

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA GESTION
DE PLATAFORMAS AUTOCARGABLES (WINCHAS)**

APLICACIÓN MÓVIL

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

EDMUNDO GABRIEL MEDINA INTRIAGO

DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO

DMQ, marzo 2023

CERTIFICACIONES

Yo, EDMUNDO GABRIEL MEDINA INTRIAGO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

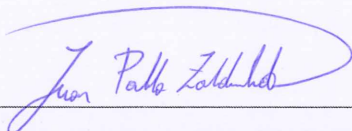


EDMUNDO GABRIEL MEDINA INTRIAGO

gabriel.medina@epn.edu.ec

gabicho.edu@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por EDMUNDO GABRIEL MEDINA INTRIAGO, bajo mi supervisión.



ING. JUAN PABLO ZALDUMBIDE PROAÑO

DIRECTOR

juan.zaldumbide@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



EDMUNDO GABRIEL MEDINA INATRIAGO

DEDICATORIA

A Dios, por darme la oportunidad de superarme y culminar mi carrera, a mis familiares y amigos por brindarme su apoyo incondicional, a mis profesores quienes a través de sus conocimientos transmitieron en mis sus saberes.

A mis hijos Emily y Neithan, quienes han sido mi mayor motivación para nunca rendirme en los estudios y poder ser un gran ejemplo para ellos.

Edmundo Gabriel Medina Intriago

AGRADECIMIENTO

Agradezco a Dios, a mi familia, y en especial a mis profesores quienes con sus conocimientos me guiaron por el camino del saber, a mi tutor ING. Juan Pablo Zaldumbide, gracias por su apoyo incondicional para poder concluir este proyecto y una etapa más en mi carrera profesional.

Edmundo Gabriel Medina Intriago

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
2 METODOLOGÍA	6
Aspectos esenciales de RAD.....	6
2.1 Metodología de Desarrollo	7
Fases de la metodología RAD	8
Planeación de los requerimientos	8
Diseño	8
Construcción o desarrollo	8
Implementación	9
Roles (<i>stakeholders</i>).....	9
Asignación de roles	10
Aplicación de la metodología	11
Planeación de los Requerimientos.....	11
2.2 Diseño de interfaces (<i>mockups</i>).....	12
Diseño de interfaces de la aplicación móvil.....	12
2.3 Diseño de la arquitectura	13
Arquitectura de Datos	13
Patrón arquitectónico	14
Patrón arquitectónico MVC (Modelo, Vista, Controlador)	14
Aplicación Móvil.....	15
2.4 Herramientas de desarrollo	16

Aplicación Móvil	16
Librerías	17
3 RESULTADOS.....	19
Iteración 0. Configuración del ambiente de desarrollo	19
Creación de la base de datos en firebase	19
Instalación de programas necesarios.....	20
Iteración 1	21
Iteración 2.....	22
Iteración 3:.....	25
Iteración 4: Pruebas y despliegue	28
4 Conclusiones	32
5 Recomendaciones	33
6 REFERENCIAS BIBLIOGRÁFICAS.....	34
7 ANEXOS	36
ANEXO I.....	37
ANEXO II.....	38
ANEXO III.....	60
ANEXO IV	61

RESUMEN

En la actualidad, las personas se movilizan grandes o pequeñas distancias utilizando medios de transporte muy importantes y exclusivos para la comunidad, ya sea este medio propio o de uso público. Los negocios que operan en este entorno han venido cambiando radicalmente con el pasar de los años, y día a día aumenta la demanda por adquirir un vehículo impulsado por una máquina.

Con el aumento de vehículos en la ciudad de Quito, se ha generado molestias en las vías de la Capital, muchas de estas molestias no solo se deben al congestionamiento, sino que también por fallas o desperfectos mecánicos en los automotores que circulan a diario, los mismos que son llevados a mecánicas por sus dueños en plataformas auto cargables (winchas). Este servicio se lo realiza a través de llamadas telefónicas, referenciadas a través de algún conocido.

Este trabajo plantea implementar una aplicación móvil capaz de gestionar los pedidos y traslados del automotor del lugar en donde sufrió el desperfecto, a una mecánica donde será reparado. La metodología por implementar en este proyecto es RAD (*Rapid Application Development*), la cual está comprendida por un desarrollo iterativo. Esta metodología comprende un proceso del desarrollo del software corto, todo esto al comprender bien los requerimientos.

Posterior a ello se detalla las interfaces, patrón arquitectónico y herramientas utilizadas para el desarrollo de la aplicación propuesta. Además del uso de una base de datos NoSQL.

Para finalizar, se mencionan las conclusiones y recomendaciones que se han obtenido a lo largo de este trabajo.

PALABRAS CLAVE: RAD, ionic-react, Firebase, plataformas auto cargables (winchas).

ABSTRACT

At the present, people move great or small distances through, some means of transport which has great and extraordinary relevance in society, whether it is owner's or public transportation. Businesses that operate in this environment have been changing radically over the years, and the demand to acquire a vehicle powered by a machine increases every day.

With vehicles increase in the city of Quito, a lot of inconvenience has been generated on the ways Capital, many of these inconveniences are not only due to congestion, but also due to failures or mechanical damage in the vehicles that circulate daily. The same ones that are taken to mechanics by their owners in self-loading platforms (winches). This service has been carried out through means such as telephone calls, referenced through an acquaintance.

In the present work, it is proposed to implement a mobile application capable of managing the orders and transfers of the vehicle from the place where it suffered the damage, to a mechanic where it will be repaired. The methodology to be implemented in this project is RAD (Rapid Application Development), which is comprised of an iterative development, in which it comprises an extremely short software development process, all this by understanding the requirements well.

After that, the interfaces, architectural pattern and tools used for the development of the proposed application are detailed. In addition to the use of a non-relational database NoSql Firebase.

Finally, the last part of this report mentions the conclusions and recommendations that have been obtained throughout this work.

KEYWORDS: RAD, ionic-react, Firebase, plataformas auto cargables (winchas).

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En los últimos años Quito ha tenido un aumento gradual demográfico y geográfico, lo cual genera por ende el crecimiento en algunos sectores, tales como el sector automotriz.

Este crecimiento hace que los quiteños busquen un medio para trasladarse a largas o cortas distancias en vehículos, ya sea propio, de uso público o través de aplicaciones móviles por medio de un teléfono inteligente. Sin embargo, muchos de estos vehículos sufren desperfectos mecánicos los cuales ocasionan malestar en las vías de la Capital, generando largas filas de tráfico.

El alcalde del Distrito Metropolitano de Quito consignó la fundación de la AMT (Agencia Metropolitana de tránsito) delegada en la libertad administrativa a las autoridades competentes, las mismas que controlan el transporte terrestre, así como, el tránsito y la seguridad vial en el área metropolitana de Quito y demás facultades discutidas en la citada resolución [1].

“Los traslados de vehículos, ya sean estos accidentados, multados o averiados se vienen realizando a través de servicios de plataformas auto cargables (winchas), los cuales son prestados por varias empresas autorizadas por las autoridades competentes. Dicho servicio consiste en acudir al llamado ya sea por parte del dueño del vehículo, la AMT (Agencia Metropolitana de Transito), GSM (Sistema de Gestión de Movilización) o ECU911 en caso de accidentes de tránsito [2]”.

Actualmente las actividades que se vienen realizando entre los usuarios dueños de vehículos, y los señores de las plataformas auto cargables, no cuentan con un procedimiento automatizado y control sistemático en donde se identifiquen tiempos estimados de respuesta para su cumplimiento. Para poder realizar un pedido de una plataforma auto cargable (wincha), se realiza a través de una llamada telefónica, referencia de algún amigo o en coordinación por parte de un agente de la AMT, por lo tanto dicho procedimiento no cuenta con una coordinación total ni manejo del tiempo establecido para cumplir la actividad requerida, todo esto genera malestar en las vías de Quito, retrasos e incumplimientos en las labores diarias de los usuarios que requieren el servicio de una plataforma auto cargable.

Por consiguiente, el presente proyecto pretende desarrollar una aplicación multiplataforma que permita realizar pedidos de plataformas auto cargables (winchas), basados en la geolocalización y que referencie a las plataformas auto cargables más cercanas al usuario, agilizando así los tiempos de respuestas ante cualquier eventualidad vehicular.

1.1 Objetivo general

Desarrollar una aplicación móvil para la autogestión de plataformas auto cargables (winchas).

1.2 Objetivos específicos

1. Establecer los requerimientos de la aplicación móvil.
2. Diseñar la arquitectura y la base de datos de la aplicación.
3. Codificar la aplicación.
4. Probar la funcionalidad de la aplicación.

1.3 Alcance

En la actualidad muchas empresas ofrecen sus servicios a través de aplicaciones móviles, como es el caso de UBER, DIDI, INDRIVE, etc. Estas permiten el traslado de usuarios de un punto **A** hacia un punto **B**. Sin embargo, los auxilios o traslados de vehículos averiados en la ciudad de Quito se los viene realizando manualmente a través de plataformas auto cargables (winchas), los cuales son prestados por varias empresas, respectivamente autorizadas por las autoridades competentes. Dicho servicio consiste en acudir al llamado telefónico o grupo de WhatsApp, ya sea por parte del dueño del vehículo, la AMT (Agencia Metropolitana de Tránsito), GSM (Sistema de Gestión de Movilización) o ECU911 en caso de accidentes de tránsito. Este procedimiento no cuenta con una coordinación total ni manejo del tiempo establecido para cumplir la actividad requeridas por parte de los usuarios, generando malestar en las vías de Quito, retrasos e incumplimientos en las labores diarias de los ciudadanos quiteños.

Por consiguiente, el presente proyecto pretende desarrollar una aplicación móvil que ayude a gestionar pedidos de plataformas auto cargables (winchas), basados en la geolocalización, esta información será suministrada por los usuarios para determinar la dirección en la cual serán recogidos los vehículos averiados, agilizando y optimizando así los tiempos de respuestas ante cualquier eventualidad vehicular, evitando contratiempos y congestionamientos en la vía pública.

La aplicación móvil contara con los siguientes roles: Proveedor de servicio de winchas y Cliente final.

Los datos proporcionados por los dueños proveedores de plataformas auto cargables (winchas), deberán contar con un permiso de funcionamiento respectivo otorgado por las autoridades competentes de la AMT para poder ofrecer sus servicios.

Posteriormente se expone la funcionalidad de la aplicación móvil.

El perfil del cliente en el aplicativo móvil permite:

Crear nuevo usuario mediante un formulario con campos concretos como son los datos personales.

Iniciar sesión mediante un usuario y contraseña.

Visualizar las winchas que se encuentran registradas en la base de datos y que además están en servicio.

Visualizar los datos del proveedor de servicio de una plataforma auto cargable wincha, además de los precios de servicio.

Visualizar la distancia entre el usuario y proveedor de servicio de winchas.

Realizar un pedido de servicio de traslado de una wincha en la cual se compartirán la ubicación del usuario.

El perfil del cliente proveedor de servicio en el aplicativo móvil permite:

Crear nuevo usuario mediante un formulario con campos concretos como son los datos personales, además de registrar los datos del vehículo.

Iniciar sesión mediante un usuario y contraseña.

Visualizar los pedidos de auxilio por parte de los usuarios finales, mismos que serán enviados al correo registrado con todos los datos requeridos del cliente final como: ubicación de recogida, datos del cliente final.

1.4 Marco Teórico

Grúa de plataforma auto cargable (wincha).

Una wincha o grúa de plataforma es un vehículo de transporte pesado, el cual es capaz de transportar autos que por desperfectos mecánicos o accidentados no puedan transportarse en sus propios ejes o avanzar largos caminos [3]. Estas plataformas se ajustan al vehículo

para que pueda subir mediante un sistema de enganches. Comúnmente este servicio es preferido por los usuarios, ya que provee mayor seguridad para los autos.

Expuesto esto, la gestión de plataformas auto cargables (winchas), a través del aplicativo móvil, es transcendental para la empresa que va a efectuarlo y una iniciativa importante para quienes lo consideren.

React

Es una librería de JavaScript que permite construir interfaces de usuario interactivas de forma muy sencilla y flexible, además de ser de código abierto y estar basado en componentes encapsulados los cuales manejan sus propios estados. Sus principales tecnologías son JavaScript (JSX), HTML y CSS las cuales sirven para crear interfaces de usuario [4].

Ionic

Es un SDK de código abierto para el desarrollo de aplicaciones híbridas las cuales están basadas en tecnologías web (HTML, CSS y JS). Es decir, el *framework* nos permite desarrollar aplicaciones iOS, Android y Web, a partir del mismo código base. Además, se integra con los principales marcos de desarrollo *front-end* como Angular, React y Vue, aunque también se puede usar JavaScript estándar [5].

Ionic – React

Es la versión nativa de *Ionic-Framework* para React, un SDK gratuito y de código abierto que impulsa millones de aplicaciones en todo el mundo.

La versión React nativa de *Ionic-Framework* facilita la creación de aplicaciones para iOS, Android, escritorio y la web como una aplicación web progresiva, ambos tienen una misma base de código y métodos de desarrollo comunes y usan la biblioteca *react-dom* común y un gran ecosistema alrededor de la plataforma web [6].

TypeScript

TypeScript es un lenguaje de programación basado en JavaScript que le brinda las mejores herramientas en cualquier situación. Este ha sido desarrollado por Microsoft para desarrollar aplicaciones de gran tamaño, además de ser multiplataforma, al igual que JavaScript, no requiere ninguna máquina virtual. [7].

Bases de datos NoSQL

Las bases de datos NoSQL son aquellas que no tienen un identificador o algo que las relacione, estas están diseñadas para tipos de datos específicos y flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son muy reconocidas por su facilidad de desarrollo, rendimiento y escalabilidad [8].

Firebase

Es una plataforma digital provista por Google, diseñada para un desarrollo de web y móvil rápido y eficiente. Firebase proporciona una base de datos para almacenar y combinar datos NoSQL almacenada en la nube, los datos se almacenan en formato JSON y se comunican en tiempo real con cada cliente conectado [9].

2 METODOLOGÍA

El enfoque ágil para el desarrollo de software hoy en día está de moda y describe los procesos de software modernos los cuales requieren de mucha disciplina. La flexibilidad se puede usar en cualquier proyecto de software, pero para que esto suceda, es importante que esté estructurado de tal manera que el equipo del proyecto pueda cambiar tareas, diseñar y entregar en el menor tiempo posible para el cliente. La metodología ágil en el contexto de trabajo de ingeniería de software, hoy en día se ha convertido en algo de moda para describir un proceso de software moderno, y requiere de mucha disciplina. La agilidad se puede aplicar a cualquier proceso de software, sin embargo, para lograrlo es esencial que este se diseñe de forma que permita al equipo del proyecto adaptar las tareas, ejecutar la planeación, realizar las entregas con énfasis en menor tiempo posible para el cliente [10].

Por ello, la metodología “ágil” busca proveer un marco de trabajo ágil para un equipo de desarrollo, con el único fin de proporcionar entregables funcionales al cliente en el menor tiempo posible.

En este capítulo se da a conocer el diseño y la implementación de la aplicación, para lo cual se procede a seguir los pasos específicos de acuerdo con la metodología aplicada, en nuestro caso la metodología RAD (*Rapid Application Development*).

Los desafíos que enfrentan las organizaciones de desarrollo de software se pueden resumir como más, mejor y más rápido. El camino de desarrollo de RAD ataca estos desafíos de frente al proporcionar un medio para desarrollar sistemas más rápido, al mismo tiempo que se reducen los costos y aumenta la calidad.

Aspectos esenciales de RAD

La metodología RAD se caracteriza por tener cuatro aspectos esenciales, en la **Fig. 1** se puede visualizar el entrelazamiento de dichos aspectos: metodología, gente, gestión y herramientas. Si llegase a faltar uno de estos no se podría desarrollar una aplicación rápida. [11]

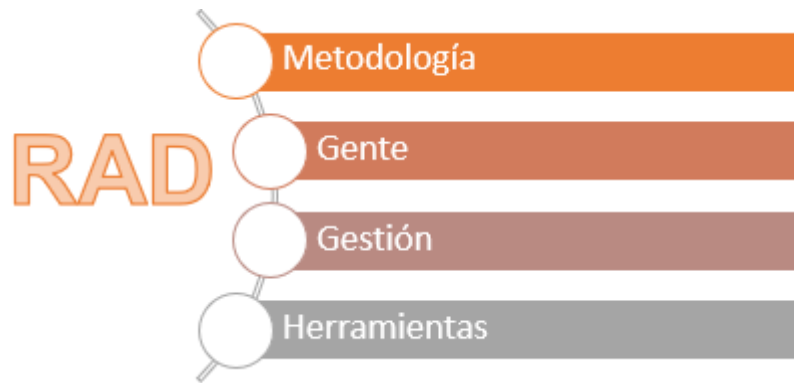


Fig. 1: Aspectos Rad

METODOLOGÍA

En el apartado 2.1 se refiere a la metodología y de su relevancia para el desarrollo rápido de aplicaciones.

2.1 Metodología de Desarrollo

RAD (*Rapid Application Development*) ataca a los desafíos de desarrollo de software como más, mejor y más rápido, proporciona un medio para desarrollar sistemas más rápido, reduciendo costos y mejorando la calidad al mismo.

Aprovecha las herramientas y técnicas automatizadas para reestructurar el proceso de desarrollo. Este proceso, se refleja en una transformación del desarrollo de los sistemas de información. RAD reemplaza los procesos de codificación y diseño manual, que dependen de las habilidades de individuos adiestrados, con diseño y codificación automatizados, que es un proceso inherentemente más estable. RAD puede así dar a una organización dedicada al desarrollo su primera base real para la mejora continua. Además de ser más estable, el desarrollo rápido de aplicaciones es un proceso más capaz, ya que es mucho más rápido y menos propenso a errores que la codificación manual.

RAD es una metodología enfocada a desarrollar aplicaciones de forma veloz, interactuando con el usuario y apoyándose en la abstracción de la información [12], la cual garantiza cumplir con las metas requeridas, reduciendo costos y tiempos de entrega.

La **Fig. 2** demuestra las cuatro fases que comprende la metodología en su ciclo de vida

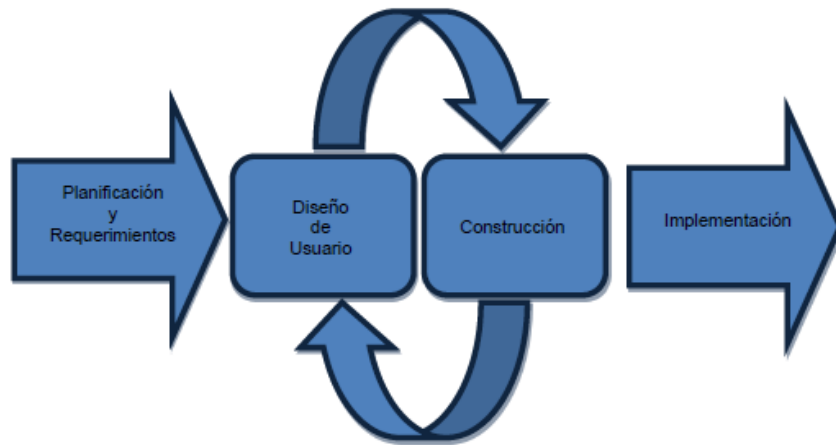


Fig. 2 Fases de RAD

Fases de la metodología RAD

Planeación de los requerimientos

Es también conocida como la etapa en la que se definen conceptos en la que se conoce las funciones del negocio y se representa las peculiaridades de la aplicación a desarrollar. La información recolectada nos permite definir el alcance del proyecto.

Diseño

En la fase de diseño se lleva a cabo reuniones con los usuarios y desarrolladores, con la finalidad de realizar modelos y prototipos referente a los requerimientos establecidos del sistema.

Dentro del marco de trabajo RAD existen las herramientas CASE, las cuales son usadas para modelar las necesidades del usuario, dándole una expectativa más real al cliente al momento de diseñar un sistema.

Construcción o desarrollo

En esta fase el equipo de desarrollo trabaja de la mano con el usuario, en la que se generan procesos y componentes, estas a su vez se les realiza pruebas e integración necesarios para el buen funcionamiento del sistema.

Implementación

En la fase de implementación incluye pruebas y capacitación del usuario final y posterior a ello la implementación de la aplicación, no sin antes haber realizado todas las pruebas necesarias.

GENTE

Roles (*stakeholders*)

El éxito de la metodología RAD, depende de la participación de personas involucradas que posean las destrezas y talentos adecuados que, combinadas con las herramientas esenciales para el desarrollo de las aplicaciones, exige un grado de esfuerzo cooperativo entre un grupo relativamente pequeño de personas.

Cada etapa de desarrollo incluye actividades que deben ser ejecutadas rápidamente.

Los actores RAD que intervienen en un proyecto son:

Sponsor

Un ejecutivo de usuarios de alto nivel que financia el sistema y se dedica tanto al valor del nuevo sistema y lograr resultados rápidamente.

User Coordinator

Designado por el patrocinador para supervisar el proyecto desde la perspectiva del usuario.

Requirements Planning

Un equipo de usuarios de alto nivel que participan en la planificación conjunta de requisitos.

User Desing Team

Un grupo de usuarios se encarga de revisar el sistema después de que se diseña. Este equipo debe estar compuesto por usuarios del equipo de planificación y usuarios con un conocimiento más detallado del sistema.

User Review Board

Un grupo de usuarios que se encargan de revisa el sistema después de la construcción y decide si las modificaciones son pertinentes antes de la transformación.

Training Manager

El ente responsable de capacitar a los usuarios para desarrollar con el nuevo sistema.

Project Manager

La persona que controla el esfuerzo de desarrollo.

Construction (SWAT) Team

The SWAT (Skilled Workers with Advanced Tools)

El equipo SWAT (trabajadores capacitados en herramientas avanzadas) es un pequeño equipo de dos a seis desarrolladores que están altamente capacitados para trabajar juntos a alta velocidad. A lograr el desarrollo más rápido posible, los miembros del equipo deben ser altamente capacitados en la metodología RAD y en el uso del conjunto de herramientas CASE elegido.

Workshop Leader

Es el especialista que organiza y conduce los talleres de requerimientos, planificación y diseño de aplicaciones conjuntas.

Asignación de roles

Los roles asignados en el equipo de trabajo del aplicativo móvil se encuentran establecidos en la **TABLA I**.

TABLA I: EQUIPO DE TRABAJO Y ASIGANCIÓN DE ROLES

MIEMBROS	ROLES
Gabriel Medina	<i>Sponsor</i>
Ing. Juan Pablo Zaldumbide. MSc.	<i>Project Manager</i>
Gabriel Medina	<i>Construction (SWAT) Team</i>

GESTIÓN

Lograr un desarrollo de alta velocidad es un proceso demasiado complejo ya que los sistemas burocráticos son extensos. La dirección RAD debe estar totalmente comprometida a gestionar el cambio de cultura, motivar al personal y departamento de TI a cumplir con los objetivos en el menor tiempo posible, utilizando medidores de rendimiento.

HERRAMIENTAS

La metodología RAD utiliza herramientas de última generación, además del talento humano hábil para lograr cumplir los objetivos rápidos y en el menor tiempo posible.

Aplicación de la metodología

De acuerdo con el flujo de trabajo propuesto por la metodología RAD, se procederá a la aplicación y desarrollo de cada fase, las distintas actividades que deben ser llevadas a cabo.

Planeación de los Requerimientos

Esta es una etapa inicial en donde los requerimientos comprende una de las fases más relevantes para el éxito en el desarrollo de un sistema, ya que al tener conocimiento de lo que el cliente desea no se malgasta tiempo ni recursos en desarrollar un sistema obsoleto, sino que se centra en las expectativas deseadas por el cliente cumpliendo así su total satisfacción.

Historias de Usuarios

Las historias de usuario se utilizan como herramienta para dar a conocer los requisitos del sistema por parte de los desarrolladores, estas historias de usuario son pequeños documentos descriptivos que describen los eventos que ocurren en el sistema, de forma sencilla, comprensible y si entrar a detalles, guía que describe una actividad que se desarrolla en el sistema, este se escribe bajo términos del cliente, no del desarrollador, de forma sencilla, clara, sin profundizar en detalles. Todos estos escritos son en base a las necesidades de cliente y no del desarrollador. [13].

En la **TABLA II**, se puede apreciar un ejemplo de historia de usuario.

TABLA II: EJEMPLO DE HISTORIA DE USUARIO INICIO DE SESIÓN

HISTORIA DE USUARIO	
Número: 2	Usuario: Cliente
Nombre de la Historia: Inicio de sesión del usuario cliente.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsables: Gabriel Medina	
Descripción: los usuarios clientes quienes utilicen la aplicación móvil pueden acceder mediante las credenciales que definieron en el momento del registro.	
Observación:	

2.2 Diseño de interfaces (*mockups*)

Diseño de interfaces de la aplicación móvil

Una vez establecidos los requerimientos y diseñado la base de datos se procede al diseño de los prototipos de la aplicación móvil, los cuales permitirán una visualización previa.

JustinMind

Es una Herramienta de diseño y creación de prototipos para aplicaciones web y móviles. Que permite crear plataformas de diseño de UI y UX todo en uno para crear activos, prototipos y simulaciones de UI [14].

DISEÑO DE APLICACIÓN MOVIL PROVEEDOR DE SERVICIO

En las **Fig. 3**, se puede visualizar la pagina principal de la aplicación móvil, en ella el usuario debera ingresar con un usuario y contraseña, previamente registrado en la aplicación, la **Fig. 4**, demuestra el formulario de registro.



Fig. 3: Inicio de Sesión



Fig. 4: Formulario de registro.

2.3 Diseño de la arquitectura

Arquitectura de Datos

De acuerdo con lo establecido en la planificación, se procedió a diseñar la base de datos, la cual está formada por un modelo no relacional en la que los datos se almacenan en forma de documentos y colecciones, por lo que la hace más flexible, en la Fig. 5 podemos apreciar los elementos de los que está compuesta en base a los requerimientos establecidos anteriormente.

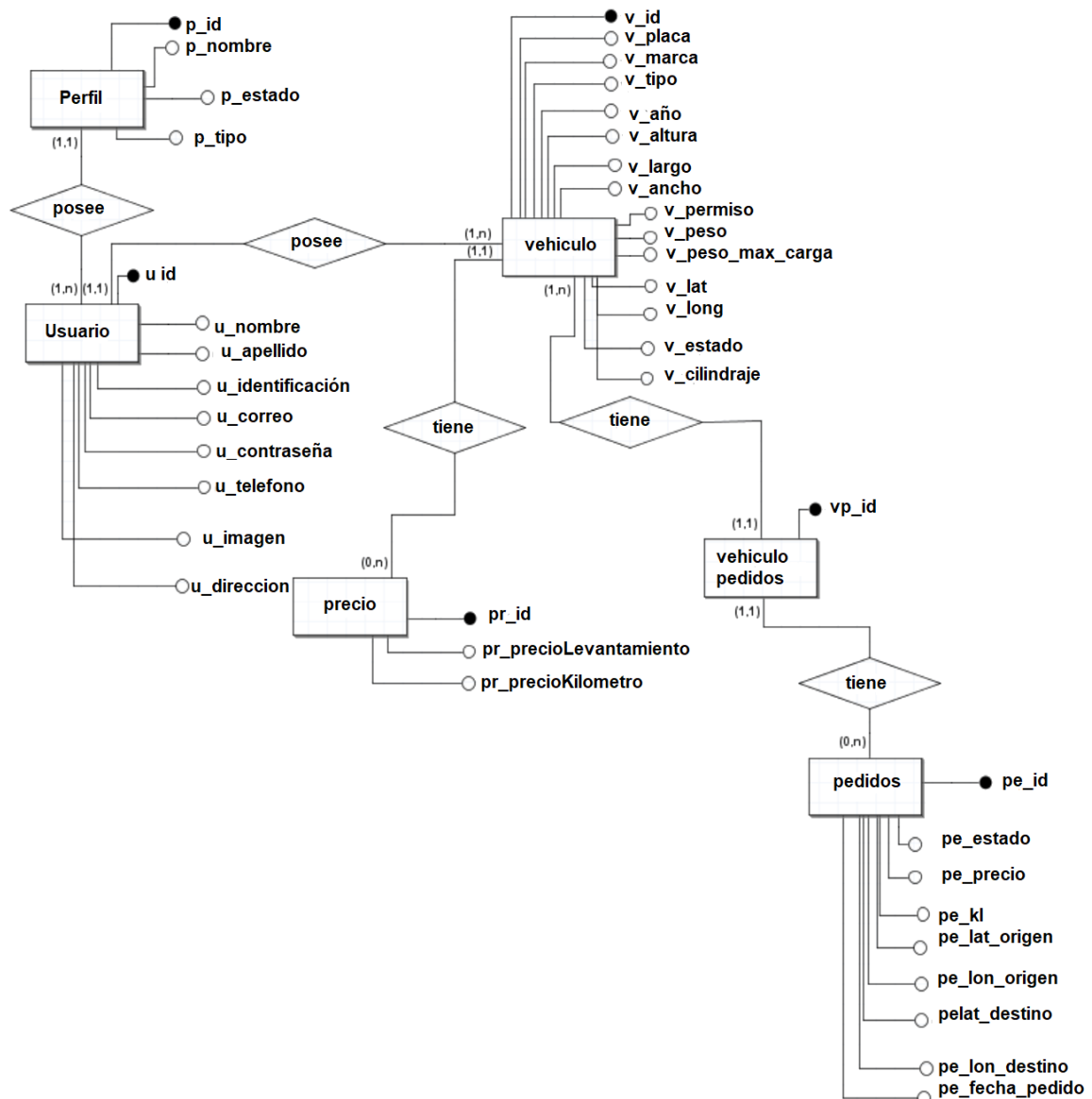


Fig. 5: Modelo No relacional de la base de Datos.

Patrón arquitectónico

A continuación, se detallan las arquitecturas utilizadas en el proyecto

Patrón arquitectónico MVC (Modelo, Vista, Controlador)

Su estructura está compuesta por módulos independientes, cada módulo cumple con una función concreta, en la que separa la lógica del sistema, la lógica de datos y la lógica de interfaz [15].

Modelo

Este módulo contiene la lógica de manejo de datos, es decir es el encargado de procesar, administrar y actualizar la información en la base de datos.

Vista

La vista es la encargada de presentar la información a los usuarios por medio de pantallas, prácticamente este módulo es conocido como el *frontend*.

Controlador

Este módulo se encarga de gestionar y procesar las peticiones, a través de este se comunican el modelo y la vista, solicitando los datos requeridos y entregándolos al usuario.

En la **Fig. 6**, podemos apreciar la composición de nuestro modelo arquitectónico.



Fig. 6: Patrón Arquitectónico modelo MVC

Aplicación Móvil

En la **Fig. 7**, se puede apreciar el diseño de la arquitectura de la aplicación móvil. En esta simula como los diferentes tipos de usuarios se comunican desde la aplicación móvil, a través del internet y se conectan con la base de datos.

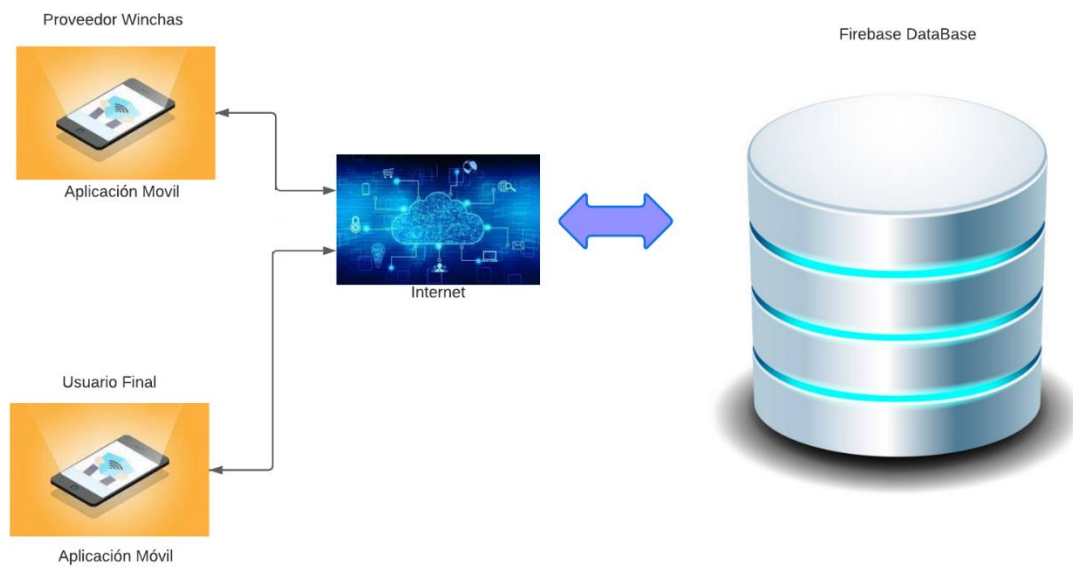


Fig. 7: Arquitectura de Aplicación Móvil

2.4 Herramientas de desarrollo

El uso de las siguientes herramientas en el desarrollo del actual proyecto se las eligió en pie a la propuesta de arquitectura planteada, en esta se detallan las herramientas a usarse para el desarrollo de la aplicación móvil demostrada en la **TABLA III**.

Aplicación Móvil

TABLA III: HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACIÓN MÓVIL

Herramienta	Justificación
Ionic - React	La aplicación es nativa ya que Ionic - React es un proyecto de interfaz de usuario y API de código abierto con varios componentes de interfaz de usuario y componentes nativos para crear aplicaciones web IOS y Android, utilizando React y tecnologías comunes en una web [16].

Visual Studio Code	Al ser un editor de código multiplataforma, nos permite trabajar con varios lenguajes de programación, además de tener integración compatible con GitHub por lo que podemos llevar un control de versiones fácilmente [17].
FireBase	FireBase nos permite desarrollar aplicaciones de calidad y de forma rápida, además de poseer diferentes funciones que se pueden adaptar a las herramientas que se usará para el desarrollo [18].
Capacitor	Capacitor es una herramienta que nos ayuda en el desarrollo de aplicaciones híbridas, a través de este se puede acceder de manera muy sencilla a las API's nativas e invocar varias funcionalidades del dispositivo.
JustinMind	Es una herramienta útil que nos permite realizar bosquejos de acuerdo con la necesidad del cliente.

Librerías

En la **TABLA IV**, se especifica las librerías utilizadas en la aplicación móvil, estas fueron necesarias para un correcto funcionamiento de la aplicación.

TABLA IV: API'S INSTALADAS PARA EL DESARROLLO DEL PROYECTO

API	Justificación
@capacitor/geolocation	La API de geolocalización proporciona formas sencillas de encontrar y verificar la posición GPS actual de un dispositivo, junto con la información de altitud, rumbo y velocidad, si está disponible.
@capacitor/camera	La API de la cámara brinda la capacidad de tomar una foto con la cámara o seleccionar una foto existente del álbum de fotos.

@ionic/react-hooks	Son un conjunto de librerías para ayudar a los desarrolladores a usar las API nativas de Capacitor.
@ionic-native/email-composer	Es un complemento que brinda acceso a la interfaz estándar que administra la edición y el envío de mensajes de correos electrónicos, en el cual se puede completar los campos de esa vista con valores iniciales.
@types/firebase	Es un tipado de código auxiliar para la API de Firebase.
@awesome-cordova-plugins	Es un conjunto seleccionado de contenedores para los complementos de Cordova que facilitan la adición de cualquier funcionalidad nativa que necesite una aplicación móvil.

3 RESULTADOS

En esta sección se analiza las iteraciones basadas en la metodología, además se detalla el progreso en el desarrollo del componente móvil.

Iteración 0. Configuración del ambiente de desarrollo

En la iteración 0, se procede a la configuración del entorno de desarrollo, en el cual se realizan las acciones necesarias para obtener un ambiente de trabajo adecuado para el desarrollo óptimo del proyecto.

Creación de la base de datos en firebase

Para la creación de la base de datos procedemos a crear una cuenta en Gmail en Google, para poder acceder los beneficios que esta ofrece, ya sean de forma gratuita o pagada.

Después de crear la cuenta Gmail nos dirigimos al navegador para ingresar a Firebase, en la **Fig. 8** demuestra la creación de la base de datos que utilizaremos para el desarrollo de la aplicación móvil.



Fig. 8: Creación de cuenta Firebase.

Instalación de programas necesarios

Procedemos a instalar las herramientas necesarias para el desarrollo de la aplicación móvil.



Fig. 9: Instalación de NodeJs

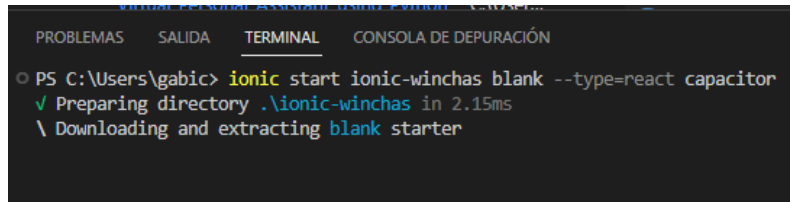
En la **Fig. 9**, encontramos una recomendación de descarga desde la página web oficial de nodejs, en esta nos recomienda instalar la versión más actual para Windows, sin embargo, se recomienda descargar una versión estable y que sea compatible con las demás herramientas de desarrollo y que se apegue a las necesidades de nuestro proyecto. En nuestro caso en la **TABLA V** podemos evidenciar la instalación de una versión anterior a la ofrecida por la página web.

TABLA V: REQUERIMIENTOS PROYECTO MÓVIL.

Requisitos	Versión	Instalación
node js	v16.14.0	https://nodejs.org/es/
Visual Studio Code	Ultima versión	https://code.visualstudio.com/download
Ionic	6.17.1	<code>npm install ionic@latest -g</code>
React - Hooks	Ultima versión	<code>npm install --save react-firebase-hooks</code>

Creación del proyecto

La **Fig. 10**, demuestra la creación de la aplicación móvil ejecutando el comando `ionic start ionic-winchas blank --type=react --capacitor`.



```
PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN
PS C:\Users\gabic> ionic start ionic-winchas blank --type=react capacitor
✓ Preparing directory .\ionic-winchas in 2.15ms
  \ Downloading and extracting blank starter
```

Fig. 10: Creación de proyecto.

Creación del repositorio GitHub

Para almacenar el código fuente procedemos crear un repositorio en GitHub, el cual nos ayudara en el manejo de versiones. En la **Fig. 11**, podemos evidenciar el proyecto creado y subido al repositorio de GitHub.

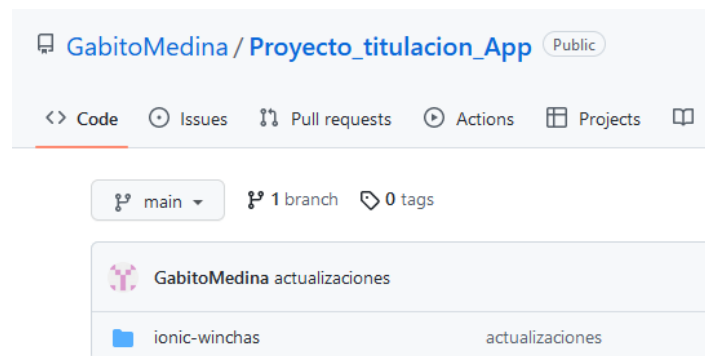


Fig. 11: Repositorio GitHub

Iteración 1

Diseño de la pantalla principal de la aplicación móvil

En la iteración 1, procedemos al diseño de la pantalla principal de la aplicación, la cual es por medio de un usuario y contraseña (véase Fig. 12), además se diseñó la pantalla en la que se diferencia los roles de usuario a registrarse (véase **Fig. 13**).



Fig. 12: Pantalla principal

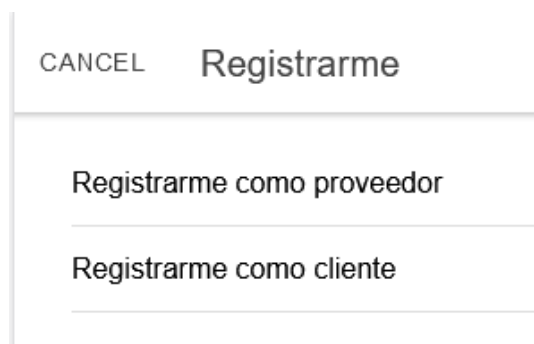


Fig. 13: Pantalla diferenciación de roles.

Iteración 2

En esta iteración se procede a realizar las actividades siguientes, cada una de ellas son necesarias para la autenticación, tanto como proveedor como cliente final en la aplicación móvil.

Módulo de registro de usuarios proveedor y cliente final CRUD

Validaciones de registro e inicio de sesión de los usuarios proveedor y cliente final.

Módulo de registro de vehículos winchas de los proveedores CRUD.

Módulo de registro de ubicación de las winchas registradas CRUD.

Módulo registro de precios de servicios de las winchas CRUD.

Visualización de las listas de winchas registradas por el proveedor.

Módulo de registro de usuarios proveedores CRUD

En la **Fig. 14**, se evidencia el registro de datos de un usuario, los cuales funcionan de manera estándar, llenando un formulario de registro apropiado con datos básicos y necesarios.



The image shows a registration form with the following fields and elements:

- Form title: Registro
- Form fields: Nombre, Apellido, Identificación, Correo electrónico, Teléfono, Dirección, Contraseña, Confirmar Contraseña.
- Buttons: A blue button labeled REGISTRARME.
- Link: ¿Ya tienes una cuenta? [Iniciar sesión](#)

Fig. 14: Pantalla de registro.

Validaciones de registro e inicio de sesión

En la **Fig. 15**, podemos ver que los campos marcados con un asterisco (*) son obligatorios para registrarse correctamente a la base de datos. Mientras que en la **Fig. 16**, evidenciamos un pequeño tramo de código, el cual valida que el email ingresado sea de formato correcto. Estas reglas son necesarias para poder acceder o registrarse en la aplicación.

Iniciar sesión'."/>

Fig. 15: Validación de correo.

```

const validateEmail = (email: string) => {
  return email.match(
    /^(?=.{1,254}$)(?=.{1,64}@)[a-zA-Z0-9!#$%&'*/+=?^_`{|}~-]+(?:\. [a-zA-Z0-9!#$%&'*/+=?^_`{|}~-]+)?$/);
};

const validate = (ev: Event) => {
  const value = (ev.target as HTMLInputElement).value;

  setIsValid(undefined);

  if (value === '') return;

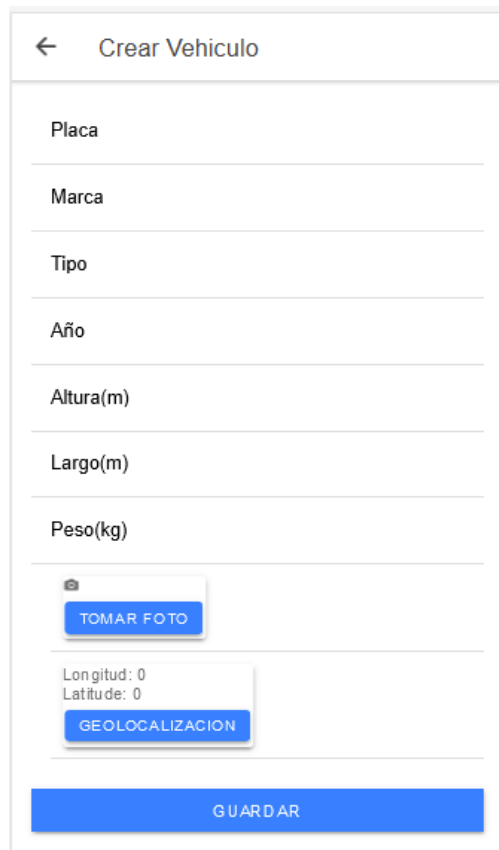
  validateEmail(value) !== null ? setIsValid(true) : setIsValid(false);
};

```

Fig. 16: Código de validación de correo.

Módulo de registro de vehículos winchas proveedores CRUD.

Cada proveedor de servicio posee una o varias winchas las cuales deberán ser registradas de manera obligatoria ya que sin estas no podrán ofrecer sus servicios, además después del registro apropiado de la wincha deberán registrar precios de servicio y ubicación en la que se encuentren estáticos a esperar un llamado de auxilio. En la Fig. 17, podemos evidenciar un registro de una plataforma auto cargable.



← Crear Vehiculo

Placa

Marca

Tipo

Año

Altura(m)

Largo(m)

Peso(kg)

TOMAR FOTO

Longitud: 0
Latitud: 0

GEOLOCALIZACION

GUARDAR

Fig. 17: Formulario de registro de vehículo.

Iteración 3:

En esta entrega las tareas siguientes están sujetas a tareas específicas que solo el usuario cliente final puede realizar una vez registrado e iniciado sesión.

Visualización de las winchas disponibles.

En la **Fig. 18**, se evidencia el listado de grúas disponibles en la aplicación móvil.

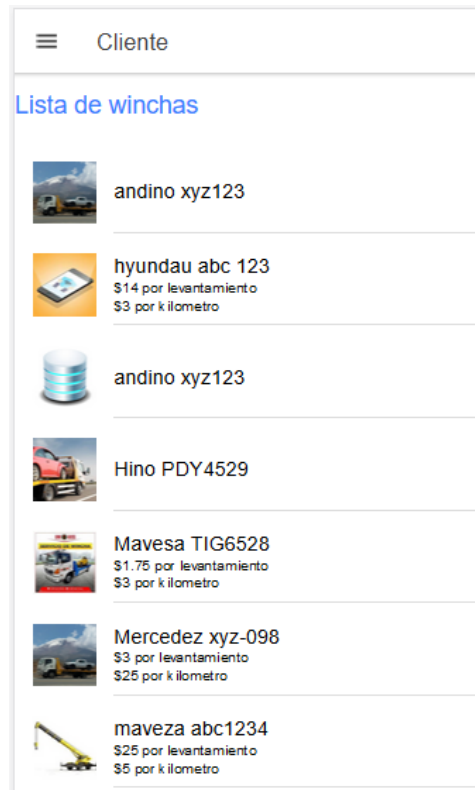


Fig. 18: Lista de grúas disponibles.

Módulo de registro de ubicación CRUD.

Una vez iniciado sesión podemos visualizar una lista de winchas registradas en la base de datos, la cual nos permitirá acceder a la ubicación en tiempo real del vehículo dañado como se muestra en la siguiente **Fig. 19**.

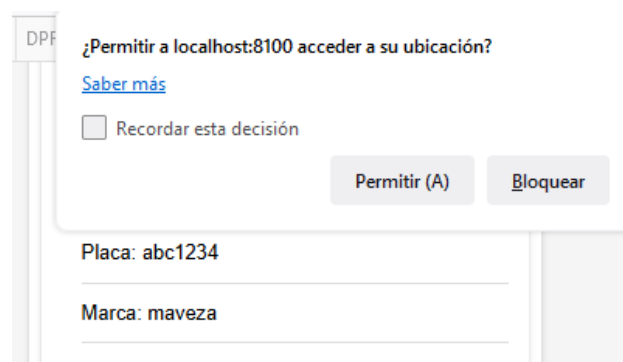


Fig. 19: Acceso a localización.

Además, en la **Fig. 20**, evidenciamos la distancia entre el usuario cliente final y el proveedor de servicios



Fig. 20: Visualización de distancia.

Módulo solicitud de servicio.

En este módulo se puede hacer una solicitud de acuerdo con las necesidades del cliente, en el cual puede llamar, comunicarse por medio de la aplicación whatsapp (véase **Fig. 21**) o enviar un email, sin embargo, al presionar el botón solicitar la aplicación enviará una solicitud a través de un email, este será el mismo que el proveedor registro al momento de suscribirse a la aplicación.

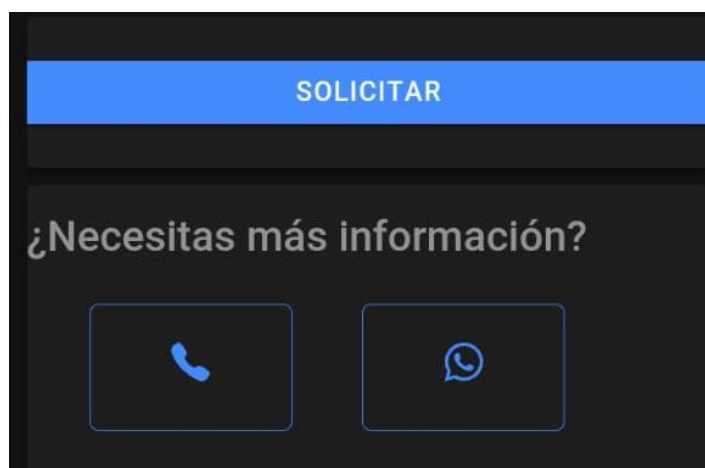


Fig. 21: Botones de comunicación

Iteración 4: Pruebas y despliegue

De acuerdo con la planificación determinada dentro de la metodología ágil, la iteración 4 está comprendida por pruebas y despliegue, las mismas que fueron ejecutadas en dispositivos móviles luego de haber terminado el ciclo de desarrollo.

Pruebas de aceptación

En la **TABLA I**, evidenciamos la prueba de aceptación en el cual el **Project Manager**, comprobó que todos los requerimientos se realizaron antes de desplegar la aplicación.

TABLA VI: Prueba de aceptación 1

Prueba de aceptación
Identificador: PA1
Identificador historia de usuario: 1 Nombre prueba de aceptación: Inicio de sesión
Descripción: El usuario debe ingresar llenando el formulario con la información registrada como lo son usuario y contraseña.
Pasos de ejecución: Ingreso de datos correspondientes. Presionar en entrar.
Resultado deseado: Después de ingresar los datos correctos el usuario puede acceder a la aplicación móvil.
Evaluación de la prueba: Se comprueba el resultado deseado.

TABLA VII: Prueba de aceptación 2

Prueba de aceptación
Identificador: PA2
Identificador historia de usuario: 3
Nombre prueba de aceptación: Registro de datos de la grúa. Dar un toque en registrar.
Descripción: El usuario debe registrar los datos de la grúa completando el formulario con la información requerida.
Pasos de ejecución: Registro de los datos correspondientes de la grúa.
Resultado deseado: Después de registrarse en la aplicación, el usuario puede ingresar a la aplicación móvil mientras espera la interfaz del usuario.
Evaluación de la prueba: Se comprueba el resultado deseado.

TABLA VIII: Prueba de aceptación 3

Prueba de aceptación
Identificador: PA3
Identificador historia de usuario: 9
Nombre prueba de aceptación: Crear solicitud de traslado
Descripción: El usuario debe ingresar a la aplicación móvil y escoger la grúa de su gusto de la lista de grúas, dar un toque en solicitar.
Pasos de ejecución:

Iniciar sesión.

Esperar que se carguen los datos de las grúas registradas.

Escoger la grúa de su gusto.

Dar un toque en solicitar.

Resultado deseado:

Una vez ingresado en la aplicación el usuario podrá visualizar un listado con las grúas registradas en la base de datos, escoger la grúa de su gusto y solicitar el servicio.

Evaluación de la prueba:

Se comprueba el resultado deseado.

Pruebas de satisfacción

Se realizaron encuestas para determinar la satisfacción de los usuarios y se evaluó. En la **Fig. 22**, se pueden ver los resultados de una de las preguntas más importantes, y al mismo tiempo se decidió que el sistema operativo más utilizado es Android.

¿Qué sistema operativo tiene tu celular?

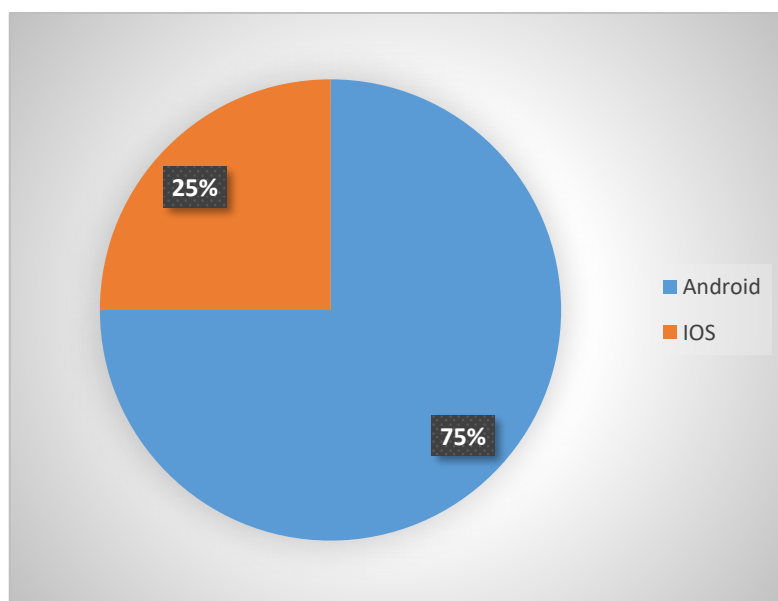


Fig. 22: Sistema operativo más popular

Pruebas de compatibilidad

La **TABLA IX** a continuación muestra los resultados de las pruebas de compatibilidad, los dispositivos fueron evaluados con varias versiones de Android de forma virtual y física.

TABLA IX: Prueba de compatibilidad en dispositivos Android

N°	Dispositivo	Versión	Virtual	Físico
1	Huawei P30 PRO	11.0	X	
2	Samsung A10	9.0.	X	
3	Motorola Moto X	5.0	X	
4	Xiaomi Redmi Note 7	10.0	X	
5	Samsung A30	11.0		X
6	Samsung	8.1.1		X
7	Samsung A50	12		X

4 CONCLUSIONES

- La metodología RAD aplicada al proyecto demostró tener un marco de desarrollo rápido y eficaz ayudándonos con la entrega de este proyecto en un tiempo sumamente corto.
- Se logro determinar los requerimientos del desarrollo gracias al trabajo en conjunto con los clientes de este proyecto, obteniendo como resultado entregas más interactivas y una retroalimentación constante.
- La herramienta de diseño de prototipos escogida JustinMind, demostró ser la adecuada para el diseño de los prototipos de este proyecto, ya que le da al cliente una experiencia más real de la aplicación.
- El desarrollo del proyecto se realizó en el framewok ionic-react, el mismo que nos generó un proyecto multiplataforma, la misma que se implementó en la plataforma Android.
- El uso de la API de geolocalización de Capacitor incorporada en el proyecto ayudo al cliente a visualizar la distancia, entre el cliente y proveedor de servicio.
- Los resultados de las pruebas de compatibilidad determinaron la importancia de usar versiones actuales de sistema operativo Android.

5 RECOMENDACIONES

- Para un uso correcto de la metodología RAD (*Rapid Application Development*), se recomienda complementar con una herramienta CASE (*Computer Aided Software Engineering*), que en español significa Ingeniería de Software Asistida por Computadora, el cual además de prototipar nos permite generar código, reduciendo costos, tiempo y de dinero. Como ejemplo tenemos la herramienta de desarrollo Android Studio.
- Para futuras versiones se recomienda usar librerías como Bootstrap, el cual le dará un diseño más responsive y elegante a la aplicación.
- Se recomienda analizar la API de Google Maps, para realizar un mejor filtro de búsqueda y geolocalización más preciso, además se recomienda usar una versión pagada ya que esta nos ofrece más beneficios desde la API.
- Para un mejor funcionamiento de la aplicación es necesario trabajar con una base de datos pagada ya que Firebase usa una capa gratuita y por tiempo limitado.
- Para un correcto funcionamiento de la aplicación móvil se recomienda usar *WebSockets*, ya que estos permiten un canal de comunicación activa entre la aplicación móvil, un servidor y el cliente.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] A. m. d. transito, «http://www.amt.gob.ec/files/resolucionA0006_22-04-2013.pdf,» 22 Abril 2013. [En línea]. [Último acceso: 15 noviembre 2022].
- [2] N. Alejandro, Interviewee, *Agente AMT*. [Entrevista]. 2 noviembre 2022.
- [3] «Grúas García,» [En línea]. Available: <https://www.gruasgarciadf.com/caracteristicas-de-la-grua-de-plataforma.html>. [Último acceso: 01 01 2023].
- [4] «React,» [En línea]. Available: <https://es.reactjs.org/docs/getting-started.html>. [Último acceso: 16 01 2023].
- [5] «Ionic,» [En línea]. Available: <https://profile.es/blog/que-es-ionic/>. [Último acceso: 16 enero 2023].
- [6] «Ionic-React,» [En línea]. Available: <https://ionic.io/blog/announcing-ionic-react>. [Último acceso: 16 enero 2023].
- [7] «IMMUNE TECHNOLOGY INSTITUTE,» [En línea]. Available: <https://immune.institute/blog/typescript-que-es-como-se-diferencia-javascript/>. [Último acceso: 16 enero 2023].
- [8] AWS, «Bases de datos NoSql,» [En línea]. Available: <https://aws.amazon.com/es/nosql/>. [Último acceso: 16 enero 2023].
- [9] «Firebase,» [En línea]. Available: <https://firebase.google.com/docs/database?authuser=0..> [Último acceso: 16 enero 2023].
- [10] R. S. Pressman, *Ingeniería del Software u enfoque práctico*, Mexico: McGraw-Hill, 2010.
- [11] [En línea]. Available: <https://www.iro.umontreal.ca>. [Último acceso: 15 enero 2023].
- [12] I. Sommerville, *Ingeniería del Software*, 7ma edición, Pearson, 2005.
- [13] R. S. Pressman, *Ingeniería de software Un enfoque práctico*, séptima edición ed., McGraw-Hill, 2010.
- [14] «JustinMind,» [En línea]. Available: <https://www.justinmind.com/>. [Último acceso: 17 enero 2023].
- [15] «Easy AppCode,» [En línea]. Available: <https://www.easyappcode.com/patron-de-diseno-mvc-que-es-y-como-puedo-utilizarlo>. [Último acceso: 17 enero 2023].
- [16] «Ionic React,» [En línea]. Available: <https://ionic.io/resources/articles/ionic-react-vs-react-native>. [Último acceso: 18 enero 2023].

- [17] [En línea]. Available: <https://visualstudio.microsoft.com/es/#vscode-section>. [Último acceso: 18 enero 2023].
- [18] [En línea]. Available: <https://keepcoding.io/blog/que-es-firebase-realtime-database/>. [Último acceso: 18 enero 2023].
- [19] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [20] [En línea]. Available: <https://ionic.io/blog/announcing-ionic-react>. [Último acceso: 16 02 2023].
- [21] «JustinMind,» 2023. [En línea]. Available: <https://www.justinmind.com/>.
- [22] [En línea]. Available: <https://visualstudio.microsoft.com/es/#vscode-section>. [Último acceso: 18 enero 2023].

7 ANEXOS

A continuación, se presenta todos los anexos, los cuales se utilizaron en todo el proceso de desarrollo de la aplicación móvil, estos se hallan divididos de la siguiente manera.

ANEXO I. Turnitin porcentaje máximo 11%.

ANEXO I. Manual técnico

ANEXO IIIANEXO IV. Manual de usuario

ANEXO IV. Manual de instalación (video)



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 3 de marzo de 2023

De mi consideración:

Yo, Juan Pablo Zaldumbide Proaño, en calidad de Director del Trabajo de Integración Curricular titulado **DESARROLLO DE APLICACIÓN MÓVIL** asociado al proyecto denominado **DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE PLATAFORMAS AUTOCARGABLES (WINCHAS)** elaborado por el estudiante **EDMUNDO GABRIEL MEDINA INTRIAGO** de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

**Juan Pablo Zaldumbide Proaño
Profesor ocasional a tiempo completo
ESFOT**

ANEXO II

Historias de Usuario

Resumen de historias de usuarios

A continuación, se desarrolla cada historia de usuario, estas historias de usuario se desarrollaron en Tablas con el fin de especificar a qué usuario está enfocado, la prioridad, el riesgo de desarrollo, una descripción corta y observaciones en el caso de ser necesario. Estas Tablas sirven para identificar el funcionamiento de la aplicación móvil.

TABLA X: Historia de usuario 1: Inicio Sesión de Usuario Proveedor

HISTORIA DE USUARIO	
Número: 1	Usuario: Cliente Final
Nombre Historia: Inicio de sesión Cliente Final	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsables: Gabriel Medina	
Descripción: Los usuarios pueden acceder a la aplicación por medio de un registro e inicio de sesión.	
Observación: Todos los usuarios pueden acceder a la aplicación siempre y cuando se hayan registrado previamente	

TABLA XI: Historia de usuario 2: Registro de datos del proveedor

HISTORIA DE USUARIO	
Número:2	Usuario: Proveedor de servicio
Nombre Historia: Registro de datos del proveedor	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	

Responsables: Gabriel Medina
<ul style="list-style-type: none"> • Descripción: Los proveedores de servicio quienes posean una plataforma auto cargable (wincha) deberán registrar los siguientes datos: • Nombre y Apellido del distribuidor. • Cédula. • Email • Contraseña
<p>Observación:</p> <p>Para registrarse en la aplicación móvil el usuario podrá escoger la opción de registrarse como proveedor de servicio de winchas.</p>

TABLA XII: Historia de Usuario: 3 Registro de datos de la Wincha

HISTORIA DE USUARIO	
Número:3	Usuario: Proveedor de servicio
Nombre Historia: Registro de datos de la wincha	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsables: Gabriel Medina	

- Descripción: Los proveedores de servicio quienes posean una plataforma auto cargable (wincha) deberán registrar los siguientes datos del vehículo.
- Placa del vehículo.
- Permiso de funcionamiento otorgado por la AMT
- Peso máximo de carga
- Altura del vehículo
- Largo del vehículo
- Ubicación del vehículo
- Foto del vehículo

Observación:

Para el registro de las unidades auto cargables es necesario contar con los permisos otorgados por AMT

TABLA XIII: Historia de usuario 4: Registro de precio

HISTORIA DE USUARIO	
Número: 4	Usuario: Proveedor de servicio
Nombre Historia: Registro de precio de Servicio.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsables: Gabriel Medina	
Descripción: los proveedores de servicio de grúas quienes utilicen la aplicación deberán registrar precios de levantamiento y precios por kilómetro recorrido.	
Observación: Los precios a registrar deberán ser referenciales.	

TABLA XIV: Historia de usuario 5: Editar información

HISTORIA DE USUARIO	
Número: 5	Usuario: Proveedor de servicio
Nombre Historia: Editar Información Registrada	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 3	
Responsables: Gabriel Medina	
Descripción: Al iniciar sesión el proveedor de servicio podrá visualizar una lista de winchas registradas por parte del proveedor, además podrá escoger en que wincha está trabajando y editar la ubicación como el precio.	
<p>Observación:</p> <p>El proveedor de servicio tendrá la posibilidad de escoger en la wincha que va a trabajar ese día, en caso de poseer varias winchas registradas en la aplicación.</p>	

HISTORIAS DE USUARIO CLIENTE FINAL

TABLA XV: Historia de usuario 6: Registro de Usuario Cliente Final

HISTORIA DE USUARIO	
Número: 6	Usuario: Cliente Final
Nombre Historia: Registro de datos del cliente final	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 4	
Responsables: Gabriel Medina	
<p>Descripción: Los clientes finales quienes quieran acceder al servicio de winchas deberán registrar los siguientes datos.</p> <ul style="list-style-type: none"> • Nombre y apellidos 	

<ul style="list-style-type: none"> • Identificación • Email • Teléfono • Contraseña
Observación:

TABLA XVI: Historia de usuario 7: Inicio Sesión de Usuario Cliente Final

HISTORIA DE USUARIO	
Número: 7	Usuario: Cliente Final
Nombre Historia: Inicio de sesión Cliente Final	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 4	
Responsables: Gabriel Medina	
Descripción: Los usuarios pueden acceder a la aplicación por medio de un registro e inicio de sesión para poder ingresar a un listado de winchas disponibles	
Observación:	
Todos los usuarios pueden acceder a la aplicación siempre y cuando se hayan registrado previamente	

TABLA XVII: Historia de usuario 8: Visualización de listado de winchas

HISTORIA DE USUARIO	
Número:8	Usuario: Cliente final
Nombre Historia: Visualización de listado de winchas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta

Iteración Asignada: 4
Responsables: Gabriel Medina
Descripción: Los usuarios finales quienes se hayan registrado podrán visualizar en la pantalla principal un listado de winchas disponibles, además al elegir una wincha disponible podrán visualizar información relevante y la distancia entre el usuario y la wincha.
Observación:

TABLA XVIII: Historia de usuario 9: Creación de solicitud de traslado

HISTORIA DE USUARIO	
Número:9	Usuario: Usuario Cliente final
Nombre Historia: Creación de solicitud de traslado.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 5	
Responsables: Gabriel Medina	
Descripción: los usuarios finales quienes utilicen la aplicación móvil pueden solicitar un servicio de traslado, el mismo que será enviado al conductor de la wincha y que este registrado en la aplicación. Al enviar una solicitud los datos del cliente final serán enviados por la aplicación móvil a través de un email.	
Observación:	

Diseño de interfaces

En esta sección se evidencia el diseño de las interfaces de la aplicación móvil, estas fueron prototipadas en base a los requerimientos establecidos.

MODULOS PROVEEDOR



Fig. 23: Pantalla principal

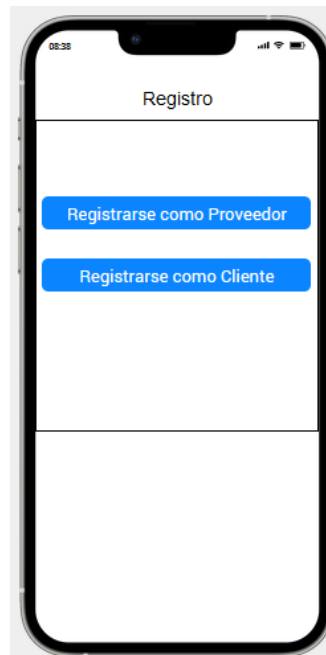


Fig. 24: Tipo de Registro



Fig. 25: Pantalla de Registro de Proveedor



Fig. 26: Pantalla de Registro de Wincha



Fig. 27: Pantalla de Actualización de ubicación



Fig. 28: Pantalla de Registro de precio

Modulo Cliente



Fig. 29: Pantalla de Registro de Cliente

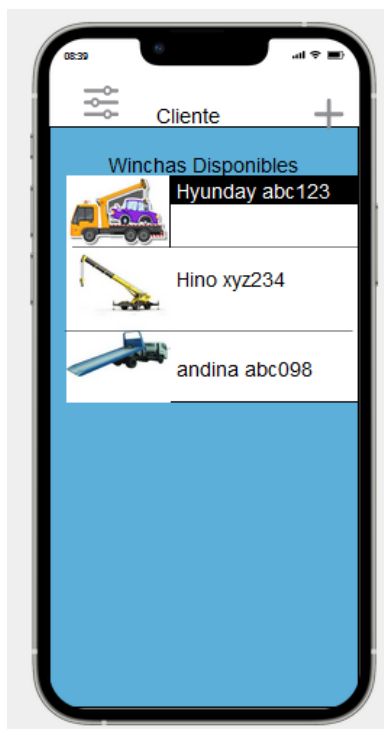


Fig. 30: Pantalla Principal de Cliente



Fig. 31: Pantalla de Información de Perfil del Proveedor



Fig. 32: Pantalla de Visualización de distancia

DISEÑO DE LA BASE DE DATOS

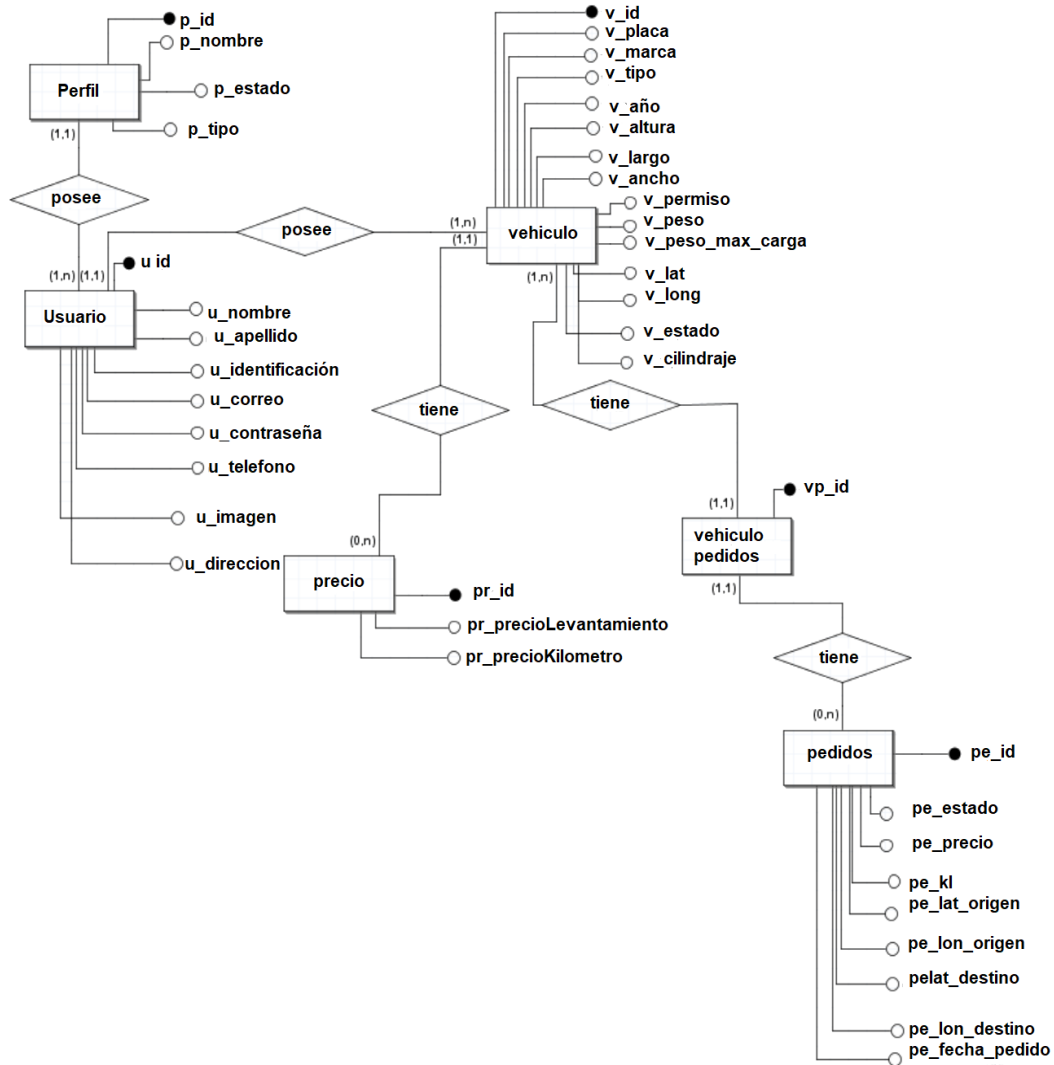


Fig. 33: Base de datos NoSql.

DISEÑO DE ARQUITECTURA

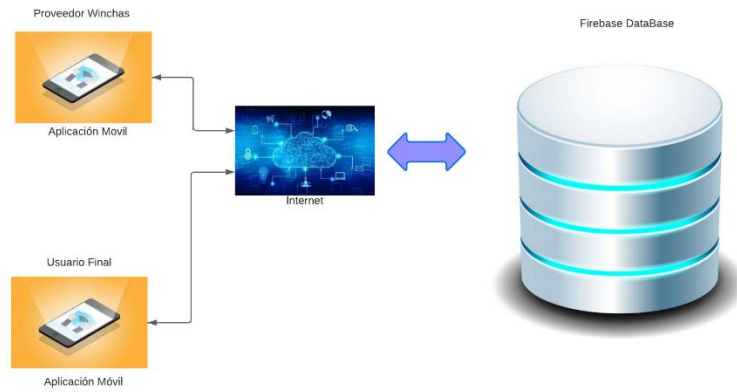


Fig. 34: Diseño Infraestructura.

Pruebas de satisfacción

Para la determinación de la satisfacción de los usuarios, se procedió a realizar encuestas, las mismas que fueron evaluadas. A continuación, las preguntas elaboradas para el usuario.

¿Qué sistema operativo tiene tu celular?

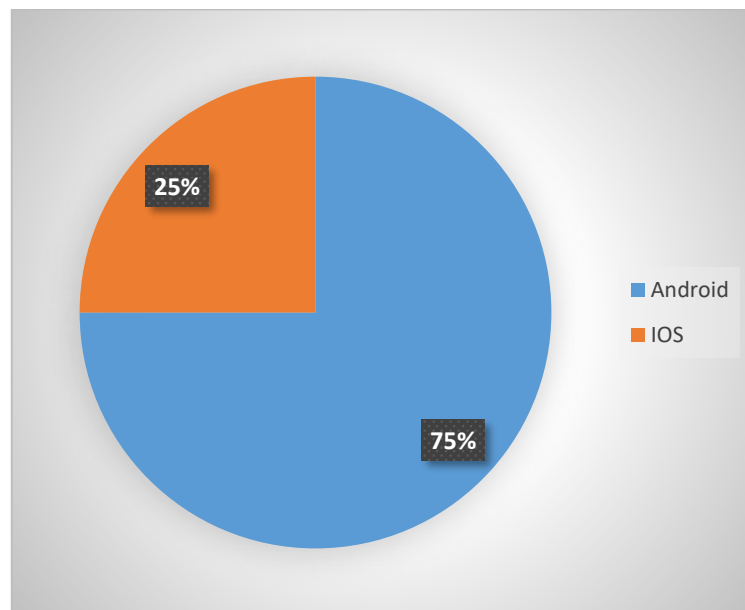


Fig. 35: Sistema operativo más popular

¿Cuál es el grado de satisfacción del aplicativo móvil?

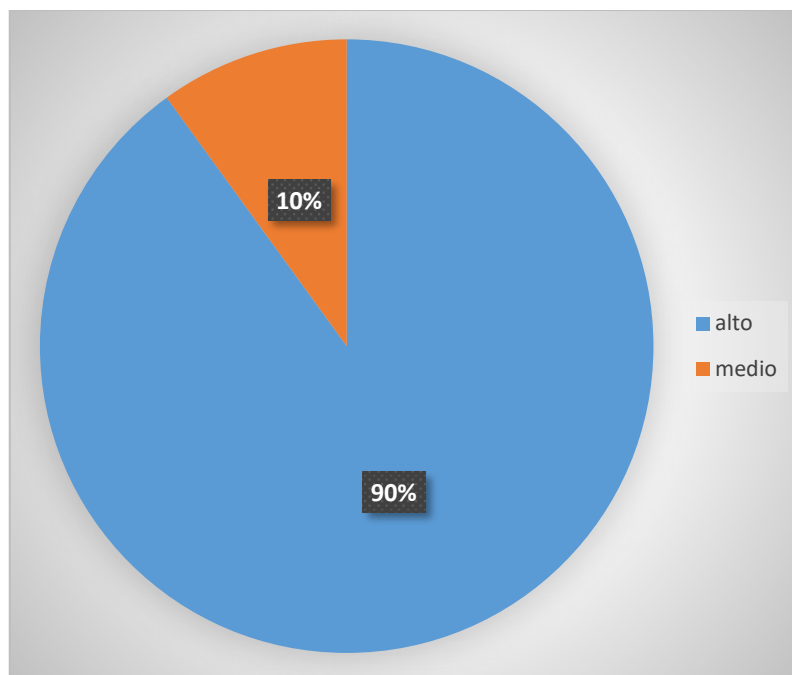


Fig. 36: Satisfacción del aplicativo móvil.

¿Estarías dispuesto a descargar el aplicativo móvil?

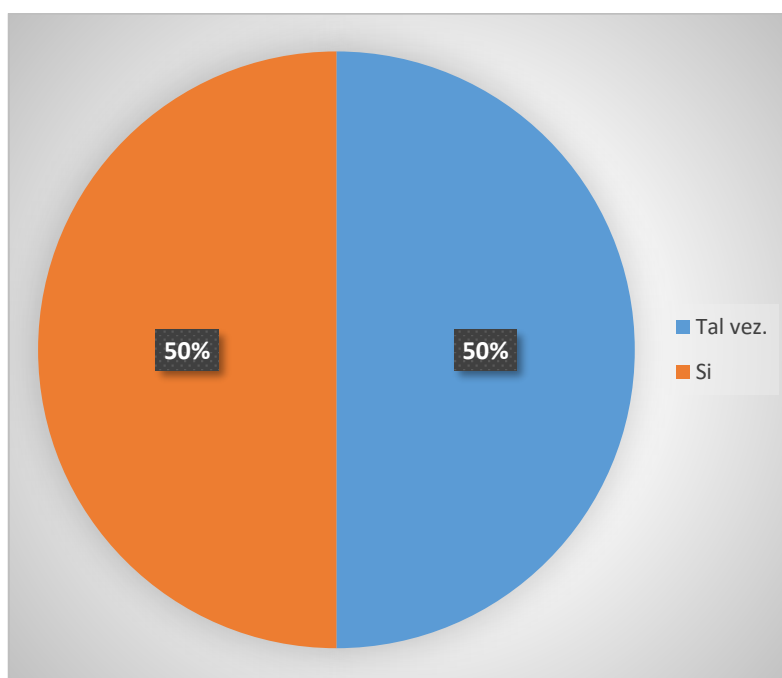


Fig. 37: Disponibilidad de descargar el aplicativo móvil

¿Al compartir tu ubicación crees que esta fue exacta al visualizar en el mapa?

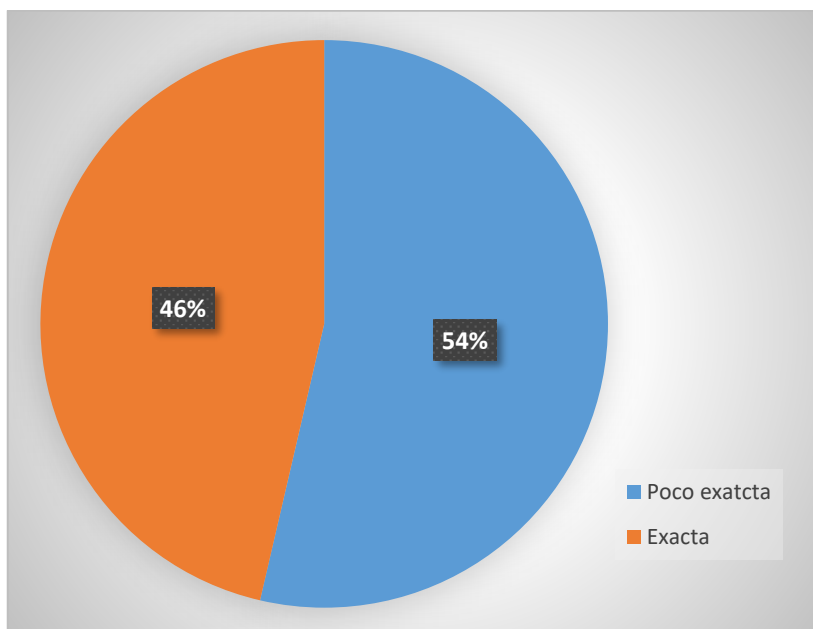


Fig. 38: Exactitud de geolocalización.

¿Te sientes más seguro al observar el perfil del proveedor de servicio?

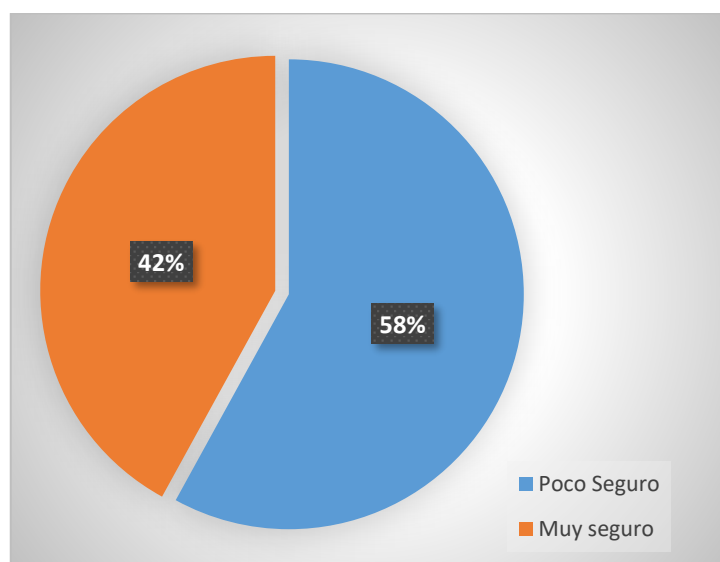


Fig. 39: Seguridad al visualizar perfil del proveedor

¿Como se siente al conocer la distancia entre el usuario final y el proveedor de servicio?

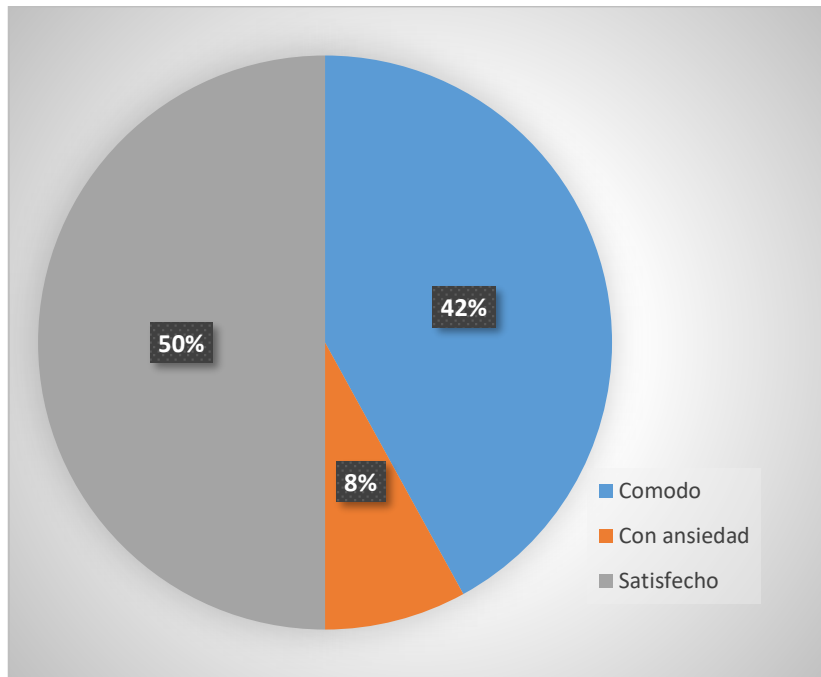


Fig. 40: Satisfacción al visualizar la distancia entre cliente y proveedor

¿Qué otra forma usarías para comunicarte con una persona a través de la aplicación?

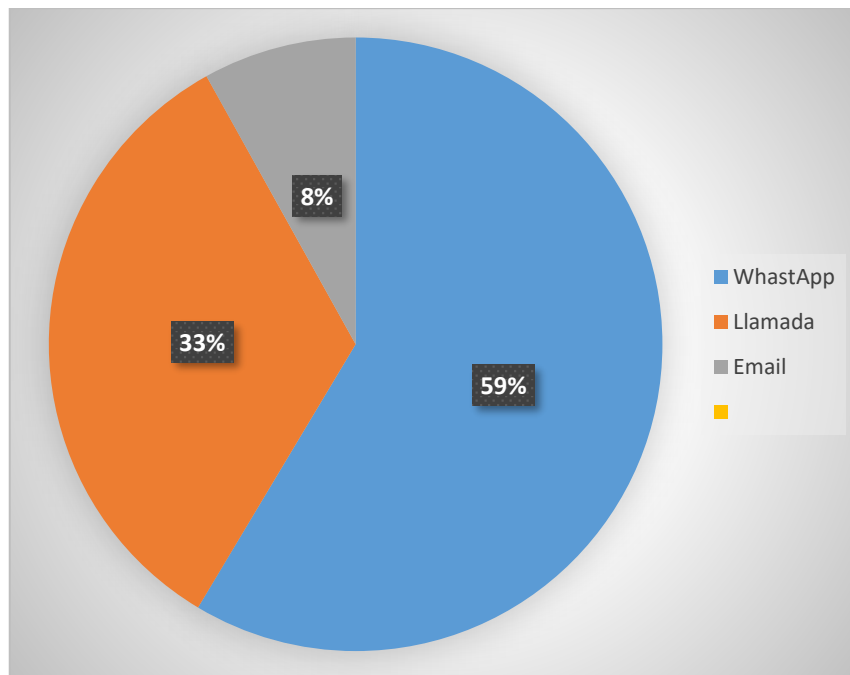


Fig. 41: Formas de comunicación.

¿Qué tan satisfecho te sentirías conocer los precios de servicios de plataformas auto cargables?

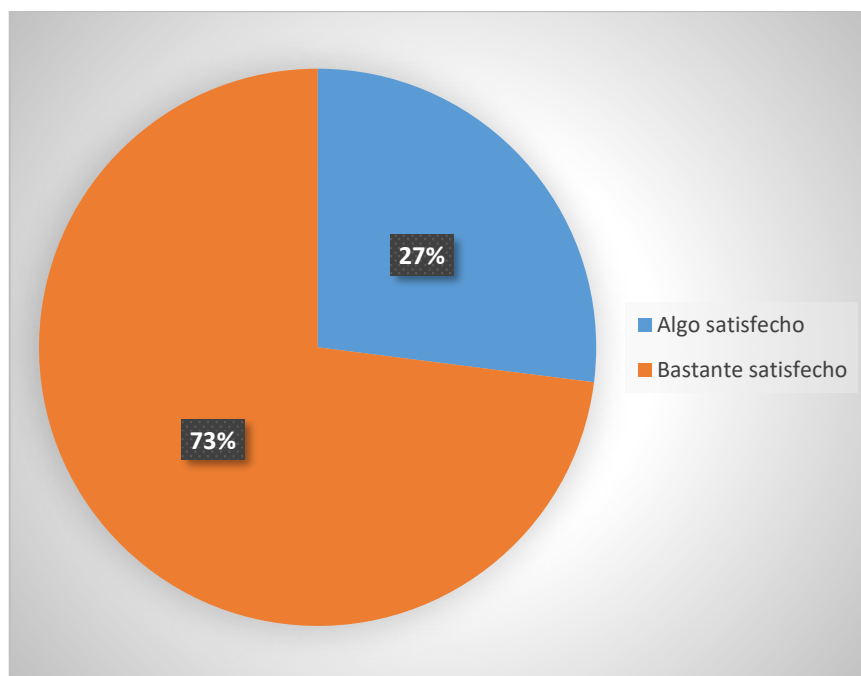


Fig. 42: Satisfacción de precios

Pruebas de compatibilidad

TABLA XIX: Prueba de compatibilidad en dispositivos Android

N°	Dispositivo	Versión	Virtual	Físico
1	Huawei P30 PRO	11.0	X	
2	Samsung A10	9.0.	X	
3	Motorola Moto X	5.0	X	
4	Xiaomi Redmi Note 7	10.0	X	
5	Samsung A30	11.0		X
6	Samsung	8.1.1		X
7	Samsung A50	12		X

PRUEBAS DE ACEPTACIÓN

TABLA XX: Prueba de aceptación 1

Prueba de aceptación
Identificador: PA1
Identificador historia de usuario: 1
Nombre prueba de aceptación: Inicio de sesión
Descripción: Los usuarios deben ingresar llenando el formulario con la información registrada como el usuario y la contraseña.
Pasos de ejecución: Ingreso de datos correspondientes. Dar un toque en Entrar.
Resultado deseado:

Una vez ingresado los datos correspondientes en la aplicación el usuario puede ingresar a la aplicación móvil.

Evaluación de la prueba:

Se comprueba el resultado deseado.

TABLA XXI: Prueba de aceptación 2

Prueba de aceptación
Identificador: PA2
Identificador historia de usuario: 2
Nombre prueba de aceptación: Registro de datos de los diferentes tipos de usuarios. Dar un toque en registrar.
Descripción: Los usuarios deben registrar los datos, llenando un formulario con la información requerida.
Pasos de ejecución: Registro de los datos personales.
Resultado deseado: Una vez registrado en la aplicación el usuario puede ingresar a la aplicación móvil.
Evaluación de la prueba: Se comprueba el resultado deseado.

TABLA XXII: Prueba de aceptación 3

Prueba de aceptación
Identificador: PA3
Identificador historia de usuario: 3
Nombre prueba de aceptación: Registro de datos de la grúa. Dar un toque en registrar.
Descripción: El usuario debe registrar los datos de la grúa llenando un formulario con la información requerida.
Pasos de ejecución: Registro de los datos correspondientes de la grúa.
Resultado deseado: Una vez registrado en la aplicación el usuario puede ingresar a la aplicación móvil esperando la interacción del usuario.
Evaluación de la prueba: Se comprueba el resultado deseado.

TABLA XXIII: Prueba de aceptación 4

Prueba de aceptación
Identificador: PA4
Identificador historia de usuario: 4
Nombre prueba de aceptación: Registro de Precio de servicio.
Descripción:

El usuario debe ingresar a la aplicación móvil y registrar en la grúa los precios de servicios de traslado, estos comprenden el precio por levantamiento y precio por kilómetro.

Pasos de ejecución:

Iniciar sesión.
Esperar que se carguen los datos de la grúa registrada.
Llenar el formulario con precios respectivos.
Dar un toque en registrar.

Resultado deseado:

Una vez ingresado en la aplicación el usuario podrá visualizar un listado con las grúas registradas en la base de datos, escoger la grúa y llenar el formulario de precios y dar un toque en registrar.

Evaluación de la prueba:

Se comprueba el resultado deseado.

TABLA XXIV: Prueba de aceptación 5

Prueba de aceptación
Identificador: PA5
Identificador historia de usuario: 5
Nombre prueba de aceptación: Crear solicitud de traslado
Descripción: El usuario debe ingresar a la aplicación móvil y escoger la grúa de su gusto de la lista de grúas, dar un toque en solicitar.
Pasos de ejecución: Iniciar sesión. Esperar que se carguen los datos de las grúas registradas. Escoger la grúa de su gusto. Dar un toque en solicitar.
Resultado deseado:

Una vez ingresado en la aplicación el usuario podrá visualizar un listado con las grúas registradas en la base de datos, escoger la grúa de su gusto y solicitar el servicio.

Evaluación de la prueba:

Se comprueba el resultado deseado.

ANEXO III

ANEXO III. Manual de usuario

El manual de usuario se encuentra disponible en el siguiente enlace:

https://github.com/GabitoMedina/Manual_de_instalacion.git

ANEXO IV

ANEXO IV. Manual de instalación

En el siguiente enlace se podrá descargar la apk de la aplicación:

https://github.com/GabitoMedina/APK_Ionic-winchas.git

En el siguiente enlace encontrara un video explicativo del Manual de Instalación:

<https://www.youtube.com/watch?v=wHMhGNLTiC>