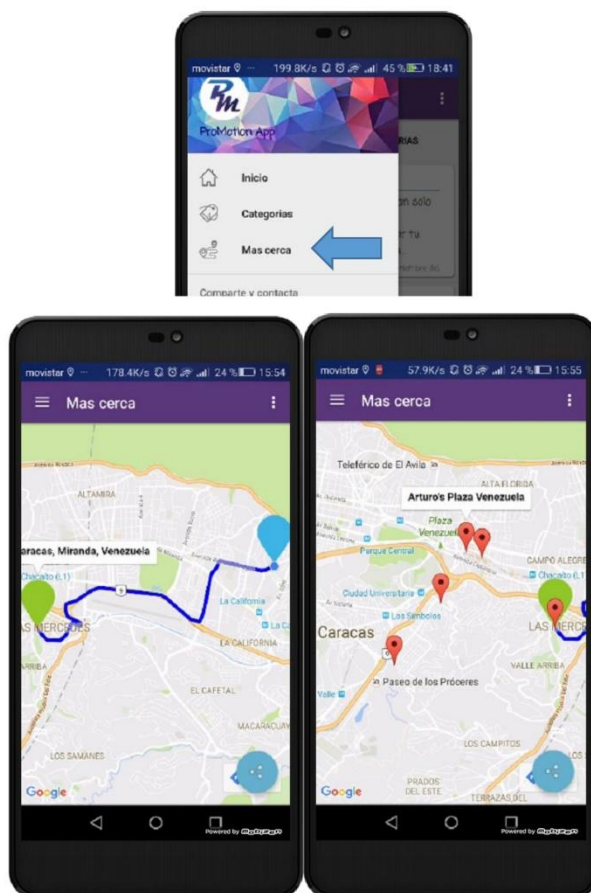
Al seleccionar una promoción vamos a poder ver el detalle de la misma que incluye su descripción, categoría a la que pertenece, precio de la promoción y validez de la misma. Así mismo encontraremos el nombre de la promoción en el banner azul y en el banner morado a la tienda que corresponde.





6 Modulo Mas Cerca

En el siguiente modulo vamos a encontrar un mapa donde veremos inicialmente la ruta de la tienda más cercana asociada a nuestro sistema, pero no solo esto, también se puede ver la ubicación exacta de cada una de las tiendas afiliadas a la aplicación.





7 Modulo Configuración

En el siguiente modulo se poseen todos los parámetros que son configurables e información de importancia con respecto a la aplicación.

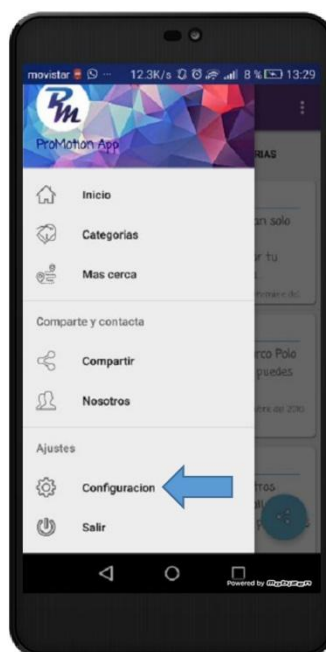


El mismo puede ser accedido desde el atajo ubicado en la parte superior derecha de las pantallas de la app o a través del menú hamburguesa ubicado a la izquierda del banner superior.

- Atajo:



- Menú hamburguesa

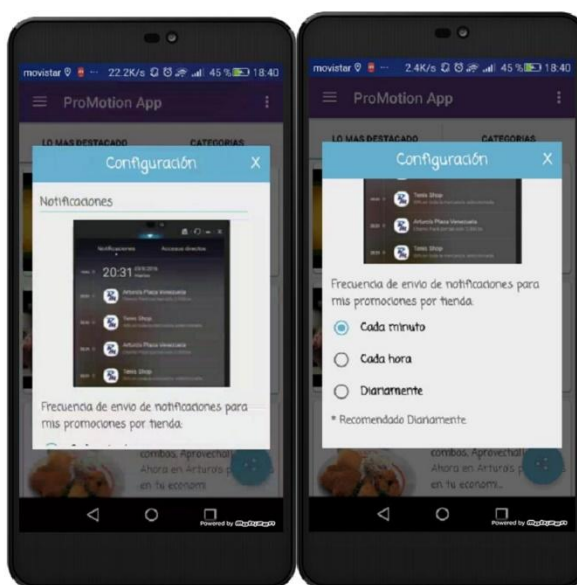




Y a su vez el mismo contiene tres secciones las cuales las podemos desplegar de la siguiente manera:

7.1 Notificaciones

En la sección de notificaciones podremos configurar la frecuencia de las notificaciones que envía la aplicación a su dispositivo móvil, de tal manera que podrá configurar si desea que estas se desplieguen cada un minuto, una hora o un día.





7.2 Actualizaciones

En esta sección podrás configurar si deseas que la aplicación se actualice solo con WIFI, siempre o nunca.



7.3 Versión de aplicación

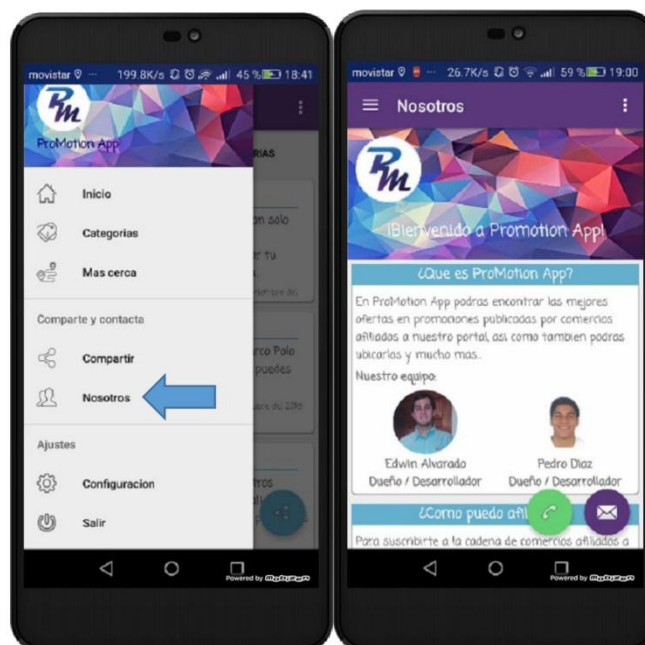
Esta sección es completamente informativa, ya que encontraras la versión de la aplicación y de esta manera podrás ver si está actualizada contra la versión que este publicada en el playstore.





8 Modulo Nosotros

En este módulo podrás encontrar la información completa de la aplicación.



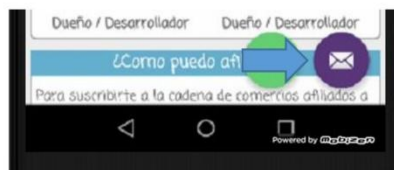
Encontraremos dos botones flotantes llamar y enviar correo:

- Llamar: esta funcionalidad permite llamar a la central telefónica donde podrás reportar inconvenientes o contactarnos para cualquier duda.





- Envío de correo: con esta funcionalidad podrás contactarnos a través de envío de correo electrónico a través de las cuentas de correo configuradas en tu dispositivo móvil de tal manera.



Y este módulo de nosotros se separa en tres secciones:

8.1 ¿Qué es ProMotion App?





8.2 ¿Cómo puedo afiliarme?



8.3 Contáctanos

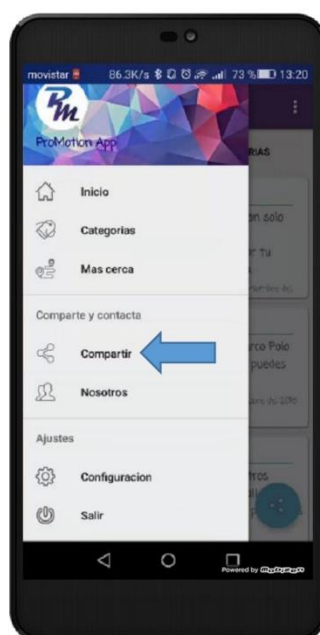




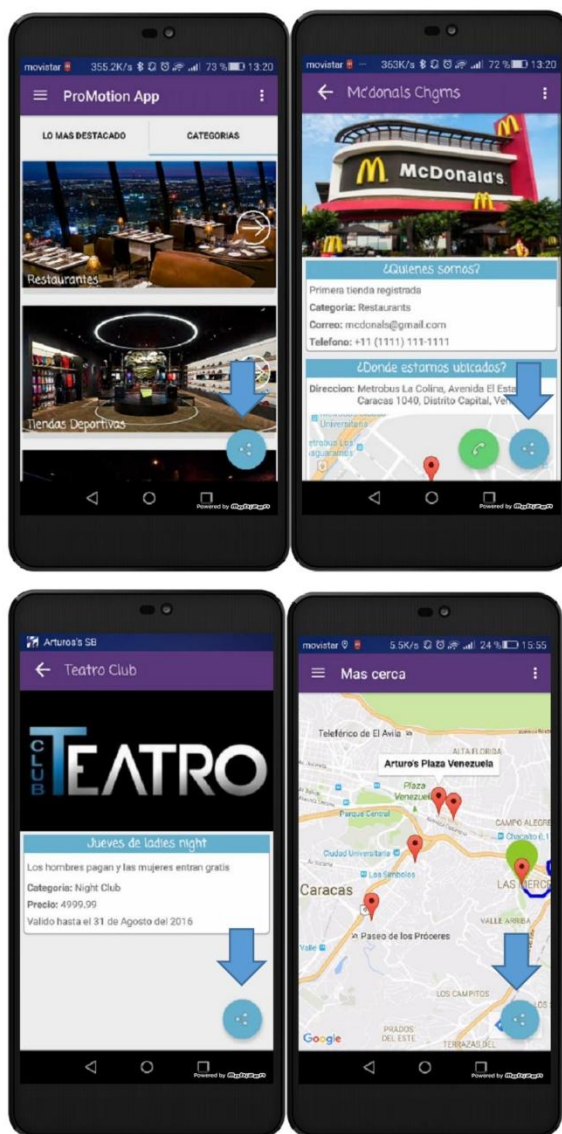
9 Compartir aplicación

Esta funcionalidad la podemos ubicar en diferentes flujos, principalmente en el menú hamburguesa y como segunda opción y más accesible en la parte inferior derecha de la mayoría de las pantallas de la app.

- Menú hamburguesa:



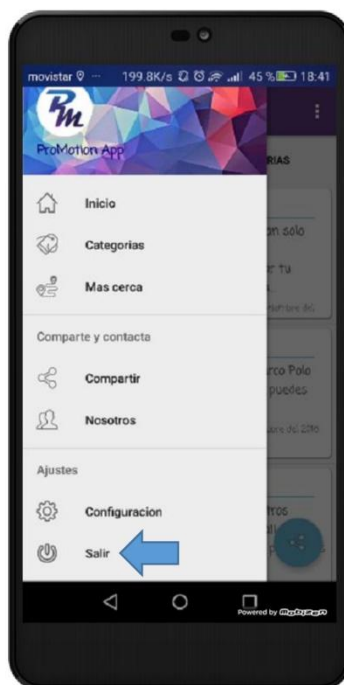
- Flujos Alternos:





10 Salir

Opción para salir de la aplicación y llevar sus procesos a background o segundo plano.



2. Manual de aplicación web app.



Manual de Usuario

Versión 1.0



Tabla de contenido

1	Descripción de aplicación.....	3
2	Login de usuario	3
3	Página Principal	4
3.1	Menú principal.....	4
3.2	Perfil	6
4	Modulo de Comercios	7
5	Modulo de Tiendas.....	8
6	Módulo de Promociones.....	11
7	Salir	12



1 Descripción de aplicación

ProMotion Web Admin es una aplicación que está pensada para todos, tanto para grandes empresarios como pequeños emprendedores, en la que podrás adquirir una gran cantidad de clientes potenciales y aumentar considerablemente tus ventas. En nuestra aplicación podrás contar con tu perfil de usuario y añadir todos tus comercios junto con sus tiendas o sucursales asociadas, indistintamente del rubro al que pertenezcan (tiendas deportivas, locales nocturnos, comida rápida, entre otros), además de poder incluir promociones a cada una de tus tiendas. Lo más importante de todo es que con un solo clic al momento de agregar y activar tus promociones, las mismas ya pueden ser visualizadas por todos los usuarios que posean la aplicación móvil de Promotion, y más interesante aun si deseas asociar un dispositivo beacon, instantáneamente los usuarios móviles de Promotion serán notificados de las promociones al momento que se encuentren cerca de las tiendas que registraste sin necesidad de entrar dentro de ella.

2 Inicio de sesión

A través del inicio de sesión, podrás ingresar al portal administrativo y ver la pagina principal de Promotion Web App.





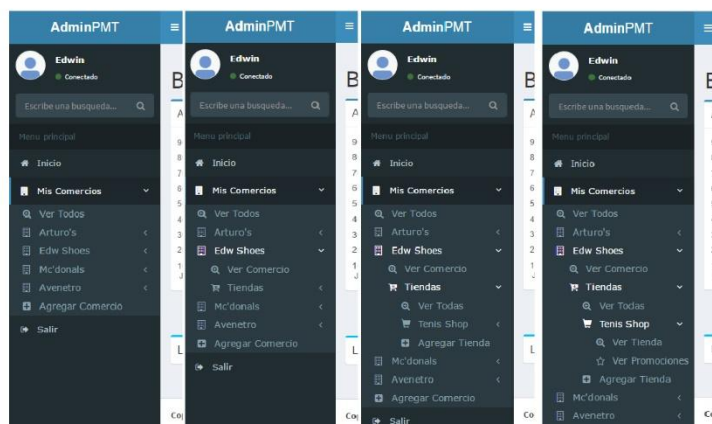
3 Página Principal

La aplicación cuenta con un módulo de inicio que se desplegará al iniciar la aplicación, en este encontraremos en primera plana varios gráficos que en un futuro pueden ser utilizados para representar estadísticas importantes sobre los comercios que administres que te permitan tomar decisiones importantes sobre tu negocio. Además se mostrará un menú al lado izquierdo el cual permitirá la navegación a través del portal y una opción en la parte superior izquierda que permitirá visualizar información de perfil del usuario que inició sesión.



3.1 Menú principal

En esta sección de la página principal se podrá observar información del perfil como el nombre, indicando que está conectado. También se contará con una barra de búsqueda. Se podrán observar los ítems principales que estarán relacionados con los comercios que tenga asociado el usuario que inició sesión, además de botones básicos como lo es Inicio que servirá para ir a la página principal del portal como la opción de salir para cerrar la sesión actual.



A través de este menú el usuario podrá navegar a través de la opción “Mis Comercios” donde podrá observar una lista de ítems desplegables que llevarán el nombre de cada uno de los comercios previamente registrados además de la opción “Ver Todos” para visualizar en detalle la lista de los comercios y la opción “Agregar Comercio” para añadir un nuevo comercio.

Si el usuario selecciona un ítem asociado a un comercio se desplegarán dos opciones: “Ver Comercio” para ver el detalle del comercio seleccionado y un ítem desplegable “Tiendas”, el cual podrá desplegar una lista de ítems también desplegables con todas las tiendas asociadas al comercio seleccionado, además de la opción “Ver Todas” para visualizar en detalle la lista de tiendas y la opción “Agregar Tienda” que servirá para añadir una tienda a ese comercio específicamente.

Si el usuario selecciona un ítem asociado de una de las tiendas se desplegarán dos opciones: “Ver Tiendas” para ver el detalle de la tienda seleccionada y “Ver Promociones” que permitirá ver todas las promociones asociadas a esa tienda.

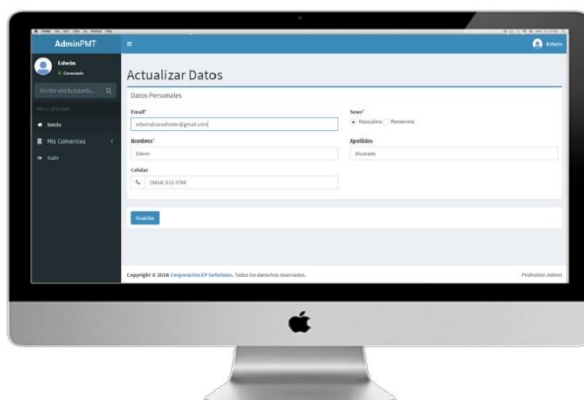


3.2 Perfil

En la parte superior derecha de la pantalla principal el usuario podrá hacer clic encima de su nombre de perfil y se desplegará un cuadro con información relacionada a su perfil, como el nombre y apellido, la cantidad de comercios de los cuales dicho usuario es propietario así como un botón para acceder al perfil y otro para cerrar la sesión.



Si el usuario presiona el botón “Perfil”, se desplegará un formulario para permitir al usuario editar datos de su perfil tales como correo, nombres, apellidos, entre otros.



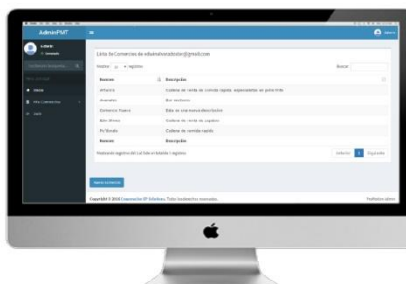


4 Modulo de Comercios

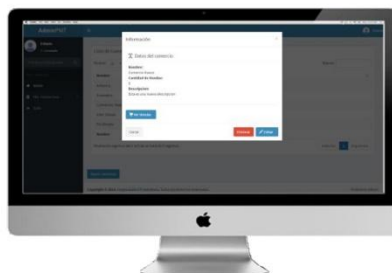
Como se explico anteriormente el usuario podrá agregar nuevos comercios de los cuales el es propietario. El usuario podrá agregar un nombre y la descripción del mismo.



Una vez guardado el nuevo comercio se desplegara una lista con todos los comercios asociados al usuario, incluyendo el comercio que se agrego previamente.



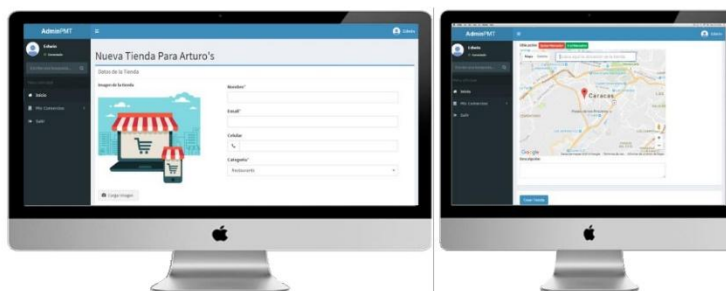
Si seleccionamos algún comercio de la lista desplegada se mostrara un detalle del comercio seleccionado con opciones para cerrar el detalle, editar y eliminar el comercio, además de un botón para ver las tiendas asociadas al comercio.



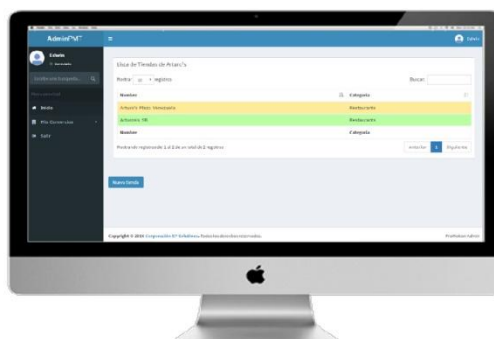


5 Modulo de Tiendas

El usuario podrá crear una tienda para un comercio en específico. La aplicación tendrá la capacidad de guardar la ubicación exacta de la tienda así como otros datos esenciales para el registro de la tienda, además de una imagen que identifique la misma.

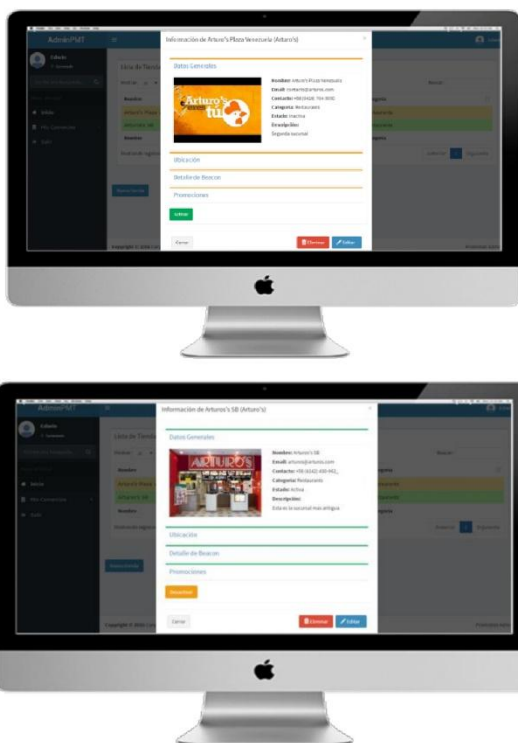


Una vez creada la nueva tienda se desplegará una lista con todas las tiendas asociadas al comercio seleccionado. En dicha lista se diferenciarán las tiendas que estén activas e inactivas, con el color verde y amarillo respectivamente.



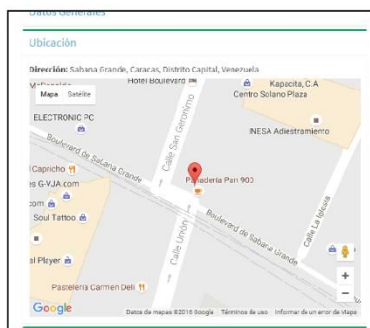


Si seleccionamos alguna tienda de la lista desplegada se mostrara el detalle de la tienda seleccionado con opciones para cerrar el detalle, editar y eliminar la tienda. En el caso de que la tienda este inactiva el detalle tendrá un aspecto de color amarillo y contendrá un botón para activar mientras que en el caso de que la tienda se encuentre activa, el detalle tendrá un aspecto verde y contendrá un botón amarillo para desactivar. El cuerpo del detalle será un acordeón que estará dividido en cuatro partes: Datos generales, Ubicación, Detalle de Beacon y Promociones.





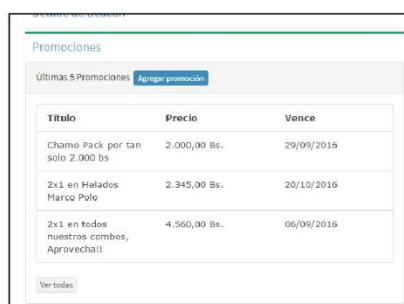
En el área de Datos generales se mostrara la información más importante de la tienda tales como nombre, correo, contacto, entre otros. En el área de ubicación de mostrara la dirección de la tienda y un mapa con la ubicación exacta de la tienda.



En el are de Detalle de Beacon se mostrar la información del dispositivo Beacon asignado a esa tienda, en caso de no poseer el dispositivo asignado en la tienda, se mostrara un aviso informándolo.



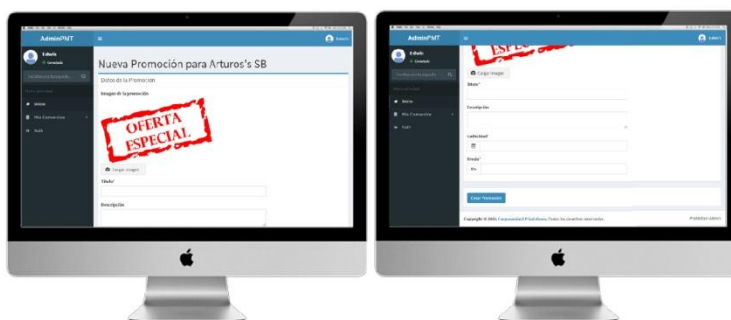
Por último para el área de Promociones se mostraran las últimas cinco promociones, ofreciendo la opción de agregar una nueva promoción y ver todas las promociones asociadas a esa tienda.



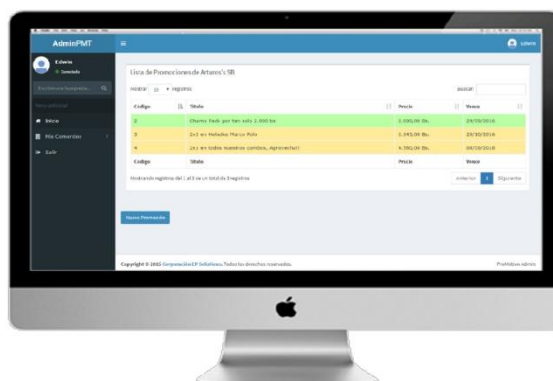


6 Módulo de Promociones

En este modulo el usuario tendrá la oportunidad de agregar promociones a una tienda que haya seleccionado previamente. Además podrá agregarle una imagen, un titulo, un precio, una descripción y una fecha de caducidad.

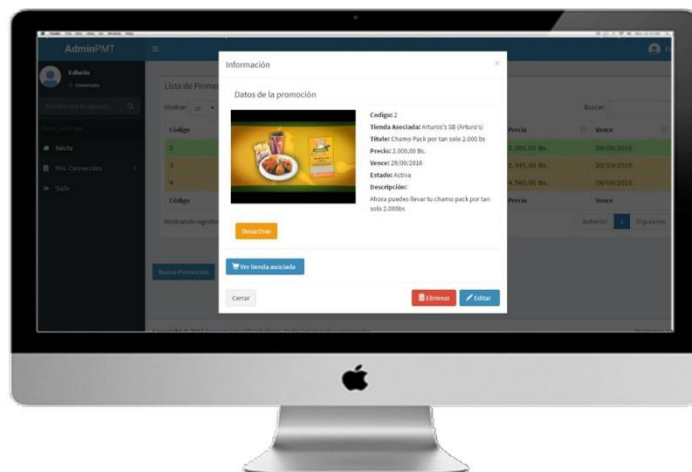


Una vez creada la nueva promoción se desplegará una lista con todas las promociones asociadas a la tienda seleccionada. En dicha lista se diferenciarán las promociones que estén activas e inactivas, con el color verde y amarillo respectivamente. Cabe destacar que solo podrá estar una promoción activa.





Si seleccionamos alguna promoción de la lista desplegada se mostrara un detalle de la promoción seleccionada con opciones para cerrar el detalle, editar y eliminar la promoción, además de un botón para ver la tienda asociada a la promoción.



7 Salir

Opción para salir de la aplicación y cerrar sesión.

