



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

APP DE VENTAS MÓVILES QUE PERMITE CONECTAR A CLIENTES
CON VENDEDORES AMBULANTES

AUTOR

Christian Rubén Aguilar Avilés

AÑO

2018



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

APP DE VENTAS MÓVILES QUE PERMITE CONECTAR A CLIENTES CON
VENDEDORES AMBULANTES

Trabajo de Titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingeniero en Sistemas de Computación e Informática

Profesor Guía

Mtr. Adonis Ricardo Rosales García

Autor

Christian Rubén Aguilar Avilés

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido el trabajo, App De Ventas Móviles Que Permite Conectar A Clientes Con Vendedores Ambulantes, a través de reuniones periódicas con el estudiante Christian Rubén Aguilar Avilés, en el semestre 2018-2, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

Adonis Ricardo Rosales García

Máster en Informática Aplicada

1756883144

DECLARACIÓN DEL PROFESOR CORRECTOR

Declaro haber revisado este trabajo, App De Ventas Móviles Que Permite Conectar A Clientes Con Vendedores Ambulantes, del estudiante Christian Rubén Aguilar Avilés, en el semestre 2018-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

Carlos Muñoz Cueva

Máster en Gerencia de Sistemas

1712981511

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

Christian Rubén Aguilar Avilés

1722949706

RESUMEN

El siguiente trabajo de titulación tiene como objetivo principal desarrollar una aplicación de ventas móviles que conecte a usuarios con vendedores ambulantes. De este objetivo general se tienen objetivos específicos, los cuales son realizar el estado del arte para el desarrollo de aplicaciones móviles con Android, desarrollar una aplicación web para gestionar reportes y configuraciones básicas de la aplicación y validar los resultados obtenidos tanto con pruebas de software como con experimentos de factibilidad. Para el desarrollo de este proyecto de titulación se escogieron las metodologías de Scrum y Lean Startup debido a que son metodologías ágiles que permiten tener retroalimentación continua para así poder corregir y direccionar de mejor manera el modelo de negocio. Con el desarrollo de este proyecto de titulación se espera que resulte menos complejo el encontrar vendedores cercanos a una determinada ubicación y optimizar el proceso de compra de productos y servicios de vendedores ambulantes.

ABSTRACT

The main objective of the following degree work is to develop a mobile sales application that connects users with street vendors. This general objective has specific objectives, which are to perform the state of the art for the development of mobile applications with Android, develop a web application to manage reports and basic configurations of the application and validate the results obtained with both software tests and with feasibility experiments. For the development of this titling project, the Scrum and Lean Startup methodologies were chosen because they are agile methodologies that allow continuous feedback to correct and direct the business model better. With the development of this titling project it is expected that it will be less complex to find vendors close to a certain location and optimize the process of purchasing products and services from street vendors.

Índice

Introducción	1
Antecedentes	1
Objetivo General.....	5
Objetivos Específicos	5
1. Capítulo I. Marco Teórico	5
1.1. Introducción	5
1.2. Metodologías	6
1.2.1. Metodología Scrum.....	6
1.2.2. Metodología Lean Startup	7
1.3. Lenguaje de Programación Y Herramientas.....	8
1.3.1. Lenguaje Java	8
1.3.2. Cloud Computing	9
1.3.3. Google Android Studio.....	10
1.3.4. Google Firebase	10
1.4. Diagrama de la Solución Y Explicación.....	11
1.5. Conclusión del Capítulo	14
2. Capítulo II. Actualidad	14
2.1. Introducción	14
2.2. Situación Actual	14
2.2.1. Encuestas	14

2.3. Diseño de la Solución.....	16
2.3.1. Historias de Usuario	16
2.3.2. Ciclo de Vida Ágil.....	24
2.3.3. Sprint	26
2.4. Conclusión del Capítulo	47
3. Capítulo III. Entorno	48
3.1. Introducción	48
3.2. Entorno Aplicación Móvil	48
3.2.1. Entorno Aplicación Móvil Vendedor	48
3.2.2. Entorno Aplicación Móvil Cliente	55
3.3. Entorno Aplicación Web.....	63
3.4. Resultados de la Aplicación Móvil	65
3.5. Análisis Económico	65
4. Conclusiones Y Recomendaciones.....	66
4.1. Conclusiones	66
4.2. Recomendaciones.....	66
Referencias	67
Anexos.....	71

Introducción

✓ Antecedentes

Para hablar del comercio hay que volver a la era neolítica, donde comenzó la actividad de la agricultura. En aquella época, esta actividad solo se la realizaba para la subsistencia de las personas que se dedicaban de lleno lo que permitió el comienzo del proceso de civilización. Posteriormente, en la edad antigua, al ser más complejo el medio de pago de las mercancías, se creó el trueque. El trueque se utilizaba para comercializar productos entre civilizaciones realizando intercambios de productos o mercancías de igual valor (Bonilla, 2013). Al llegar a la edad media, se crean rutas comerciales, las cuales se utilizaban para reactivar la economía en las diferentes regiones. Después, llegó la edad del descubrimiento en la cual, al crear las rutas comerciales, se pudo descubrir el continente americano y la India, lo que permitió obtener mayor variedad de productos y mercancías (Trujillo, 2015). Ya para el siglo XX se llega a la época de la globalización, donde se crean zonas de libre comercio para así rebajar costos de producción y que los comerciantes puedan ser competitivos entre sí (Bonilla, 2013).

Y para el siglo XXI, llega la era tecnológica donde se crea el comercio electrónico, también denominado *eCommerce* (Trujillo, 2015). En ese entonces, las empresas introdujeron un nuevo canal para realizar ventas que se basaba en Internet. En sus inicios, el comercio electrónico permitió realizar las ventas de bienes o servicios a través de la *World Wide Web* (www) empleando servicios de pago electrónico como lo son las aprobaciones de las tarjetas de crédito al momento de realizar las compras (Bonilla, 2013).

Actualmente, si se quiere referir al comercio electrónico, se lo puede hacer como *mCommerce*. Esto se debe a que el uso de aplicaciones móviles ha ido creciendo a un ritmo acelerado y que los dispositivos móviles han ido mejorando sus tecnologías de conectividad y una mayor confianza por parte de sus usuarios al momento de utilizarlos. Además, las plataformas de pago han ido creciendo y actualmente se

pueden encontrar en cualquier dispositivo móvil como, por ejemplo, Android Pay, Samsung Pay o Apple Pay (Oliveras, 2017).

En los últimos años, las aplicaciones móviles y aplicaciones web han ganado bastante popularidad. Cada día, las personas utilizan más sus teléfonos inteligentes para realizar gestiones, comprar productos, navegar por redes sociales. Las aplicaciones móviles han llegado para facilitar la vida a las personas. En la actualidad, gracias a la evolución de la tecnología las personas cuentan con una gran variedad de herramientas que buscan sacar la máxima eficiencia a las actividades diarias. Últimamente, prefieren utilizar aplicaciones móviles sobre una página o aplicación web debido a su facilidad de uso y comodidad (Alfocea, 2017).

Existen diferentes tipos de aplicaciones, los cuales son: Aplicaciones Web, Aplicaciones Nativas y Aplicaciones Híbridas (Pimienta, s.f.). Las aplicaciones web son las que más se utilizan debido a que se pueden acceder desde cualquier dispositivo sin importar el sistema operativo que utilizan, basta con tener un navegador y acceder a ellas. Dentro de sus ventajas, no requieren de aprobaciones para su publicación, cuentan con mayor número de plataformas soportadas, su código fuente puede ser reutilizado. Por otro lado, sus desventajas son que no utilizan de manera óptima los recursos del dispositivo, su rendimiento es menor, su interacción con el Hardware del dispositivo es limitada. Las aplicaciones nativas son aquellas que se desarrollan para un sistema operativo específico, como Android, iOS. Estas aplicaciones son exclusivamente desarrolladas para dispositivos móviles, es decir, teléfonos inteligentes o tabletas que utilizan los sistemas operativos mencionados. Sus ventajas son que pueden ser publicadas en las tiendas de Android (Play Store) e iOS (App Store), brindan una mejor experiencia para el usuario, no necesitan todo el tiempo una conexión a Internet. Sus desventajas son que pueden ser utilizadas solamente en el sistema operativo para el que fueron desarrolladas, costos y tiempo de desarrollo altos, aprobación para ser publicadas. Las aplicaciones híbridas son aquellas que tienen características de las dos aplicaciones anteriormente detalladas. Estas aplicaciones son

multiplataformas, utilizan las tecnologías web y corren localmente en un dispositivo sin necesidad de estar conectadas a Internet todo el tiempo. Este tipo de aplicaciones al igual que las nativas, pueden ser distribuidas en las tiendas Android e iOS (Pimienta, s.f.).

Dentro de estas aplicaciones, es posible conectarse utilizando redes sociales como Facebook, Twitter, Google+ o LinkedIn. Las categorías más populares para las aplicaciones son juegos, redes sociales, productividad, salud, educación, viajes, multimedia, entretenimiento. Cada día existen más aplicaciones que ayudan en las actividades diarias de las personas, por ejemplo, existen aplicaciones como Uber o Airbnb donde los propios usuarios son los que ayudan a estas empresas a mantenerse. Estas aplicaciones conectan usuarios que ofrecen un servicio con otros usuarios que quieren utilizar esos servicios. Para el año 2018 se espera que el mercado de aplicaciones móviles genere ciento diez mil millones de dólares. Este mercado es muy competitivo ya que, actualmente existen 5.5 millones de aplicaciones (García, 2017).

Las aplicaciones pueden ser gratuitas en su totalidad, otras son aplicaciones con compras in-app, es decir, dentro de la aplicación se puede pagar por mejores características en la aplicación. Existen también aplicaciones que muestran publicidad dentro de ella y aplicaciones bajo suscripción que, a través de un pago mensual, se pueden utilizar características avanzadas de la aplicación. Dentro de estas aplicaciones, se pueden realizar compras y mediante el ingreso de información de tarjetas de crédito o débito, se pueden pagar productos o servicios. Además, se pueden agregar plataformas de pago como PayPal o PayPhone que, mediante el ingreso del usuario y contraseña de esta plataforma, ya se puede realizar la compra directamente sin ingresar datos de tarjetas de crédito en las aplicaciones móviles, o aplicaciones web (Iglesias, 2016).

Aplicaciones como Uber, Cabify o AdomicilioYa se mantienen en el mercado aun cuando son gratuitas debido a que utilizan el método de permitir descargar la aplicación sin costo, pero para utilizar completamente su aplicación, brindan

servicios extras que tienen un costo adicional. Uber o Cabify, brindan el servicio de taxis a sus usuarios, sin embargo, los mismos usuarios se convierten en choferes de otros usuarios. Mientras que la aplicación de AdomicilioYa, brinda el servicio de llevar pedidos de comida a sus usuarios hacia la ubicación donde se encuentren por un costo adicional a su pedido. Estos costos resultan ser comisiones que por el uso de la aplicación se cobra a sus usuarios. De los valores que se cobran, dependiendo la aplicación varían los porcentajes que se lleva la empresa y el conductor en caso de Uber y Cabify, y repartidor en el caso de AdomicilioYa (Álvarez, 2017).

Las necesidades de la población a medida que pasa el tiempo han ido incrementando haciendo que resulte complicado obtener directamente algunos servicios o productos específicos. Sin embargo, existen personas que realizan recorridos ofreciendo diferentes productos, por ejemplo, venta de cilindros de gas GPL, venta de botellones de agua mineral, compra y recolección de chatarra o lavandería.

Si bien es cierto, existen puntos fijos de venta de los productos, sin embargo, resulta más sencillo conocer que persona puede entregarlo a domicilio mediante una aplicación móvil. En la actualidad, el proceso de compra de estos productos o el uso de otros servicios se limita a cuando el vendedor pase por la ubicación del cliente, lo que significa una complejidad moderada para adquirir uno de estos productos o servicios. Algunos vendedores utilizan pitos, canciones y/o anuncios para dar a conocer lo que están vendiendo o simplemente para avisar que están en el sector, lo cual resulta molesto para la mayoría de las personas ya que estos ruidos se encuentran en un volumen alto y que, además no han solicitado ningún producto o servicio.

Atendiendo a la situación anterior, se considera que el **problema** a resolver es: ¿Cómo integrar los servicios de venta ambulante de productos con una tecnología que permita su consumo a domicilio? Donde se tienen los siguientes objetivos:

✓ Objetivo General

Desarrollar una aplicación de ventas móviles que conecte a usuarios con vendedores ambulantes.

✓ Objetivos Específicos

- Realizar el estado del arte para el desarrollo de aplicaciones móviles con Android.
- Desarrollar una aplicación web para gestionar reportes y configuraciones básicas de la aplicación.
- Validar los resultados obtenidos tanto con pruebas de software como con experimentos de factibilidad.

Con el desarrollo de la aplicación móvil, se espera que el proceso de compra de estos productos mejore y optimice el tiempo de vendedores y clientes, ya que al poseer una herramienta que permita visualizar los vendedores cercanos a la ubicación de un cliente, este no debe estar pendiente del vendedor una gran cantidad de tiempo para no perder la oportunidad de comprar el producto.

1. Capítulo I. Marco Teórico

1.1. Introducción

En el siguiente capítulo, se describirán las metodologías y herramientas que se van a utilizar para desarrollar el proyecto. Se describirán las metodologías de Scrum y Lean Startup, las cuales son utilizadas por equipos de desarrollo para realizar reuniones semanales en las que se trata los avances del proyecto y realizar diferentes pruebas antes de lanzar un nuevo producto recibiendo retroalimentación, respectivamente. Así también, se describirá el funcionamiento de las herramientas y como se van a relacionar con las aplicaciones.

1.2. Metodologías

1.2.1. Metodología Scrum

Scrum es una de las metodologías ágiles que tiene como objetivo reducir la complejidad de los proyectos y así lograr satisfacer las necesidades del cliente (Francia, 2017). Con Scrum se busca la aplicación de buenas prácticas para trabajar colaborativamente para obtener el mejor resultado posible (Pérez, 2015).

Scrum cuenta con eventos, los cuales son bloques de tiempo que tienen una duración máxima y que no pueden ser acortados ni alargados. Estos eventos se los llama Sprint. Al final de cada Sprint, se tiene como objetivo obtener un entregable funcional que se presentará al cliente y este lo validará dando retroalimentación al equipo para el siguiente Sprint (Schwaber, Ken; Sutherland, Jeff, 2016, pp. 8,9).

En Scrum existen roles como El Equipo Scrum, el cual está conformado por el Dueño de Producto, el Equipo de Desarrollo y el Scrum Master. El Equipo Scrum es un equipo autoorganizado y multifuncional que no depende de terceras personas ni son dirigidos por personas externas al equipo. El Equipo Scrum busca tener creatividad, flexibilidad y aumentar la productividad del trabajo. El Dueño de Producto es el encargado de aumentar el valor del producto y del trabajo realizado por el Equipo Scrum. Por su parte el Equipo de Desarrollo es aquel que se encarga de entregar un producto terminado al cliente al final de cada Sprint, el cual es potencialmente puesto en producción. Y el Scrum Master es que se encarga de que Scrum se entienda y se adopte en el equipo y en el proyecto. Además, es el encargado de ser el nexo entre el Equipo Scrum y las personas externas al equipo y lograr una comunicación eficaz entre todas las partes interesadas (Schwaber, Ken; Sutherland, Jeff, 2016, pp. 5,6).

Los beneficios de utilizar Scrum son la reducción del costo de desarrollo de un proyecto cuando este se ha incrementado demasiado, la reducción del tiempo de entrega del proyecto, resolución de problemas complejos, facilidad de adaptación del equipo de trabajo al estilo que plantea, creación de un ambiente laboral

colaborativo, motivación del equipo de trabajo, retroalimentación continua por parte del cliente, actualización del estado del proyecto hacia al cliente, maximiza la productividad del equipo de trabajo y maximiza el retorno de inversión (ROI) de la empresa (Queralt, 2014).

Por lo tanto, al utilizar esta metodología, se podrá visualizar y medir el avance del proyecto mediante entregables funcionales, los cuales se entregarán idealmente de manera semanal y así como también recibir retroalimentación para mejorar el proyecto.

1.2.2. Metodología Lean Startup

Lean Startup es una metodología que se centra en la creación de nuevas empresas, un producto o servicio. El objetivo de Lean Startup es reducir lo más posible el riesgo, los gastos innecesarios de tiempo, dinero y recursos. La metodología Lean Startup fue ideada por Eric Ries y fue basada en otras metodologías como la de Toyota, Just Intime, la cual tiene como objetivo reducir todas las perdidas posibles en recursos e inclusive en los procesos de manufactura de la empresa (Touza, 2017).

El modelo de esta metodología consiste en presentar prototipos rápidos al cliente para obtener una visión más amplia del producto final, reducir los gastos y realizar encuestas para analizar el mercado y así poder determinar la factibilidad del producto o servicio. Al momento de realizar prototipos rápidos y lanzarlos al mercado, Lean Startup busca poder medir los resultados que se obtuvieron con ese producto mínimo. Además, ayuda a que la empresa pueda aprender de manera rápida el mercado, que es lo que quiere la gente y como, con un corto ciclo iterativo y así poder mejorar hasta encontrar el producto final que satisfaga las necesidades de una o un grupo de personas (Scipion, s.f.).

Por lo tanto, las tres fases de esta metodología son: construir, medir, aprender. Se construye un producto mínimo viable centrado en una hipótesis, se miden resultados utilizando métricas de acuerdo con el producto lanzado y finalmente se obtiene

información con detalles que servirán para seguir mejorando el producto (Sánchez, 2016).

Sus principales ventajas son eliminar o reducir el riesgo de mayor inversión, visualizar resultados a corto plazo, crear un producto que cumpla con las necesidades de la población y estructurar de manera sólida una idea (Hernández, 2016).

Después de conocer cómo funciona la metodología Lean Startup, con su aplicación dentro de este proyecto, se podrá obtener retroalimentación real por parte de usuarios y así poder direccionar de mejor manera el modelo de negocio para que las aplicaciones puedan tener un efecto positivo dentro de la sociedad.

1.3. Lenguaje de Programación Y Herramientas

1.3.1. Lenguaje Java

Java es un lenguaje de programación que utiliza la programación orientada a objetos (POO), además, es un lenguaje derivado de otro lenguaje de programación llamado C. Para su estructura utiliza paquetes que contienen las clases. Dentro de las clases se encuentran los métodos y variables (Perry, 2012). Es muy utilizado en la actualidad en las empresas en la programación de sistemas, aplicaciones cliente, aplicaciones cliente / servidor o en aplicaciones web. Por otro lado, Java es utilizado para programar aplicaciones móviles para el sistema operativo Android, que pertenece a Google (Villalobos, 2010).

Las ventajas de utilizar Java para la programación son la programación orientada a objetos, la que se asemeja a la forma de pensar de un humano; el programador no necesita liberar memoria ya que el lenguaje mismo se encarga de esa tarea al momento que el programador necesita más memoria, existen varios editores (IDE's) en los que se puede programar, es un lenguaje muy robusto y es independiente de la plataforma en la que se encuentra el programador, es decir, multiplataforma (Cárdenas, 2016), es software libre, es decir, no es necesario pagar una licencia

para utilizarlo, posee una gran comunidad de desarrolladores que aportan en foros y ofrecen ayuda a otros desarrolladores (Villalobos, 2010).

1.3.2. Cloud Computing

Cloud Computing o solamente Cloud, es una tecnología que permite a usuarios, pequeñas, medianas y grandes empresas acceder a servicios de software, infraestructura y procesamiento de datos a través de Internet de manera remota. Además, es la entrega de servicios bajo demanda a través de Internet con un sistema de pago dependiendo el consumo realizado (Lamata, 2012).

Existen diferentes modelos de Cloud que son, SaaS (Software Como Servicio), IaaS (Infraestructura Como Servicio) y PaaS (Plataforma Como Servicio). SaaS se centra en proveer software para que sus usuarios lo utilicen directamente desde una aplicación web o una aplicación móvil. Los datos son almacenados en la infraestructura de las grandes empresas como Google, Amazon, IBM o Microsoft. IaaS provee el servicio de infraestructura (Hardware) para que las empresas utilicen recursos computacionales como servidores, redes o almacenamiento de datos con un costo según su consumo. Y PaaS provee lo que es Hardware y Software a las empresas para que solo se concentren en implementar sus aplicaciones y llegar al mercado de manera más rápida y eficaz (Argudo, 2015)

Actualmente, el uso de Cloud Computing resulta para las empresas un beneficio enorme ya que, los gastos de inversión se convierten en gastos variables debido a que se paga según el consumo que tiene la empresa y no tienen que gastar en su propia infraestructura. Por otro lado, las empresas pueden escalar sus aplicaciones para ofrecer su servicio a más usuarios sin ningún problema. También, mediante el uso de estas tecnologías, las aplicaciones y servicios pueden ser mundiales y no limitarse a una región específica. Otra ventaja es, la disponibilidad de información, es decir, cualquier persona autorizada puede acceder a la información desde cualquier parte del mundo y desde cualquier dispositivo, y existe un nivel mayor de colaboración entre equipos en proyectos (Oviedo, 2015).

1.3.3. Google Android Studio

Android Studio es el entorno de desarrollo (IDE) de aplicaciones móviles para el sistema operativo Android y pertenece a Google. Además, cuenta con un emulador de teléfonos móviles de Google, que sirve para probar las aplicaciones que están en desarrollo en caso de no tener un teléfono físico. Android Studio también sirve para desarrollar aplicaciones para relojes (*Smart Watch*), incluso se pueden desarrollar aplicación para Android TV o Android Car (Android, s.f.).

La principal ventaja de este IDE es que, al ser propietario Google, así como también del sistema operativo Android, se vuelve el IDE ideal para desarrollar aplicaciones. Android Studio ofrece diferentes herramientas que se pueden integrar con otras herramientas como Firebase, para desarrollar aplicaciones nativas. Incluso en el mismo IDE, se pueden realizar las pruebas necesarias para verificar que la aplicación pueda funcionar sin problemas en los teléfonos móviles (Android, s.f.).

Al utilizar la herramienta Android Studio, se puede desarrollar aplicaciones nativas para el Sistema Operativo Android, por lo que resultará menos complicado su integración al momento de realizar pruebas.

1.3.4. Google Firebase

Firebase es una plataforma de desarrollo de aplicaciones móviles, propiedad de Google. Firebase se encuentra disponible para las diferentes plataformas como lo son Android, iOS y Web. Firebase es una plataforma utilizada en la Nube (Cloud), lo que permite acceder a la información desde cualquier dispositivo (Zamora, 2016).

Firebase, además de ser una plataforma de desarrollo, es un conjunto de herramientas que tiene como objetivo crear aplicaciones de alta calidad, escalable para los usuarios, obtener estadísticas en tiempo real del funcionamiento de la aplicación y lograr obtener ingresos, ya sea mediante el cobro de un valor por la descarga o el uso de un aplicación o a través de publicidad (Ruiz, 2017).

Firebase es una herramienta muy útil para los proyectos que deben ser realizados en un tiempo limitado gracias a que posee su propia API (Interfaz de Programación

de Aplicaciones) que resulta intuitiva y es sostenida en un SDK (Kit de Desarrollo de Software) y es una herramienta ágil debido a que se pueden integrar API de terceras partes a una aplicación, por ejemplo, Google Maps (Sanz, 2017).

Las principales herramientas que provee Firebase a los desarrolladores son Base de Datos en Tiempo Real, Autenticación de Usuarios, Almacenamiento, Laboratorio de Test, Informes Sobre Fallos, Monitoreo de Rendimiento y Notificaciones (Peña, 2018).

Entre sus principales ventajas están la seguridad (respaldo de Google), integración de infraestructura, sincronización de datos y medición y cuantificación de la información de la aplicación, útil para proyectos que no tienen una fuerte inversión inicialmente y facilidad para el desarrollo en tiempo limitado (El Asri, 2016).

El uso de Firebase como una base de datos, una herramienta de almacenamiento y al estar basada en la Nube, permite obtener escalabilidad y alta disponibilidad de información para el uso de las aplicaciones. Además, al ser una herramienta que pertenece a una gran empresa como Google, resultará más fácil la integración con otras herramientas de Google, como Android Studio.

1.4. Diagrama de la Solución Y Explicación

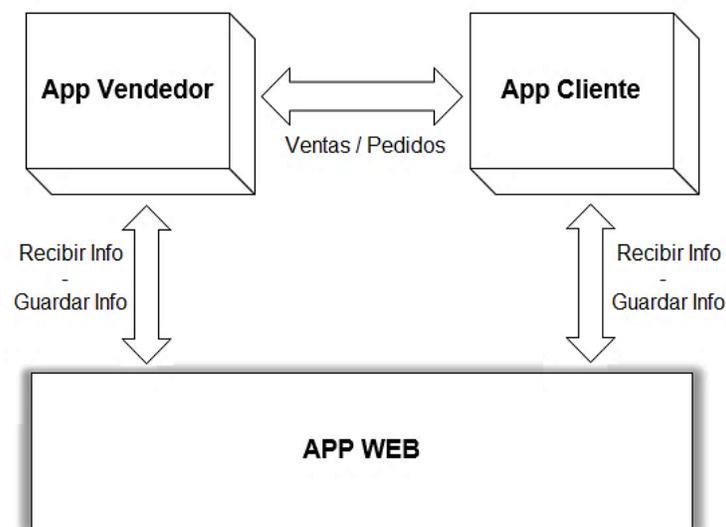


Figura 1.- Diagrama de la estructura general del proyecto.

La estructura del proyecto consiste en el desarrollo de tres aplicaciones: una app para los vendedores, una app para clientes y una app web que servirá como panel de control para mostrar los vendedores y clientes registrados, así como también los pedidos realizados a través de la aplicación y calificaciones registradas de los clientes sobre el servicio de los vendedores.



Figura 2.- Estructura de la aplicación para Vendedores

En la aplicación de vendedor, el Vendedor podrá establecer su estado de activo y no activo para poder recibir los pedidos. También, mientras se encuentre activo, se podrá visualizar su ubicación en el mapa de su aplicación. Además, al momento de recibir un pedido, recibirá una alerta en la que se mostrará el tiempo y distancia hacia la ubicación del Cliente, así como la dirección de su ubicación. Posteriormente, tendrá la posibilidad de aceptar el pedido o cancelarlo. Al llegar a la ubicación del cliente, se enviará una notificación para hacerle saber que debe recibir el producto. Una vez entregado, se finalizará la entrega.



Figura 3.- Estructura de la aplicación para Clientes

En la aplicación de cliente, el Cliente podrá visualizar los vendedores disponibles que se encuentren cerca de su ubicación. Posteriormente, podrá realizar el pedido donde encontrará al vendedor más cercano. Al momento que el Vendedor llegue a la su ubicación, recibirá una notificación para que retire el pedido. Finalmente, el Vendedor finalizará el pedido y el Cliente deberá calificar el servicio.



Figura 4.- Estructura de la aplicación Web

El administrador podrá visualizar un listado de todos los usuarios registrados a través de las aplicaciones. Además, podrá visualizar el listado por separado de

Vendedores y Clientes registrados, con su respectiva información como nombres, apellidos, teléfono.

1.5. Conclusión del Capítulo

En conclusión, al utilizar las metodologías Scrum y Lean Startup, se puede llevar un proceso de desarrollo del proyecto más productivo y organizado. Además, al utilizar las herramientas de desarrollo pertenecientes a Google como lo son Android Studio y Firebase, se podrá tener alta disponibilidad de información y escalabilidad para atender las peticiones de los usuarios con el respaldo de una gran empresa. Y lo más importante, son herramientas que, para el desarrollo de este proyecto, son gratuitas.

2. Capítulo II. Actualidad

2.1. Introducción

En el siguiente capítulo, se mostrará los resultados de las encuestas realizadas a vendedores y a clientes utilizando la metodología de Lean Startup. Con la aplicación de esta metodología, se busca conocer la situación actual del mercado para así poder direccionar de manera correcta y efectiva el proyecto.

2.2. Situación Actual

2.2.1. Encuestas

NOTA: El modelo de las encuestas se encuentra en la Sección [ANEXOS](#).

A continuación, se muestran los resultados obtenidos una vez realizadas las encuestas.



Figura 5.- Resultados de las encuestas realizadas a vendedores ambulantes

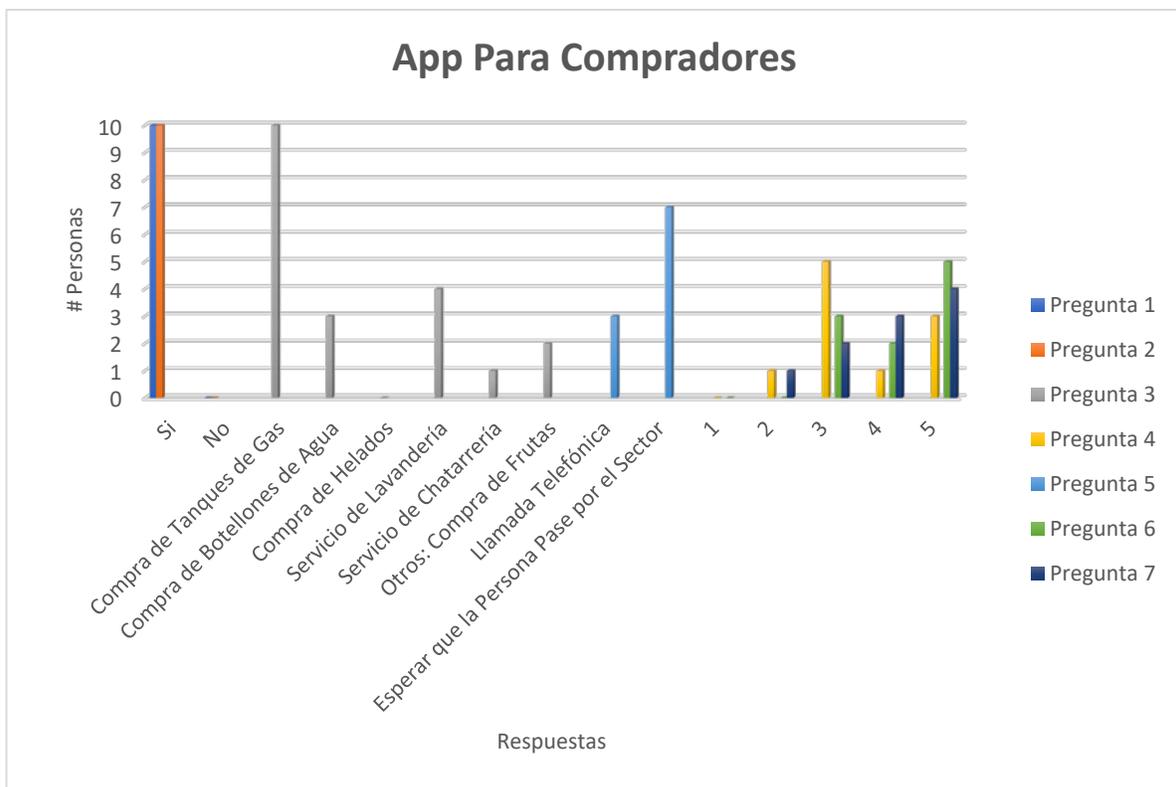


Figura 6.- Resultados de las encuestas realizadas a compradores

Como se pudo apreciar después de haber consultado con algunos vendedores y clientes, existe interés en el uso de las aplicaciones para mejorar el servicio de ventas y servicios ambulantes como venta de tanques de gas, venta de botellones de agua, servicio de lavandería. Al realizar las aplicaciones se optimizará este proceso y resultará más fácil poder encontrar vendedores cercanos a una determinada ubicación de los clientes.

2.3. Diseño de la Solución

2.3.1. Historias de Usuario

Una historia de usuario es una descripción de cada funcionalidad que va a tener un sistema o una aplicación. Estas historias de usuario tienen como característica principal que se detalla la o las personas que van a ser las responsables realizar la tarea, la estimación de tiempo para realizarla, la prioridad que tiene la funcionalidad en el desarrollo del proyecto y una validación al momento de probar la funcionalidad con el cliente o usuario. Estas historias de usuario pueden ser modificadas mínimamente a lo largo del desarrollo del proyecto y pueden cambiar el orden de presentación o de prioridad según las necesidades del cliente, previamente acordadas y planificadas. (Quijano, 2017)

A continuación, se presentan las definiciones de los elementos de una historia de usuario.

- **ID:** identificador de la historia de usuario, **único** para la funcionalidad o trabajo.
- **Usuario:** de dónde se obtuvo.
- **Nombre de historia:** nombre descriptivo de la historia de usuario.
- **Prioridad en negocio:** sistema de priorización que nos permite determinar el orden en el que las historias de usuario deben de ser implementadas.
- **Riesgo en desarrollo:** riesgo **técnico o funcional** asociado a la implementación de la historia de usuario.

- **Estimación:** estimación del esfuerzo necesario ideal de desarrollo de la historia de usuario. Según convenga, el equipo también puede utilizar unidades de desarrollo conocidas como puntos de historia (estas unidades representan el esfuerzo de desarrollo/persona que se estipule al comienzo del proyecto).
- **Sprint asignado:** puede ser útil para organización del propietario del producto incluir el **número de sprint** en el que previsiblemente se vaya a realizar la historia.
- **Responsable:** en casos en que queramos sugerir la persona que pueda implementar la historia de usuario. Recordar que en Scrum es en último término el equipo autogestionado quién distribuye y por tanto asigna las tareas.
- **Descripción:** descripción sintetizada de la historia de usuario. El estilo puede ser libre, según mejor nos funcione, debe responder a tres preguntas: ¿Quién se beneficia? ¿Qué se quiere? y ¿Cuál es el beneficio? Se recomienda seguir el siguiente patrón que garantiza que la funcionalidad está descrita a un alto nivel y de una manera no demasiado extensa:

Como [rol del usuario], quiero [objetivo], para poder

- **Validación:** pruebas de aceptación consensuadas con el cliente o usuario. Estas son las pruebas que el código debe superar para dar como finalizada la implementación de la historia de usuario.

Tabla 1

Historia de usuario 1

Historia De Usuario	
Id: 1	Usuario: Christian Aguilar
Nombre De Historia: Login & Layout Mapa Vendedor	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A

Estimación: Una Semana	Sprint Asignado: 1
Responsable: Christian Aguilar	
Descripción: Como vendedor, deseo visualizar mi ubicación actual en el mapa, para poder recibir pedidos.	
Validación: Una vez registrado el vendedor, al acceder a la aplicación puede visualizar su posición exacta en el mapa.	

Tabla 2

Historia de usuario 2

Historia De Usuario	
Id: 2	Usuario: Christian Aguilar
Nombre De Historia: Login & Layout Mapa Cliente	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 1
Responsable: Christian Aguilar	
Descripción: Como cliente, deseo visualizar mi ubicación en el mapa, para poder recibir mi pedido.	
Validación: Una vez registrado el cliente, al acceder a la aplicación puede visualizar su posición exacta en el mapa.	

Tabla 3

Historia de usuario 3

Historia De Usuario	
Id: 3	Usuario: Christian Aguilar
Nombre De Historia: Realizar Pedido Cliente	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 4
Responsable: Christian Aguilar	

Descripción: Como cliente, deseo tener un botón, para poder realizar un pedido.
Validación: Al ingresar a la aplicación, el cliente puede hacer clic en el botón “Realizar Pedido” para que aparezca la opción solicitar servicio. Posteriormente al hacer clic en este botón comienza la búsqueda del vendedor más cercano para solicitar el servicio.

Tabla 4

Historia de usuario 4

Historia De Usuario	
Id: 4	Usuario: Christian Aguilar
Nombre De Historia: Encontrar y Mostrar Vendedores Disponibles	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 3
Responsable: Christian Aguilar	
Descripción: Como cliente, deseo visualizar en el mapa los vendedores disponibles, para poder recibir mi pedido del vendedor más cercano.	
Validación: Al momento de ingresar a la aplicación, el cliente puede observar su ubicación actual y a su vez puede visualizar en el mapa los vendedores disponibles.	

Tabla 5

Historia de usuario 5

Historia De Usuario	
Id: 5	Usuario: Christian Aguilar
Nombre De Historia: Notificación Pedido Vendedor	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 5
Responsable: Christian Aguilar	
Descripción: Como vendedor, deseo recibir una alerta, para poder entregar el pedido recibido.	

Validación: El momento en que un cliente realiza un pedido, encuentra al vendedor más cercano a su ubicación. En el instante en que el pedido se asocia al vendedor más cercano, el vendedor recibe una alerta con información del cliente como la dirección, la distancia y el tiempo hacia dicha ubicación.

Tabla 6

Historia de usuario 6

Historia De Usuario	
Id: 6	Usuario: Christian Aguilar
Nombre De Historia: Control Vendedores Online & Offline	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 2
Responsable: Christian Aguilar	
Descripción: Como vendedor, deseo tener la opción de estar disponible o no disponible, para poder decidir el horario de venta.	
Validación: Al momento de ingresar a la aplicación, el vendedor visualiza un switch el cual sirve para cambiar el estado de En Línea a Fuera de Línea y viceversa.	

Tabla 7

Historia de usuario 7

Historia De Usuario	
Id: 7	Usuario: Christian Aguilar
Nombre De Historia: Aceptar & Cancelar Pedido Vendedor	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 5
Responsable: Christian Aguilar	
Descripción: Como vendedor, deseo tener la opción de aceptar o rechazar un pedido, para poder controlar las ventas.	

Validación: Al momento de recibir un pedido, al vendedor le aparece una alerta con información del cliente y dos botones con los cuales podrá aceptar o cancelar dicho pedido.

Tabla 8

Historia de usuario 8

Historia De Usuario	
Id: 8	Usuario: Christian Aguilar
Nombre De Historia: Notificación Cuando Vendedor Llegue al Punto	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 6
Responsable: Christian Aguilar	
Descripción: Como cliente, deseo recibir una alerta o notificación cuando el vendedor llegue a mi punto, para poder recibir el pedido.	
Validación: Al momento en que el vendedor llega a la ubicación del cliente, este recibe una notificación push la cual avisa que debe salir a recibir su pedido.	

Tabla 9

Historia de usuario 9

Historia De Usuario	
Id: 9	Usuario: Christian Aguilar
Nombre De Historia: Layout Salir & Cambio de Contraseña	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 6
Responsable: Christian Aguilar	
Descripción: Como vendedor o cliente, deseo tener la opción de modificar mi contraseña, para tener mayor seguridad.	

Validación: Cuando se ingresa a la pantalla principal, cada una de estas aplicaciones cuenta con un menú lateral el cual contiene diferentes opciones. En el menú tiene la posibilidad de cerrar sesión y cambiar contraseña. Al momento en que selecciona actualizar contraseña, se despliega una ventana en la cual el usuario debe ingresar la contraseña actual, la nueva contraseña e ingresarla nuevamente para confirmar el cambio. Después de actualizar la contraseña, el usuario puede cerrar sesión e ingresar nuevamente ingresando la nueva contraseña.

Tabla 10

Historia de usuario 10

Historia De Usuario	
Id: 10	Usuario: Christian Aguilar
Nombre De Historia: Layout Actualizar Información Vendedor	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 7
Responsable: Christian Aguilar	
Descripción: Como vendedor, deseo poder modificar mi información, para poder corregirla en caso de estar errónea.	
Validación: Cuando se ingresa a la pantalla principal, la aplicación cuenta con un menú lateral el cual contiene diferentes opciones. En el menú tiene la posibilidad de actualizar el perfil el vendedor. Al momento en que selecciona actualizar perfil, se despliega una ventana en la cual el vendedor debe ingresar un nuevo nombre o el mismo y seleccionar una imagen de su teléfono. Cuando selecciona una imagen, esta se sube a la base de datos para actualizar en el perfil del vendedor. Después de confirmar, el vendedor debe cerrar sesión y volver a ingresar para visualizar el cambio.	

Tabla 11

Historia de usuario 11

Historia De Usuario	
Id: 11	Usuario: Christian Aguilar
Nombre De Historia: Layout Actualizar Información Cliente	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 8
Responsable: Christian Aguilar	
Descripción: Como cliente, deseo poder modificar mi información, para poder corregirla en caso de estar errónea.	
Validación: Cuando se ingresa a la pantalla principal, la aplicación cuenta con un menú lateral el cual contiene diferentes opciones. En el menú tiene la posibilidad de actualizar el perfil el cliente. Al momento en que selecciona actualizar perfil, se despliega una ventana en la cual el cliente debe ingresar un nuevo nombre o el mismo y seleccionar una imagen de su teléfono. Cuando selecciona una imagen, esta se sube a la base de datos para actualizar en el perfil del cliente. Después de confirmar, el cliente debe cerrar sesión y volver a ingresar para visualizar el cambio.	

Tabla 12

Historia de usuario 12

Historia De Usuario	
Id: 12	Usuario: Christian Aguilar
Nombre De Historia: Mantener la Sesión Aun Cerrando la App	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 7
Responsable: Christian Aguilar	
Descripción: Como vendedor o cliente, deseo poder mantener mi sesión abierta, para poder utilizar la aplicación aun cuando la cierre.	

Validación: Al momento de iniciar sesión, el usuario puede cerrar la aplicación totalmente y la sesión se mantiene en el último estado, Es decir, si el usuario tenía iniciada la sesión en la aplicación, al momento de volver a ingresar, no hace falta volver a ingresar sus datos.

Tabla 13

Historia de usuario 13

Historia De Usuario	
Id: 13	Usuario: Christian Aguilar
Nombre De Historia: Layout Rating Vendedor	
Prioridad En Negocio:	Riesgo En Desarrollo: N/A
Estimación: Una Semana	Sprint Asignado: 7
Responsable: Christian Aguilar	
Descripción: Como cliente, deseo poder calificar al vendedor, para poder ver posteriormente las calificaciones de cada vendedor según su nivel de servicio con otros clientes.	
Validación: Al momento en el que el vendedor finaliza la entrega, en la aplicación del cliente se despliega una ventana para calificar el servicio y al vendedor mediante estrellas y un comentario.	

2.3.2. Ciclo de Vida Ágil

Durante el desarrollo del proyecto, se planificaron reuniones semanales las cuales son los llamados Sprint. En cada Sprint, se presentará un avance funcional de las aplicaciones de acuerdo con las historias de usuario definidas anteriormente. A continuación, se describen las historias de usuario que se implementaron en cada Sprint:

- Sprint #1:
 - Historia de usuario 1.
 - Historia de usuario 2.
- Sprint #2:
 - Historia de usuario 6.
- Sprint #3:
 - Historia de usuario 4.
- Sprint #4:
 - Historia de usuario 3.
- Sprint #5:
 - Historia de usuario 5.
 - Historia de usuario 7.
- Sprint #6:
 - Historia de usuario 8.
 - Historia de usuario 9.
- Sprint #7:
 - Historia de usuario 10.
 - Historia de usuario 12.
 - Historia de usuario 13.
- Sprint #8:
 - Historia de usuario 11.

Posteriormente, se tienen reuniones de revisión de los avances acordados en cada Sprint. En estas reuniones se valida cada avance funcional de las aplicaciones y se da retroalimentación para implementarlo en el siguiente Sprint.

2.3.3. Sprint

2.3.3.1. Sprint #1

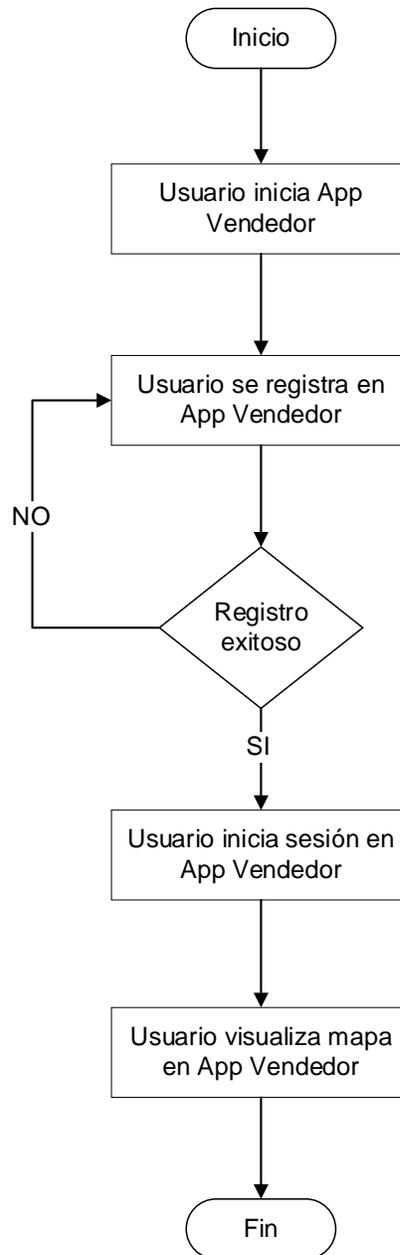


Figura 7.- Flujo correspondiente a la historia de usuario 1

Vendedores
<ul style="list-style-type: none"> - UUIDVendedor: varchar - avatarUrlVendedor: varchar - nombresVendedor: varchar - apellidosVendedor: varchar - telefonoVendedor: number - correoVendedor: varchar - contraseñaVendedor: varchar - tipoServicioVendedor: varchar - ratingVendedor: double

Figura 8.- Modelo de datos correspondiente a la historia de usuario 1

```

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

firebaseStorage = FirebaseStorage.getInstance();
storageReference = firebaseStorage.getReference();

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    activity, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);

View navigationHeaderView = navigationView.getHeaderView(0);
TextView txtNombre = navigationHeaderView.findViewById(R.id.txtNombres);
TextView txtStars = navigationHeaderView.findViewById(R.id.txtStars);
CircleImageView imgAvatar = navigationHeaderView.findViewById(R.id.img_avatar);

txtNombre.setText(Common.usuarioActual.getNombres());
txtStars.setText(Common.usuarioActual.getRating());
if(Common.usuarioActual.getAvatarUrl() != null && !TextUtils.isEmpty(Common.usuarioActual.getAvatarUrl()))
    Picasso.with(this).load(Common.usuarioActual.getAvatarUrl()).into(imgAvatar);

```

Figura 9.- Porción de código correspondiente a la historia de usuario 1

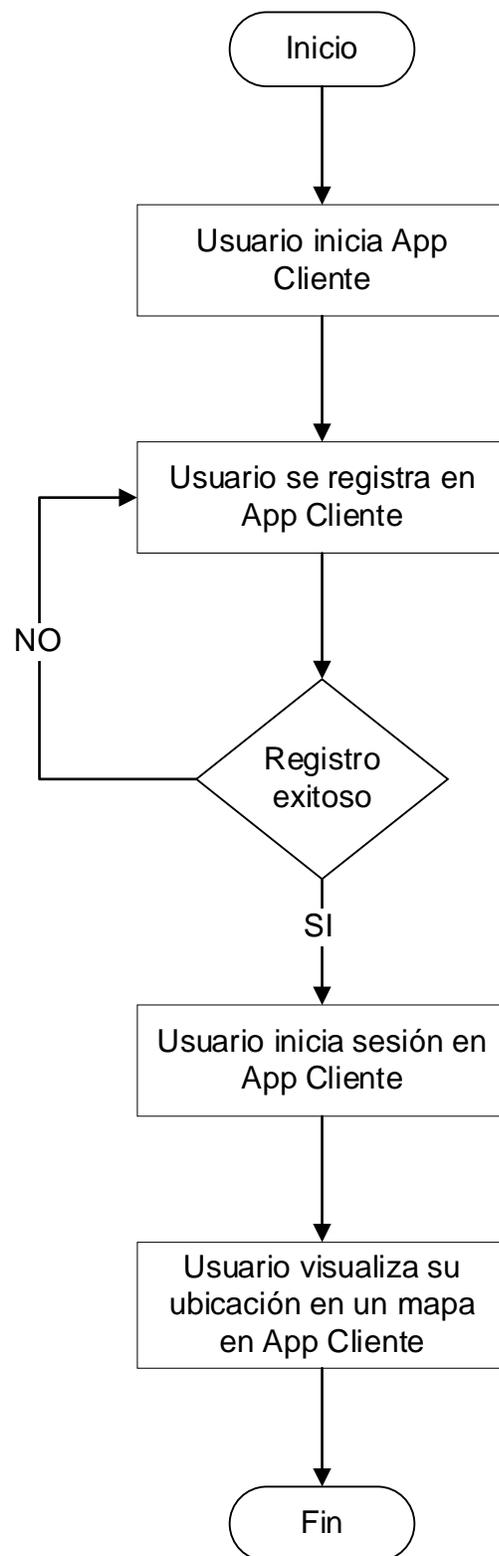


Figura 10.- Flujo correspondiente a la historia de usuario 2

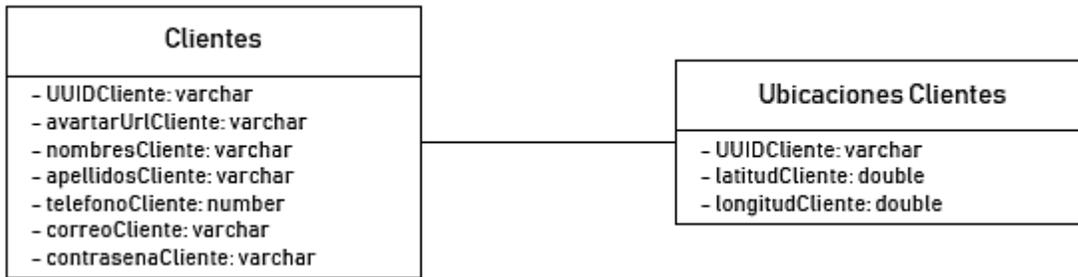


Figura 11.- Modelo de datos correspondiente a la historia de usuario 2

```

firebaseStorage = FirebaseStorage.getInstance();
storageReference = firebaseStorage.getReference();

mService = Common.getFCMService();

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    activity, this, drawer, toolbar, "Open navigation drawer", "Close navigation drawer");
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);

View navigationHeaderView = navigationView.getHeaderView(0);
TextView txtNombre = navigationHeaderView.findViewById(R.id.txtNombres);
CircleImageView imgAvatar = navigationHeaderView.findViewById(R.id.img_avatar);

txtNombre.setText(Common.usuarioActual.getNombres());
if(Common.usuarioActual.getAvatarUrl() != null && !TextUtils.isEmpty(Common.usuarioActual.getAvatarUrl()))
    Picasso.with(this).load(Common.usuarioActual.getAvatarUrl()).into(imgAvatar);

mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
mapFragment.getMapAsync(onMapReadyCallback: this);
  
```

Figura 12.- Porción de código correspondiente a la historia de usuario 2

2.3.3.2. Sprint #2

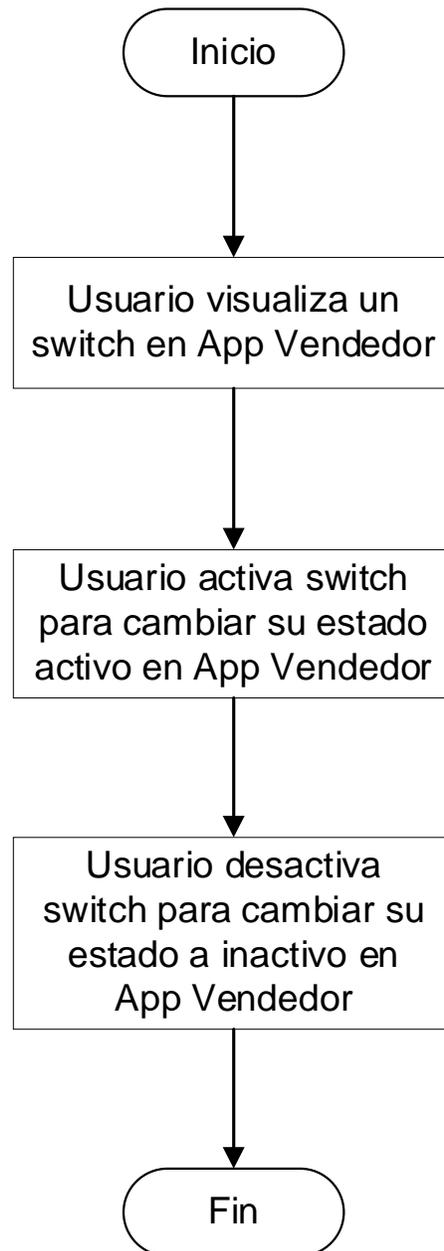


Figura 13.- Flujo correspondiente a la historia de usuario 6

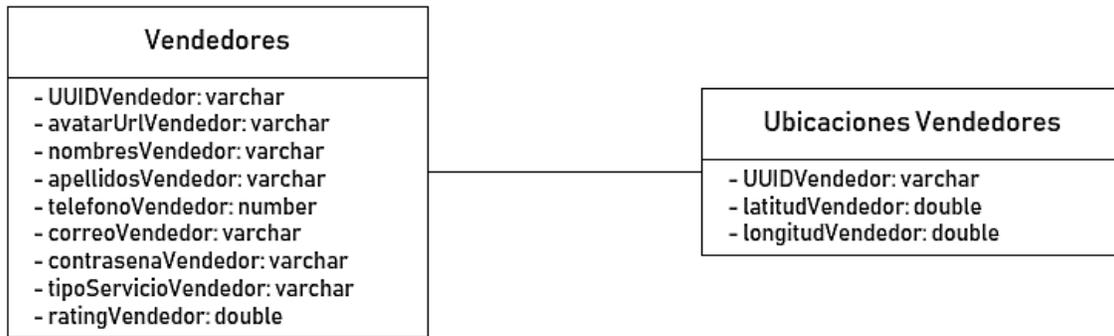


Figura 14.- Modelo de datos correspondiente a la historia de usuario 6

```

location_switch = (MaterialAnimatedSwitch)findViewById(R.id.location_switch);
location_switch.setOnCheckedChangeListener((isOnline) -> {
    if(isOnline){
        FirebaseDatabase.getInstance().goOnline();
        startLocationUpdates();
        displayLocation();
        Toast.makeText( context: Bienvenida.this, text: "En línea.", Toast.LENGTH_SHORT).show();
    }
    else{
        FirebaseDatabase.getInstance().goOffline();
        stopLocationUpdates();
        if(mCurrent != null)
            mCurrent.remove();
        mMap.clear();
        handler = new Handler();
        handler.removeCallbacks(drawPathRunnable);
        Toast.makeText( context: Bienvenida.this, text: "Fuera de línea.", Toast.LENGTH_SHORT).show();
    }
}
  
```

Figura 15.- Porción de código correspondiente a la historia de usuario 6

2.3.3.3. Sprint #3

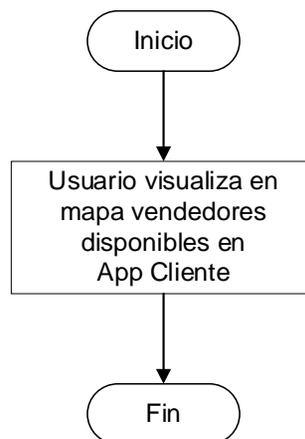


Figura 16.- Flujo correspondiente a la historia de usuario 4

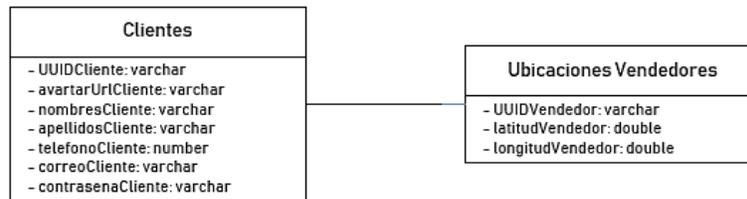


Figura 17.- Modelo de datos correspondiente a la historia de usuario 4

```

private void mostrarVendedoresDisponibles() {
    mMap.clear();
    mMap.addMarker(new MarkerOptions().position(new LatLng(mLastLocation.getLatitude(), mLastLocation.getLongitude()));

    DatabaseReference vendedorUbicacion = FirebaseDatabase.getInstance().getReference(Common.ubicVendedores);
    GeoFire gfVendedorUbicacion = new GeoFire(vendedorUbicacion);

    GeoQuery geoQuery = gfVendedorUbicacion.queryAtLocation(new GeoLocation(mLastLocation.getLatitude(), mLastLocation.getLongitude()));
    geoQuery.removeAllListeners();

    geoQuery.addGeoQueryEventListener(new GeoQueryEventListener() {
        @Override
        public void onKeyEntered(String key, final GeoLocation location) {
            FirebaseDatabase.getInstance().getReference(Common.infoVendedores).child(key).addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    Usuario vendedor = dataSnapshot.getValue(Usuario.class);

                    mMap.addMarker(new MarkerOptions()
                        .position(new LatLng(location.getLatitude(), location.getLongitude()))
                        .flat(true)
                        .title(vendedor.getNombres() + " " + vendedor.getApellidos())
                        .snippet("Rating: " + vendedor.getRating())
                        .icon(BitmapDescriptorFactory.fromResource(R.drawable.vehiculo)));
                }
            });
        }
    });
}
  
```

Figura 18.- Porción de código correspondiente a la historia de usuario 4

2.3.3.4. Sprint #4

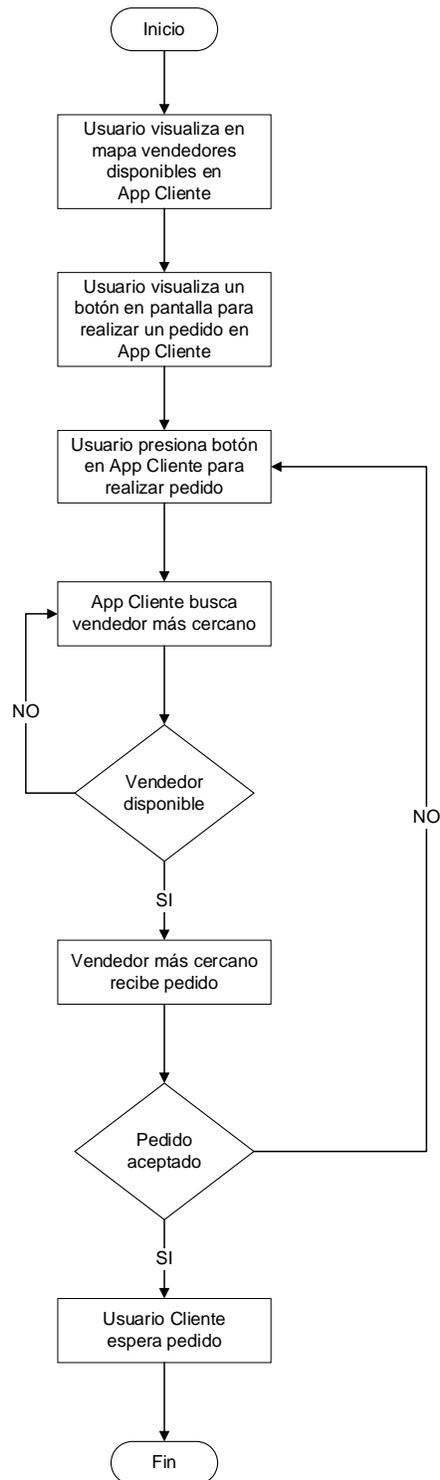


Figura 19.- Flujo correspondiente a la historia de usuario 3

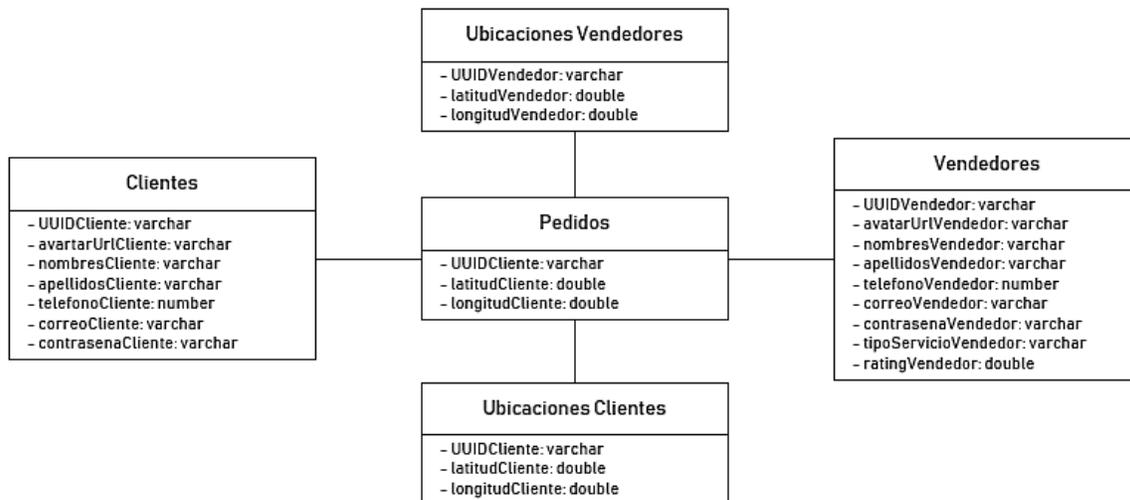


Figura 20.- Modelo de datos correspondiente a la historia de usuario 3

```
private void encontrarVendedor() {
    final DatabaseReference ubicacionVendedor = FirebaseDatabase.getInstance().getReference(Common.ubicVendedores);
    GeoFire gfUbicacionVendedor = new GeoFire(ubicacionVendedor);

    final GeoQuery geoQuery = gfUbicacionVendedor.queryAtLocation(new GeoLocation(mLastLocation.getLatitude(), mLastLocation.getLongitude()));
    geoQuery.removeAllListeners();
    geoQuery.addGeoQueryEventListener(new GeoQueryEventListener() {
        @Override
        public void onKeyEntered(String key, GeoLocation location) {
            //Si encuentra vendedor
            if(!Common.vendedorDisponible){
                Common.vendedorDisponible = true;
                Common.idVendedor = key;
                btnPedido.setClickable(true);
                btnPedido.setText("SOLICITAR SERVICIO");
                Toast.makeText(context, "Bienvenida, this", Toast.LENGTH_SHORT).show();
            }
        }
    });

    mService.getPath(requestApi).enqueue(new Callback<String>() {
        @Override
        public void onResponse(Call<String> call, Response<String> response) {
            try {
                JSONObject jsonObject = new JSONObject(response.body().toString());

                JSONArray routes = jsonObject.getJSONArray( "routes");

                JSONObject object = routes.getJSONObject( index 0);

                JSONArray legs = object.getJSONArray( "legs");

                JSONObject legsObject = legs.getJSONObject( index 0);

                JSONObject distance = legsObject.getJSONObject("distance");
                txtDistancia.setText(distance.getString( name: "text"));

                JSONObject time = legsObject.getJSONObject("duration");
                txtTiempo.setText(time.getString( name: "text"));
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });
}
```

Figura 21.- Porción de código correspondiente a la historia de usuario 3

2.3.3.5. Sprint #5



Figura 22.- Flujo correspondiente a la historia de usuario 5

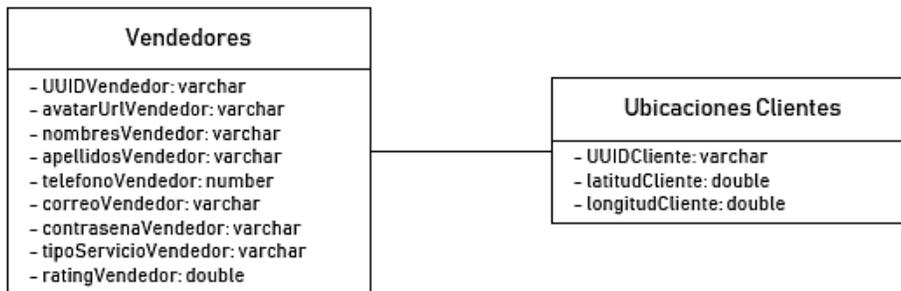


Figura 23.- Modelo de datos correspondiente a la historia de usuario 5

```

mediaPlayer = MediaPlayer.create( context: this, R.raw.ringtone);
mediaPlayer.setLooping(true);
mediaPlayer.start();

if(getIntent() != null){
    lat = getIntent().getDoubleExtra( name: "lat", defaultValue: -1.0);
    lng = getIntent().getDoubleExtra( name: "lng", defaultValue: -1.0);
    idCliente = getIntent().getStringExtra( name: "cliente");

    getDirection(lat, lng);
}
  
```

Figura 24.- Porción de código correspondiente a la historia de usuario 5

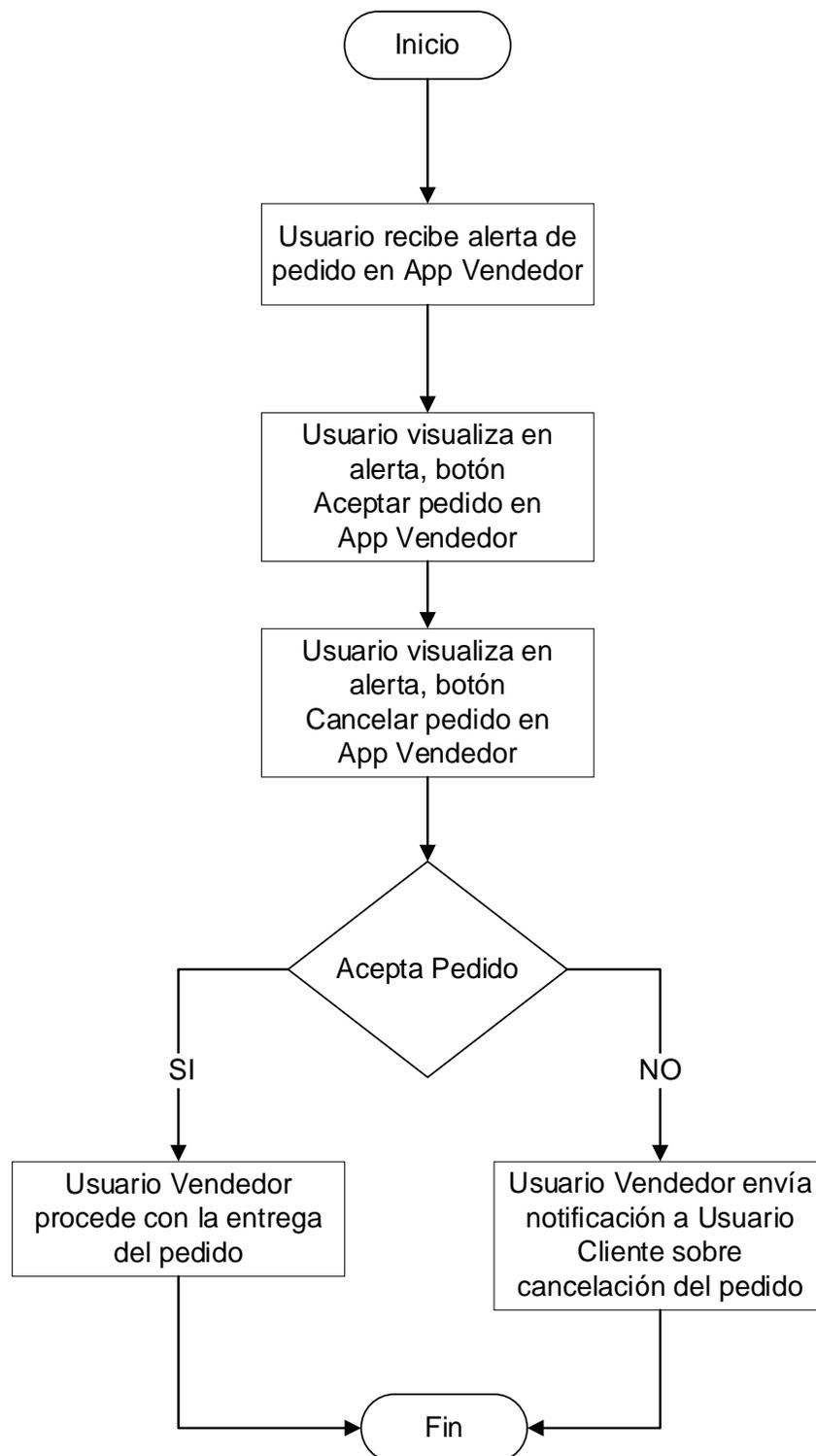


Figura 25.- Flujo correspondiente a la historia de usuario 7

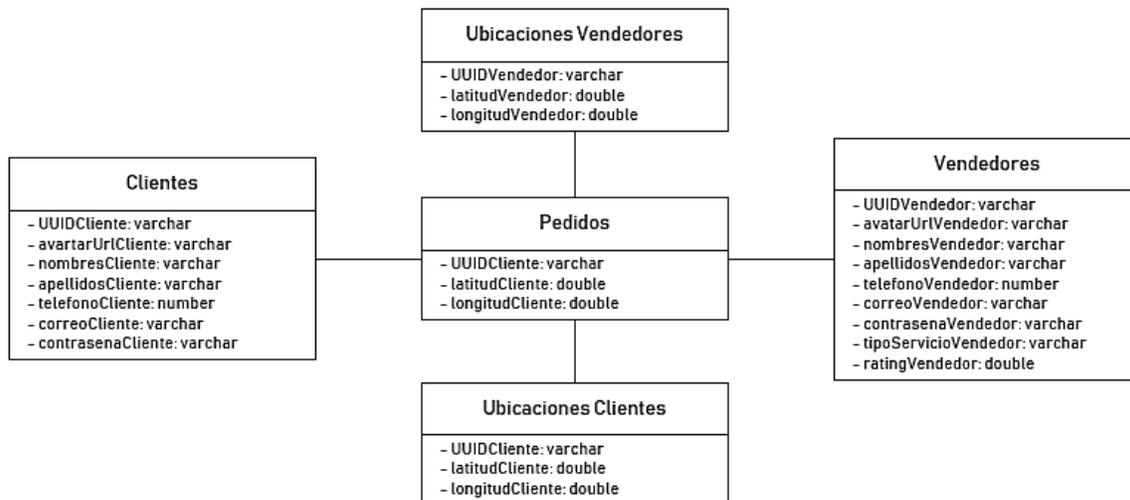


Figura 26.- Modelo de datos correspondiente a la historia de usuario 7

```

mService    = Common.getGoogleAPI();
mFCMService = Common.getFCMService();

txtTiempo    = (TextView) findViewById(R.id.txtTiempo);
txtDistancia = (TextView) findViewById(R.id.txtDistancia);
txtDireccion = (TextView) findViewById(R.id.txtDireccion);
btnAcepta    = (Button) findViewById(R.id.btnAcepta);
btnCancela   = (Button) findViewById(R.id.btnCancela);

btnAcepta.setOnClickListener((view) -> {
    Intent intent = new Intent( packageContext: Pedido.this, Seguimiento.class);
    intent.putExtra( name: "lat", lat);
    intent.putExtra( name: "lng", lng);
    intent.putExtra( name: "idCliente", idCliente);

    startActivity(intent);
    finish();
});

btnCancela.setOnClickListener((view) -> {
    if(!TextUtils.isEmpty(idCliente))
        cancelarPedido(idCliente);
});

```

Figura 27.- Porción de código correspondiente a la historia de usuario 7

2.3.3.6. Sprint #6

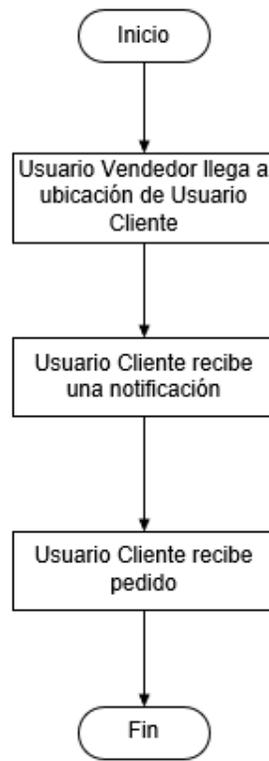


Figura 28.- Flujo correspondiente a la historia de usuario 8

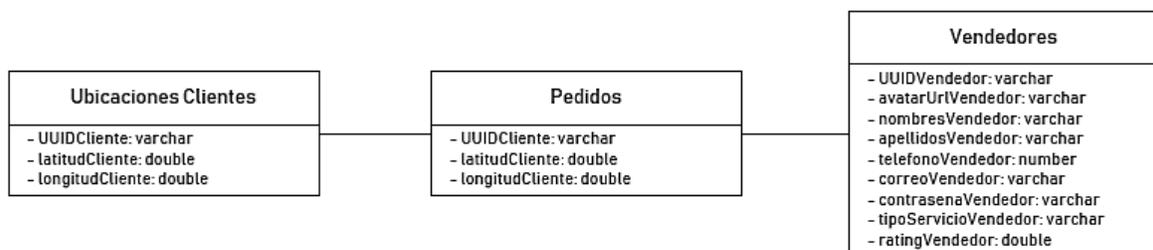


Figura 29.- Modelo de datos correspondiente a la historia de usuario 8

```

private void enviarNotificacionFinEntrega(String idCliente) {
    Token token = new Token(idCliente);
    Notification notification = new Notification( title: "Entrega", idCliente);
    Sender sender = new Sender(notification, token.getToken());
}
  
```

Figura 30.- Porción de código correspondiente a la historia de usuario 8

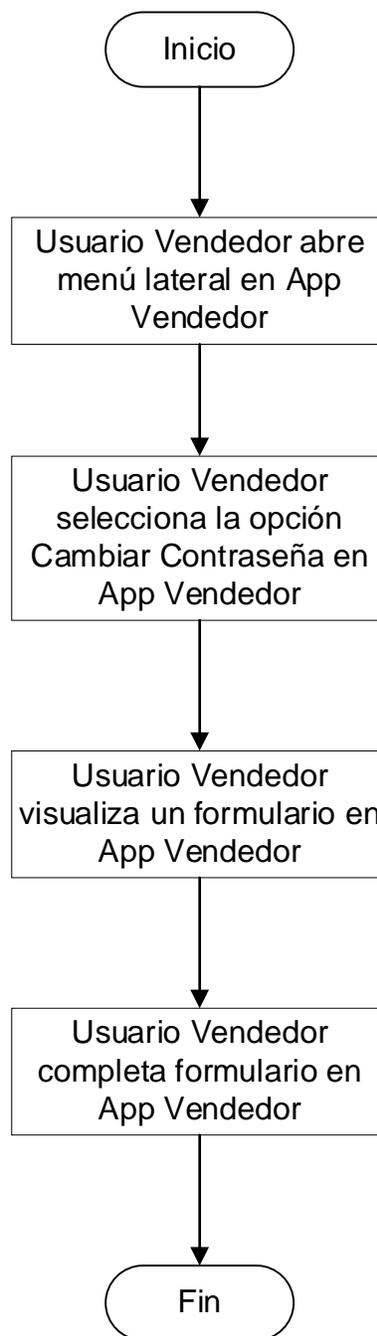


Figura 31.- Flujo correspondiente a la historia de usuario 9

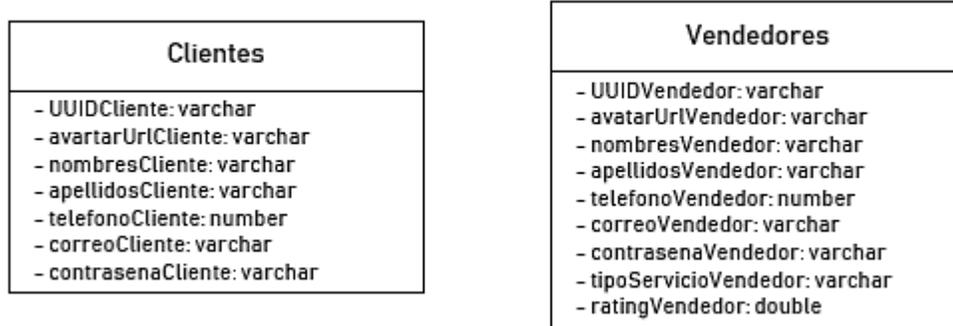


Figura 32.- Modelo de datos correspondiente a la historia de usuario 9

```

edtNuevaContrasena.getText().toString().equals(edtConfirmarContrasena.getText().toString()) {
    String email = FirebaseAuth.getInstance().getCurrentUser().getEmail();

    AuthCredential credential = EmailAuthProvider.getCredential(email, edtContrasena.getText().toString());
    FirebaseAuth.getInstance().getCurrentUser().reauthenticate(credential)
        .addOnCompleteListener((task) -> {
            if(task.isSuccessful()){
                FirebaseAuth.getInstance().getCurrentUser().updatePassword(edtConfirmarContrasena.getText().toString())
                    .addOnCompleteListener((task) -> {
                        if(task.isSuccessful()){
                            Map<String, Object> contrasena = new HashMap<>();
                            contrasena.put("contrasena", edtConfirmarContrasena.getText().toString());
                            DatabaseReference infoVendedor = FirebaseDatabase.getInstance().getReference(Common.infoVendedor)
                                .child(FirebaseAuth.getInstance().getCurrentUser().getUid()).updateChildren(contrasena)
                                    .addOnCompleteListener((task) -> {
                                        if(task.isSuccessful())
                                            Toast.makeText(context, "Contraseña cambiada.", Toast.LENGTH_SHORT);
                                        else
                                            Toast.makeText(context, "Contraseña cambiada pe", Toast.LENGTH_SHORT);
                                        cargandoDialog.dismiss();
                                    });
                        } else {
                            Toast.makeText(context, "Cambio de contraseña fallido.", Toast.LENGTH_SHORT);
                        }
                    });
            } else {
                cargandoDialog.dismiss();
            }
        });
}

```

Figura 33.- Porción de código correspondiente a la historia de usuario 9

2.3.3.7. Sprint #7



Figura 34.- Flujo correspondiente a la historia de usuario 10

Vendedores
<ul style="list-style-type: none"> - UUIDVendedor: varchar - avatarUrlVendedor: varchar - nombresVendedor: varchar - apellidosVendedor: varchar - telefonoVendedor: number - correoVendedor: varchar - contraseñaVendedor: varchar - tipoServicioVendedor: varchar - ratingVendedor: double

Figura 35.- Modelo de datos correspondiente a la historia de usuario 10

```
private void mostrarActualizarInfoDialog() {
    final AlertDialog.Builder dialog = new AlertDialog.Builder(context: this);
    dialog.setTitle("ACTUALIZAR DATOS");
    dialog.setMessage("Completa la información.");

    LayoutInflater inflater = LayoutInflater.from(this);
    View layout_acualizar_info = inflater.inflate(R.layout.layout_actualizar_info, root: null);

    final MaterialEditText edtANombre = layout_acualizar_info.findViewById(R.id.edtANombre);
    final ImageView cargar_imagen = layout_acualizar_info.findViewById(R.id.cargar_imagen);
    cargar_imagen.setOnClickListener((view) -> { seleccionarImagen(); });

    dialog.setView(layout_acualizar_info);

    dialog.setPositiveButton(text: "ACTUALIZAR", (dialogInterface, i) -> {
        dialogInterface.dismiss();

        final SpotsDialog cargandoDialog = new SpotsDialog(context: Bienvenida.this, R.style.Custom);
        cargandoDialog.show();

        String aNombre = edtANombre.getText().toString();

        Map<String, Object> actualizarNombre = new HashMap<>();
        if(!TextUtils.isEmpty(aNombre))
            actualizarNombre.put("nombres", aNombre);
    });
}
```

Figura 36.- Porción de código correspondiente a la historia de usuario 10

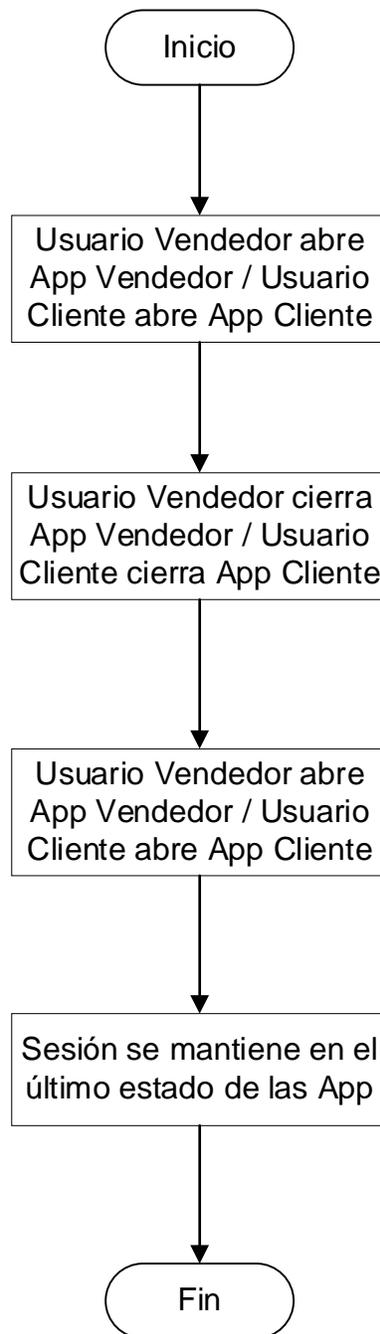


Figura 37.- Flujo correspondiente a la historia de usuario 12

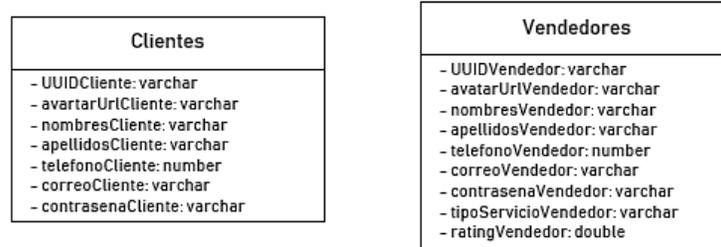


Figura 38.- Modelo de datos correspondiente a la historia de usuario 12

```
private void autoInicioSesion(String user, String pwd) {
    final SpotsDialog cargandoDialog = new SpotsDialog( context: Inicio.this, R.style.Custom);
    cargandoDialog.show();

    auth.signInWithEmailAndPassword(user, pwd)
        .addOnSuccessListener((OnSuccessListener) (authResult) -> {
            cargandoDialog.dismiss();
            FirebaseDatabase.getInstance().getReference(Common.infoVendedores).child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                .addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        Common.usuarioActual = dataSnapshot.getValue(Usuario.class);
                        startActivity(new Intent( packageContext: Inicio.this, Bienvenida.class));
                        cargandoDialog.dismiss();
                        finish();
                    }
                });
        });
}
```

Figura 39.- Porción de código correspondiente a la historia de usuario 12

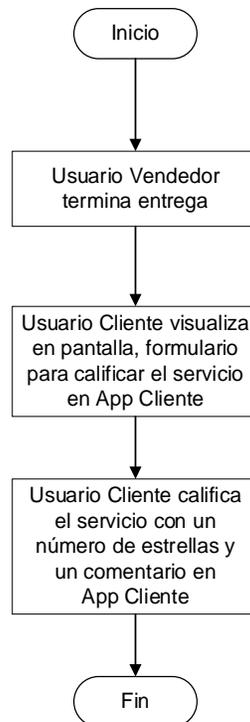


Figura 40.- Flujo correspondiente a la historia de usuario 13

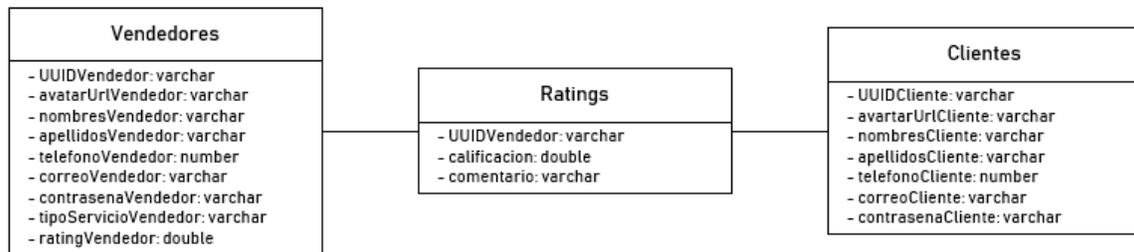


Figura 41.- Modelo de datos correspondiente a la historia de usuario 13

```

private void calificarServicio(final String idVendedor) {
    final SpotsDialog cargandoDialog = new SpotsDialog(context: Rating.this, R.style.Custom);
    cargandoDialog.show();

    Rate rate = new Rate();
    rate.setCalificacion(String.valueOf(ratingStars));
    rate.setComentario(edtComentario.getText().toString());

    ratingRef.child(idVendedor).push().setValue(rate)
        .addOnCompleteListener((task) -> {
            ratingRef.child(idVendedor)
                .addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        double averageStars = 0.0;
                        int count = 0;
                        for(DataSnapshot postSanpshot : dataSnapshot.getChildren()){
                            Rate rate = postSanpshot.getValue(Rate.class);
                            averageStars += Double.parseDouble(rate.getCalificacion());
                            count++;
                        }
                        double totalStars = averageStars/count;
                        DecimalFormat df = new DecimalFormat(pattern: "#.#");
                        String valueUpdate = df.format(totalStars);

                        Map<String, Object> calificacionVendedor= new HashMap<>();
                        calificacionVendedor.put("rating", valueUpdate);
                    }
                })
        })
}
  
```

Figura 42.- Porción de código correspondiente a la historia de usuario 13

2.3.3.8. Sprint #8

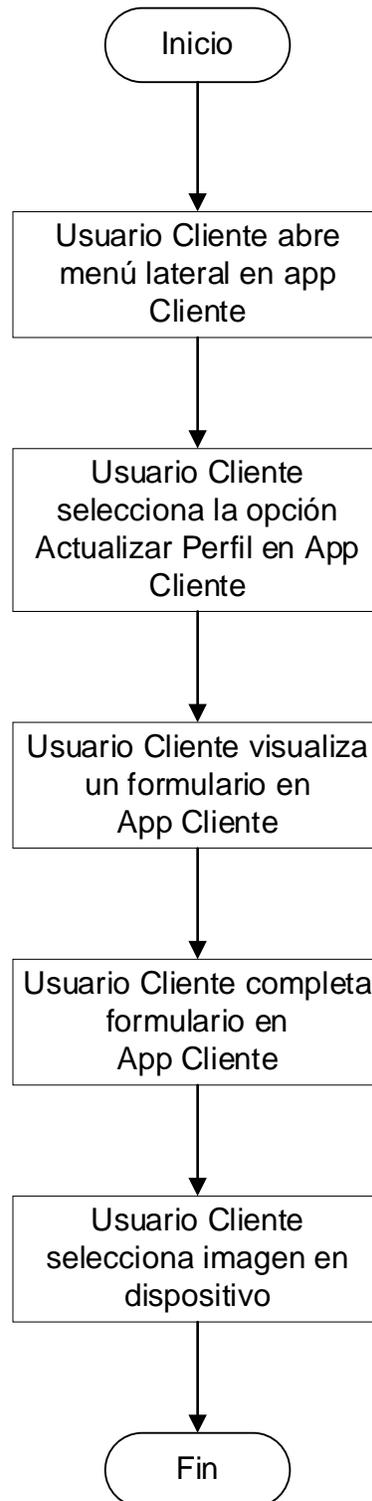


Figura 43.- Flujo correspondiente a la historia de usuario 11

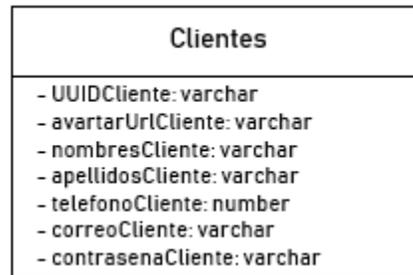


Figura 44.- Modelo de datos correspondiente a la historia de usuario 11

```
private void mostrarActualizarInfoDialog() {
    final AlertDialog.Builder dialog = new AlertDialog.Builder( context, this);
    dialog.setTitle("ACTUALIZAR DATOS");
    dialog.setMessage("Completa la información.");

    LayoutInflater inflater = LayoutInflater.from(this);
    View layout_acualizar_info = inflater.inflate(R.layout.layout_actualizar_info, root: null);

    final MaterialEditText edtANombre = layout_acualizar_info.findViewById(R.id.edtANombre);
    final ImageView cargar_imagen = layout_acualizar_info.findViewById(R.id.cargar_imagen);
    cargar_imagen.setOnClickListener((view) -> { seleccionarImagen(); });

    dialog.setView(layout_acualizar_info);

    dialog.setPositiveButton( text: "ACTUALIZAR", (dialogInterface, i) -> {
        dialogInterface.dismiss();

        final SpotsDialog cargandoDialog = new SpotsDialog( context: Bienvenida.this, R.style.Custom);
        cargandoDialog.show();

        String aNombre = edtANombre.getText().toString();

        Map<String, Object> actualizarNombre = new HashMap<>();
        if(!TextUtils.isEmpty(aNombre))
            actualizarNombre.put("nombres", aNombre);
    });
}
```

Figura 45.- Porción de código correspondiente a la historia de usuario 11

2.4. Conclusión del Capítulo

Como se detalló anteriormente, el uso de la metodología Scrum a través de los Sprint, resulta muy útil para el desarrollo del proyecto debido a que se pueden visualizar y probar los entregables funcionales de las aplicaciones móviles de Vendedor y Cliente y así recibir retroalimentación para ir mejorando el proyecto en cada Sprint.

3. Capítulo III. Entorno

3.1. Introducción

A continuación, se mostrará el entorno de las aplicaciones móviles de Vendedor y Cliente, así como también la aplicación web de administración. Además, se describirán los resultados del funcionamiento de ambas aplicaciones.

3.2. Entorno Aplicación Móvil

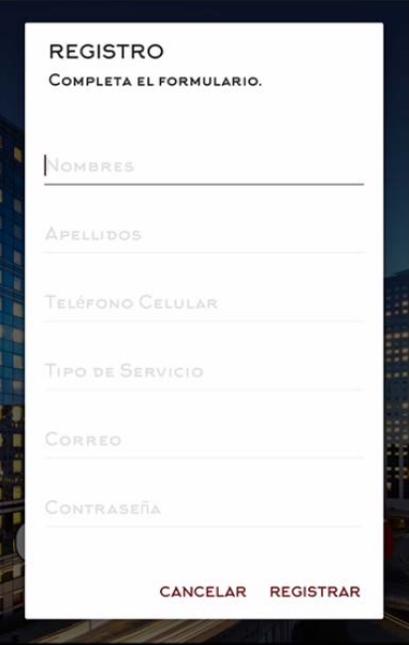
3.2.1. Entorno Aplicación Móvil Vendedor

Al iniciar la aplicación de Vendedor, se desplegará la siguiente pantalla, donde tiene las opciones ingresar con un vendedor existente o registrar uno nuevo.



Figura 46.- Pantalla inicial de la aplicación de Vendedor

Al presionar el botón de registro, se mostrará un formulario donde se registrará la información del nuevo vendedor como nombres, apellidos, teléfono celular, el tipo de servicio que ofrece (Gas o Lavandería), correo electrónico y contraseña.



REGISTRO
COMPLETA EL FORMULARIO.

NOMBRES

APELLIDOS

TELÉFONO CELULAR

TIPO DE SERVICIO

CORREO

CONTRASEÑA

CANCELAR REGISTRAR

Figura 47.- Pantalla de registro para nuevos vendedores

Al presionar el botón de ingreso, el vendedor deberá ingresar su correo electrónico y contraseña para pasar a la pantalla principal de la aplicación vendedor.



INGRESO
INGRESA TUS DATOS.

CORREO

CONTRASEÑA

CANCELAR INGRESAR

INGRESO REGISTRO

Figura 48.- Pantalla de ingreso para vendedores registrados

Después de haberse autenticado, se mostrará la siguiente pantalla con un mapa, donde el vendedor visualizará un switch para cambiar su estado (online / offline).



Figura 49.- Pantalla principal de la aplicación vendedor, en estado offline.

Al momento en que el vendedor cambie su estado a online, en la pantalla se mostrará su ubicación actual en el mapa y podrá recibir pedidos.

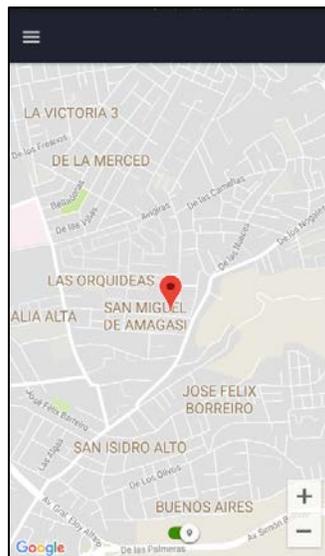


Figura 50.- Ubicación actual del vendedor, en estado online

Si presiona el icono ☰ que se encuentra en la parte superior izquierda, se mostrará un menú lateral con información del vendedor como imagen, nombres y calificación promedio, así como también las opciones de actualizar su perfil, cambiar su contraseña, contacto, ayuda y cerrar sesión.

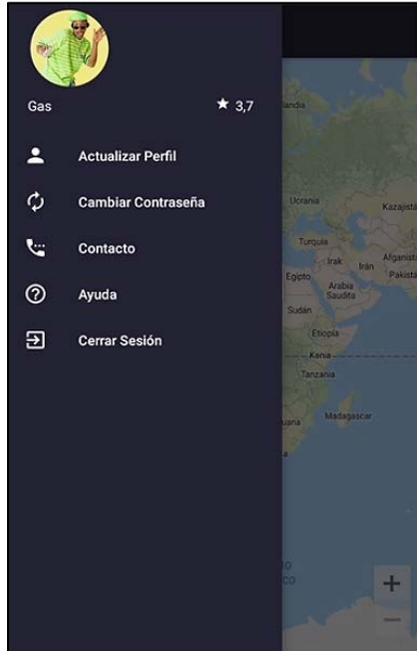


Figura 51.- Pantalla con el menú lateral

Al momento de seleccionar la opción de actualizar perfil, el vendedor tendrá la posibilidad de actualizar sus nombres y subir una imagen de perfil.



Figura 52.- Pantalla para actualizar perfil de vendedor

Al momento de seleccionar la opción de actualizar contraseña, el vendedor deberá ingresar su contraseña actual, la nueva contraseña y confirmar la nueva contraseña.



Figura 53.- Pantalla para actualizar contraseña del vendedor

Al momento de seleccionar la opción de contacto, se mostrará un cuadro informativo con los datos de contacto con el desarrollador de la aplicación.

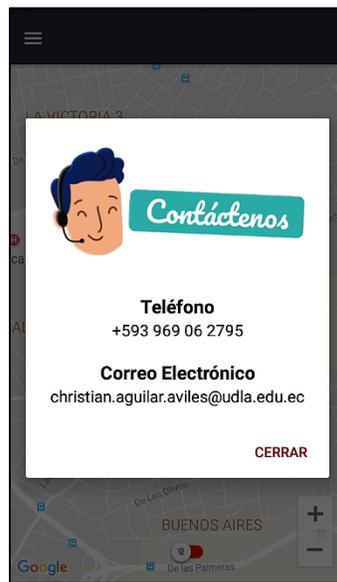


Figura 54.- Pantalla de contacto de la aplicación

Al momento de seleccionar la opción de ayuda, el vendedor podrá leer preguntas frecuentes respecto a la aplicación.

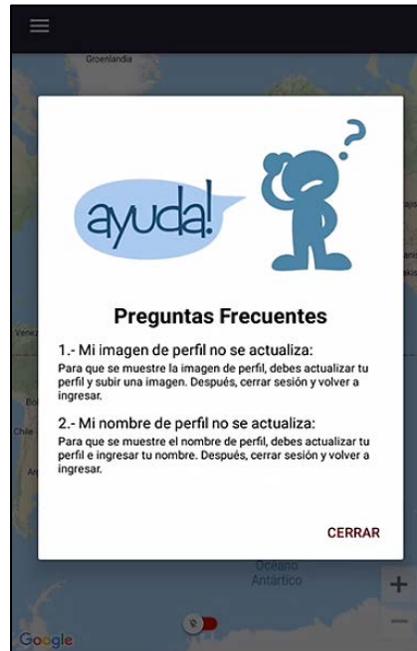


Figura 55.- Pantalla de ayuda de la aplicación vendedor

El momento en el que un cliente realice un pedido, al vendedor más cercano se le mostrara una pantalla alerta, la cual contiene información como la dirección del cliente, el tiempo y distancia hacia la ubicación del cliente. También, tendrá la opción de cancelar o aceptar el pedido.



Figura 56.- Pantalla de recepción del pedido

Al momento de aceptar el pedido, al vendedor se le mostrará una pantalla con un mapa, la cual contiene la ruta hacia la ubicación del cliente.

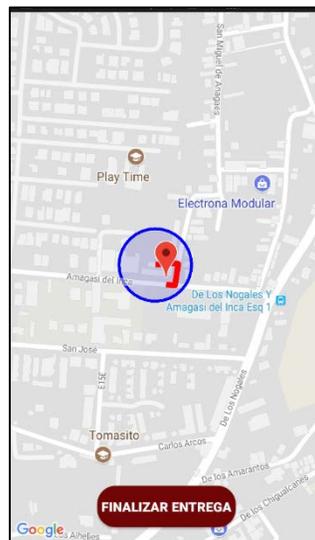


Figura 57.- Pantalla con la ruta hacia la ubicación del cliente

Al momento que el vendedor llegue a la ubicación del cliente, deberá presionar el botón finalizar entrega para volver a la pantalla principal.

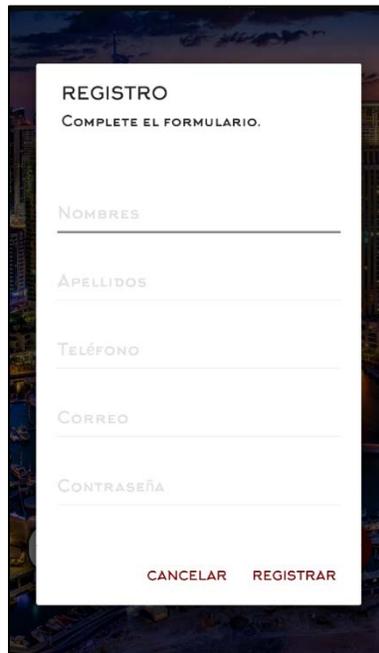
3.2.2. Entorno Aplicación Móvil Cliente

Al iniciar la aplicación de Cliente, se desplegará la siguiente pantalla, donde tiene las opciones ingresar con un cliente existente o registrar uno nuevo.



Figura 58.- Pantalla inicial de la aplicación de Vendedor

Al presionar el botón de registro, se mostrará un formulario donde se registrará la información del nuevo cliente como nombres, apellidos, teléfono celular, correo electrónico y contraseña.



REGISTRO
COMPLETE EL FORMULARIO.

NOMBRES

APELLIDOS

TELÉFONO

CORREO

CONTRASEÑA

CANCELAR REGISTRAR

Figura 59.- Pantalla de registro para nuevos clientes

Al presionar el botón de ingreso, el cliente deberá ingresar su correo electrónico y contraseña para pasar a la pantalla principal de la aplicación cliente.



INGRESO
INGRESE SUS DATOS.

CORREO

CONTRASEÑA

CANCELAR INGRESAR

INGRESO REGISTRO

Figura 60.- Pantalla de ingreso para clientes registrados

Después de haberse autenticado, se mostrará la siguiente pantalla con un mapa, donde el cliente visualizará su ubicación actual. Además, el cliente visualizara el tipo de servicio disponible para realizar el pedido.



Figura 61.- Pantalla principal de la aplicación cliente

Posteriormente, si existen vendedores disponibles, el cliente podrá visualizarlos en el mapa. Los vendedores serán reconocidos con el icono de un vehículo y según el tipo de servicio que ofrecen.

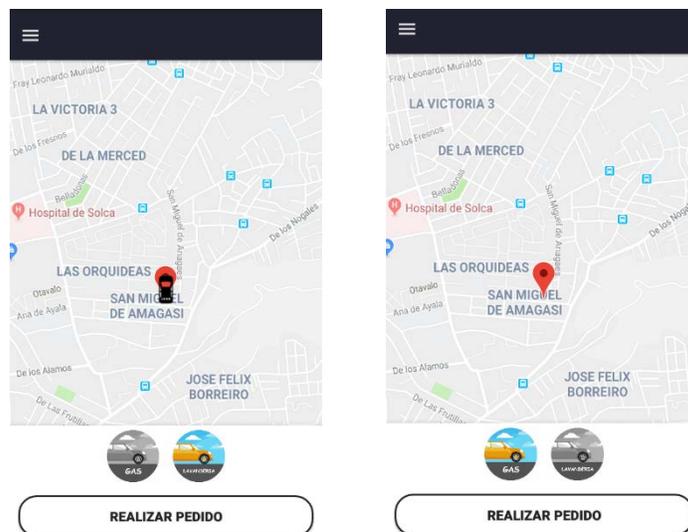


Figura 62.- Visualización de vendedores disponibles según tipo de servicio

Si presiona el icono ☰ que se encuentra en la parte superior izquierda, se mostrará un menú lateral con información básica del cliente como su imagen de perfil y nombres, así como también las opciones de actualizar su perfil, cambiar su contraseña, contacto, ayuda y cerrar sesión.

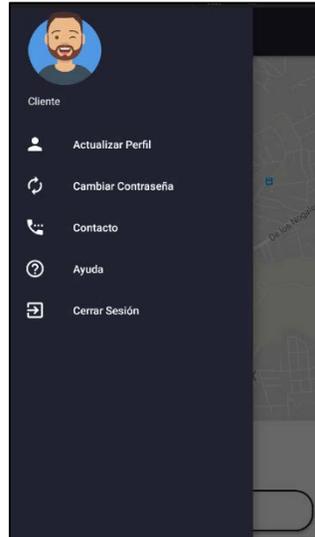


Figura 63.- Pantalla con el menú lateral

Al momento de seleccionar la opción de actualizar perfil, el cliente tendrá la posibilidad de actualizar sus nombres y subir una imagen de perfil.

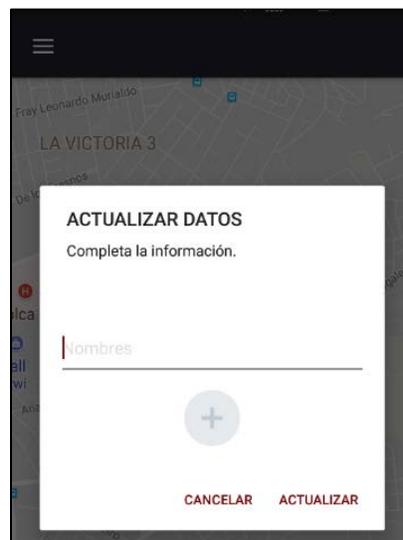


Figura 64.- Pantalla para actualizar perfil de cliente

Al momento de seleccionar la opción de actualizar contraseña, el cliente deberá ingresar su contraseña actual, la nueva contraseña y confirmar la nueva contraseña.

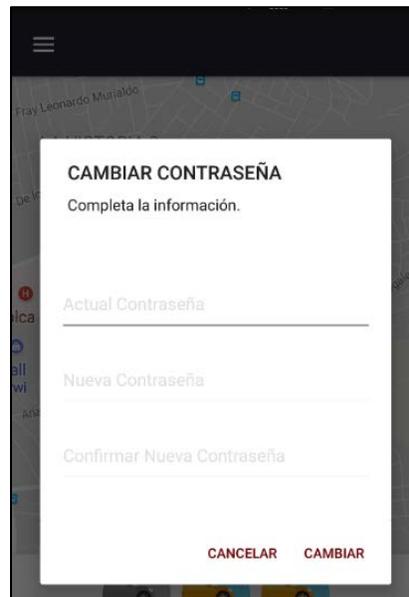


Figura 65.- Pantalla para actualizar contraseña del cliente

Al momento de seleccionar la opción de contacto, se mostrará un cuadro informativo con los datos de contacto con el desarrollador de la aplicación.



Figura 66.- Pantalla de contacto de la aplicación

Al momento de seleccionar la opción de ayuda, el cliente podrá leer preguntas frecuentes respecto a la aplicación.

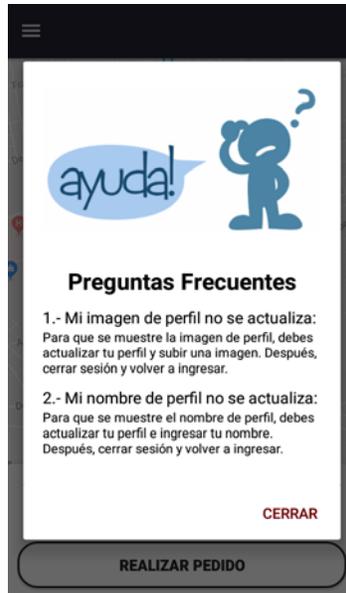


Figura 67.- Pantalla de ayuda de la aplicación cliente

El momento en que el cliente presiona el botón “Realizar Pedido”, buscará el vendedor más cercano y establecerá su punto de entrega del pedido.



Figura 68.- Buscando vendedor más cercano y punto de entrega establecido

Cuando encuentre al vendedor más cercano, el cliente tendrá la opción de “Solicitar Producto”.



Figura 69.- Solicitud del servicio

Después de solicitar el servicio, el cliente esperará que el vendedor llegue a su ubicación. El botón mostrará mensaje “Esperando Entrega”.



Figura 70.- Espera de entrega

Cuando el vendedor haya llegado al lugar de entrega, se enviará una notificación al teléfono del cliente avisando que debe salir a recibir su pedido.

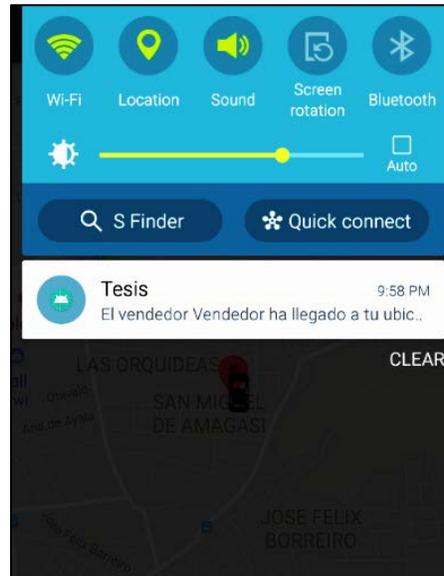


Figura 71.- Notificación de llegada del vendedor

Cuando el vendedor finalice la entrega, en la aplicación de cliente se mostrará una pantalla para calificar el servicio y realizar un comentario.

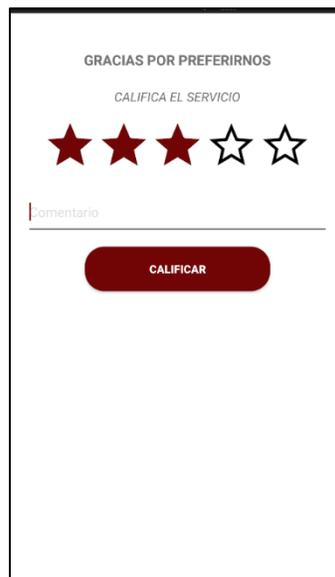


Figura 72.- Pantalla de calificación del servicio

Finalmente, después de realizar la calificación se mostrará nuevamente la pantalla principal para poder realizar un nuevo pedido.



Figura 73.- Pantalla principal de la aplicación cliente

3.3. Entorno Aplicación Web

Al entrar en la aplicación web, el administrador deberá ingresar sus credenciales para poder visualizar la pantalla de administración.

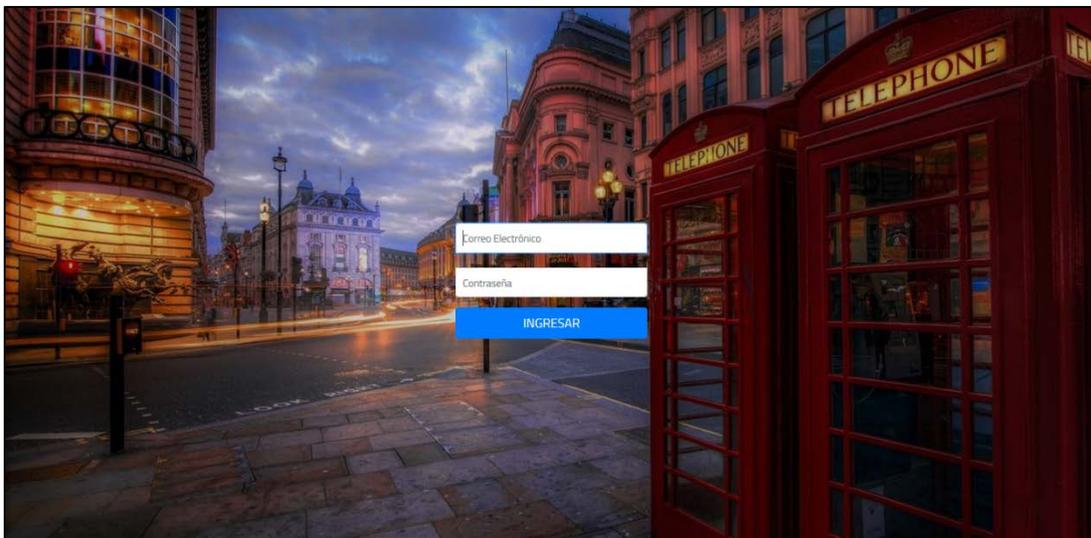
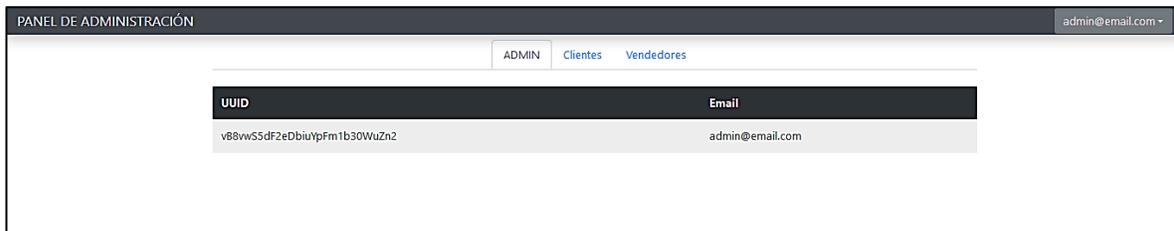


Figura 74.- Pantalla inicial aplicación web

Posteriormente, el administrador podrá visualizar las cuentas de administrador que se encuentran registradas en la base de datos.



PANEL DE ADMINISTRACIÓN admin@email.com

ADMIN **Cientes** Vendedores

UUID	Email
vB8vwS5dF2eDbiuYpFm1b30WuZn2	admin@email.com

Figura 75.- Panel de administración para usuarios administrador

Si se da clic en la pestaña Clientes, el administrador podrá visualizar los clientes registrados en la aplicación. Además, podrá visualizar información de cada uno como nombres, apellidos, teléfono y el correo electrónico.



PANEL DE ADMINISTRACIÓN admin@email.com

Admin **CLIENTES** Vendedores

UUID	Nombres	Apellidos	Telefono	Correo
xRGzFW30AhbrgqrtEnVr7LA0FOA3	Cliente	Test	0987654132	cliente@email.com

Figura 76.- Panel de administración para visualizar clientes registrados

Y si se da clic en la pestaña Vendedores, el administrador podrá visualizar los vendedores registrados en la aplicación. Además, podrá visualizar información de cada uno como nombres, apellidos, teléfono, rating, el tipo de servicio que ofrecen y el correo electrónico.



PANEL DE ADMINISTRACIÓN admin@email.com

Admin Clientes **VENDEDORES**

UUID	Nombres	Apellidos	Telefono	Tipo de Servicio	Rating	Correo
OTEieanCptaPvGdCtqrwo0Fxp8v2	Vendedor	Test	0987654321	Tanques de Gas	3.9	vendedor@email.com

Figura 77.- Panel de administración para visualizar clientes registrados

3.4. Resultados de la Aplicación Móvil

La validación se realizó con pruebas funcionales al final de cada Sprint, durante las cuales fueron aceptados los incrementos. En cada prueba funcional, se verificó que las aplicaciones respondan a los requerimientos que se definieron al inicio del proyecto en las historias de usuario.

3.5. Análisis Económico

Después de un análisis, se concluyó que el desarrollo del proyecto tendría un costo aproximado de \$9,200.00. A continuación, el detalle:

Tabla 14

Costo aproximado del desarrollo del proyecto

Descripción	Costo	Unidad	Total
Desarrollador	\$ 15.00	600 horas	\$ 9,000.00
Licencias	\$ 25.00	1	\$ 20.00
Servicios	\$ 45.00	4 meses	\$ 180.00
		Total	\$ 9,200.00

4. Conclusiones Y Recomendaciones

4.1. Conclusiones

Se realizó el estado del arte del desarrollo de aplicaciones móviles para el Sistema Operativo Android del cual se concluyó que la mejor opción para el desarrollo del proyecto fue la utilización de las metodologías Scrum y Lean Startup debido a que es un proyecto sin información histórica. Además, también se decidió utilizar las herramientas de desarrollo Firebase para la base de datos, como lenguaje de programación Java y Android Studio como IDE para el desarrollo de las aplicaciones móviles.

Se desarrolló una aplicación web en la cual un usuario administrador podrá visualizar los usuarios de las aplicaciones móviles como Clientes y Vendedores, así como también la información básica de cada uno de ellos. Después de un análisis, se decidió que la mejor opción para desarrollar la aplicación web es el uso del lenguaje de programación JavaScript y *Node JS* para el *backend* y HTML y CSS para el *frontend*.

La validación se realizó con pruebas funcionales al final de cada Sprint, durante las cuales fueron aceptados los incrementos. El costo aproximado del desarrollo fue de \$9,200.00 teniendo en cuenta honorarios, costo de servicios y licencias.

4.2. Recomendaciones

Atendiendo a la posibilidad de crecimiento del negocio que gestiona, se recomienda implementar nuevos tipos de servicios para venta ambulante. El sistema ha sido desarrollado para soportar el incremento de estas funciones.

Referencias

- Alfocea, J. (2017). *Se Dispara el Uso de Aplicaciones Móviles*. Recuperado el 08 de mayo de 2017 de: <https://www.androidsis.com/se-dispara-uso-aplicaciones-moviles/>
- Álvarez, L. A. (2017). *Uber y Cabify: Tan Iguales, Tan Distintos*. Recuperado el 14 de diciembre de 2017 de: <http://www.elmundo.es/economia/2017/12/14/5a302ebb22601d52378b4633.html>
- Android. (s.f.). *Características Android Studio*. Recuperado el 15 de diciembre de 2017 de: <https://developer.android.com/studio/features.html>
- Android. (s.f.). *Conoce Android Studio*. Recuperado el 15 de diciembre de 2017 de: <https://developer.android.com/studio/intro/index.html>
- Argudo, M. (2015). *Tipos de Cloud Computing*. Recuperado el 17 de noviembre de 2017 de: <https://openwebinars.net/blog/tipos-de-cloud-computing/>
- Bonilla, M. (2013). *Historia del Comercio*. Recuperado 10 de octubre de 2017 de: <https://es.slideshare.net/MarcelaBonilla/historia-del-comercio-26996173>
- Cárdenas, D. (2016). *Ventajas y Desventajas de Programar en JAVA*. Recuperado el 20 de diciembre de 2017 de: <http://adictoalcodigo.blogspot.com/2016/07/ventajas-y-desventajas-de-programar-en.html>
- El Asri, L. (2016). *Alexandru Simonescu: Firebase Ayuda a Quienes Deben Desarrollar Rápido*. Recuperado el 08 de diciembre de 2017 de: <http://katade.com/2016/08/31/alexandru-simonescu-firebase-desarrollar/>
- Francia, J. (2017). *¿Qué es Scrum?* Recuperado el 03 de noviembre de 2017 de: <https://www.scrum.org/resources/blog/que-es-scrum>

- García, É. (2017). *El Mercado de Aplicaciones Móviles Seguirá Imbatible*. Recuperado 15 de diciembre de 2017 de: <https://www.xatakamovil.com/aplicaciones/el-mercado-de-aplicaciones-moviles-seguira-imbatile-se-espera-que-genere-110-mil-millones-en-2018>
- Hernández, L. (2016). *5 Razones para Utilizar el Método Lean Startup*. Recuperado el 10 de febrero de 2018 de: <https://www.entrepreneur.com/article/271613>
- Iglesias, A. (2016). *Cómo Monetizar Aplicaciones Móviles*. Recuperado el 20 de enero de 2018 de: <http://www.ticbeat.com/tecnologias/como-monetizar-aplicaciones-moviles-y-ganar-dinero-con-ellas/>
- Lamata, E. (2012). *Cloud Computing*. Recuperado el 18 de octubre de 2017 de: <http://www.eleconomista.es/gestion-empresarial/noticias/4087167/07/12/Cloud-Computing-que-es-para-que-sirve-y-cuales-son-sus-aplicaciones-.html>
- Oliveras, P. (2017). *La Evolución del Comercio Electrónico: del eCommerce al mCommerce*. Recuperado el 10 de septiembre de 2017 de: <http://blog.servilia.com/la-evolucion-del-comercio-electronico-del-ecommerce-al-mcommerce>
- Oviedo, E. (2015). *Ventajas del Cloud Computing*. Recuperado el 20 de octubre de 2017 de: <https://www.contunegocio.es/tecnologia/10-ventajas-del-cloud-computing/>
- Peña, J. (2018). *¿Qué Es Firebase y Qué Nos Aporta?* Recuperado el 03 de marzo de 2018 de: <https://arpentechnologies.com/es/blog/mobile-applications/que-es-firebase-y-que-nos-aporta/>
- Pérez, L. (2015). *¿Cómo Ayuda la Metodología Scrum a la Gestión de Proyectos de TI?* Recuperado el 20 de noviembre de 2017 de: <https://searchdatacenter.techtarget.com/es/cronica/Como-ayuda-la-metodologia-Scrum-a-la-gestion-de-proyectos-de-TI>

- Perry, S. (2012). *Conceptos Básicos del Lenguaje Java*. Recuperado el 20 de diciembre de 2017 de: <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introtojava1/index.html>
- Pimienta, P. (s.f.). *Tipos de Aplicaciones Móviles y Sus Características*. Recuperado el 10 de diciembre de 2017 de: <https://deideaaapp.org/tipos-de-aplicaciones-moviles-y-sus-caracteristicas/>
- Queralt, E. (2014). *¿Qué es el Método SCRUM y que Beneficios y Ventajas va a Aportar a Mi Negocio?* Recuperado el 19 de abril de 2018 de: <https://www.openiebs.com/blog/que-es-scrum-beneficios-ventajas-para-negocio-agile-scrum/>
- Quijano, J. (2017). *Historias de Usuario, Una Forma Natural de Análisis Funcional*. Recuperado el 15 de marzo de 2018 de: <https://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>
- Ruiz, M. (2017). *¿Qué es Firebase de Google?* Recuperado el 16 de enero de 2018 de: <https://openwebinars.net/blog/que-es-firebase-de-google/>
- Sánchez, X. (2016). *Metodología Lean Startup: Aprender a Crear Tu Empresa Con Más éxito*. Recuperado el 15 de enero de 2018 de: <https://www.emprenderalia.com/aprende-a-crear-tu-empresa-con-lean-startup/>
- Sanz, J. (2017). *Firestore: Qué Es y Por Qué Integrarlo en Nuestras APPs*. Recuperado el 05 de enero de 2018 de: <https://javiersanzrozalen.wordpress.com/2017/01/09/firebase-que-es-y-por-que-integrarlo-en-nuestras-apps/>
- Schwaber, Ken; Sutherland, Jeff. (2016). *La Guía de Scrum*. Recueprado el 19 de abril de 2018 de: *La Guía Definitiva de Scrum*.

Schwaber, Ken; Sutherland, Jeff. (2016). *La Guía de Scrum*. Recuperado el 19 de abril de 2018 de: *La Guía Definitiva de Scrum*.

Scipion, F. (s.f.). *¿Podemos Realmente Confiar en la Metodología Lean Startup para Montar un Negocio en la Nube?* Recuperado el 01 de diciembre de 2017 de: <https://www.lifestylealcuadrado.com/metodologia-lean-startup/>

Touza, R. (2017). *Metodología Lean Startup ¿Qué Es? y ¿Puede Ayudar a Mi Empresa?* Recuperado 11 de noviembre de 2017 de: <https://robertotouza.com/metodologia-lean-startup-que-es-puede-ayudar-empresa>

Trujillo, J. (2015). *Linea del Tiempo de la Historia del Comercio en el Mundo*. Recuperado el 10 de noviembre de 2017 de: <https://es.slideshare.net/josealdot/linea-del-tiempo-de-la-historia-del-comercio-en-el-mundo>

Villalobos, J. (2010). *Introducción a Java*. Recuperado el 05 de octubre de 2017 de: <http://codigoprogramacion.com/cursos/java/47-introjava.html>

Zamora, J. (2016). *¿Qué es Firebase? La Mejorada Plataforma de Desarrollo de Google*. Recuperado el 05 de enero de 2018 de: <https://elandroidelibre.lespanol.com/2016/05/firebase-plataforma-desarrollo-android-ios-web.html>

Anexos

¿Posee un teléfono inteligente? *

Si

No

¿Posee un plan de teléfono con Internet? *

Si

No

¿Cuál es el método más utilizado para la venta de su producto?

*

Llamada telefónica

Recorrido

¿Cuántos clientes en promedio tiene diariamente? *

Entre 1 - 10 Clientes

Entre 10 - 20 Clientes

Más de 20 Clientes

¿Tiene un sector específico de trabajo? *

Si

No

¿Estaría dispuesto a desplazarse una distancia moderada para acceder a un comprador seguro? *

	1	2	3	4	5	
Nada Probable	<input type="radio"/>	Muy Probable				

¿Estaría dispuesto a vender su producto en línea (Internet)? *

	1	2	3	4	5	
Nada Probable	<input type="radio"/>	Muy Probable				

¿Qué tan complejo le resulta conseguir clientes? *

	1	2	3	4	5	
Nada Complejo	<input type="radio"/>	Muy Complejo				

¿Estaría dispuesto a utilizar una aplicación móvil para vender su producto? *

	1	2	3	4	5	
Nada Probable	<input type="radio"/>	Muy Probable				

¿Estaría dispuesto a dar una comisión del precio del producto a quien lo venda? *

	1	2	3	4	5	
Nada Probable	<input type="radio"/>	Muy Probable				

Anexo1.- Encuesta realizada a vendedores ambulantes

¿Posee un teléfono inteligente?

- Si
- No

¿Posee Internet en su teléfono?

- Si
- No

¿Cuál es el producto o servicio que más utiliza?

- Compra de tanques de gas
- Compra de botellones de agua
- Compra de helados
- Servicio de lavandería
- Servicio de chatarrería
- Otros: _____

¿Qué tan complejo es conseguir el producto o servicio?

- | | | | | | | |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|--------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Nada Complejo | <input type="radio"/> | Muy Complejo |

Anexo 2.- Encuesta realizada a compradores

