

UNIVERSIDAD NACIONAL DE CHIMBORAZO



FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

Proyecto de Investigación previo a la obtención del Título de Ingeniero en Sistemas y
Computación

TRABAJO DE TITULACIÓN

DESARROLLO DE UN APLICATIVO MÓVIL MEDIANTE EL MANEJO DE
STREAMS DE DATOS ASÍNCRONOS PARA ENCOMIENDAS DE LA EMPRESA
VIPCAR RIOBAMBA

AUTOR:

Byron Rubén Velástegui Solís

TUTOR:

PhD. Miryan Estela Narváez Vilema

Riobamba - Ecuador

2020

VEREDICTO DE LA INVESTIGACIÓN

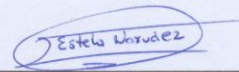
Los miembros del Tribunal de Graduación del proyecto de investigación de título: **“DESARROLLO DE UN APLICATIVO MÓVIL MEDIANTE EL MANEJO DE STREAMS DE DATOS ASÍNCRONOS PARA ENCOMIENDAS DE LA EMPRESA VIPCAR RIOBAMBA”**, presentado por: Byron Rubén Velástegui Solís, dirigida por: PhD. Miryan Estela Narváez Vilema.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso de custodia en la biblioteca de la facultad de Ingeniería de la UNACH.

Para constancia lo expuesto firman:

PhD. Miryan Estela Narváez Vilema

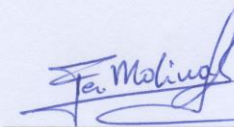
Director de Proyecto



Firma

PhD. Fernando Molina

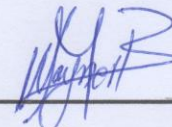
Miembro de Tribunal



Firma

Ing. Wayner Xavier Bustamante

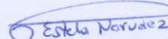
Miembro de Tribunal



Firma

AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación corresponde exclusivamente a: Byron Rubén Velástegui Solís con la dirección de la PhD. Miryan Estela Narváez Vilema y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”



PhD. Miryan Estela Narváez Vilema

060357677-8

Tutor de Proyecto de Investigación



Byron Rubén Velástegui Solís

050349904-8

Autor del Proyecto de Investigación

DEDICATORIA

Dedico este trabajo de investigación a mi Familia, amigos y mi novia por ser mi guía y fortaleza, a mi madre y mi padre que con mucho cariño, voluntad y paciencia me ha permitido llegar a cumplir una meta más en mi vida, enseñándome a creer lo que sueñas lo puedes llegar a cumplir en la vida, a mi familia y amigos que me han apoyado y brindado su ayuda en todo momento y a mi novia que con su aliento y ayuda me dio fortaleza para lograr mis metas.

BYRON RUBEN VELÁSTEGUI SOLIS

AGRADECIMIENTO

Agradezco a mis padres por alentarme y guiarme en la vida, a mis hermanos que en los momentos más difíciles me han sacado una sonrisa para seguir adelante y a mi novia que esta todos los días motivándome llenándome de alegría para continuar con mis sueños y cumplir más metas, aunque sean difíciles, pero jamás van a ser imposibles.

BYRON RUBEN VELÁSTEGUI SOLIS

ÍNDICE

VEREDICTO DE LA INVESTIGACIÓN.....	II
AUTORÍA DE LA INVESTIGACIÓN	III
DEDICATORIA.....	IV
AGRADECIMIENTO	V
RESUMEN	X
INTRODUCCIÓN.....	1
CAPITULO I.....	3
1. PLANTEAMIENTO DE PROBLEMA	3
1.1. Problema.....	3
1.2. Justificación.....	4
1.3. OBJETIVOS.....	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos.....	5
CAPÍTULO II.....	6
2. MARCO TEÓRICO	6
2.1. Aplicaciones Móviles.....	6
2.2. Tipos de Aplicaciones Móviles	6
2.3. Sistema Operativo Android	7
2.4. Android Studio.....	7
2.5. SQL Server.....	8
2.6. Firebase	8
2.7. Azure.....	9
2.8. Programación Reactiva	9
2.9. Extensiones Reactivas.....	11
2.10. Streams Reactivos	11
2.11. Ejecución Asíncrona	12

2.12.	Visual Studio.....	13
2.14.	Servicios Web	14
2.15.	Patrón Modelo Vista Controlador.....	14
2.16.	IBM SPSS Statistics 25.....	15
CAPÍTULO III		16
3.	METODOLOGÍA.....	16
CAPÍTULO IV		25
4.	RESULTADOS Y DISCUSIÓN	25
CAPÍTULO V.....		33
5.	CONCLUSIONES Y RECOMENDACIONES	33
5.1.	CONCLUSIONES	33
5.2.	RECOMENDACIONES	34
BIBLIOGRAFÍA		35
ANEXOS		37
ANEXO I:.....		38
ANEXO II:.....		39
ANEXO III:		45
ANEXO IV:		57

ÍNDICE DE TABLAS

Tabla 1: Requerimientos del Usuario	17
Tabla 2: Datos estadísticos de la batería utilizada en los dispositivos móviles	28
Tabla 3: Datos estadísticos de la memoria RAM utilizada en los dispositivos móviles.....	29
Tabla 4: Datos estadísticos del CPU utilizado en los dispositivos móviles	31
Tabla 5: Tablas de datos recolectados en los dispositivos móviles.....	39

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Procesos de ejecución Síncrona.....	10
Ilustración 2: Procesos de ejecución Asíncrona.....	10
Ilustración 3: Ejecución de un aplicativo Síncrono.....	12
Ilustración 4: Ejecución de una aplicación Asíncrona	13
Ilustración 5: Interrelación entre los elementos del patrón MCV.....	15
Ilustración 6: Caso de Uso General Aplicación Cliente.....	19
Ilustración 7: Caso de Uso General Aplicación Administrador	19
Ilustración 8: Caso de Uso General la Aplicación Repartidor	20
Ilustración 9: Diagrama de Componentes Aplicación VipCar	20
Ilustración 10: Diagrama de clases aplicativo VipCar Riobamba.....	22
Ilustración 11: Arquitectura de la Aplicación	23
Ilustración 12: Proceso de los datos en SPSS	24
Ilustración 13: Uso de Batería al utilizar la aplicación VipCar en los dispositivos móviles.	25
Ilustración 14: Uso del CPU al utilizar la aplicación VipCar en los dispositivos móviles.	26
Ilustración 15: Uso de la RAM al ejecutar la aplicación VipCar en los dispositivos móviles...	27
Ilustración 16: Batería utilizada por los dispositivos	28
Ilustración 17: Memoria RAM utilizada por los dispositivos móviles.....	30
Ilustración 18: Uso del CPU en los dispositivos móviles	31
Ilustración 19: Diagrama de la Base de Datos	38
Ilustración 20: Login y registro en la aplicación.....	45
Ilustración 21: Pagina de bienvenida a los usuarios.....	46
Ilustración 22: Menú del Administrador.....	47
Ilustración 23: Página de registro de un nuevo repartidor.....	48
Ilustración 24: Página de Cambio de contraseña administrador.....	49
Ilustración 25: Menú dinámico del Repartidor	50
Ilustración 26: Pagina de pedidos en proceso.	51
Ilustración 27: Pagina de pedidos recibidos.....	52
Ilustración 28: Pagina del detalle de los pedidos solicitados.	53
Ilustración 29: Alerta de confirmación del pedido.....	54
Ilustración 30: Pagina de pedidos entregados.	55
Ilustración 31: Menú de los clientes.	56
Ilustración 32: Pagina para describir el pedido.	56
Ilustración 33: Plataforma Azure.....	57

RESUMEN

El uso de streams de datos asíncronos o también conocida como programación reactiva proporciona tiempos de respuesta rápidos y consistentes.

La presente investigación tuvo como objetivo principal desarrollar una aplicación móvil para automatizar la recepción y envío de encomiendas que la empresa VipCar ofrece en la ciudad de Riobamba, con la finalidad de mejorar sustancialmente los procesos que hasta el momento funcionan de una manera no óptima en materia de eficiencia. Se implementaron nuevas tecnologías, abordando paradigmas de la programación asíncrona, permitiendo que la aplicación sea reactiva, es decir tener componentes activos y listos para recibir eventos.

Para el desarrollo del aplicativo móvil se utiliza streams de datos asíncronos, los cuales permitirán disminuir el uso de recursos, cooperando que sea una aplicación escalable, y brindara facilidad a la alta disponibilidad de datos. La aplicación móvil de encomiendas para la empresa VipCar Riobamba, posee una interfaz clara y sencilla mediante la cual los usuarios podrán realizar sus pedidos más eficientemente, optimizando tiempo en la ejecución de cada proceso, se llevará registros de tiempos en que son atendidos cada pedido para garantizar una constante optimización del servicio y atención inmediata a los usuarios. Además, permitirá obtener información más precisa, acerca de los clientes que contraten algún servicio ofertado.

El sistema fue evaluado en diez diferentes dispositivos móviles por diez días, se obtuvieron datos en primer plano y en segundo plano de ejecución, estos datos fueron analizados con la ayuda del software estadístico IBM SPSS demostrando que la aplicación ocupa la mínima cantidad de recursos: Memoria RAM, Batería y CPU.

Palabras Clave: Programación reactiva, streams de datos asíncronos, aplicativo móvil, servicio de encomiendas, eventos.

ABSTRACT

The use of asynchronous data streams or known also as reactive programming that provides fast and consistent response times.

The main objective of this research was to develop a mobile application to automate the reception and sending of orders that the company VipCar offers in the city of Riobamba, in order to substantially improve the processes that so far work in a non-optimal way about efficiency terms. New technologies were implemented, addressing paradigms of asynchronous programming, allowing the application to be reactive, that is, to have active components ready to receive events.

For the development of the mobile application, asynchronous data streams are used, which will reduce the use of resources, helping to make it a scalable application, and will facilitate a high availability of data. The mobile application for the company VipCar Riobamba, has a clear and simple interface through which users can make their orders more efficiently, optimizing time in the execution of each process, records of the time each order is served will be kept to ensure a constant optimization of service and immediate attention to users. In addition, it will allow to obtain more precise information, about the clients that contract some offered service.

The system was evaluated in ten different mobile devices for ten days, data was obtained in the foreground and in the background of execution, these data were analyzed with the help of IBM SPSS statistical software showing that the application occupies the minimum amount of resources: RAM, Battery and CPU memory.

Keywords: Reactive programming, asynchronous data streams, mobile application, delivery service, events.

Translation reviewed by:



MsC. Edison Damian
English Professor

INTRODUCCIÓN

La programación asincrónica se ha convertido en un tema importante de discusión en los últimos años, especialmente cuando se habla de ejecutar simultáneamente múltiples operaciones. Esta programación ha aparecido para organizar el trabajo entre las diferentes unidades de procesamiento de hardware disponible en el dispositivo. La programación asincrónica viene al rescate para resolver los problemas que podrían surgir de este nuevo paradigma de procesamiento y mejorar el rendimiento y el tiempo de respuesta en las aplicaciones. (Vasconcelos, 2016)

Android es un sistema operativo open source para teléfonos móviles, desarrollado por Open Handset Alliance (OHA) bajo autorización de google. Android se basa en un kernel Linux y se distribuye bajo una licencia Apache license 2.0. (Benbourahla, 2015)

Las aplicaciones se conocen comúnmente como "aplicaciones móviles" o simplemente "aplicaciones". En esta era de rápidos avances tecnológicos, las aplicaciones móviles se han convertido en una de las principales herramientas que se usa diariamente tanto en la vida personal como profesional. Las aplicaciones se pueden descargar fácilmente desde plataformas de distribución de aplicaciones móviles de forma gratuita o de compra. Estas aplicaciones desempeñan papeles clave para facilitar muchas aplicaciones que son fundamentales en nuestra sociedad actual, como comunicación, educación, negocios, entretenimiento, medicina, finanzas, viajes, servicios públicos y sociales. (Siuhi & Mwakalonge, 2016)

En el presente proyecto se propone un sistema de encomiendas para la empresa VipCar Riobamba, debido a la universalización de los dispositivos móviles los cuales posibilita la comunicación interpersonal ubicua y la realización de actividades ligadas a los equipos informáticos. Se propone el desarrollo de un sistema de encomiendas para los pedidos que se realizan a la empresa VipCar, el sistema se implementara en dispositivos con sistema operativo Android por ser el más utilizado por la comunidad. (Pérez, 2019)

La aplicación estará enfocada en facilitar la entrega de encomiendas que soliciten los clientes. La empresa proporcionará un servicio de calidad a los usuarios, pues se ha tomado en consideración las necesidades, requerimientos y preferencias de los clientes, el proceso del servicio va desde la recepción de encomiendas hasta la entrega de estos en los lugares de destino. Los alcances de usufructuar este proyecto es que se implemente en

gran medida en otras empresas en el futuro, puesto a que es independiente y se puede adaptar a cada empresa que desee este servicio. El aplicativo móvil estará enfocado en tres divisiones que son: administrador, repartidor y cliente, los mismos tendrán acceso a las interfaces de acuerdo al rol que posean.

El documento está organizado de la siguiente manera:

El capítulo I, detalla el problema, la justificación, el objetivo general y los objetivos específicos de la investigación.

El capítulo II, presenta una descripción general del marco teórico relacionado con la investigación.

El capítulo III, describe la metodología aplicada durante el desarrollo de la investigación.

El capítulo IV, da a conocer los resultados y la discusión de la investigación.

El capítulo V, establece las conclusiones y recomendaciones del estudio.

CAPITULO I

1. PLANTEAMIENTO DE PROBLEMA

1.1. Problema

Controlar el envío y entrega de paquetes desde un celular en tiempo real ya es posible y está dominando el mundo, gracias a las plataformas móviles que permite conectar a dos personas para realizar la entrega de encomiendas. Es una manera rápida y muy usada para enviar diferentes tipos de productos.

En Ecuador existen varias aplicaciones móviles para solicitar encomiendas de las cuales se señala las más utilizadas: Glovo que es española y Uber Eats de Estados Unidos, son aplicaciones mayormente utilizadas a nivel mundial y en Ecuador se utilizan en las ciudades grandes (Quito, Cuenca, Guayaquil). En la ciudad de Riobamba para solicitar encomiendas se utilizan las compañías de taxis, estas cooperativas no cuentan con aplicaciones propias que permitan agilizar la solicitud de los clientes, el servicio de encomiendas ha ido en auge por la ventaja de poder recibir sus pedidos en la puerta de su casa.

La falta de información y control del envío y entrega de paquetes sobre las encomiendas solicitadas a la empresa VipCar Riobamba, ha causado la pérdida de clientes, la manera de informar de las solicitudes de encomiendas se realiza mediante la aplicación WhatsApp, misma que en la actualidad no es de gran utilidad por la cantidad de pedidos que ingresa. Este proceso no es nada óptimo para la empresa, muchos de los usuarios optan por alguna otra compañía que les brinde el mismo servicio y les atiende de forma más rápida y eficaz.

Para solucionar la problemática se desarrolló una aplicación móvil para solventar dicho servicio, esta aplicación tendrá un alto impacto, debido a que se podrá llegar a tener más clientes porque la aplicación estará disponible en cualquiera parte, optimizando así los procesos de atención al cliente. Por ser una aplicación móvil cualquier persona que tenga acceso a Internet podrá utilizarla en cualquier lugar que se encuentre.

1.2. Justificación

Hoy en día, en Internet se puede encontrar todo tipo de contenido. El número de tiendas online es enorme, cualquier producto que se busque por raro que sea, tiene un precio y se encuentra al alcance de un clic, sin importar el país de origen. La psicosis que existía hace unos años de “si no me llega”, ha desaparecido casi por completo (aunque lamentablemente siga habiendo casos puntuales). Sin embargo, si hay un punto negativo en las compras por la red, es sin duda, el tiempo de espera hasta recibir el paquete. Es obvio que en tiendas tipo Amazon o cualquier otra, ese tiempo es menor, sobre todo si se decide acelerar el proceso pagando más por un envío preferente, incluso si eres un comprador habitual pierdes la cuenta del tiempo que ha pasado desde la compra e incluso la llegada de paquetes al cabo de horas, días o varias semanas.

En la actualidad para ahorrar tiempo las distintas empresas de transporte cuentan con aplicaciones móviles para Android e iOS. Las empresas que deciden contar con aplicaciones móviles que se adaptan a sus necesidades van en aumento porque les ofrecen beneficios para el crecimiento empresarial. (Pérez, 2019)

La compañía VipCar Riobamba tiene una gran acogida en el servicio de encomiendas que realizan a diario, aquellas personas que trabajan de repartidores han estado realizando la receptación de la información mediante aplicaciones no dedicadas a ello, teniendo varios inconvenientes y el proceso no se desarrolla rápidamente. Por ese motivo se desarrolló un aplicativo móvil, que permite que los datos ingresen de forma rápida y de manera específica para cada cliente.

La creación del aplicativo móvil para la empresa VipCar Riobamba fomentará el crecimiento de la empresa aportando una mayor factibilidad al momento de solicitar encomiendas, llevando al aumento de clientes que utilicen el aplicativo y el servicio. La aplicación tiene varios módulos que gestiona eficazmente la información de los clientes y del servicio que oferta la empresa, con el propósito de mejorar todos los procesos que hoy en día se demoran demasiado y a menudo son ineficientes.

1.3. OBJETIVOS

1.3.1. Objetivo General

Desarrollar un aplicativo móvil mediante el manejo de Streams de datos asíncronos para encomiendas de la empresa VipCar Riobamba.

1.3.2. Objetivos Específicos

- Analizar las técnicas y herramientas para Streams de datos asíncronos en aplicaciones móviles.
- Desarrollar del aplicativo móvil para encomiendas en la empresa Vip Car Riobamba.
- Evaluar el desempeño de la aplicación móvil en dispositivos Android.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Aplicaciones Móviles

Son programas diseñados para ser ejecutados en dispositivos móviles, que permiten resolver una o varias tareas en específico. Son similares a los conocidos procesadores de texto, las hojas de cálculo, los programas de diseño y edición de video de los ordenadores de escritorio, pero con una complejidad menor y optimizadas para el contexto móvil. (Serna & Pardo, 2016)

2.2. Tipos de Aplicaciones Móviles

Aplicaciones Nativas

Son aquellas que están desarrolladas para ejecutarse sobre un sistema operativo o plataforma específicos y se desarrollan en el lenguaje nativo de cada dispositivo, de ahí su nombre, estas aplicaciones están escritas en un lenguaje de programación propio, lo que permite acceder a todos los recursos del hardware del dispositivo móvil, como la cámara de fotos y el GPS. (Hereter & Zanini, 2016)

Aplicaciones Híbridas

Se trata de una aplicación web desarrollada con los estándares HTML, CSS y JavaScript, entre otros, la cual es empaquetada bajo un conjunto de reglas y parámetros que permite instalarla en un dispositivo como cualquier aplicación nativa, luego al ser ejecutada, la aplicación híbrida utilizara el motor del navegador web, ocultando su menú, su barra de direcciones y de herramientas para poder simular que la WebApp es una aplicación nativa. (Luna, 2016)

Las aplicaciones híbridas tienen muchos beneficios sobre las aplicaciones nativas puras, específicamente en términos de soporte de plataformas, velocidad de desarrollo y acceso a código de terceros. (Fennema et al., 2017)

2.3. Sistema Operativo Android

Inicialmente pensado para teléfonos móviles y Tablet, hasta que poco a poco fue dominando mercados y ahora se puede encontrar en televisores, relojes, e incluso coches. La gran ventaja de Android es que está basado en Linux, un sistema operativo libre y multiplataforma, en julio de 2005 es comprado por Google, que saca su primera versión del sistema operativo el 5 de noviembre de 2007, aunque los primeros terminales con Android no saldrían hasta 2008. (Orozco, 2018)

2.4. Android Studio

Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de APPS para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas APPS para Android, como las siguientes: (Android, 2020)

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros.
- Compatibilidad con C++ y NDK.
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine.

2.5. SQL Server

Es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos. Proporciona nuevas soluciones de copia de seguridad y recuperación ante desastres, así como de arquitectura híbrida con Windows Azure, lo que permite a los clientes utilizar sus actuales conocimientos con características locales que aprovechan los centros de datos globales de Microsoft. (Microsoft, 2019)

Características

- Soporte para procedimientos almacenados.
- Permite gestionar información de servidores de bases de datos de terceros.
- Incluye potente entorno gráfico para administración, permitiendo el uso de comandos MML y DDL de manera gráfica.
- Escalabilidad, seguridad y estabilidad.
- Soporte para transacciones.
- Cliente-servidor, donde los datos e información se albergan en el servidor y los clientes o terminales de la red acceden solo a la información.

Incluye una versión reducida denominada MSDE orientado a proyectos más pequeños como el mismo gestor de base de datos, además Microsoft SQL Server se establece como una alternativa de Microsoft a otros poderosos sistemas gestores de bases de datos como Sybase ASE, Oracle, PostgreSQL o MySQL. (Santamaría & Hernández, 2017)

Brinda a las organizaciones herramientas efectivas para proteger, desbloquear y escalar el poder de sus datos, desde computadoras de escritorio, teléfonos y tabletas, hasta centros de datos y nubes privadas y públicas. (Mistry & Misner, 2012)

2.6. Firebase

Es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2012 y adquirida por Google en 2014. Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de

usuarios y dando resultado también a la obtención de una mayor monetización. (Google, 2016)

2.7. Azure

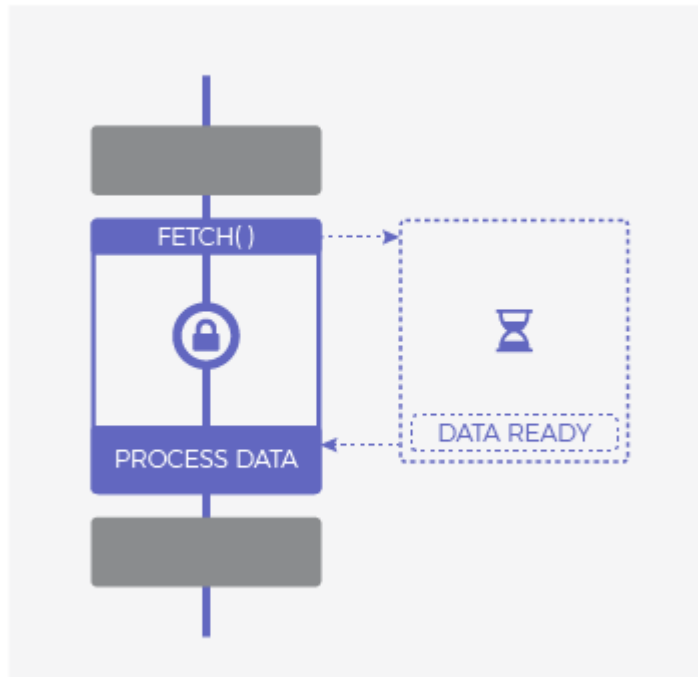
Es un conjunto integral de servicios en la nube que los desarrolladores y los profesionales de TI utilizan para crear, implementar y administrar aplicaciones a través de la red global de centros de datos. (Microsoft Azure, 2019)

2.8. Programación Reactiva

Es otro paradigma de programación, está basado en torno a la capacidad de expresar fácilmente los flujos de datos y la propagación automática de cambios. (Sikora, 2017) La programación reactiva se trata de la propagación del cambio. También se conoce como programación declarativa, donde se expresa la intención y el estado de la aplicación como dinámicamente determinado por los cambios en los factores subyacentes. (Sharma, 2018)

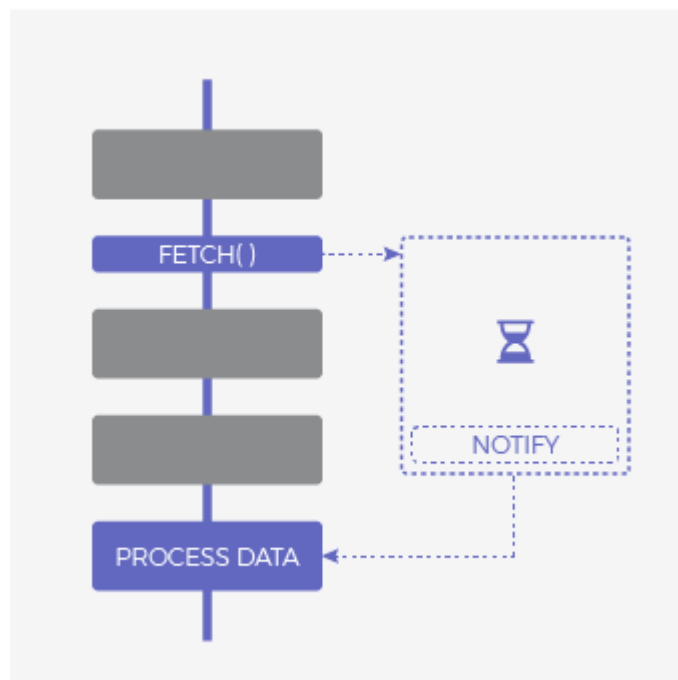
En consecuencia, si un programa reacciona a los cambios en los factores subyacentes, se le puede llamar reactivo. Los programas reactivos se pueden construir utilizando técnicas imperativas, como las devoluciones de llamada. Esto puede estar bien para un programa que tiene un solo evento. Sin embargo, para aplicaciones donde están ocurriendo cientos de eventos, esto podría conducir fácilmente al infierno de devolución de llamada; se puede tener numerosas devoluciones de llamada confiando el uno en el otro, y sería realmente difícil determinar cuáles se estaban ejecutando. Como resultado, se requerirá un nuevo conjunto de abstracciones que permitan construir a la perfección interacciones asíncronas impulsadas por eventos a través de un límite de red. Hay bibliotecas en diferentes idiomas imperativos, como Java, que proporcionan estas abstracciones. Estas bibliotecas se denominan extensiones reactivas. (Sharma, 2018)

Ilustración 1: Procesos de ejecución Síncrona



Fuente: LEMONCODE - Javascript Asíncrono: La guía definitiva (2018).
Recuperado de: <https://lemoncode.net/lemoncode-blog/2018/1/29/javascript-asincrono>

Ilustración 2: Procesos de ejecución Asíncrona



Fuente: LEMONCODE - Javascript Asíncrono: La guía definitiva (2018).
Recuperado de: <https://lemoncode.net/lemoncode-blog/2018/1/29/javascript-asincrono>

2.9. Extensiones Reactivas

Conocidas como ReactiveX, permiten expresar los eventos asincrónicos en una aplicación como un conjunto de secuencias observables. (Sharma, 2018)

La ReactiveX API consta de los siguientes componentes:

- Observables: interpreta el concepto central de Reactivex. Simbolizan las secuencias de elementos emitidos y generan eventos que se propagan a los suscriptores previstos.
- Observador: cualquier aplicación puede expresar su intención para eventos publicados por un observable creando un observador y suscribiéndose al observable respectivo
- Operadores: permiten transformar, combinar y manipular las secuencias de elementos emitidos por observables. (Sharma, 2018)

2.10. Streams Reactivos

Las secuencias reactivas son una especificación que determina el conjunto mínimo de interfaces requeridas para construir el procesamiento asincrónico de un gran volumen de datos sin límites. Es una especificación dirigida a JVM y JavaScript runtime. El objetivo principal de la especificación de flujos reactivos es estandarizar el intercambio de datos de flujo a través de un límite asíncrono de aplicaciones. (Sharma, 2018)

La API consta de cuatro interfaces:

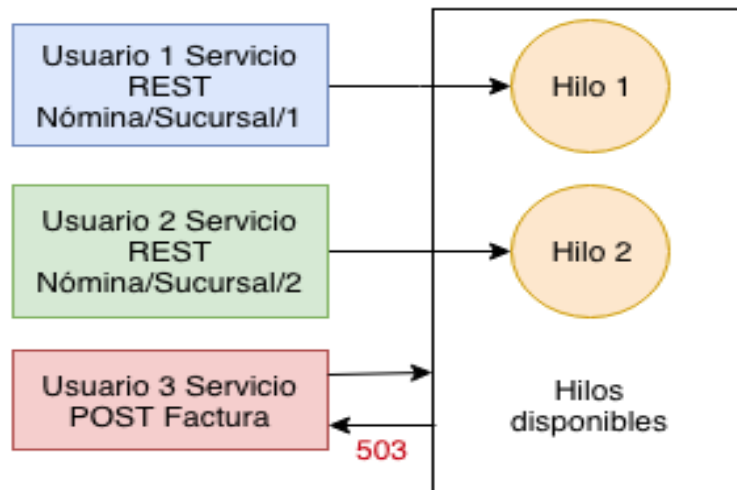
- Editor: es responsable de la generación de un número ilimitado de eventos asincrónicos y de enviar esos eventos a los suscriptores asociados.
- Suscriptor: es un consumidor de los eventos publicados por un editor. El suscriptor recibe eventos de suscripción, datos, finalización y error. Puede elegir realizar acciones en cualquiera de ellos.
- Suscripción: es un contexto compartido entre el editor y el suscriptor, con el propósito de mediar el intercambio de datos entre los dos. La suscripción está disponible solo con el suscriptor y le permite controlar el flujo de eventos del editor. La síntesis se invalida si hay un error o una finalización. Un suscriptor también puede cancelar las suscripciones para cerrar su transmisión.

- Procesador: representa una etapa de procesamiento de datos entre un suscriptor y un editor. En consecuencia, están obligados mutuamente. El procesador tiene que obedecer el contrato entre el editor y el suscriptor. Si hay un error, debe propagarlo de nuevo al suscriptor. (Sharma, 2018)

2.11. Ejecución Asíncrona

Se refiere a la capacidad de ejecutar tareas sin tener que esperar para terminar primero las tareas ejecutadas previamente. El modelo de ejecución desacopla las tareas, de modo que cada una de ellas se puede realizar simultáneamente, utilizando el hardware disponible. La API de Streams reactiva ofrece eventos de forma asíncronica. Un editor puede generar datos de eventos en forma de bloqueo síncronico. Por otro lado, cada uno de los controladores en el evento puede procesar los eventos de forma sincronizada. Sin embargo, la publicación de eventos debe ocurrir de forma asíncronica. El suscriptor no debe bloquearlo mientras procesa eventos. (Sharma, 2018)

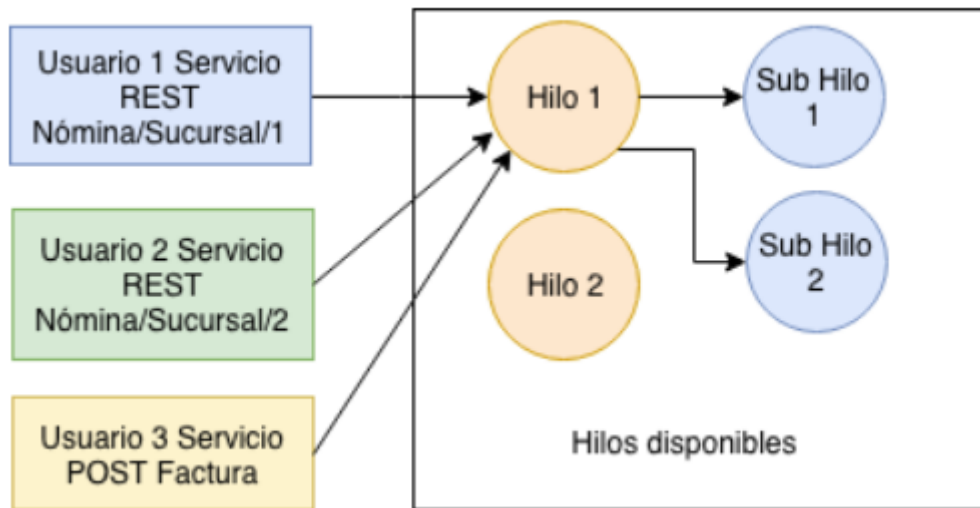
Ilustración 3: Ejecución de un aplicativo Síncrono



Fuente: Servicios REST con ASP.NET Core y Entity Framework Core - Programación asíncrona (2018).

Recuperado de: <https://abi.gitbook.io/net-core/3.-servicios-rest/3.9-programacion-asincrona>

Ilustración 4: Ejecución de una aplicación Asíncrona



Fuente: Servicios REST con ASP.NET Core y Entity Framework Core - Programación asíncrona (2018).

Recuperado de: <https://abi.gitbook.io/net-core/3.-servicios-rest/3.9-programacion-asincrona>

2.12. Visual Studio

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todo el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Además, los lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. (Microsoft, 2016)

2.13. Node JS

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables.

Esto contrasta con el modelo de concurrencia más común de hoy en día, en el que se emplean hilos del Sistema Operativo. Las redes basadas en hilos son relativamente ineficientes y muy difíciles de usar. Además, los usuarios de Node.js están libres de preocuparse por el bloqueo del proceso, ya que no existe. Casi ninguna función en

Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea. Por ello, es muy propicio desarrollar sistemas escalables en Node.js. (Foundation, 2020)

2.14. Servicios Web

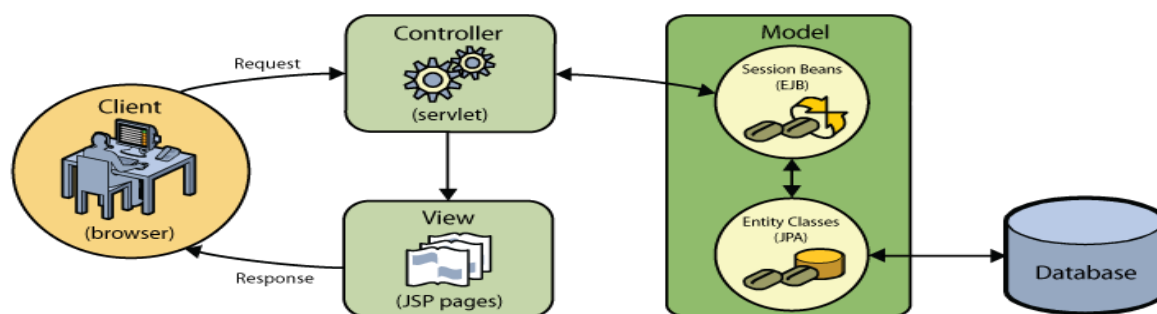
Un servicio web expone funcionalidad a un consumidor, es una URL programable y proporciona mecanismos para invocar operaciones de forma remota a través de internet, está basado en estándares web (HTTP, XML, SOAP, WSDL, UDDI), puede implementarse en cualquier plataforma, actuando como caja negra. (Pelechano, 2020)

2.15. Patrón Modelo Vista Controlador

Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas.

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, 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, la Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo. El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. (Díaz & Fernández, 2012)

Ilustración 5: Interrelación entre los elementos del patrón MCV



Fuente: NetBeans E-commerce Tutorial - Designing the Application (2015).
Recuperado de <https://netbeans.org/kb/docs/javaee/ecommerce/design.html>

2.16. IBM SPSS Statistics 25

El software estadístico SPSS (Statistical Package for the Social Sciences), es uno de los programas mayormente usados a nivel mundial, los procesos estadísticos que incluye ayudan a entidades que necesitan analizar datos para aplicaciones prácticas o diversas necesidades de investigación. Además, ofrece crear vínculos con herramientas office para finalmente permitir efectuar análisis estadísticos muy complejos.

Usos Potenciales

Permite crear un archivo en forma estructurada, además de organizar una base de datos que puede ser examinada con diferentes métodos estadísticos, permite realizar y capturar análisis de datos sin depender de otros programas y por último hace posible la transformación de un banco de datos creado en Excel a una de SPSS.

Ventajas y Desventajas

El software hace uso de distintos cuadros de diálogo que permiten determinar las acciones a tomar seleccionando análisis útiles, sin embargo, el usuario al no contar con una experiencia previa haciendo uso de SPSS hace complicado escoger las opciones de análisis que este provee. (Castañeda et al, 2010)

CAPÍTULO III

3. METODOLOGÍA

La metodología del proyecto se enfoca en una investigación cuantitativa, permitiendo realizar un análisis estadístico con los datos recolectados, para evaluar el desempeño de la aplicación VipCar Riobamba se instaló en 10 dispositivos con sistemas operativo Android. El *monitor de información* (todos los dispositivos móviles tienen) fue la herramienta que proporciono la información necesaria para medir el rendimiento de cada dispositivo.

3.1. Tipo y Diseño de Investigación

- 3.1.1. Descriptiva: permitió recolectar datos que expusieron la situación actual del objeto de estudio detallando la información obtenida de los diferentes dispositivos móviles con Android.
- 3.1.2. Transversal: conlleva a una comparación de muestras independientes obtenidas en un periodo de diez días en diferentes dispositivos móviles con sistemas operativos Android de distintas versiones.
- 3.1.3. El tipo de estudio es cuasi experimental dado que busco evaluar los recursos que utilizo el aplicativo móvil VipCar Riobamba al momento de encontrarse en primer plano y segundo plano de ejecución en los distintos dispositivos móviles con sistema operativo Android.

3.2. Unidad de Análisis

La ejecución de la aplicación se lo llevo a cabo durante 10 días, y los datos se obtuvieron al momento que se encuentra en primer plano y segundo plano de ejecución. Los datos obtenidos fueron registrados y analizados con la ayuda del software estadístico SPSS.

3.3. Tamaño de Muestra

La aplicación se instaló en 10 dispositivos móviles con diferentes versiones de sistema operativo Android.

3.4. Técnica de Recolección de Datos

3.4.1. Entrevistas

Se optó por la entrevista informal con el objetivo de determinar los requerimientos y las funcionalidades del aplicativo móvil para proceder con su desarrollo. Se realizó la entrevista al Presidente y Gerente de la empresa VipCar Riobamba, permitiendo identificar la situación actual de la empresa y los problemas que se solucionarían al implementar el aplicativo.

3.5. Técnicas de Análisis e interpretación de la información

Se utilizó la metodología en Cascada para plantear las fases del proyecto de forma secuencial, concretando reuniones para obtener los requerimientos y la funcionalidad que tendrá el aplicativo móvil, la información recolectada ha permitido establecer una planificación funcional para cumplir con los requerimientos del producto que solicita la empresa.

3.6. Pasos de la metodología en Cascada

3.6.1. Análisis.

Requerimientos del cliente: Se obtuvo los requerimientos del usuario y las necesidades a cubrir por el aplicativo móvil, mediante entrevistas y reuniones de innovación con el presidente y el gerente de la empresa VipCar Riobamba.

Tabla 1: Requerimientos del Usuario

REQUERIMIENTO FUNCIONAL (RF)	DESCRIPCIÓN DEL REQUERIMIENTO
RF1	La aplicación funcionará en dispositivos móviles con sistema operativo Android.
RF2	La aplicación contará con diferentes tipos de usuarios: clientes, administradores y repartidores.
RF3	Los nuevos clientes se podrán registrar directamente en la aplicación, mientras que el administrador y el repartidor solo podrán ser registrados por un administrador principal.
RF4	El cliente al iniciar sesión, observará la pantalla de bienvenida y un botón de menú desplegable.
RF5	El cliente podrá solicitar y cancelar pedidos realizados.

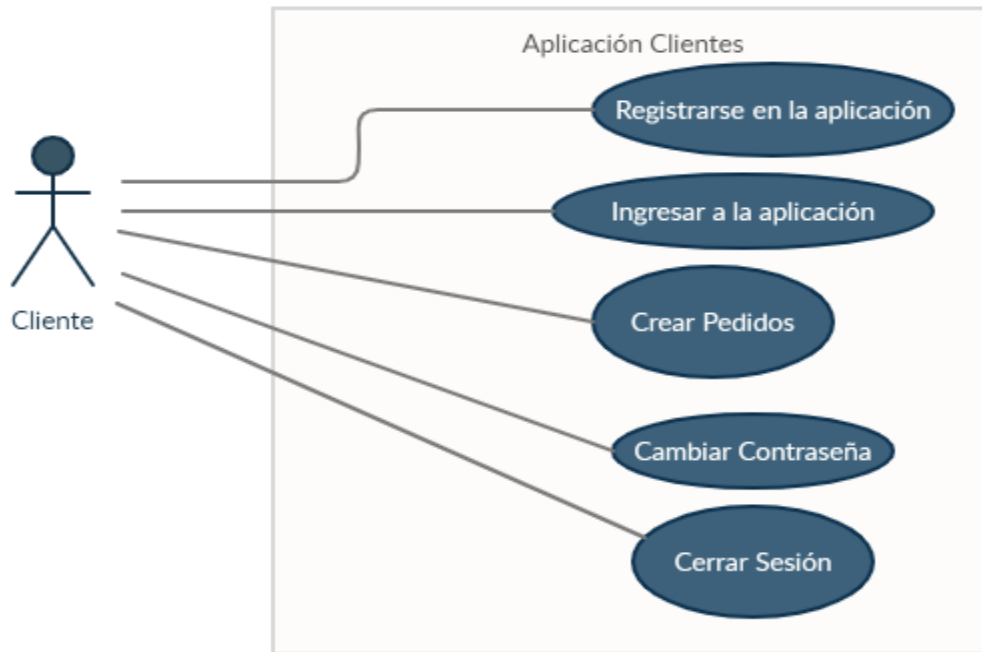
RF6	La aplicación permite enviar notificaciones a los repartidores de pedidos realizados por los clientes, mostrando dirección y datos del pedido en la barra de notificaciones.
RF7	Los usuarios repartidores cuenta con opciones de visualización de pedidos recibidos, pedidos aceptados y pedidos finalizados, un repartidor podrá aceptar un pedido y se le asignara a esa unidad y ningún otro repartidor podrá aceptar el mismo pedido.
RF8	El administrador podrá registrar y eliminar a repartidores y administradores.
RF9	El repartidor podrá llamar directamente desde la aplicación al cliente.
RF10	Los repartidores al aceptar un pedido enviaran el tiempo estimado de llegada y el número de la unidad que transportara el pedido.

Realizado por: El Autor

REQUERIMIENTO NO FUNCIONAL (RNF)	DESCRIPCIÓN DEL REQUERIMIENTO
RNF1	La aplicación necesita mínimo 20 megas de almacenamiento disponible para poder ser instalada y pueda funcionar.
RNF2	La aplicación debe ser fácil para su utilización.
RNF3	Las interfaces de la aplicación deben ser amigables e intuitivas.
RNF4	La aplicación debe ser fácil de descargar e instalar.
RNF5	La aplicación debe mantener los datos almacenados y seguros en la plataforma AZURE.
RNF6	La aplicación debe ser fácil de analizar y modificar para corregir posibles fallas.

Realizado por: El Autor

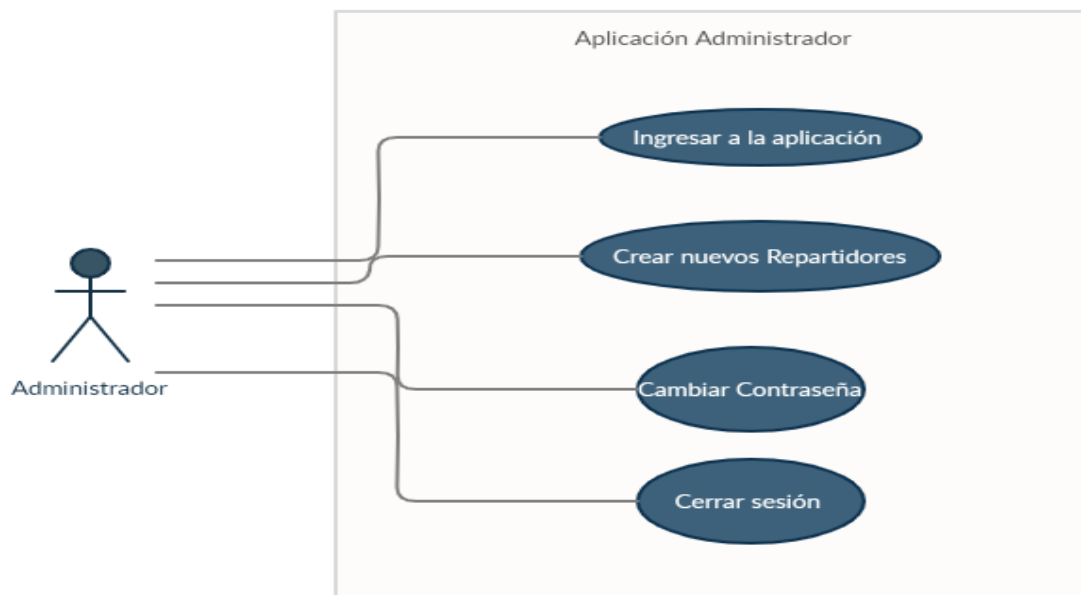
Ilustración 6: Caso de Uso General Aplicación Cliente



Realizado por: El Autor

En la ilustración 6 se observa la interacción entre el cliente y la aplicación de pedidos VipCar Riobamba.

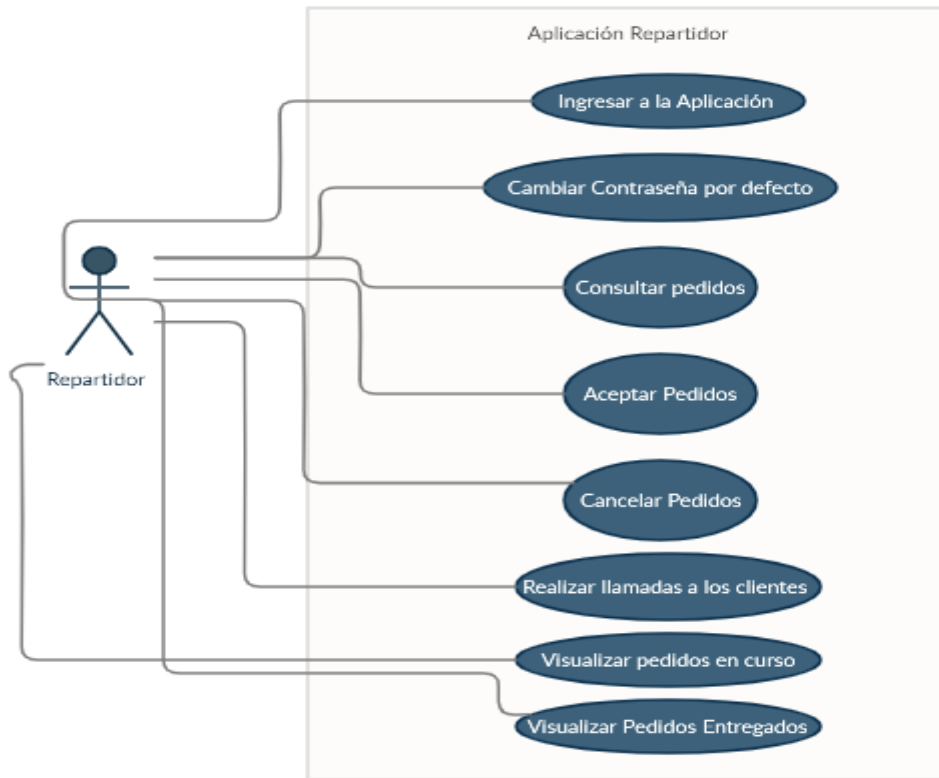
Ilustración 7: Caso de Uso General Aplicación Administrador



Realizado por: El Autor

En la ilustración 7 se observa de manera general la relación del administrador y la aplicación VipCar Riobamba.

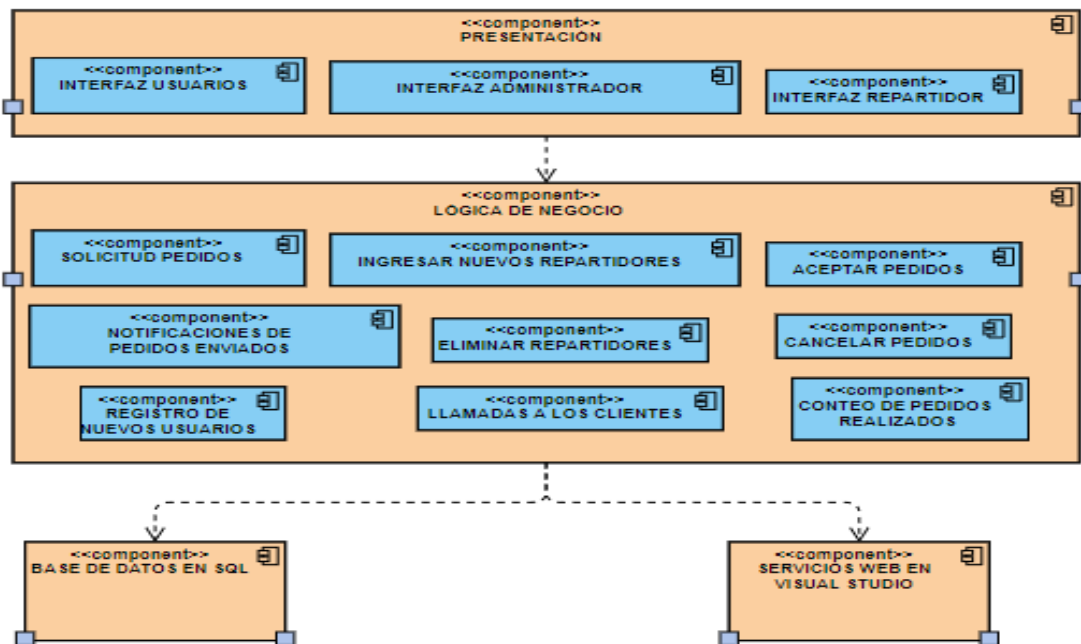
Ilustración 8: Caso de Uso General la Aplicación Repartidor



Realizado por: El Autor

En la ilustración 8 se observa como el agente repartidor interactúa en el escenario de la aplicación móvil VipCar Riobamba.

Ilustración 9: Diagrama de Componentes Aplicación VipCar



Realizado por: El Autor

En la ilustración 9 se observa los elementos físicos del sistema y las relaciones.

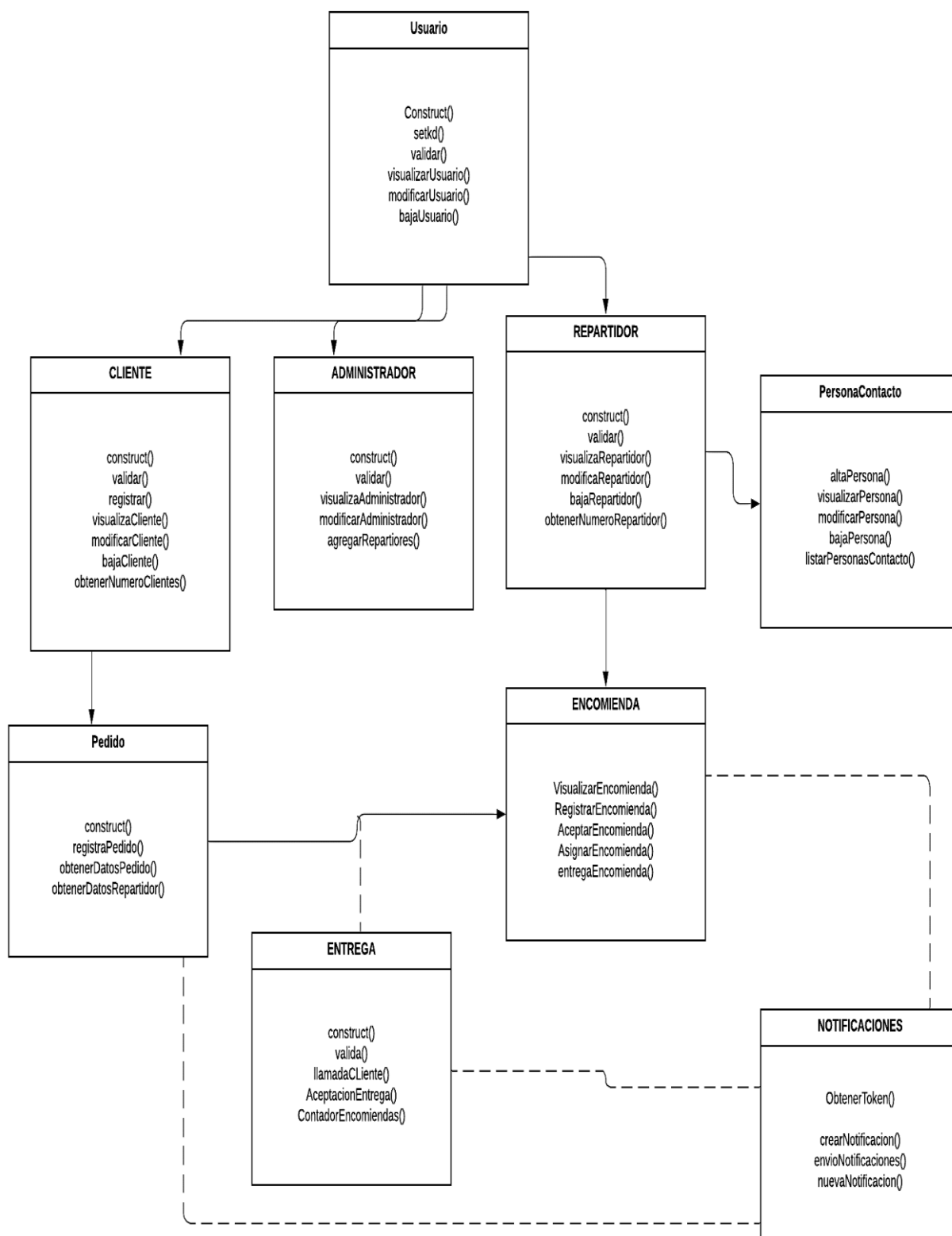
3.6.2. Diseño.

Se efectuó el modelado de la base de datos analizando los campos necesarios para el funcionamiento adecuado al almacenar los datos del aplicativo móvil. Utilizando los requerimientos solicitados se realizó un bosquejo del aplicativo móvil, el mismo fue presentado y analizado por los directivos de la empresa, se presenta las interfaces gráficas del proyecto bajo los requerimientos que se plantearon (Anexo 3).

3.6.3. Desarrollo.

Después de haber levantado la información necesaria y las funciones que debe realizar el aplicativo móvil; se inició a programar utilizando el framework IONIC para el desarrollo del aplicativo móvil, para la creación de los servicios web se utilizó el IDE Visual Studio, la plataforma digital Firebase para crear las notificaciones en cloud messaging la cual se conecta al aplicativo móvil en tiempo real, el gestor de datos SQL server para la base de datos (ver Anexo 1) y la plataforma digital AZURE para administrar los servicios web de la aplicación y la conexión con la base de datos.

Ilustración 10: Diagrama de clases aplicativo VipCar Riobamba



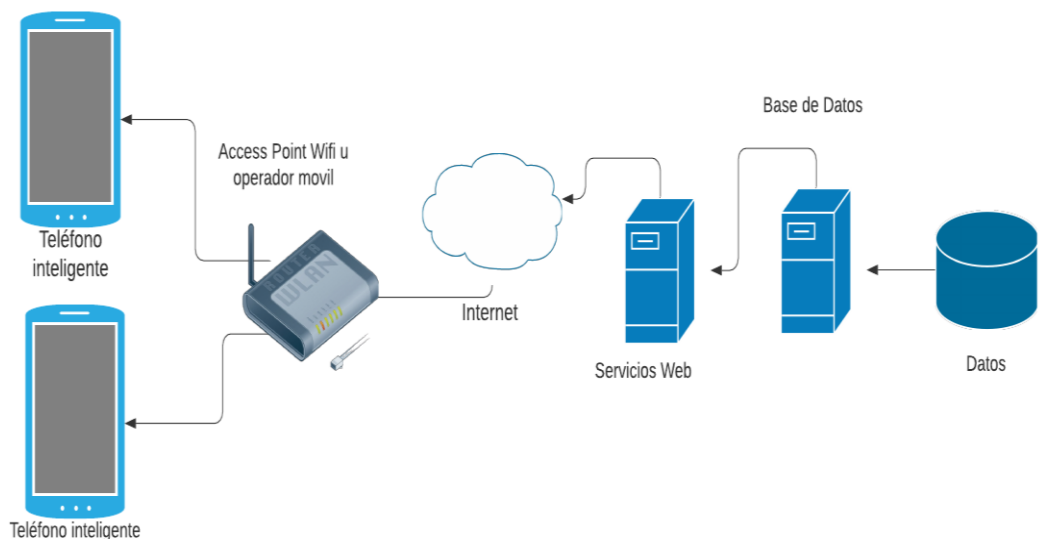
Realizado por: El Autor

La ilustración 10 presenta el diagrama estructural de la aplicación VipCar Riobamba mostrando sus atributos, las relaciones entre los objetos y sus clases.

3.6.4. Pruebas.

Al concluir el desarrollo del aplicativo móvil se comprueba el funcionamiento, se descarga el paquete de instalación (Android Application Package - APK) para instalar en los dispositivos móviles, se evidencia las mediciones de los recursos que consume (CPU, memoria RAM y la batería) al ejecutarse la aplicación VipCar.

Ilustración 11: Arquitectura de la Aplicación



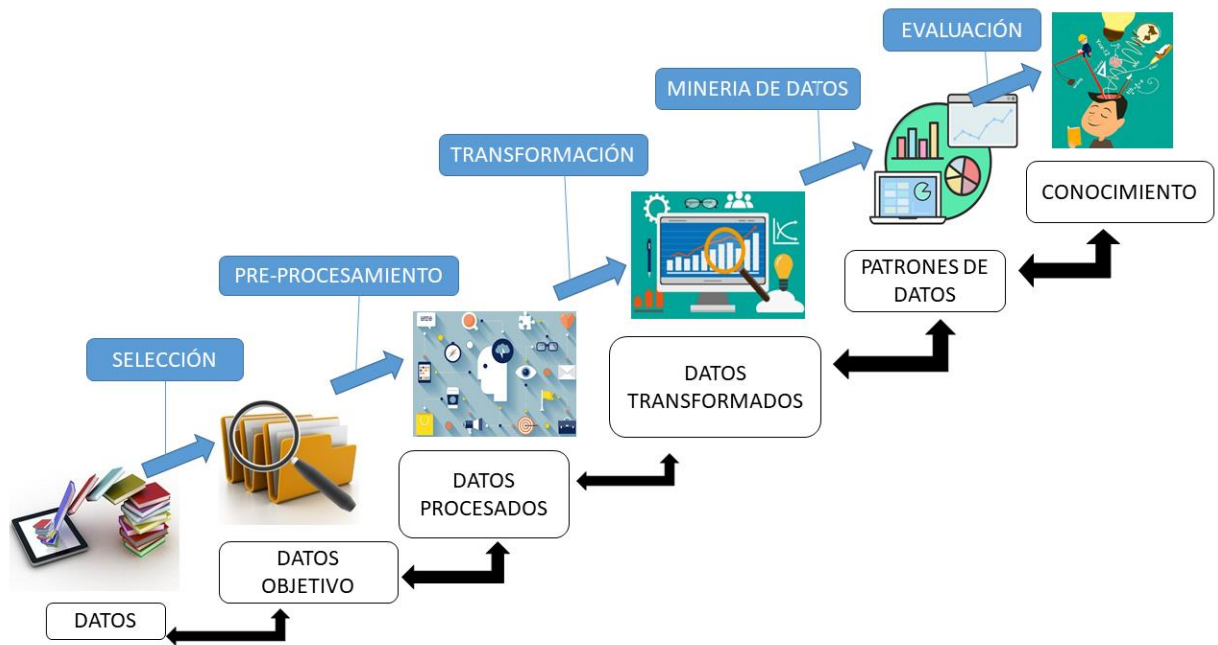
Realizado por: El Autor

3.7. Análisis de datos

Para asegurar la calidad de los datos se recolectaron en dos escenarios: 1) primer plano y 2) segundo plano, permitiendo observar de mejor manera los servicios que se están consumiendo, para disminuir el error al momento de recolectar los datos, se cerraron todas las aplicaciones que se encontraban en segundo plano de ejecución, dejando abierta solo VipCar Riobamba.

Mediante la herramienta estadística SPSS se analizaron los datos (anexo 2), que fueron proporcionados por el monitor de información que viene instalado en todos los dispositivos con sistema operativo Android, permitiendo conocer los recursos que están siendo utilizados por el aplicativo móvil al momento de encontrarse ejecutando en primer plano y segundo plano respectivamente.

Ilustración 12: Proceso de los datos en SPSS



Realizado por: El Autor

$$\sum_{i=1}^{10} \frac{x_i}{10}$$

X = Datos obtenidos de los servicios consumidos

i = número desde que literal se va a iniciar

10 = número de literales

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

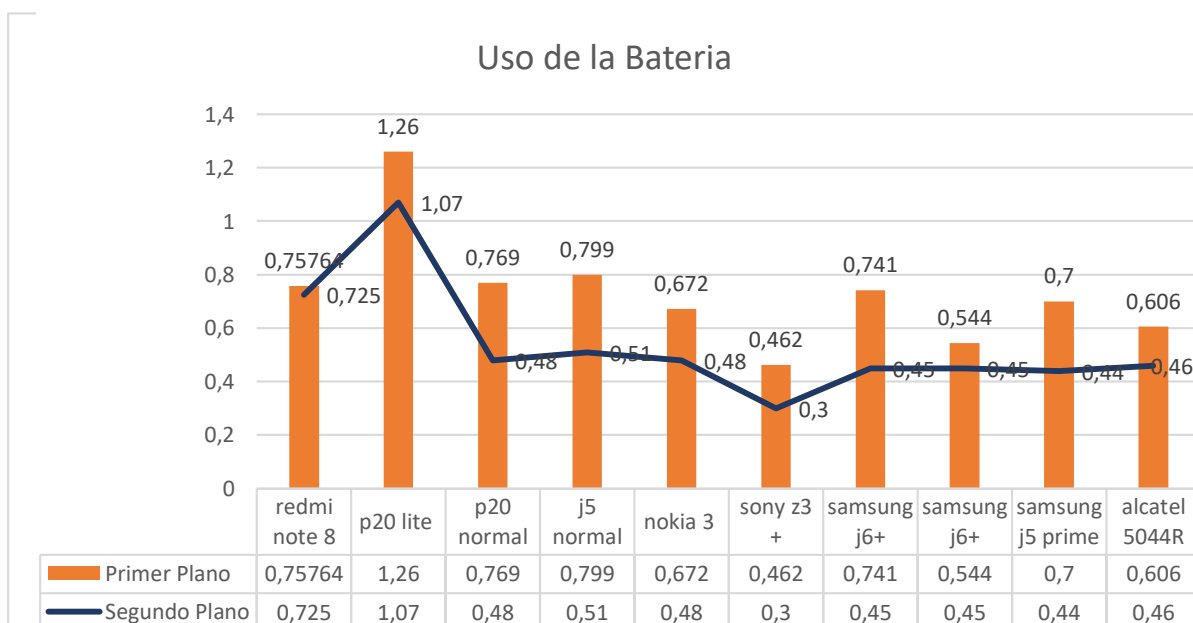
En este capítulo se establece los datos obtenidos mediante el monitor de información con el que cuentan los dispositivos móviles con sistema operativo Android, las pruebas se realizaron en diez dispositivos móviles, durante diez días en diferentes intervalos de tiempo, lo que permitió evaluar el desempeño de la aplicación VipCar, los datos se obtuvieron en 3 escenarios:

1. Aplicación en primer plano de ejecución (interacción entre la aplicación y el usuario en la pantalla principal).
2. Aplicación en segundo plano de ejecución (son los procesos de las aplicaciones que se encuentran ejecutando cuando el dispositivo móvil no se encuentra en uso).
3. Teléfonos móviles sin la aplicación VipCar instalada.

Durante los 10 días se obtuvieron 900 datos, mismos que se analizaron con el software estadístico BPM SPSS.

4.1. Datos de la batería al ejecutar la aplicación en primer y segundo plano.

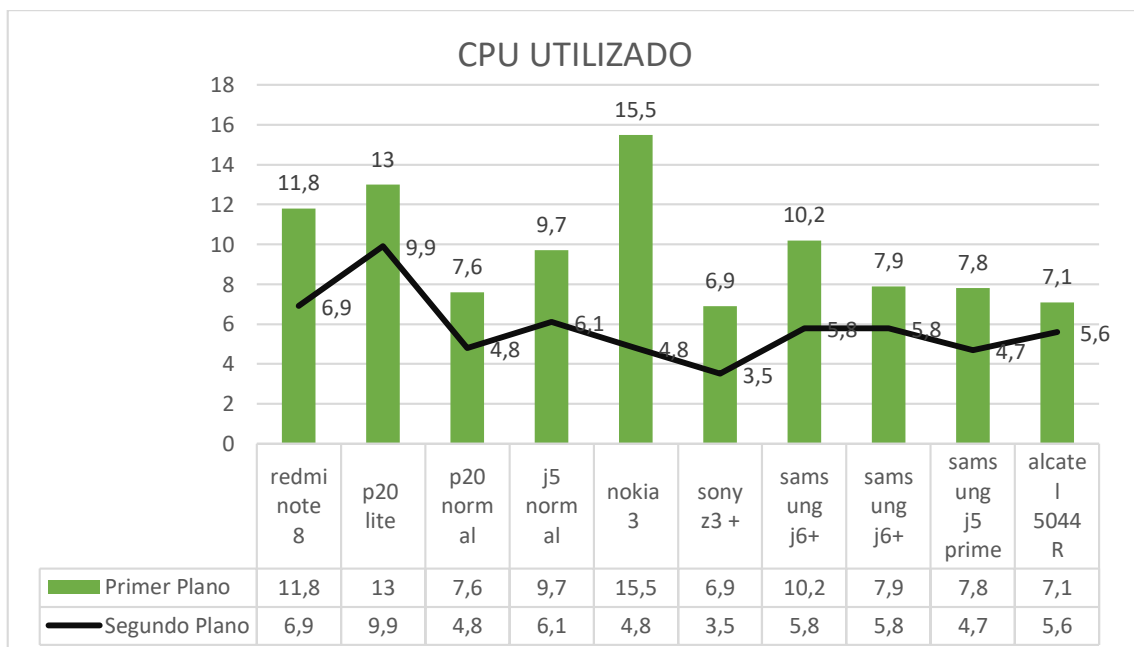
Ilustración 13: Uso de Batería al utilizar la aplicación VipCar en los dispositivos móviles.



Realizado por: El Autor

La ilustración 2 muestra los resultados obtenidos de la utilización de la batería promedio al momento de realizar pedidos y recibir las notificaciones durante 10 días de recolección de datos, se observa que 9 de 10 dispositivos móviles cuentan con un consumo de batería baja al momento de interacción con el usuario (primer plano), demostrando un porcentaje menor al 1% de consumo; reduciendo aún más al momento de encontrarse ejecutando en segundo plano, mientras que el dispositivo p20 lite consume 1.26% llevándolo a ser el dispositivo que más batería utiliza al momento de ejecutar la aplicación en primer y segundo plano.

Ilustración 14: Uso del CPU al utilizar la aplicación VipCar en los dispositivos móviles.

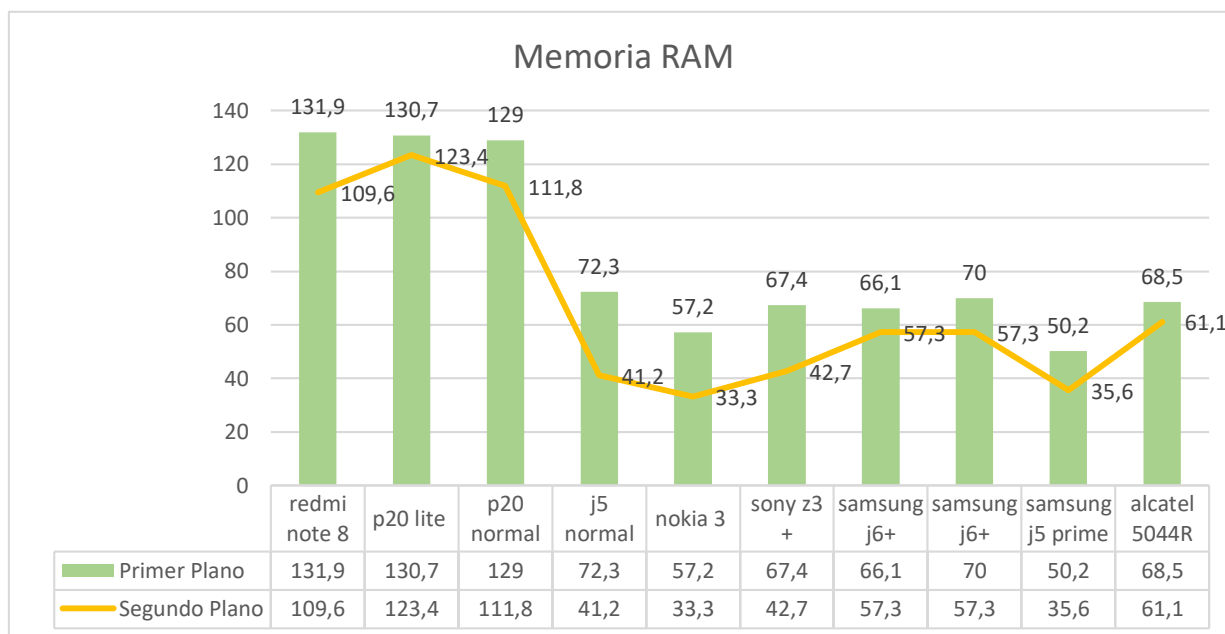


Realizado por: El Autor

La ilustración 3 expone los datos obtenidos sobre el uso del CPU durante los 10 días en diferentes dispositivos móviles, se observa que el dispositivo que utiliza más el CPU es el Nokia 3 con un 15.5% al momento que está ejecutándose la aplicación en primer plano, en segundo plano el aplicativo se adapta y finaliza todos los servicios que no utiliza y permite que se reduzca a 4.8% de uso; el dispositivo Sony z3+ utilizó un 6.9% en primer

plano y en segundo plano con un 3.5% demostrando ser un dispositivo que optimiza el uso del recurso (CPU), del total de dispositivos utilizados para la recolección de la información el 60% de los dispositivos cuentan con un consumo de CPU menor al 10% mientras que el 40% supera el 10% de uso de CPU.

Ilustración 15: Uso de la RAM al ejecutar la aplicación VipCar en los dispositivos móviles.



Realizado por: El Autor

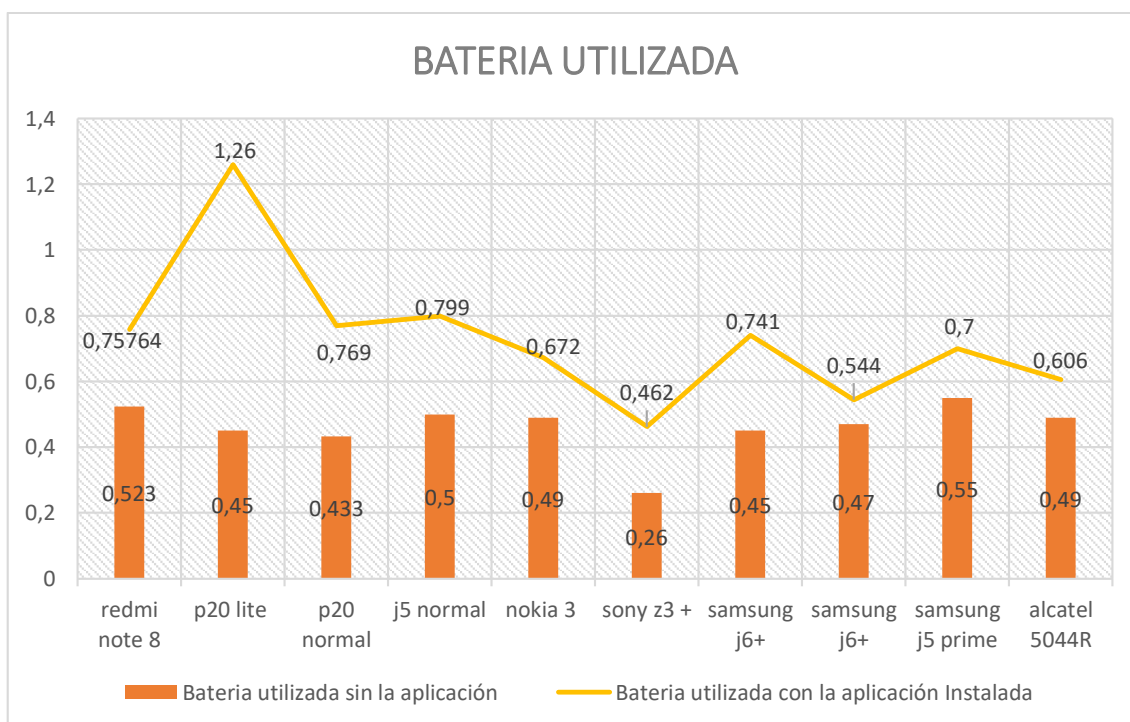
La Ilustración 4, presenta los datos recolectados del uso de la memoria RAM, se observan notables diferencias de consumo, los dispositivos móviles con más memoria RAM son los que consumen este recurso en mayor cantidad al momento de ejecutar la aplicación VipCar, se observa que el dispositivo Redmi Note 8 consume 131.9 Mb siendo el consumo más alto, mientras que el porcentaje más bajo en consumo es del Samsung j5 prime con un 50.2 Mb en primer plano; demostrando que la aplicación se adapta a los recursos de cada dispositivo móvil y utiliza solo los servicios necesarios en su ejecución, cuando el aplicativo se encuentra en segundo plano el dispositivo que menos memoria RAM consume es el Nokia 3 con un 33.3 Mb, el porcentaje más alto en consume es del Huawei p20 lite con un 123.4 Mb. El 70% de los dispositivos tienen un consumo entre 30 Mb y 75 Mb, mientras que el 30% de dispositivos cuenta con un consumo alto de RAM que va desde los 129 Mb hasta los 131.9 Mb.

Tabla 2: Datos estadísticos de la batería utilizada en los dispositivos móviles

Dispositivos móviles con el aplicativo VipCar instalado, uso de Batería.		Dispositivos móviles sin el aplicativo VipCar instalado, uso de Batería.	
Modelo del dispositivo	Datos estadísticos	Modelo del dispositivo	Datos estadísticos
Redmi note 8	0,75764	Redmi note 8	0,523
P20 lite	1,26	P20 lite	0,45
P20 normal	0,769	P20 normal	0,433
J5 normal	0,799	J5 normal	0,5
Nokia 3	0,672	Nokia 3	0,49
Sony z3 +	0,462	Sony z3 +	0,26
Samsung j6+	0,741	Samsung j6+	0,45
Samsung j6+	0,544	Samsung j6+	0,47
Samsung j5 prime	0,7	Samsung j5 prime	0,55
Alcatel 5044R	0,606	Alcatel 5044R	0,49

Realizado por: El Autor

Ilustración 16: Batería utilizada por los dispositivos



Realizado por: El Autor

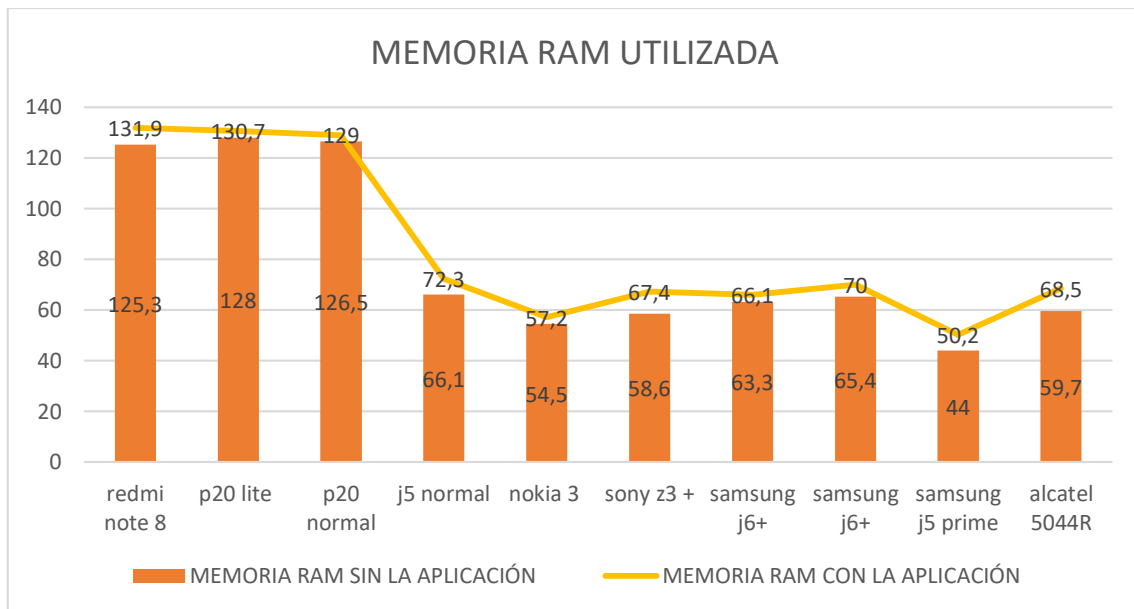
En la ilustración 5, presenta los datos recolectados del uso de la batería de los dispositivos móviles, en dos escenarios: 1) Dispositivos que tienen instalada la aplicación y 2) Dispositivos que no cuentan con la aplicación instalada, donde se observan una mínima diferencias en los dos casos, el consumo más pronunciado es del dispositivo móvil P20 lite cuando se encuentra instalada la aplicación con un consumo del 1.26% de la batería, mientras que el más bajo es el dispositivo Sony Z3+ con el 0.462% de consumo. Los dispositivos cuando no cuentan con la aplicación instalada tienen un consumo bajo, el porcentaje más alto que presenta es el dispositivo móvil Samsung J5 prime con un 0.55%, el porcentaje más bajo es el Sony Z3+ con un 0.26% de batería utilizada, demostrando que no aumenta el consumo de la batería en grandes proporciones al momento de tener ejecutando la aplicación VipCar Riobamba.

Tabla 3: Datos estadísticos de la memoria RAM utilizada en los dispositivos móviles

Dispositivos móviles con el aplicativo VipCar instalado, uso memoria RAM.		Dispositivos móviles sin el aplicativo VipCar instalado, uso memoria RAM.	
Modelo del dispositivo	Datos estadísticos	Modelo del dispositivo	Datos estadísticos
Redmi note 8	131,9	Redmi note 8	125,3
P20 lite	130,7	P20 lite	128
P20 normal	129	P20 normal	126,5
J5 normal	72,3	J5 normal	66,1
Nokia 3	57,2	Nokia 3	54,5
Sony z3 +	67,4	Sony z3 +	58,6
Samsung j6+	66,1	Samsung j6+	63,3
Samsung j6+	70	Samsung j6+	65,4
Samsung j5 prime	50,2	Samsung j5 prime	44
Alcatel 5044R	68,5	Alcatel 5044R	59,7

Realizado por: El Autor

Ilustración 17: Memoria RAM utilizada por los dispositivos móviles



Realizado por: El Autor

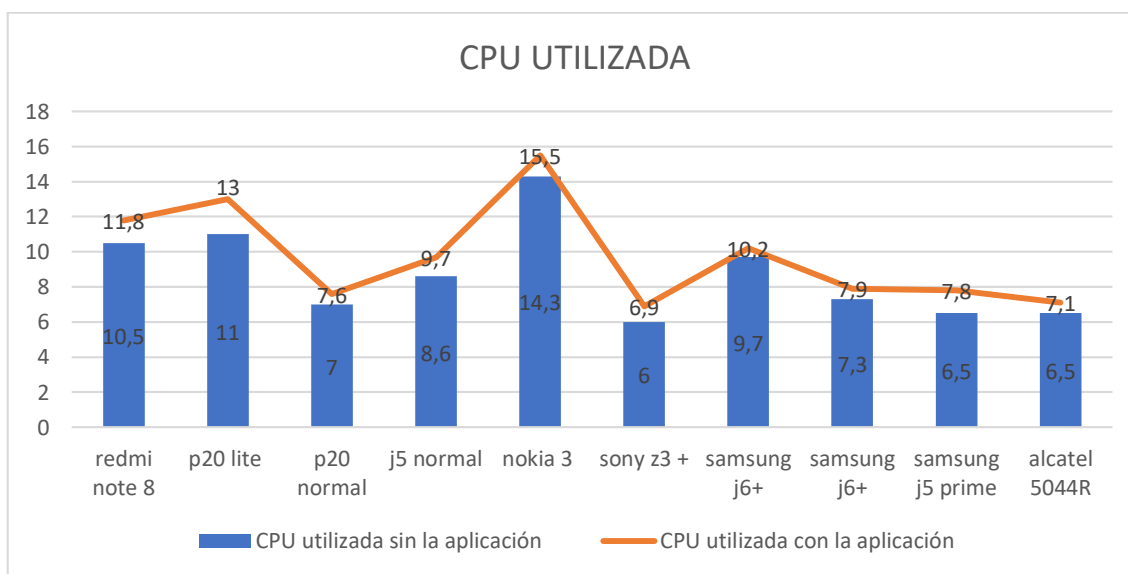
La ilustración 6, presenta los datos recolectados del uso de la memoria RAM en dos ámbitos diferentes: 1) Aplicación instalada en el dispositivo y 2) Dispositivo móvil sin la aplicación instalada, se observan mínimas diferencias de consumo, los dispositivos Redmi Note 8 consume 131.9 MB al ejecutar la aplicación y cuando el dispositivo móvil no cuenta con el aplicativo instalado consume 125.3 Mb notando una diferencia de 6.6 Mb de aumento siendo el segundo dispositivo con más consumo de memoria RAM. El dispositivo con el porcentaje más alto en consumo de memoria RAM al ejecutar el aplicativo es el Sony Z3+ con 67.4 MB en ejecución y al momento de no tener instalada la aplicación utiliza 58.6 Mb teniendo un incremento de 8.8 Mb al ejecutar el aplicativo móvil VipCar Riobamba.

Tabla 4: Datos estadísticos del CPU utilizado en los dispositivos móviles

Dispositivos móviles con el aplicativo VipCar instalado, uso CPU.		Dispositivos móviles sin el aplicativo VipCar instalado, uso CPU.	
Modelo del dispositivo	Datos estadísticos	Modelo del dispositivo	Datos estadísticos
Redmi note 8	11,8	Redmi note 8	10,5
P20 lite	13	P20 lite	11
P20 normal	7,6	P20 normal	7
J5 normal	9,7	J5 normal	8,6
Nokia 3	15,5	Nokia 3	14,3
Sony z3 +	6,9	Sony z3 +	6
Samsung j6+	10,2	Samsung j6+	9,7
Samsung j6+	7,9	Samsung j6+	7,3
Samsung j5 prime	7,8	Samsung j5 prime	6,5
Alcatel 5044R	7,1	Alcatel 5044R	6,5

Realizado por: El Autor

Ilustración 18: Uso del CPU en los dispositivos móviles



Realizado por: El Autor

La ilustración 7 muestra los datos obtenidos del uso del CPU de cada dispositivo móvil, se realiza la prueba en dos escenarios: 1) Aplicativo instalado en el dispositivo y 2) Dispositivo sin el aplicativo instalado, el uso del CPU no se modifica gravemente tiene un consumo bajo debido a los streams de datos asíncronos que permite desactivar funciones del aplicativo cuando no se encuentran en uso, donde la mayoría de dispositivos móviles no supera el 1.1% cuando se encuentra en ejecución la aplicación VipCar. Los dispositivos que superan en consumo de CPU con un mínimo porcentaje son Redmi note 8 con 1.3% y el segundo dispositivo es el P20 lite con el 2% al tener en ejecución el aplicativo móvil.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Los streams de datos asíncronos aumenta el throughput soportado, permitiendo realizar más tareas y minimizar recursos, los streams de datos asíncronos también permite ejecutar varios procesos al mismo tiempo, esto ayudó a que la aplicación no pese mucho y al instalar en los dispositivos móviles utiliza los servicios de forma independiente, permitiendo ser una aplicación escalable, la herramienta utilizada para implementar streams de datos asincrónicos fue visual Studio Code herramienta que permite de manera fácil y efectiva integrar una programación híbrida, disminuyendo los recursos que se utilizan, y se pueda consumir en tres diferentes plataformas como son: Android, IOS y como plataformas WEB.
- El desarrollo del aplicativo VipCar Riobamba, está permitiendo a la empresa manipular los pedidos y llevar un control adecuado de trabajadores y clientes, permite a los clientes realizar los pedidos de forma fácil y efectiva, el cliente recibirá un mensaje de confirmación cuando el pedido sea aceptado donde incluye la hora promedio que llegará el pedido y el nombre del repartidor.
- La aplicación VipCar Riobamba está desarrollada en angular versión 8.1.3, en la cual se vinculó los streams de datos asincrónicos permitiendo que la aplicación active y desactive los flujos de datos, mejorando su funcionalidad en dispositivos de gama baja, media y alta, proporcionando reducción en los recursos al momento de ejecutar la aplicación en primer plano, la aplicación no supero más del 1.3%, en el consumo de batería, 15.5% en el uso del CPU y en la memoria RAM utilizo 131.9 Mb, mientras que en segundo plano la aplicación solo consumía un máximo de 1.07% en batería, el CPU un 9.9% y la memoria RAM llego hasta el 123.4 Mb, esto se adapta de acuerdo al teléfono y sus características, visto que en dispositivos de gama media y alta los servicios consumen más recursos, utilizan más memoria RAM pero debido a los datos asíncronos permite que los recursos de la aplicación que no se utilicen se cierren de manera automática permitiendo que funcione de manera más ágil y sin consumir recursos en dispositivos de gama baja, permitiendo que el desempeño sea óptimo para dispositivos Android que vayan desde 1 Gb en RAM.

5.2. RECOMENDACIONES

- Utilizar los Streams de datos asíncronos en las aplicaciones móviles brinda una mayor escalabilidad al momento de implementar nuevos procesos u actualizaciones necesarias, es necesario tener conocimientos de la forma de ejecución de los procesos para no tener problemas al momento de la implementación.
- Visual Studio code permite de forma fácil e intuitiva la programación con Streams de datos asíncronos, al momento de programar se debe instalar y actualizar todos los códec necesarios para no tener problemas al momento de ejecutar el aplicativo.
- Realizar soporte y actualizaciones a la aplicación VipCar Riobamba, para que su funcionamiento se encuentre en óptimas condiciones, lo cual ayudara a que el cliente se encuentre satisfecho al momento de utilizarla.

BIBLIOGRAFÍA

- D'Souza, D., & Wills, A. (1998). *Objects, Components, and Frameworks with UML*. Massachusetts : Addison Wesley Longman, Inc.
- Android, S. (01 de 05 de 2020). *Developers*. Obtenido de https://developer.android.com/studio/intro/?gclid=Cj0KCQjwgo_5BRDuARIsA DDEntR3OfDQIRfNBMIQofnc12uPn-jEj2cpYP0ZJI_FXxMmRbNs2eX3jQgaAmuWEALw_wcB&gclsrc=aw.ds
- Benbourahla, N. (2015). *Android 5: Principios del desarrollo de aplicaciones Java*. Barcelona: Eni.
- Cardador, A. (2015). *Dimensionar, instalar y optimizar el hardware. IFCT0510*. España : IC Editorial.
- Castañeda et al. (2010). *Procesamiento de datos y análisis estadísticos utilizando SPSS*. Porto Alegre: ediPUCRS.
- Díaz, Y., & Fernández, Y. (2012). Patrón Modelo-Vista-Controlador. *Telemática*, 47-57. Obtenido de <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>
- Fennema, M., Herrera, S., palavecino, R., Budán, P., Rosenzvaig, F., Najar, P., . . . Saavedra, E. (2017). Aproximaciones para el Desarrollo Multiplataforma y Mantenimiento de Aplicaciones. *XIX Workshop de Investigadores en Ciencias de la Computación* (págs. 446-450). Buenos Aires: Sedici.
- Foundation, O. (10 de 01 de 2020). *node js*. Obtenido de <https://nodejs.org/es/about/>
- Google. (2016). *Firestore expands to become unified app platform*. Obtenido de <https://firebase.googleblog.com/2016/05/firebase-expands-to-become-unified-app-platform.html>
- Hereter, L., & Zanini, . (2016). *jQuery Mobile: Diseño y desarrollo de aplicaciones para smartphones y tablets*. (C. A. Corp, Ed.) Argentina: RedUsers.
- Luna, F. (2016). *Desarrollo web para dispositivos móviles: Herramientas para diseñar y programar WebApps*. Argentina: RedUsers.
- Marchioni, F. (2018). *Practical Java EE Development on WildFly: Quickstart guide for developing, deploying and securing Java EE applications on WildFly application server*. sicilia: ITBuzzPress.
- Microsoft . (2019). *Microsoft* . Obtenido de <https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

- Microsoft. (2016). *msdn.microsoft.com*. Obtenido de Microsoft: <https://msdn.microsoft.com/es-es/library/mt299001.aspx>
- Microsoft Azure. (2019). *Microsoft Azure*. Obtenido de <https://azure.microsoft.com/es-es/overview/what-is-azure/>
- Mistry, R., & Misner, S. (2012). *Introducing Microsoft® SQL Server 2012*. Microsoft Press.
- Orozco, G. (2018). *Iniciación a Android en Kotlin. Casos prácticos*. España: Ediciones Paraninfo, S.A.
- Pelechano, V. (08 de 01 de 2020). *researchgate.net*. Obtenido de Servicios Web. Estándares, extensiones y perspectivas de futuro. : https://www.researchgate.net/publication/228634068_Servicios_Web_Estandares_extensions_y_perspectivas_de_futuro
- PEÑA, S. D. (2017). *UF1469 - SGBD e instalación*. Madrid: Paraninfo, S.A.
- Pérez, Á. (5 de 10 de 2019). Empresa VipCar Riobamba. (B. Velástegui, Entrevistador)
- Santamaría, J., & Hernández, J. (2017). *SQL SERVER VS MySQL*. 1. Obtenido de <https://iessanvicente.com/colaboraciones/sqlserver.pdf>
- Serna, S., & Pardo, C. (2016). *Diseño de interfaces en aplicaciones móviles* (1 ed.). España: RA-MA.
- Sharma, R. (2018). *Hands-On Reactive Programming with Reactor: Build reactive and scalable microservices using the Reactor framework*. Birmingham: Packt Publishing Ltd.
- Sikora, M. (2017). *PHP Reactive Programming*. Birmingham: Packt Publishing Ltd.
- Siuhi, S., & Mwakalonge, J. (2016). Opportunities and challenges of smart mobile applications in transportation. *ScienceDirect*, 582-592.
- Vasconcelos, H. (2016). *Programación asincrónica de Android Segunda edición*. Birmingham: Packt Publishing.

ANEXOS

ANEXO I:

Diagrama de la Base de Datos

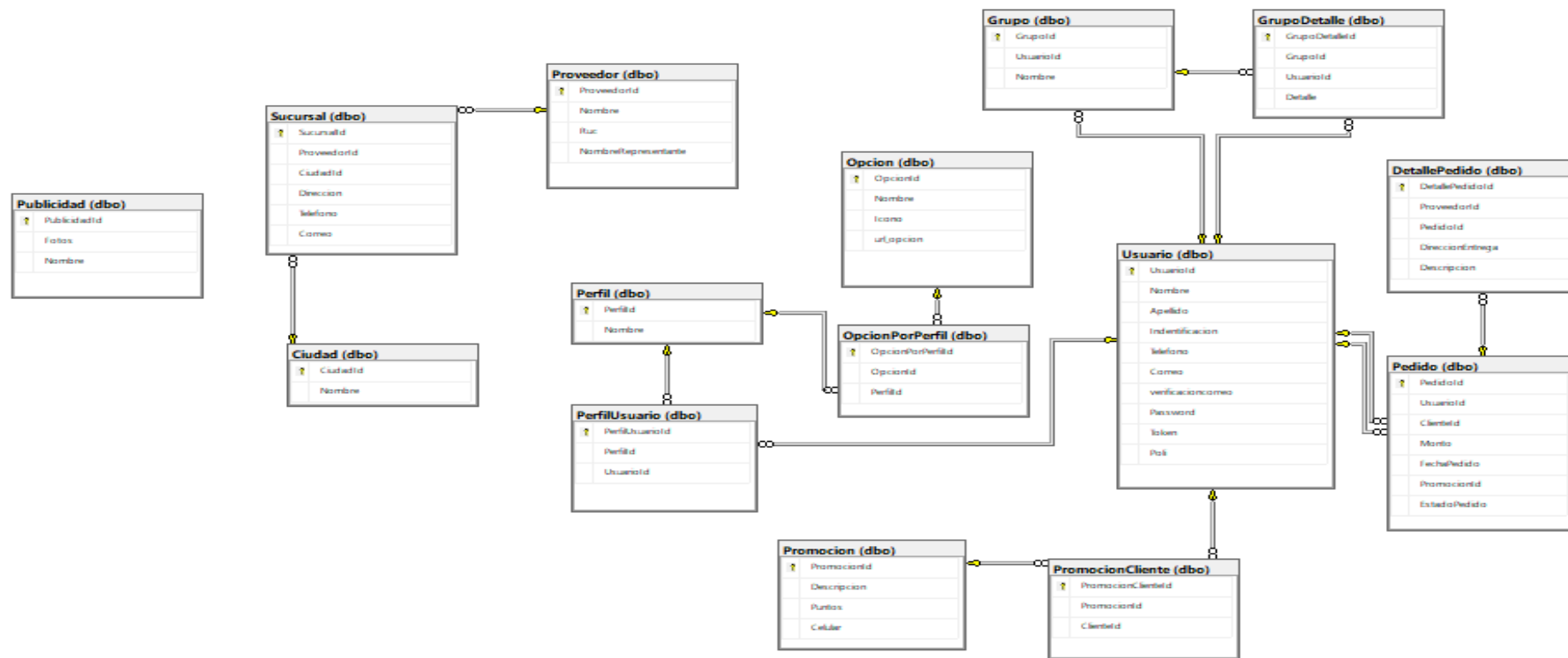


Ilustración 19: Diagrama de la Base de Datos

Realizado por: El Autor

ANEXO II:

Tabla 5: Tablas de datos recolectados en los dispositivos móviles

PRIMER PLANO			SEGUNDO PLANO		
REDMI NOTE 8					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.64	123	8	0.58	116	5
0.66	121	11	0.63	115	6
0.6	122	11	0.59	117	6
0.72	121	7	0.58	115	5
0.7	121	9	0.61	117	6
0.79	135	13	0.7	125	7
1.05	145	15	0.92	135	9
0.96	143	14	0.87	132	8
1.15	146	16	0.95	125	9
0.94	142	14	0.82	115	8

PRIMER PLANO			SEGUNDO PLANO		
P20 LITE					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.60	131	6	0.4	124	4
0.9	133	10	0.85	126	8
1.15	130	10	1.05	123	10
1.1	137	13	1.02	127	10
1.16	136	12	1.07	126	10
1.6	129	14	1.27	122	11
1.53	131	14	1.23	124	11
1.56	126	15	1.24	124	11

1.6	124	17	1.27	119	12
1.4	130	19	1.3	123	12

PRIMER PLANO			SEGUNDO PLANO		
P20 NORMAL					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.58	150	5	0.3	110	3
0.63	120	8	0.5	112	5
0.93	131	8	0.7	115	7
0.58	120	6	0.4	110	4
0.61	117	7	0.5	112	5
0.8	128	8	0.7	115	7
0.92	151	9	0.6	114	6
0.87	132	8	0.3	110	3
0.95	125	9	0.4	110	4
0.82	115	8	0.4	110	4

PRIMER PLANO			SEGUNDO PLANO		
J5 NORMAL					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.61	70	9	0.3	40	6
0.66	75	7	0.4	43	4
0.71	70	10	0.3	40	6
0.86	73	10	0.6	41	6
0.99	71	9	0.7	40	6
0.85	73	12	0.6	43	8
0.86	74	10	0.6	43	6

0.61	70	8	0.3	40	5
0.86	73	9	0.6	41	7
0.98	74	13	0.7	41	7

PRIMER PLANO			SEGUNDO PLANO		
NOKIA 3					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.6	50	9	0.4	33	5
0.71	55	12	0.5	37	6
0.62	49	15	0.4	31	4
0.62	53	15	0.4	34	5
0.69	53	13	0.6	34	5
0.66	55	17	0.5	34	5
0.6	69	21	0.4	29	3
0.65	60	17	0.5	31	4
0.77	58	17	0.7	32	5
0.8	70	19	0.4	38	6

PRIMER PLANO			SEGUNDO PLANO		
SONY Z3 +					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.4	62	5	0.2	40	3
0.61	66	6	0.4	45	4
0.49	60	6	0.4	41	3
0.65	63	5	0.2	42	3
0.44	66	6	0.3	41	3
0.35	66	7	0.2	46	4
0.44	78	9	0.4	44	4
0.44	71	8	0.4	40	3
0.6	77	9	0.3	49	5
0.5	65	8	0.2	39	3

PRIMER PLANO			SEGUNDO PLANO		
SAMSUNG J6 +					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.68	63	8	0.4	50	5
0.85	66	9	0.5	55	6
1.05	61	8	0.5	52	5
0.56	66	8	0.4	54	5
0.57	62	10	0.4	57	6
0.88	67	9	0.6	54	5
0.82	68	12	0.6	63	7
0.6	69	13	0.3	64	7
0.7	69	10	0.4	62	6
0.7	70	15	0.4	62	6

PRIMER PLANO			SEGUNDO PLANO		
SAMSUNG J6 +					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.51	66	7	0.4	50	5
0.63	71	9	0.5	55	6
0.59	68	7	0.5	52	5
0.44	68	7	0.4	54	5
0.46	69	8	0.4	57	6
0.71	63	6	0.6	54	5
0.69	74	9	0.6	53	7
0.4	70	9	0.3	54	7
0.5	78	8	0.4	62	6
0.51	73	9	0.4	62	6

PRIMER PLANO			SEGUNDO PLANO		
SAMSUNG J5 PRIME					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.7	48	8	0.4	36	5
0.71	50	8	0.4	37	5
0.7	46	6	0.4	33	4
0.65	46	6	0.3	33	4
0.75	41	5	0.5	28	3
0.75	55	8	0.5	38	5
0.73	52	9	0.5	36	5
0.76	55	11	0.6	40	6
0.75	53	9	0.5	39	5
0.5	56	8	0.3	36	5

PRIMER PLANO			SEGUNDO PLANO		
ALCATEL 5044R					
BATERIA	MEMORIA RAM	CPU	BATERIA	MEMORIA RAM	CPU
0.6	73	7	0.4	65	6
0.6	81	7	0.4	70	7
0.52	55	6	0.4	50	4
0.7	56	6	0.5	50	4
0.53	68	8	0.4	55	5
0.69	79	8	0.5	73	7
0.5	71	7	0.4	70	7
0.71	67	8	0.6	66	6
-0.71	80	9	0.6	63	6
0.5	55	5	0.4	49	4

ANEXO III:

VISTAS DE LA APLICACIÓN MÓVIL

La aplicación móvil VipCar Riobamba cuenta con tres distintas sesiones para sus usuarios que son: Administrador, repartidor y cliente; para ingresar un nuevo administrador y repartidor es necesario que un usuario con credenciales de administrador registre desde la plataforma de nuevos administradores o repartidores, mientras que para ser un nuevo cliente, al ingresar a la aplicación debe registrarse e ingresar los datos correspondientes, finalmente para ingresar a las interfaces proporcionadas ingresar el correo electrónico y contraseña.

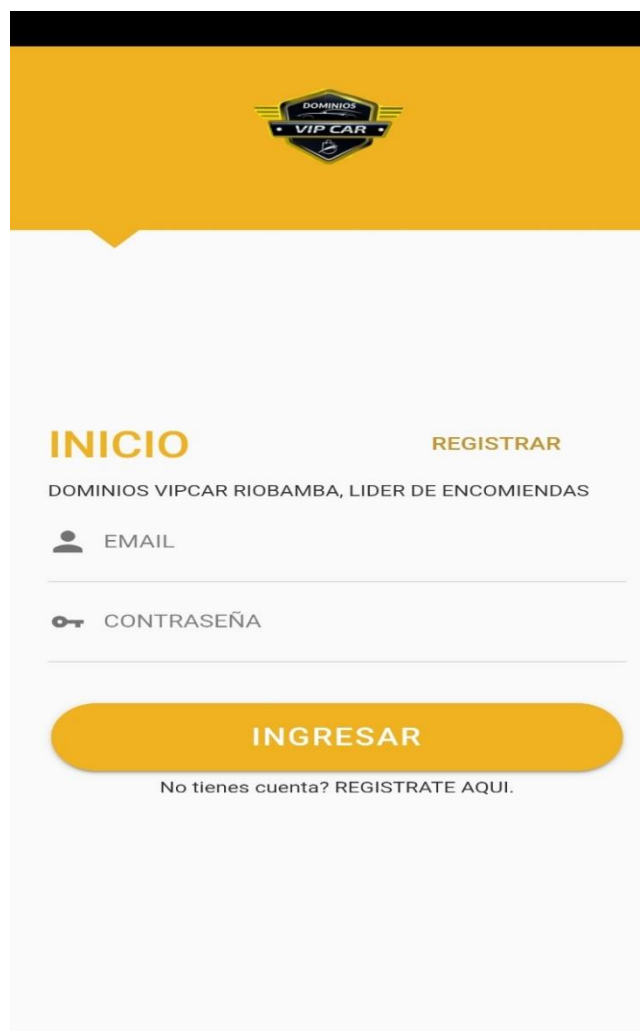


Ilustración 20: Login y registro en la aplicación

Realizado por: El Autor

En la ilustración 6 se aprecia la pestaña de login donde se puede realizar el ingreso a la cuenta de usuario o realizar un nuevo registro para ser un nuevo cliente de la empresa.

Interfaz de bienvenida para los usuarios.



Ilustración 21: Pagina de bienvenida a los usuarios

Realizado por: El Autor

En la ilustración 7 se observa la pestaña de bienvenida al momento de ingresar a la sesión, la cual es una pestaña por default, esta interfaz se muestra a todos los usuarios: administradores, taxistas y clientes.

Vista del Administrador en el aplicativo móvil

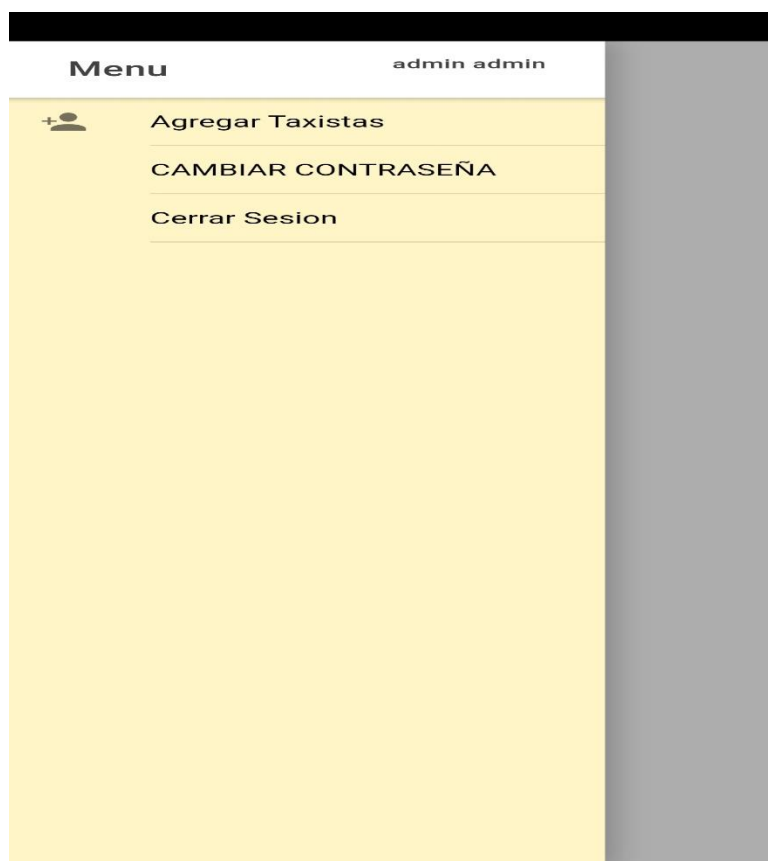


Ilustración 22: Menú del Administrador

Realizado por: El Autor

En la ilustración 8 se puede apreciar el menú dinámico desde donde el administrador podrá internarse a la pestaña “Agregar Taxistas” para incorporar nuevos repartidores o taxistas, cuenta con opciones de cambiar su contraseña o cerrar la sesión.

Vista de la página de registro para un nuevo repartidor.

Nombre

Apellido

Celular

Cedula

Contraseña

Confirmar Contraseña

Correo electronico

Confirmar correo eletronico

Poli N^a


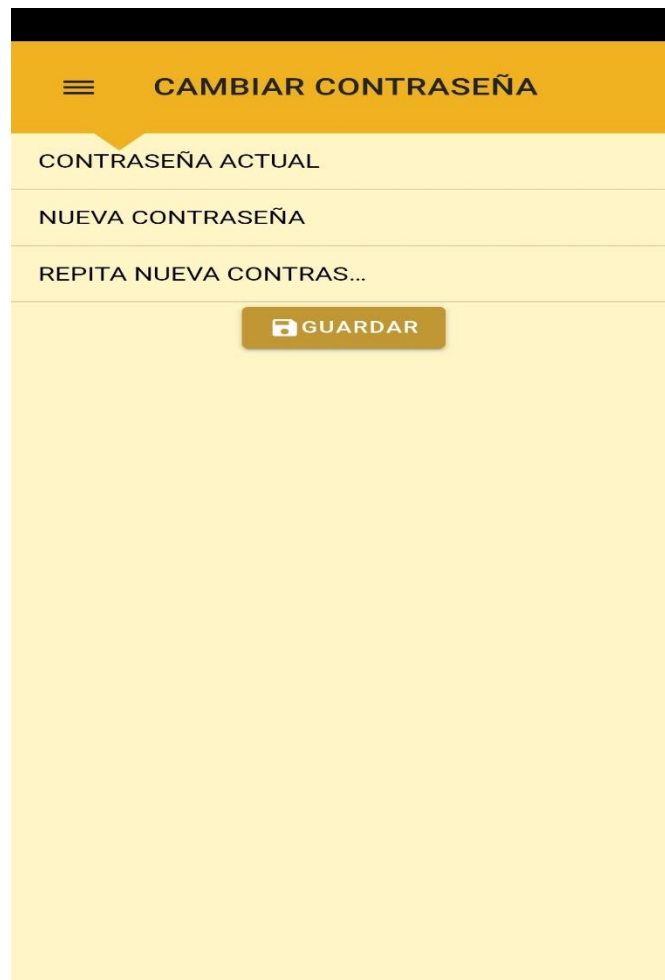
 GUARDAR

Ilustración 23: Página de registro de un nuevo repartidor

Realizado por: El Autor

En la ilustración 9 se observa la pestaña “Registrar Taxista” en la sesión del administrador, donde se ingresan los datos de los nuevos repartidores que van a trabajar en la empresa, esta opción solo se encuentra activa para los usuarios administradores permitiendo dar seguridad y control de nuevos ingresos de trabajadores para la empresa.

Vista de la página para el cambio de contraseña



☰ CAMBIAR CONTRASEÑA

CONTRASEÑA ACTUAL

NUEVA CONTRASEÑA

REPITA NUEVA CONTRAS...

GUARDAR

Ilustración 24: Página de Cambio de contraseña administrador

Realizado por: El Autor

La ilustración 10 se aprecia la pestaña de “Cambio de Contraseña” en la sesión del administrador, en la cual se puede realizar el cambio de la contraseña del usuario activo.

Vista del menú de los repartidores.

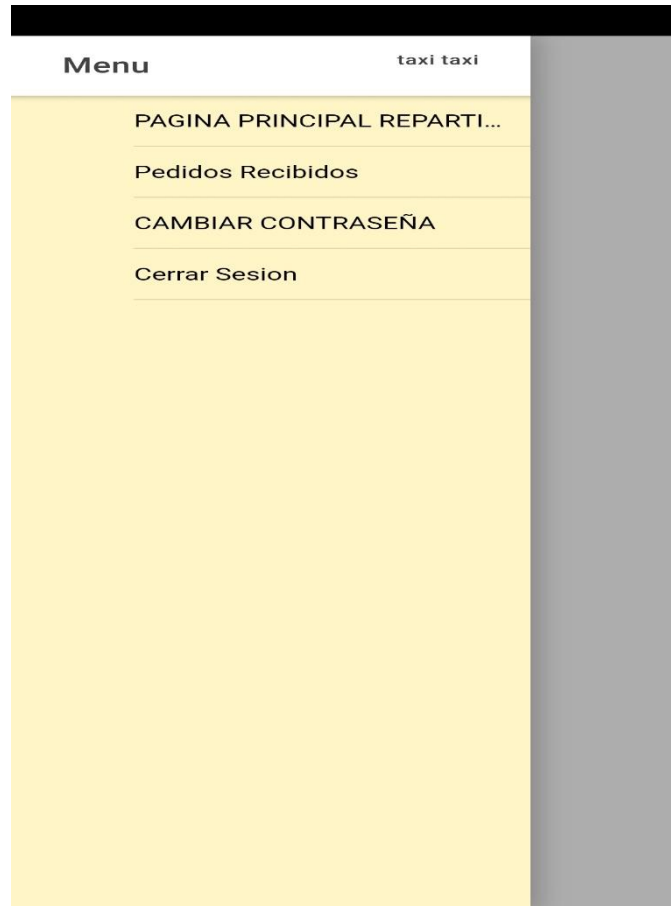


Ilustración 25: Menú dinámico del Repartidor

Realizado por: El Autor

En la ilustración 11 se aprecia las opciones del menú dinámico para las sesiones de los repartidores, se observan las opciones de: pedidos recibidos, página principal, cambiar contraseña y cerrar sesión.

Vista de los pedidos aceptados y que se encuentran en proceso por los repartidores

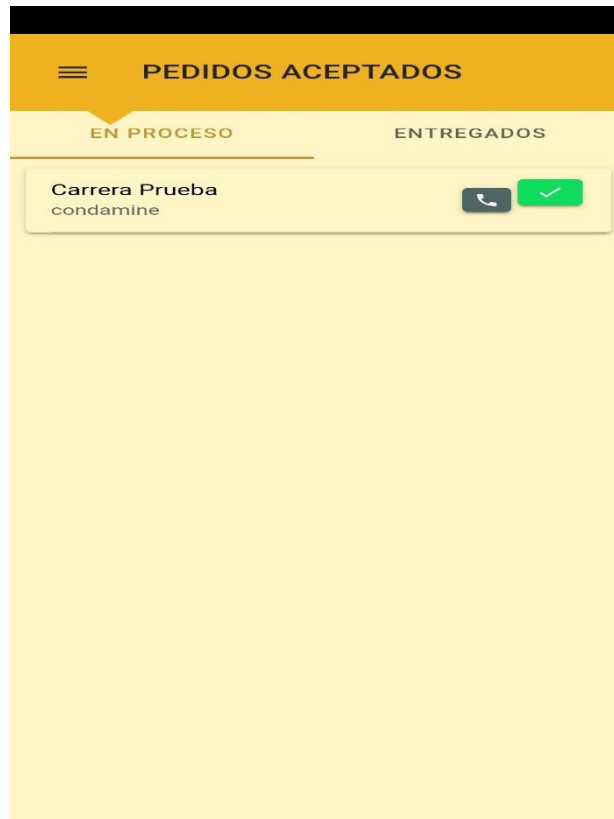


Ilustración 26: Pagina de pedidos en proceso.

Realizado por: El Autor

En la ilustración 12 se contempla los pedidos que son aprobados por los repartidores que se encuentran en proceso de expendio, en esta pestaña se cuenta con la alternativa de realizar llamadas al cliente para informarle que su pedido se encuentra en la dirección de entrega y el botón de visto para la confirmación de entrega.

Vista de la página donde se reciben los pedidos solicitados.

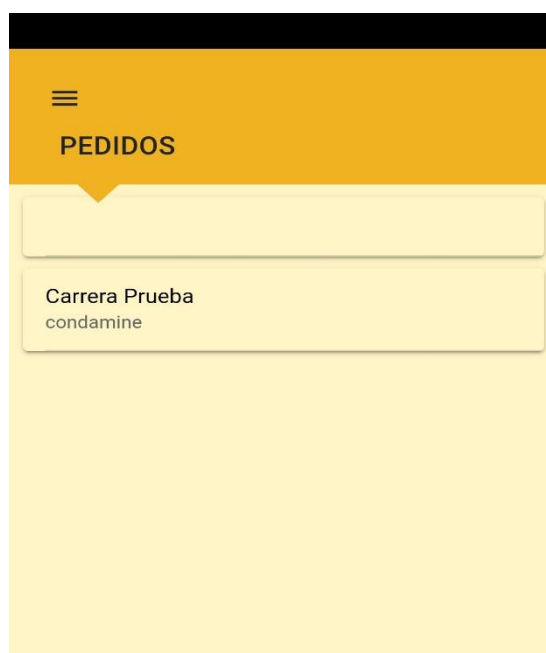


Ilustración 27: Pagina de pedidos recibidos.

Realizado por: El Autor

En la ilustración 13 se aprecia las peticiones recibidas, las cuales se encuentran disponibles para que los repartidores de turno acepten, cuando el pedido es aceptado por un repartidor automáticamente se elimina el pedido de esta pestaña y es asignado al repartidor que aprobó el pedido.

Vista de los detalles del pedido realizado.

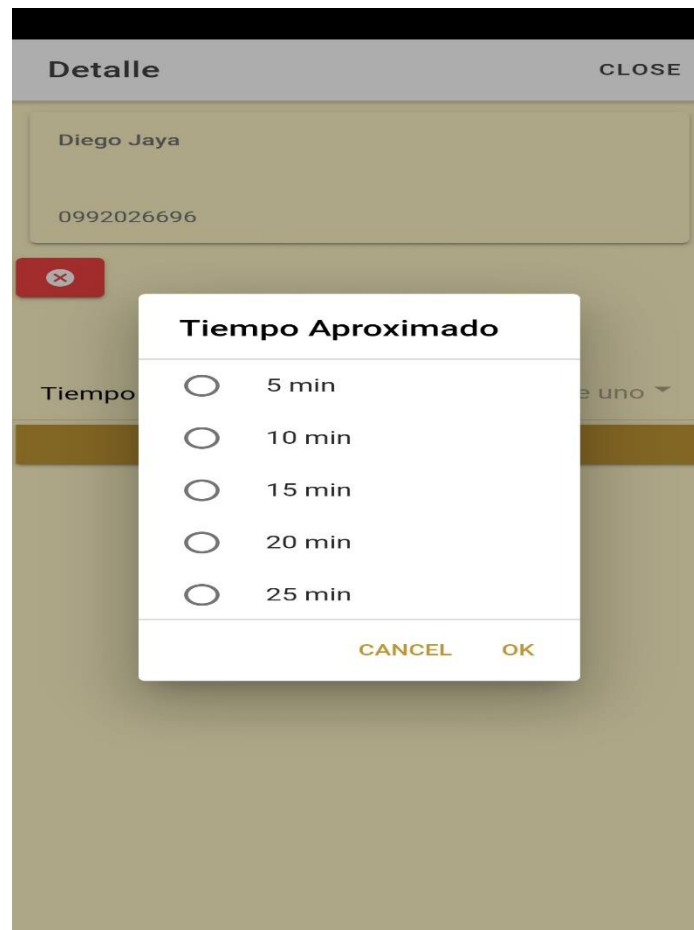


Ilustración 28: Pagina del detalle de los pedidos solicitados.

Realizado por: El Autor

La ilustración 14 muestra los detalles de los pedidos que han solicitado los clientes, donde se observa el nombre, dirección y el número de teléfono del cliente, desde esta interfaz el repartidor puede aprobar o rechazar el pedido, cuando el repartidor acepta el pedido tiene que situar el tiempo aproximado de llegada a la ubicación del cliente, al mismo instante de aceptar se remite una notificación al cliente sobre su pedido que llegara en un tiempo determinado y el número de identificador del repartidor que está realizando el pedido.

Vista de la Alerta de confirmación del pedido.

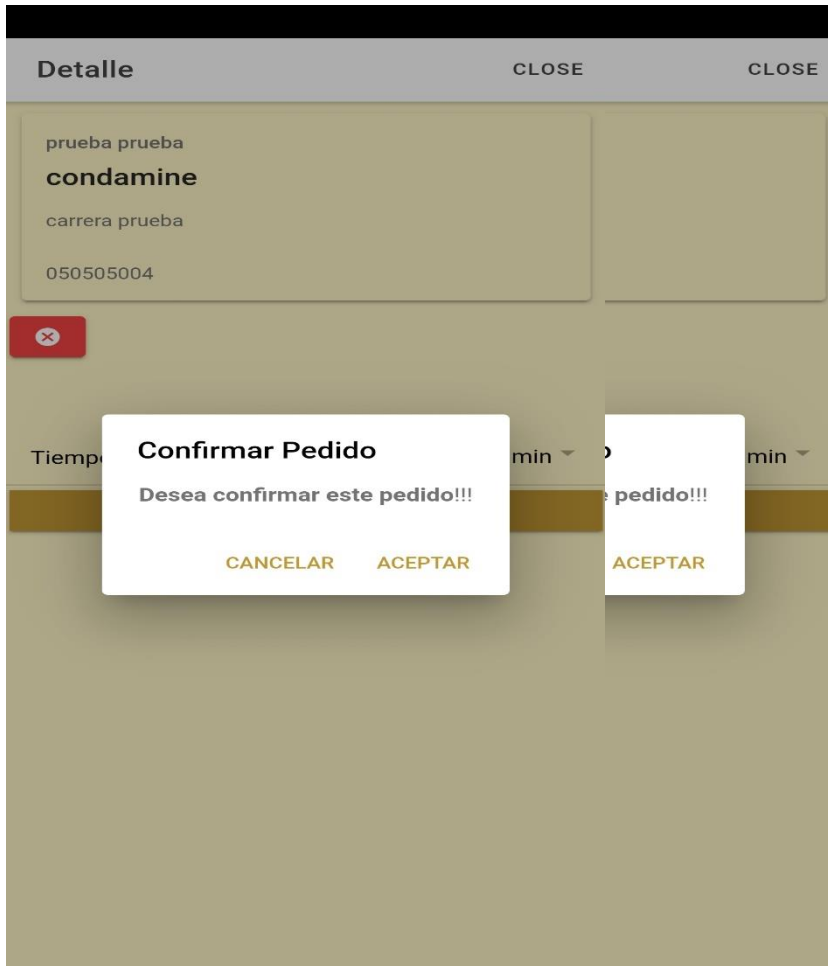


Ilustración 29: Alerta de confirmación del pedido.

Realizado por: El Autor

En la ilustración 15 se aprecia una alerta de confirmación, donde se procederá a reafirmar el pedido o cancelar, esto permite la seguridad de no aceptar pedidos por error.

Vista de la página con los pedidos entregados de un repartidor determinado

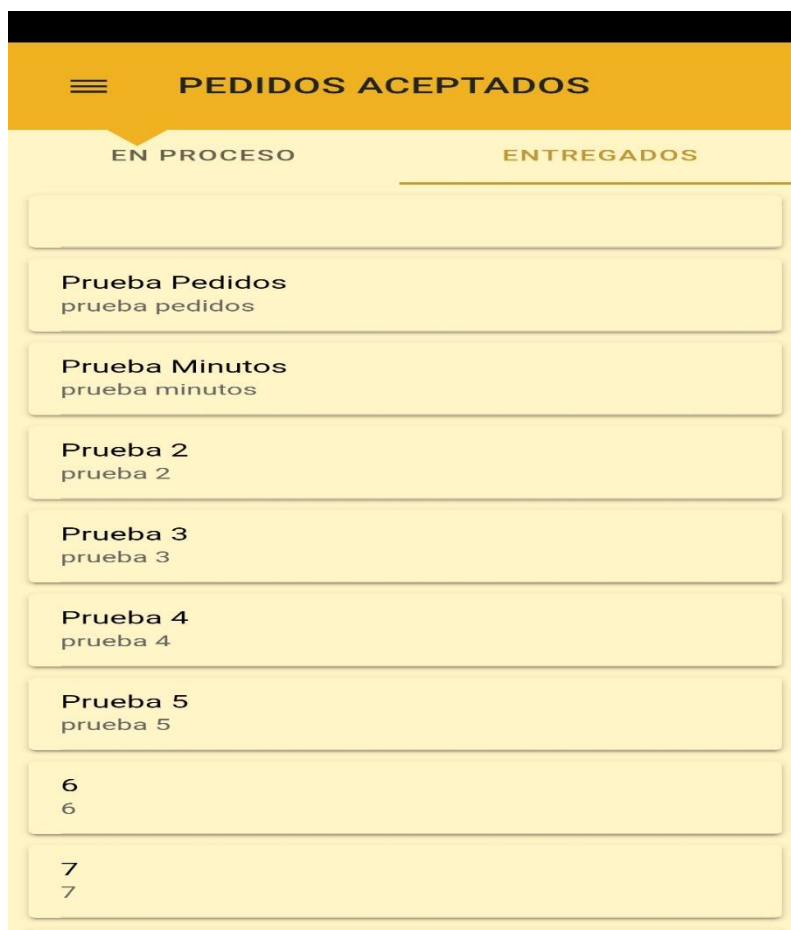


Ilustración 30: Pagina de pedidos entregados.

Realizado por: El Autor

La ilustración 16 muestra todos los pedidos que fueron aceptados por un repartidor y ya se realizó su respectiva entrega al cliente.

Vista del menú de los clientes.



Ilustración 31: Menú de los clientes.

Realizado por: El Autor

La ilustración 17 muestra el menú dinámico en la interfaz de los clientes que cuenta con tres opciones como: interfaz principal de pedidos, cambiar contraseña y cerrar la sesión.

Vista de la página principal de los clientes.



Ilustración 32: Pagina para describir el pedido.

Realizado por: El Autor

En la ilustración 18 se observa que el cliente puede describir su pedido y la dirección de entrega, al momento de presionar el botón enviar se desplegara una alerta de seguridad para reforzar la seguridad de pedidos solicitados.

ANEXO IV:

Base de datos y servicios web en Azure.

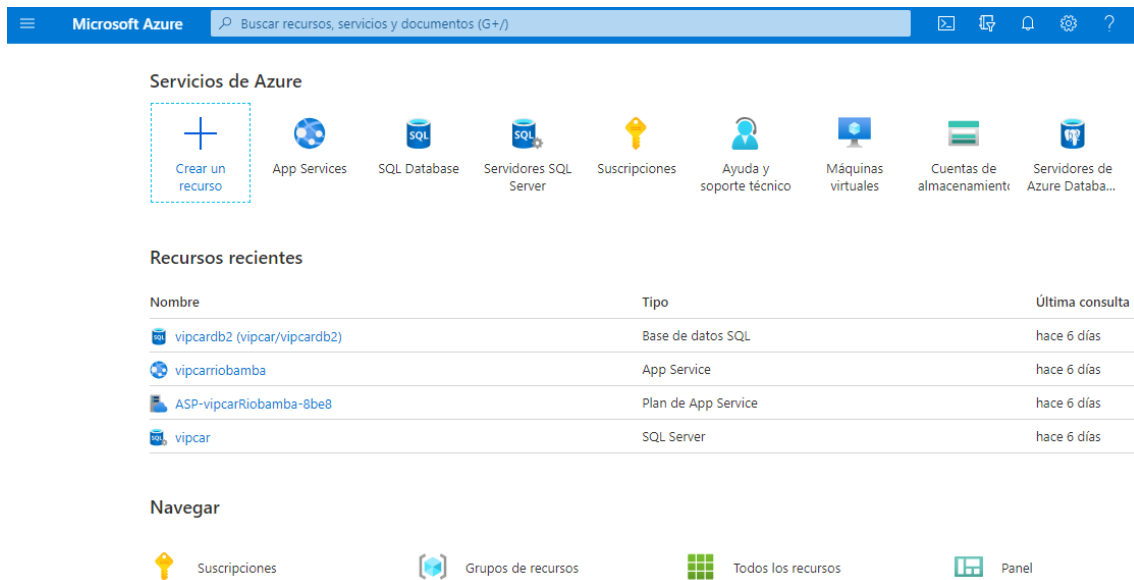


Ilustración 33: Plataforma Azure

Realizado por: El Autor

En la ilustración 19 se aprecia la plataforma de AZURE en donde se encuentra desplegada la base de datos y los servicios web de la aplicación para que se mantenga en conexión a toda hora.