



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA
E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS

Tema:

DESARROLLO DE UNA APLICACIÓN MOVIL MULTIPLATAFORMA PARA
GESTIONAR AGENTES MOTORIZADOS EN LA EMPRESA DE
ENCOMIENDAS GABITOS EXPRESS.

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo la
obtención del título de Ingeniera en Sistemas Computacionales e Informáticos.

ÁREA: Software

LINEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Sylvia Inés Guanoluisa Taco

TUTOR: Ing. David Omar Guevara Mg.

Ambato - Ecuador

marzo - 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: DESARROLLO DE UNA APLICACIÓN MOVIL MULTIPLATAFORMA PARA GESTIONAR AGENTES MOTORIZADOS EN LA EMPRESA DE ENCOMIENDAS GABITOS EXPRESS, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Sylvia Inés Guanoluisa Taco, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2022.

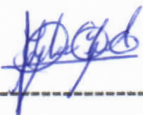
Ing. David Omar Guevara Mg.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado DESARROLLO DE UNA APLICACIÓN MOVIL MULTIPLATAFORMA PARA GESTIONAR AGENTES MOTORIZADOS EN LA EMPRESA DE ENCOMIENDAS GABITOS EXPRESS, es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2022.



Sylvia Inés Guanoluisa Taco

C.C. 1804404315

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por la señorita Sylvia Inés Guanoluisa Taco, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **DESARROLLO DE UNA APLICACIÓN MOVIL MULTIPLATAFORMA PARA GESTIONAR AGENTES MOTORIZADOS EN LA EMPRESA DE ENCOMIENDAS GABITOS EXPRESS**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, marzo 2022.

Ing. Pilar Urrutia, Mg.

PRESIDENTA DEL TRIBUNAL

Ing. Edison Alvarez

PROFESOR CALIFICADOR

PhD. Julio Balarezo

PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2022.



Sylvia Inés Guanoluisa Taco

C.C. 1804404315

AUTOR

DEDICATORIA

Dedico este trabajo primero a Dios que es fundamental en mi vida, a mi madre y hermano por motivarme para culminar mis estudios.

Sylvia

AGRADECIMIENTO

Agradezco a Dios por permitirme cumplir este sueño junto a mi madre y mi hermano.

A mi madre Blanquita, a mi hermano Yulais y a mi amiga Lety, por su apoyo incondicional.

A todos los docente de la FISEI por guiar este arduo camino con paciencia y dedicación para convertirme en una profesional con bases necesarias y ser una ayuda en la sociedad, aportando mis habilidades y conocimientos adquiridos.

Sylvia

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
APROBACIÓN TRIBUNAL DE GRADO	iv
DERECHOS DE AUTOR.....	v
DEDICATORIA	vi
AGRADECIMIENTO.....	vii
ÍNDICE DE TABLAS	xi
INDICE DE FIGURAS.....	xiv
RESUMEN.....	xvi
ABSTRACT	xvii
CAPITULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de Investigación	1
1.2 Antecedentes Investigativos.....	1
1.2.1 Contextualización del problema.....	1
1.2.2 Fundamentación Teórica.....	3
1.2.2.1 Dispositivo móvil.....	3
1.2.2.2 Aplicación móvil (App)	3
1.2.2.2.1 Aplicación nativa	3
1.2.2.2.2 Aplicación Web.....	4
1.2.2.2.3 Aplicación híbrida.....	4
1.2.2.2.4 Comparación de tipos de aplicación móvil	4
1.2.2.3 Framework desarrollo multiplataforma.....	5
1.2.2.3.1 Ionic.....	6
1.2.2.3.2 React Navite.....	6
1.2.2.3.3 Comparación de Frameworks	7
1.2.2.4 Herramientas Web.....	8
1.2.2.4.1 Node.js	8
1.2.2.4.2 HTML5	8
1.2.2.4.3 Capacitor	8
1.2.2.4.4 Angular.....	9
1.2.2.4.5 API REST.....	9
1.2.2.4.7 Push Notifications	10

1.2.2.5 Geolocalización.....	10
1.2.2.6 Sistema de Posicionamiento Global (GPS).....	10
1.2.2.7 API de Geolocalización.....	11
1.2.2.8 Google Maps Platform	11
1.2.2.8.1 Google Maps Platform – Maps	11
1.2.2.8.2 Google Maps Platform – Routes.....	12
1.2.2.8.3 Google Maps Platform – Places.....	12
1.2.2.9 Hosting	12
1.2.2.10 Base de Datos.....	13
1.2.2.10.1 Base de Datos Relacional.....	13
1.2.2.10.2 Base de Datos No relacional	13
1.2.2.10.3 Comparación de base de datos	14
1.2.2.10.4 Mongodb	15
1.2.2.10.5 Google Cloud Firestore	15
1.3 Objetivos	16
1.3.1 Objetivo General	16
1.3.2 Objetivos Específicos:.....	16
CAPITULO II	17
METODOLOGIA	17
2.1 Materiales	17
2.2 Métodos	17
2.2.1 Modalidad de Investigación.....	17
2.2.2 Población y Muestra	17
2.2.3 Recolección de la Información	19
2.2.4 Procesamiento y Análisis de Datos.....	19
2.2.4.1 Resultados de la encuesta.....	20
CAPÍTULO III.....	21
RESULTADOS Y DISCUSIÓN.....	21
3.1 Análisis y discusión de los resultados.	21
3.1.1 Metodologías de desarrollo de software	21
3.1.1.1 Metodologías tradicionales	21
3.1.1.2 Metodologías ágiles	21
3.1.1.3 Metodología Scrum.....	22
3.1.1.4 Metodología Kanban.....	24

3.1.15 Metodología Extreme Programming (XP)	24
3.1.1.6 Comparación de Metodologías ágiles	26
3.1.2 Tecnologías de desarrollo de Back-end.....	27
3.1.2.1 Firebase	27
3.1.2.1.1 Servicios de Firebase.....	27
3.1.2.2 Node Js.....	28
3.1.2.3 Comparación de tecnologías back-end.....	29
3.1.3 Desarrollo de la propuesta	29
3.1.3.1 Fase I: Planificación.....	29
3.1.3.1.1 Resultado de la entrevista.....	30
3.1.3.1.2 Definición de roles	30
3.1.3.1.3 Historias de Usuario	30
3.1.3.1.4 Tareas	40
3.1.3.1.5 Estimación de Historias de Usuario	55
3.1.3.1.6 Plan de Entrega	57
3.1.3.2 Fase II: Diseño	58
3.1.3.2.1 Arquitectura de la aplicación	58
3.1.3.2.2 Diseño de la base de datos	62
3.1.3.2.3 Tarjetas CRC (Clase-Responsabilidad-Colaboración).....	65
3.1.3.2.4 Diseño de Interfaces.....	68
3.1.3.3 Fase III: Codificación.....	74
3.1.3.3.1 Modelos.....	74
3.1.3.3.2 Servicios.....	75
3.1.3.3.3 Métodos aplicación móvil	81
3.1.3.3.4 Functions (Funciones de firebase)	83
3.1.3.3.5 Desplegar el código en hosting firebase.....	84
3.1.3.4 Fase IV: Pruebas	86
3.1.3.4.1 Pruebas de Aceptación.....	86
CAPITULO IV	94
CONCLUSIONES Y RECOMENDACIONES.....	94
4.1 Conclusiones.....	94
4.2 Recomendaciones	95
Bibliografía	96

ÍNDICE DE TABLAS

Tabla 1. Comparación de tipos de aplicación móvil	4
Tabla 2. Comparación de Frameworks	7
Tabla 3. Comparación de base de datos	14
Tabla 4. Población a investigarse.....	18
Tabla 5. Comparación de Metodologías de Desarrollo de Software	22
Tabla 6. Comparación de Metodologías ágiles	26
Tabla 7. Comparación de tecnologías back-end	29
Tabla 8. Definición de roles	30
Tabla 9. Plantilla de Historia de Usuario	31
Tabla 10. Historia de usuario Pantalla de inicio	32
Tabla 11. Historia de usuario Iniciar sesión.....	32
Tabla 12. Historia de usuario Recuperar clave	32
Tabla 13. Historia de usuario Registrar cliente.....	33
Tabla 14. Historia de usuario Actualizar perfil cliente	33
Tabla 15. Historia de usuario Registrar motorizado	33
Tabla 16. Historia de usuario Dar de baja motorizado.....	34
Tabla 17. Historia de usuario Mostrar motorizados.....	34
Tabla 18. Historia de usuario CRUD productos	34
Tabla 19. Historia de usuario Actualizar perfil motorizado.....	35
Tabla 20. Historia de usuario Mostrar productos (Tienda)	35
Tabla 21. Historia de usuario Mostrar productos seleccionados (carrito)	35
Tabla 22. Historia de usuario Ingresar descripción producto pedido.....	36
Tabla 23. Historia de usuario Mostrar pedidos	36
Tabla 24. Historia de usuario Asignar motorizado	36
Tabla 25. Historia de usuario Mostrar pedidos asignados	37
Tabla 26. Historia de usuario Actualizar estado de pedido.....	37
Tabla 27. Historia de usuario Mostar ubicación motorizado	37
Tabla 28. Historia de usuario Recorrido diario.....	38
Tabla 29. Historia de usuario Calificar Motorizado.....	38
Tabla 30. Historia de usuario Reporte pedidos	38
Tabla 31. Historia de usuario Reporte motorizados.....	39
Tabla 32. Historia de usuario Mostrar menú.....	39
Tabla 33. Historia de usuario Desplegar la aplicación en el hosting	39
Tabla 34. Diseño de las interfaces para pantalla de inicio	40
Tabla 35. Codificar interfaces	40
Tabla 36. Codificar servicio firestore.....	40
Tabla 37. Diseño de interfaz de usuario para iniciar sesión.....	41
Tabla 38. Implementación de métodos de autenticación.....	41
Tabla 39. Implementación de métodos de seguridad	41
Tabla 40. Diseño interfaz de usuario para recuperar clave	41
Tabla 41. Implementación del servicio para recuperar clave.....	42
Tabla 42. Diseño interfaz de usuario para registro cliente.....	42
Tabla 43. Diseño interfaz de usuario para registro cliente.....	42

Tabla 44. Implementación del servicio de google maps	43
Tabla 45. Implementación del servicio de creación de clientes	43
Tabla 46. Diseño de interfaz para actualizar perfil	43
Tabla 47. Implementación del servicio para actualizar perfil cliente	43
Tabla 48. Diseño de interfaz de usuario para registrar motorizado	44
Tabla 49. Implementación del servicio de subida de imágenes al servidor	44
Tabla 50. Implementación del servicio para registrar motorizados	44
Tabla 51. Diseño de interfaz de usuario para dar de baja motorizado	45
Tabla 52. Implementación del servicio para dar de baja motorizado	45
Tabla 53. Diseño de interfaz para mostrar motorizados	45
Tabla 54. Implementación del servicio para mostrar motorizados	45
Tabla 55. Diseño de interfaz para CRUD productos.....	46
Tabla 56. Implementación de los servicios para CRUD productos	46
Tabla 57. Diseño de interfaz para actualizar perfil motorizado	46
Tabla 58. Implementación del servicio para actualizar perfil motorizado	47
Tabla 59. Diseño de interfaz para mostrar productos (Tienda).....	47
Tabla 60. Implementación del servicio para mostrar productos	47
Tabla 61. Diseño de interfaz para mostrar productos seleccionados (carrito)	48
Tabla 62. Implementación del servicio para guardar el pedido.	48
Tabla 63. Diseño de interfaz para ingresar descripción pedido producto	48
Tabla 64. Implementación del servicio para ingresar descripción pedido producto. .	48
Tabla 65. Diseño de interfaz para mostrar pedidos.....	49
Tabla 66. Implementación del servicio para mostrar pedidos.....	49
Tabla 67. Diseño de interfaz para asignar motorizado.....	49
Tabla 68. Implementación del servicio para asignar motorizado	50
Tabla 69. Diseño de interfaz para mostrar pedidos asignados	50
Tabla 70. Implementación del servicio para mostrar pedidos asignados.....	50
Tabla 71. Diseño de interfaz para actualizar estado pedido.....	51
Tabla 72. Implementación del servicio para actualizar estado pedido.....	51
Tabla 73. Implementación del servicio de notificaciones	51
Tabla 74. Mostar ubicación motorizado.....	51
Tabla 75. Implementación del servicio para mostrar ubicación motorizado	52
Tabla 76. Diseño de interfaz para mostrar recorrido diario del motorizado	52
Tabla 77. Implementación del servicio para mostrar recorrido diario del motorizado	52
Tabla 78. Diseño de interfaz para calificar motorizado	53
Tabla 79. Implementación del servicio para calificar motorizado	53
Tabla 80. Diseño de interfaz para mostrar reporte pedidos	53
Tabla 81. Implementación del servicio para mostrar reporte pedidos	54
Tabla 82. Diseño de interfaz para mostrar reporte motorizados	54
Tabla 83. Implementación del servicio para mostrar reporte motorizados	54
Tabla 84. Diseño de interfaz para mostrar menú	54
Tabla 85. Implementación del menú.....	55
Tabla 86. Desplegar la aplicación en el hosting de firebase	55
Tabla 87. Estimación de Historias de usuario.....	56
Tabla 88. Plan de Entrega	57

Tabla 89. Tarjeta CRC - Ubicación.....	66
Tabla 90. Tarjetas CRC - Cliente.....	66
Tabla 91. Tarjeta CRC - Producto.....	66
Tabla 92. Tarjeta CRC - Autenticación.....	66
Tabla 93. Tarjeta CRC - Carrito.....	67
Tabla 94. Tarjeta CRC - Pedido.....	67
Tabla 95. Tarjeta CRC - Motorizado.....	67
Tabla 96. Tarjeta CRC – Calificación.....	67
Tabla 97. Prueba de inicio de sección.....	86
Tabla 98. Prueba de registrar cliente.....	86
Tabla 99. Prueba de registrar motorizado.....	87
Tabla 100. Prueba de CRUD productos.....	87
Tabla 101. Prueba de mostrar productos. (Tienda).....	87
Tabla 102. Prueba de mostrar productos seleccionados (carrito).....	88
Tabla 103. Prueba de registrar descripción producto pedido.....	88
Tabla 104. Prueba de mostrar pedidos.....	89
Tabla 105. Prueba de mostrar ubicación motorizado.....	89
Tabla 106. Prueba de asignar motorizado.....	89
Tabla 107. Prueba de mostrar pedidos asignados.....	90
Tabla 108. Prueba de actualizar estado de pedido.....	90
Tabla 109. Prueba de mostrar motorizados.....	90
Tabla 110. Prueba de actualizar perfil cliente.....	91
Tabla 111. Prueba de actualizar perfil motorizado.....	91
Tabla 112. Prueba de calificar motorizado.....	91
Tabla 113. Prueba de recorrido diario.....	92
Tabla 114. Prueba de reporte pedidos.....	92
Tabla 115. Prueba de reporte motorizados.....	92
Tabla 116. Prueba Mostrar menú.....	93
Tabla 117. Prueba dar de baja motorizado.....	93
Tabla 118. Prueba recuperar clave.....	93

INDICE DE FIGURAS

Figura 1. Arquitectura de la aplicación.....	58
Figura 2. Node.js.....	59
Figura 3. Visual Studio Code.....	59
Figura 4. Instalar ionic global.....	59
Figura 5. Crear el proyecto.....	60
Figura 6. Correr el proyecto.....	60
Figura 7. Realizar un build a la aplicación.....	60
Figura 8. Agregar a la plataforma Android.....	60
Figura 9. Agregar a la plataforma IOS.....	60
Figura 10. Estructura del Proyecto Ionic.....	61
Figura 11. Colección de clientes.....	63
Figura 12. Colección de motorizados.....	63
Figura 13. Colección de productos.....	64
Figura 14. Sub-colección de pedidos.....	64
Figura 15. Sub-colección de carrito.....	65
Figura 16. Reglas firebase.....	65
Figura 17. Interfaz de inicio de la aplicación.....	68
Figura 18. Interfaz para iniciar sesión.....	68
Figura 19. Interfaz para registrar el cliente.....	69
Figura 20. Interfaz para registrar motorizado.....	69
Figura 21. Interfaz para mostrar productos.....	70
Figura 22. Interfaz del carro de compras.....	70
Figura 23. Interfaz para el CRUD Productos.....	71
Figura 24. Interfaz para visualizar la ubicación del motorizado.....	71
Figura 25. Interfaz para recuperar clave.....	72
Figura 26. Interfaz para asignar el motorizado.....	72
Figura 27. Interfaz para mostrar los pedidos.....	73
Figura 28. Interfaz de pedidos asignados.....	73
Figura 29. Interfaz de Menú.....	74
Figura 30. Código modelos.....	75
Figura 31. Servicio firestore.....	76
Figura 32. Servicio Firestorage.....	77
Figura 33. Servicio FirebaseAuth.....	77
Figura 34. Servicio Carrito.....	78
Figura 35. Servicio Googlemaps.....	79
Figura 36. Servicio motorizados.....	80
Figura 37. Servicio Notificaciones.....	81
Figura 38. Método para mostrar pedidos.....	81
Figura 39. Método de autenticación.....	82
Figura 40. Método de obtener ubicación.....	83
Figura 41. Funciones de firebase.....	84
Figura 42. Instalar herramienta angular PWA.....	84
Figura 43. Login en Firebase.....	85

Figura 44. Ejecutar firebase init	85
Figura 45. Despliegue hosting firebase	85
Figura 46. Gabitos Express publicado	85
Figura 47. Habilitar la persistencia	85

RESUMEN EJECUTIVO

Actualmente, las aplicaciones móviles tienen gran acogida a nivel mundial, por su facilidad de uso y acceso, destinados para diferentes actividades y en constante desarrollo. Por tal motivo, se han convertido en herramientas necesarias e indispensables para la vida diaria o empresarial de una persona.

El presente proyecto de investigación detalla el desarrollo de una aplicación móvil multiplataforma para gestionar a los agentes motorizados en base a los pedidos, así como la disponibilidad y ubicación a través de la geolocalización, además permite visualizar las notificaciones después de cada proceso, con el objetivo de mejorar el servicio que ofrece la Empresa de Encomiendas Gabitos Express.

El proyecto está desarrollado mediante la Metodología XP, apta para el desarrollo de proyectos pequeños de poca duración, y su entorno de desarrollo con la tecnología Ionic en su versión 5, al igual que Angular para el lado del cliente, Capacitor se utilizó para acceder a funciones nativas Android e iOS, por otro lado, se utilizó la tecnología Firebase en el lado del servidor; aportando servicios extras como Firestore Database utilizada como base de datos NoSql, Storage como servicio de almacenamiento y su hosting para el despliegue de una aplicación PWA. Los recursos utilizados para la visualización del mapa y geolocalización fue mediante la implementación de API's que ofrece google maps.

Palabra Clave: Ionic, geolocalización, angular, firebase, API's, capacitor

ABSTRACT

Currently, the mobile applications are very well received worldwide, for their ease of use and access, intended for different activities and in constant development. For this reason, they have become necessary and indispensable tools for the daily or business life of a person

This research project details the development of a multiplatform mobile application to manage motorized agents based on orders, as well as availability and location through geolocation, as well as allowing notifications to be displayed after each process, with the objective to improve the service offered by the Gabitos Express Parcel Company.

The project is developed using the XP Methodology, suitable for the development of small projects of short duration, and its development environment with Ionic technology in its version 5, as well as Angular for the client side, Capacitor was used to access native Android and iOS functions, on the other hand, Firebase technology was used on the server side; providing extra services such as Firestore Database used as a NoSql database, Storage as a storage service and its hosting for the deployment of a PWA application. The resources used for the visualization of the map and geolocation were through the implementation of APIs offered by google maps.

Keywords: Ionic, geolocation, angular, firebase, API's, capacitor

CAPITULO I

MARCO TEÓRICO

1.1 Tema de Investigación

DESARROLLO DE UNA APLICACIÓN MOVIL MULTIPLATAFORMA PARA GESTIONAR AGENTES MOTORIZADOS EN LA EMPRESA DE ENCOMIENDAS “GABITOS EXPRESS”.

1.2 Antecedentes Investigativos

1.2.1 Contextualización del problema

En los últimos años, el incremento del uso de dispositivos móviles al igual que las aplicaciones se han convertido en herramientas indispensables para las personas en su vida cotidiana, lo que conduce a desarrollar aplicaciones móviles con distintos y nuevos frameworks basados en diversos lenguajes de programación, cuyo objetivo es acelerar el proceso de desarrollo y la reutilización de código ya existente.

La expectativa consiste en que sea ejecutable en la mayor cantidad de dispositivos, es decir, que sea multiplataforma, combinado lo mejor de la programación nativa y web, esto brindara un entorno confiable para el desarrollo de aplicaciones capaces de usar de forma eficiente los recursos del dispositivo y este sea utilizado en el mayor número de plataformas. Reduciendo el tiempo y costo por mantenimiento o crecimiento del software desarrollado.

Uno de los frameworks más utilizado para el desarrollo de aplicaciones móviles multiplataforma actualmente es IONIC, caracterizado por utilizar HTML5 (Hyper Text Markup Language) como lenguaje de programación para las vistas de las aplicaciones front-end, se puede integrar marcos como Angular, React o Vue para la programación de controladores y funcionalidades, lo cual permite crear aplicaciones rápidas y altamente interactivas.

Según el Instituto Nacional de Estadística y Censos (INEC), Ecuador registra un porcentaje de 76.8% de personas que tienen un teléfono inteligente en el periodo fiscal 2019, esto indica que la probabilidad de aceptación del desarrollo de aplicaciones móviles sea cada vez más considerable, el mismo que abre paso al desarrollo e implementación de estas aplicaciones móviles en diferentes áreas como salud, educación y a nivel empresarial, identificando necesidades para mejorar la productividad en cada uno de los procesos a realizar [1].

En la Provincia de Tungurahua, cantón Ambato existen pocas empresas que brindan el servicio de encomiendas, siendo la Empresa de Encomiendas Gabitos Express una de ellas, la empresa realiza todo el proceso de forma manual, llevando un registro en papel y apoyándose con la aplicación Whatsapp, mensajes de texto o por llamadas telefónicas para registrar la encomienda y de la misma manera la coordinación y gestión de los agentes motorizados, provocando retraso en la entrega de las encomiendas o en muchas ocasiones asignar el mismo pedido a varios motorizados, todo ello, por desconocer si el motorizado está disponible, si mantiene encomiendas asignadas o si ya se encuentra en proceso de entrega, ya que en casos extremos no se puede localizar a los agentes motorizados, por tanto se ha considerado oportuno desarrollar una aplicación móvil multiplataforma que cumpla con estos requerimientos, agilizando y optimizando el trabajo requerido.

Por lo antes mencionado, es necesario dar una solución al problema establecido mediante el desarrollo de un prototipo de aplicación móvil multiplataforma en la empresa de encomiendas “Gabitos Express”. El aplicativo brindará las facilidades para la gestión de los agentes motorizados en la entrega de encomiendas, además de adaptarse a cualquier entorno de trabajo sin importar la plataforma en la que se utilice, la aplicación brindaría un gran apoyo en este proceso que es un eje fundamental en el desarrollo de la empresa.

1.2.2 Fundamentación Teórica

1.2.2.1 Dispositivo móvil

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, que posee varias funciones, entre ellas podemos mencionar, el procesamiento e intercambio de información, la conexión a alguna red, además se puede instalar diversas aplicaciones móviles según la necesidad, todo esto a través de una memoria interna e ilimitada.

Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea (sms), agenda, juegos, cámara fotográfica, acceso a Internet, reproducción de video y mp3 e incluso GPS [2].

1.2.2.2 Aplicación móvil (App)

Una aplicación móvil o app (en inglés) es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tablets y otros dispositivos móviles. Generalmente se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry 10, entre otros [3].

1.2.2.2.1 Aplicación nativa

Una aplicación nativa es aquella que han sido desarrollada con el software que ofrece cada sistema operativo de las compañías propietarias, llamado genéricamente Software Development Kit o SDK [4].

También ofrecen un diseño acorde al sistema operativo en el que se ejecuta, tomando la librería de gráficos propios de cada plataforma. Ofreciendo una mejor experiencia de usuario ya que estos están acostumbrados a la interfaz de su sistema operativo
Aplicación Web [5].

1.2.2.2 Aplicación Web

Una aplicación web está desarrollada mayormente con HTML y CSS, con el objetivo de ser ejecutada en cualquier navegador de Internet y en cualquier dispositivo móvil o pc, sin ser instalada, debido a que los archivos de configuración, así como los datos de la aplicación se guardan en la nube llamados servidores web (hosting).

Es imprescindible el internet para hacer uso de estas aplicaciones al igual que una URL (Localizador de Recursos Uniforme) para acceder a ella, al no estar instalada la aplicación en el dispositivo móvil no se puede aprovechar al máximo los componentes del mismo [6].

1.2.2.3 Aplicación híbrida

Se denomina aplicación híbrida a la combinación entre aplicación web y aplicación móvil nativa. Su desarrollo es parecido a una aplicación web usando HTML, CSS (Cascading Style Sheets) y JavaScript, cuando ya está terminada se procede a compilar obteniendo algo similar a una aplicación nativa. Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS, y distribuir las en cada una de sus tiendas [4].

Las aplicaciones híbridas, encapsulan la misma mediante el webkit de cada plataforma nativa; logrando publicarlas para Android en la tienda Google Play Store y para iOS en App Store, sin ningún problema. Además, este tipo de aplicación permite integrar API's de cada plataforma de manera simple, accediendo a funcionalidades como notificaciones push, cámara, GPS, sensores, entre otras, propias de las aplicaciones nativas [7].

1.2.2.4 Comparación de tipos de aplicación móvil

Tabla 1. Comparación de tipos de aplicación móvil

Aplicación	Nativa	Híbrida	Web
Lenguaje de desarrollo	Net, Java, C, Objective C, Swift	Html, Css, Javascript	Html, Css, Javascript

Tiempo de desarrollo	Alto: Es necesario desarrollar una app para cada sistema operativo	Bajo: Se desarrolla una sola vez.	Bajo: Se desarrolla una sola vez.
Rendimiento	Optimo	Optimo a excepción de aplicaciones que utilizan mucho hardware, como juegos.	Baja, Su desempeño depende de la velocidad de internet.
Accedo a recursos del dispositivo	Completo	Prácticamente completo.	Pueden acceder a pocos componentes.
Costo	Alto costo	Bajo el costo	Bajo Costo
Interfaz de usuario	Optima	Optima	Baja
Publicación App Store	SI	SI	NO

Elaborado por: El Investigador

En la tabla 1. Se analizó las características expuestas de cada tipo de aplicación móvil, con el fin de justificar la elección de la aplicación híbrida, considerando como la mejor opción ya que satisface los requerimientos de la empresa Gabito's Express, a bajo costo y tiempo de desarrollo.

1.2.2.3 Framework desarrollo multiplataforma

Se conoce como framework de desarrollo multiplataforma a un esquema de reutilización de software conformado por componentes y relaciones entre estos, por ejemplo: la abstracción de clases, objetos o componentes que la conforman; además, provee diferentes componentes de conexión a base de datos relacionales o no relacionales [8].

Actualmente, existen muchos frameworks para el desarrollo de aplicaciones móvil multiplataforma destinados a videojuegos, realidad aumentada, inteligencia artificial, aplicaciones en tiempo real, geolocalización, etc.

Debido a la simplicidad, eficiencia, seguridad y bajo costo, que un Framework basado en JavaScript, HTML5 y CSS ofrece a los desarrolladores, Frameworks como IONIC

y React Native, tienen una acogida gran escala para el desarrollo de aplicaciones multiplataforma.

1.2.2.3.1 Ionic

Ionic es un framework de código abierto, diseñado para integrarse a la perfección con los mejores marcos como Angular, React, Vue o incluso ningún marco con JavaScript vanilla. Los componentes de Ionic están escritos en HTML, CSS y JavaScript, lo que facilita la creación de interfaces de usuario modernas y de alta calidad [9].

Ionic proporciona el entorno necesario para desarrollar aplicaciones híbridas, mediante Capacitor, una herramienta fundamental para implementar en dispositivos móviles o en una pc como aplicación web de forma nativa, con un solo código [10].

Además, Ionic trabaja bajo el concepto MVC (Modelo-Vista-Controlador), un patrón de diseño que separa los datos, la lógica y las interfaces de usuario; cuenta con información bien documentada y con una comunidad muy activa.

No se recomienda utilizar el framework para el desarrollo de video juegos u otras aplicaciones que utilicen cantidad de recursos, ya que presentará bajo rendimiento a diferencia de una aplicación nativa [9].

1.2.2.3.2 React Native

React Native de Facebook ha causado una gran sensación en la industria del desarrollo de aplicaciones móviles. Es un marco JavaScript completo que permite a los desarrolladores crear software multiplataforma, similar a las aplicaciones nativas; se basa en la librería de JavaScript React para la creación de componentes visuales, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas, en este caso iOS y Android.

Es decir, en lugar de desarrollar una aplicación web híbrida o en HTML5, lo que se obtiene como resultado es una aplicación real nativa, indistinguible que se podrá desarrollar el código en Objective-C o Java.

React Native usa el mismo paradigma fundamental de construcción de bloques de UI (componentes visuales con los que interacciona el usuario) que las aplicaciones nativas reales de Android e iOS, pero gestiona la interacción entre los mismos utilizando las capacidades de JavaScript y React. Además, React Native ofrece una experiencia de desarrollo similar al de la Web, como es el caso de la recarga en tiempo real de su aplicación durante el desarrollo, algo que es muy agradable [11].

1.2.2.3 Comparación de Frameworks

Tabla 2. Comparación de Frameworks

Aplicación	IONIC	React Native
Categoría	Web Híbrido	Nativo Híbrido
Multiplataforma	Si	Si
Reutilización de código	Si: Una base de código, la base de código de la interfaz de usuario permanece igual.	No: diferentes bases de código de interfaz de usuario mientras más sea más personalizable.
Plataformas compatibles	Android, iOS y aplicación web progresiva.	Android, iOS y aplicación web progresiva.
Inversión	Mínima inversión en personal y tiempo	Mediana inversión en personal y tiempo
Acceso a API / Funciones nativas	Si: a través de complementos	Si: a través de complementos
Comunidad	Extensa	Mediana

Elaborado por: El Investigador

En la tabla 2. Se analizó varias características con el fin de justificar la elección del framework Ionic, donde se destaca la reutilización de código, mismo que facilita personalizar la interfaz de usuario libremente con una sola base de código, a diferencia de React Native mientras más se desee personalizar se trabajará con tres bases de códigos separadas, dos para la administración de interfaz y una para el controlador.

1.2.2.4 Herramientas Web

1.2.2.4.1 Node.js

Node.js es básicamente un framework para implementar operaciones de entrada y salida. Está basado en eventos, streams y construido encima del motor de Javascript V8, que es con el que funciona el Javascript de Google Chrome.

Node.js puede soportar miles de conexiones al mismo tiempo mediante el bucle de eventos, al ser un framework de código abierto con un lenguaje de programación Javascript [12].

1.2.2.4.2 HTML5

Se conoce como HTML5 a la nueva versión del lenguaje que se usa para estructurar páginas web, que surge como una evolución lógica de las especificaciones anteriores con los siguientes objetivos: separar totalmente la información, y la forma de presentarla, resumir, simplificar y hacer más sencillo el código utilizado, incorporar nuevas etiquetas semánticas, páginas compatibles con todos los navegadores web incluyendo los de los teléfonos móviles y otros dispositivos utilizados en la actualidad para navegar en Internet [13].

Dado que Ionic es un framework basado en HTML5, necesita de un componente para ejecutar el proyecto, como una aplicación nativa, esto lo podemos conseguir utilizando un contenedor como Cordova o Capacitor. Se pone a consideración Capacitor debido a las nuevas características implementadas.

1.2.2.4.3 Capacitor

Capacitor es un framework de código abierto que permite "transformar" una WebApp en una App Mobile o Desktop mediante el uso de WebView. Capacitor se integran completamente con Ionic y Angular. No solo desde los schematics, sino también al momento de compilar, no es necesario hacer ajustes en los scripts como en el caso de Apache Cordova.

Es compatible con todos los Plugins de Cordova y los de Ionic a la vez, lo que permite utilizar Capacitor sin riesgos de compatibilidad con algún plugin en particular [14].

1.2.2.4.4 Angular

Angular es un framework Javascript muy potente, adecuado para el desarrollo de aplicaciones front-end moderna, creado por Google, cuya finalidad es facilitar el desarrollo de aplicaciones web SPA (Single Page Application) y ofrecer herramientas para trabajar con elementos de una web de manera más sencilla y óptima; lo que hace perfecto para el desarrollo de aplicaciones altamente interactivas, además contempla una gama de herramientas y servicios que hacen que la ejecución de Ionic sea bastante simple.

Angular usa TypeScript para la programación. TypeScript es una extensión del lenguaje Javascript, que agrega características importantes para las aplicaciones empresariales, como el tipado de datos o los decoradores; la compilación de TypeScript a Javascript compatible con los navegadores se realiza mediante las herramientas del Angular CLI, por lo que no agrega ninguna dificultad en el flujo de desarrollo [15].

1.2.2.4.5 API REST

API REST (Representational State Transfer) es un conjunto de reglas que permite la comunicación e integración de datos o servicios entre aplicaciones o dispositivos; utilizando métodos HTTP como: POST, con este método se puede insertar datos en el servidor; con GET se puede consultar los datos, con DELETE se puede borrar la información del servidor y con el método PUT se puede realizar modificaciones [16].

Una API REST ofrece [17]:

- **Escalabilidad:** Al mantener separado la parte del cliente y el servidor, es posible integrar recursos de manera óptima.
- **Multiplataforma:** Los métodos de API REST devuelven los datos en formato JSON, lo que permite pintar dichos datos sin ningún problema, independientemente del fron-end que esté utilizando.

1.2.2.4.6 Visual Studio Code

Se conoce a Visual Studio Code como un editor de código liviano, pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js, además tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) [18] .

1.2.2.4.7 Push Notifications

Se denomina push notification a los mensajes emergentes que se visualiza en los dispositivos al visitar un sitio web o al utilizar una APP. Las push notifications proporcionan el mismo comportamiento tanto en una APP y en un sitio web, es un medio por el cual se puede notificar o avisar a los usuarios de los cambios realizados, sin recargar la página o volver a ingresar a la app [19].

1.2.2.5 Geolocalización

Se denomina geolocalización a la capacidad de conocer la posición geográfica, las coordenadas, o ubicación de un objeto ya sea un teléfono móvil, tablet, ordenador, coche entre otros. Cualquier tipo de teléfono móvil puede ser localizado, ya que contamos con varios modos de geolocalizar.

Mediante las coordenadas geográficas como la latitud y longitud se puede encontrar a un dispositivo, utilizando el GPS, Wi-Fi, torres telefónicas o radiofrecuencia, los cuales aportan la información para identificar la ubicación [20].

1.2.2.6 Sistema de Posicionamiento Global (GPS)

GPS es una red compuesta por al menos 30 satélites que orbitan alrededor de la Tierra. Los teléfonos móviles incorporan receptores de GPS. Al menos 4 de estos satélites son visibles para nuestro dispositivo y cada satélite emite una señal sobre su ubicación cada cierto tiempo. Teniendo en cuenta la latitud, longitud, altura y tiempo se calcula la ubicación. Mientras más satélites tomen parte del proceso, más exacta será la triangulación.

A medida que la tecnología avanza, las empresas presentan nuevos dispositivos móviles, los cuales en su gran mayoría ya integran GPS [21].

1.2.2.7 API de Geolocalización

Se conoce a una API de geolocalización como una herramienta que permite ubicar dispositivos con gran precisión en cualquier lugar de la superficie de la Tierra. Para ello, se basa en señales de teléfonos móviles, GPS y WiFi que son detectadas por el dispositivo.

API (Application Programming Interface) es un conjunto de herramientas que se utiliza para comunicar e integrar diferentes sistemas [22].

1.2.2.8 Google Maps Platform

Google presenta una plataforma donde enlista diferentes APIs, preparadas para ser implementadas, en aplicaciones móviles o en sitios web.

1.2.2.8.1 Google Maps Platform – Maps

Presenta a Maps y Street View, que permiten integrar mapas estáticos o interactivos.

A continuación se enlista las APIs que ofrece Google Maps [23]:

- _ SDK de Maps para iOS.
- _ SDK de Maps para Android.
- _ API Street View.
- _ API Static de Maps.
- _ URLs de Maps.
- _ API JavaScript de Maps.

Las Google Maps APIs se pueden incorporar mediante el SDK o peticiones HTTP y el API KEY obtenido mediante la consola de desarrolladores de google. Mencionando que no todas las APIs de Google son gratuitas.

1.2.2.8.2 Google Maps Platform – Routes

Permite administrar rutas ya sea para planear una entrega u otra actividad relacionada de manera eficiente y en tiempo real.

- **Directions:** Proporciona rutas en tiempo real.
- **Distance Matrix:** Muestra el tiempo y distancia en referencia algún sitio.
- **Roads:** Permite crear itinerarios.

1.2.2.8.3 Google Maps Platform – Places

- **Place Details:** Detalla la información de un lugar en específico.
- **Current Place:** Identificar un sitio en base a cierta información.
- **Find Place:** Encontrar un sitio en base al número de celular u otra información.
- **Llenado automático:** Visualiza las recomendaciones de sitios en el momento de la búsqueda
- **Geocoding:** Convierte las coordenadas geográficas en direcciones y vice-versa.
- **Geolocation:** Visualiza el sitio exacto del dispositivo mediante una red Wifi.
- **Time Zone:** Visualiza la zona horaria sin importar donde se encuentre.

Las APIs que ofrece google Places son [23]:

- _ API Places.
- _ SDK do Places para iOS.
- _ SDK do Places para Android.
- _ API Geocoding.
- _ Places Library, API JavaScript de Maps.
- _ API Geolocation.
- _ API Time Zone.
- _ API Elevation.

1.2.2.9 Hosting

Se conoce como hosting al sitio donde se aloja el dominio y todo el contenido de la página web. El mismo debe proporcionar medidas de seguridad que garanticen el

rendimiento y la protección de ataques cibernéticos a la información almacenada en el mismo.

Otro punto muy importante, el hosting debe proporcionar las medidas de seguridad adecuadas, evitando posibles ataques informáticos [24].

1.2.2.10 Base de Datos

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite, una base de datos es un sistema de archivos electrónico [25].

1.2.2.10.1 Base de Datos Relacional

Las bases de datos relacionales (SQL) son el tipo que más tiempo lleva en el mercado de la tecnología y son las más utilizadas tradicionalmente. Su composición está hecha con bases de datos llenas de tablas con filas que contienen campos estructurados.

La organización de estas bases de datos son a través de identificadores, los mismos que sirven para integrarse o realizar una consulta; un registro es un sistema completo de campos y un archivo es una colección de registros. La base de datos relacional más usada y conocida es MySQL junto con Oracle, seguida por SQL Server y PostgreSQL, entre otras [26].

Para modificar algún dato, en este tipo de base de datos, después de determinar su estructura, tiende a presentar complicaciones por sus tablas relacionadas.

1.2.2.10.2 Base de Datos No relacional

Una base de datos NoSQL como su nombre lo indica es una base de datos no relacional que se utiliza para almacenar y recuperar los datos. Funcionan mejor con big data y aplicaciones web en tiempo real. Aquí los datos están en una estructura en forma de árbol. Por ejemplo, la nueva información se agrega como una rama, y luego las sub-ramas y puede seguir agregando.

Estas bases de datos nos ofrecen un esquema flexible, con diferentes modelos de datos como documentos, clave valor o gráficos, los mismos que pueden ser en memoria y búsqueda. Presentan una gran acogida por la facilidad de desarrollar, y el resultado obtenido en referencia a la funcionalidad y rendimiento [26].

1.2.2.10.3 Comparación de base de datos

Tabla 3. Comparación de base de datos

Base de datos	Sql	No Sql
Tipo de esquema	Esquema estático	Esquema dinámico
Rendimiento	Medio	Alto
Almacenamiento	La información se organiza en tablas separadas.	La información se organiza normalmente mediante documentos.
Migración	Implica un proceso muy amplio.	Se pueden adaptarse a otro tipo de sistema de manera sencilla
Escalabilidad	Escalan verticalmente.	Escalan horizontalmente.
Identificador	Es necesario un identificador.	No depende de un identificador.
SQL	Se recomiendan utilizar para consultas complejas.	Se recomiendan utilizar para consultas sencillas.
Manipulación de datos	Lenguaje específico que utiliza instrucciones Select, Insert y Update, p. ej., SELECT fields FROM table WHERE.	A través de APIs orientadas a objetos

Elaborado por: El Investigador

En la tabla 3. Se analizó varios parámetros determinando como opción favorable a la base de datos No Sql, por basarse en un esquema dinámico siendo ideales para manejar grandes volúmenes de datos que se generan rápidamente, almacenándolos mediante documentos. Además funcionan muy bien en aplicaciones a tiempo real, cumpliendo el requisito del proyecto a desarrollar.

Entre las bases de datos NoSql podemos mencionar Mongodb y Firebase con más acogida, para este tipo de proyectos

1.2.2.10.4 Mongodb

Es un sistema de base de datos NoSQL multiplataforma de licencia libre, orientado a documentos de esquema libre, lo que implica que cada registro puede tener un esquema de datos distinto, (los atributos no tiene que repetirse entre los diferentes registros).

En MongoDB, cada registro o conjunto de datos se denomina documento, que pueden ser agrupados en colecciones, (equivalente a las tablas de las bases de datos relacionales pero sin estar sometidos a un esquema fijo). Se pueden crear índices para algunos atributos de los documentos [27].

1.2.2.10.5 Google Cloud Firestore

Cloud Firestore es una base de datos NoSQ, flexible y escalable que ofrece Firebase, trabaja a través de documentos, los mismo que contienen campos y se almacenan como colecciones y subcolecciones, para facilitar realizar consultas, siendo perfecta para crear aplicaciones móviles y web utilizando los SDKs tanto para iOS y Android.

Además, Cloud Firestore incorpora un método para trabajar de manera normal sin conexión a internet, con una base descargada en su cache del cliente hasta obtener una conexión optima, otro añadido que nos ofrece es la integración con otros servicios como Cloud Functions y otras bibliotecas de código abierto [28].

Mediante un análisis las bases de datos NoSql propuestas para el desarrollo del proyecto, se decide optar por Google Cloud Firestore, debido a su estructura de base de datos, la integración con otras funcionalidades de firebase y la recuperación de los datos en tiempo real, perfecto para aplicaciones que necesitan manejar en varios dispositivos.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar una Aplicación móvil multiplataforma para la gestión de agentes motorizados en la empresa de encomiendas “Gabitos Express”.

1.3.2 Objetivos Específicos:

- Analizar el proceso actual de gestión de agentes motorizados y pedidos para el desarrollo de la aplicación móvil multiplataforma.
- Identificar una metodología adecuada para la creación de la aplicación móvil multiplataforma para la gestión de agentes motorizados en la empresa de encomiendas “Gabito Express”.
- Implementar una Aplicación móvil multiplataforma para la gestión de agentes motorizados en la empresa de encomiendas “Gabitos Express”.

CAPITULO II

METODOLOGIA

2.1 Materiales

Los materiales utilizados para el desarrollo del presente proyecto fueron artículos científicos, libros, revistas y sitios web como fuentes documentales bibliográficas; además se realizó una entrevista basada en un cuestionario de preguntas con el personal de la empresa para la recolección de información respecto a los procesos que realiza la empresa. De igual manera se aplicaron los conocimientos adquiridos durante la etapa académica que implica la carrera.

2.2 Métodos

2.2.1 Modalidad de Investigación

Modalidad Bibliográfica o Documentada: La investigación es bibliográfica porque permite ampliar y profundizar diferentes enfoques, teorías, conceptualizaciones y criterios de diversos autores sobre el problema en estudio, basándose en documentos referentes a base de datos y sistemas automatizados.

Modalidad de Campo: La investigación tendrá la modalidad de campo porque se visitara el establecimiento con el fin de recolectar información relevante de manera directa sobre el proceso del servicio, para cumplir con los objetivos propuestos del trabajo de estudio.

2.2.2 Población y Muestra

Población

La población es un conjunto de elementos que contienen ciertas características que se pretenden estudiar. Por esa razón, entre la población y la muestra existe un carácter inductivo desde lo particular a lo general, esperando que la parte observada denominada como muestra sea lo más representativa de la realidad siendo la población, para un correcto estudio [29].

En determinados casos de investigación no es posible obtener la muestra de una población, es donde se aplica métodos de muestreo no probabilísticos, los cuales permiten seleccionar casos característicos de una población limitando la muestra sólo a estos casos. Se utiliza en escenarios en las que la población es muy variable y consiguientemente la muestra es muy pequeña.

Para extraer la muestra se debe:

- Definir claramente la población que sirve de base para la muestra.
- Disponer de un registro de la población, es decir, una lista de sus elementos (marco muestral).
- Determinar el tamaño de la muestra, para obtener el resultado al menor costo, menor tiempo y con el personal indispensable.
- Lograr que la muestra sea representativa, es decir, que refleje las características de la población, en la misma proporción.
- Aplicar en la muestra los procedimientos e instrumentos de recolección de información.

Tabla 4. Población a investigarse

Detalle	Frecuencia
Propietario	1
Jefe de logística	1
Agentes motorizados	15
Reporte de pedidos	1
Principales clientes del periodo 2020	5
TOTAL	23

Elaborado por: El Investigador

Muestra

Es un subconjunto o parte del universo o población en que se llevará a cabo la investigación. Hay procedimientos para obtener la cantidad de los componentes de la muestra como fórmulas, lógica y otros. La muestra es una parte representativa de la población [30].

La muestra es la que puede determinar la problemática, siendo capaz de generar los datos que identifican las fallas dentro del proceso. Es decir, representa una parte de la población objeto de estudio.

La presente población es considerada muy pequeña por lo cual no se va a considerar el cálculo de la muestra, además dadas las características de la investigación se requiere la participación de todos los informantes.

2.2.3 Recolección de la Información

Para la recolección de información se utilizó la técnica de la entrevista donde se determinó los requerimientos tanto funcionales como técnicos, identificando detalladamente las necesidades del administrador así como de los motorizados y clientes; Además se aplicó la técnica de observación donde se identifica con más precisión el proceso que realizan cotidianamente.

Durante y después de la revisión de requerimientos, acorde a la metodología seleccionada, se realizó retroalimentaciones con la Jefa de Logística, para ratificar la percepción de los requisitos anteriormente expresados.

2.2.4 Procesamiento y Análisis de Datos

Para el análisis de los requerimiento de la aplicación se procedió a realizar una entrevista al propietario, la jefa de logística y motorizados de la empresa basado en preguntas desarrolladas de manera estructurada con el propósito de obtener directamente la información, ya que será la primera fuente de información para recopilar los requisitos para el desarrollo de la aplicación. **Anexo 1**

Posteriormente en base a la metodología seleccionada, se identificara de mejor manera los requerimientos para el desarrollo del presente proyecto.

2.2.4.1 Resultados de la encuesta

El objetivo de la empresa es brindar el servicio a todo tipo de cliente sin distinción ya que para la empresa el factor fundamental es el cliente y ponen a su disposición el servicio de pedidos o encomiendas.

Actualmente la empresa no cuenta con ninguno tipo de aplicación para la gestión de motorizado o pedidos, por ello indica que la aplicación debe funcionar en cualquier dispositivo móvil independientemente del sistema operativo, una aplicación donde el administrador pueda registrar a los motorizados, asignar a los motorizados y visualizar la ubicación de los mismo, visualizar los pedidos, que el cliente pueda registrar y consultar los pedidos, así como calificar la entrega de los motorizados; en cuanto a los motorizados puedan ver los pedidos que se le asignaron y los acepte, cambiando el estado del pedido para todos los usuarios.

El propietario ratifica que la aplicación es una aporte fundamental, ya que al conocer mediante una notificación la entrega de un pedido, ya no sería necesario contactarse preguntando el proceso del mismo e identificar su disponibilidad, de la misma manera al conocer la ubicación del motorizado en relación a un pedido, se reducirá tiempos destinados para otras actividades muy ajenas al trabajo, evidentemente se mejorara el servicio hacia los clientes y la gestión de los motorizados.

Además menciona estar totalmente de acuerdo en apoyar económicamente para llevar la aplicación a producción, recalando que su objetivo es brindar un mejor servicio a nuestros clientes, integrando la tecnología.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados.

3.1.1 Metodologías de desarrollo de software

3.1.1.1 Metodologías tradicionales

Las metodologías de desarrollo tradicionales imponen una disciplina de trabajo fundamentada en la documentación sobre el proceso de desarrollo de software, se realiza una planificación global y total de todo el trabajo a realizar, para comenzar con el ciclo de desarrollo de software.

Otra de las características importantes son los altos costos al implementar un cambio al no ofrecer una buena solución para proyectos donde el entorno es volátil.

Entre las principales metodologías tradicionales tenemos a RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y cumplir con un plan de proyecto, definido en la fase inicial del desarrollo del proyecto [31].

3.1.1.2 Metodologías ágiles

Las metodologías ágiles presentan como principal particularidad la flexibilidad, los proyectos en desarrollo son subdivididos en proyectos más pequeños, además son altamente colaborativos e incluye una comunicación constante con el usuario, siendo así más adaptable a los cambios.

De hecho, el cambio de requerimientos por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constante [32].

Existen actualmente una serie de metodologías que responden a las características de las metodologías ágiles y cada vez están teniendo más adeptos. Quienes tiene el mismo objetivo, de desarrollar software en el menor tiempo posible y con el menor costo.

Tabla 5. Comparación de Metodologías de Desarrollo de Software

Metodología Tradicional	Metodología Ágil
Orientado a procesos	Orientado a personas
Proceso estático	Proceso dinámico
Se basa en un proyecto global	Un proyecto dividido en proyectos pequeños.
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el proyecto	Entregas constantes de software
Documentación extensa	Poca documentación

Elaborado por: El Investigador

En la tabla 5. Se analizó varios parámetros los cuales indican que la Metodología Ágil es la indicada para el desarrollo del presente proyecto, al ser un proceso dinámico con constante contacto con el cliente, el cual reduce un porcentaje considerable de error en el desarrollo, el mismo que se realiza en fases incrementales con iteraciones de lapsos cortos.

A continuación se detalla las metodologías más utilizadas:

3.1.1.3 Metodología Scrum

Scrum es un proceso creado para generar varios procesos y técnicas con el propósito de mejorar continuamente el software a desarrollar, el equipo y el entorno donde se realiza el trabajo [32].

Roles de Scrum

A continuación se detalla los roles de la metodología Scrum [33]:

- **Product owner:** Es el encargado de llevar al máximo el trabajo realizado por el equipo de desarrollo, además él debe mantener constante comunicación con el cliente.
- **Scrum Master:** Delegado para eliminar problemas presentados un sprint.
- **Scrum Team:** Son los encargados de valorar las actividades o tareas del product backlog.
- **Stakeholders:** Son básicamente todas las demás personas que intervienen en el proyecto pero que no forman parte del Equipo de Scrum, por ejemplo los Stakeholders que son los usuarios finales.

Reuniones de Scrum

Las reuniones que se realizan en la metodología Scrum son las siguientes [33]:

- **Sprint:** Al sprint se le considera como el corazón de la Metodología Scrum, planificado en un intervalo de tiempo no mayor a un mes.
- **Planificación del Sprint:** Es una reunión donde se define las actividades o tareas a realizar en base a preguntas como que y como se va a realizar dichas tareas.
- **Ejecución de sprint:** Reunión diaria donde se expone las actividades anteriores y las presentes, dentro de un intervalo de tiempo no mayor a 15 minutos.
- **Revisión de sprint:** Es una reunión con el cliente donde se presenta el desarrollo y funcionamiento del proyecto, en un lapso no mayor a 4 horas.
- **Retrospectiva de sprint:** Es un punto fundamental para revisar o identificar mejoras para el proyecto.

Artefactos

- **Product backlog**

Se denomina product backlog a conjunto de tareas o actividades de un proyecto, donde cada una de ellas es definida con un tiempo estimado según la prioridad establecida por el Product Owner en conjunto con el cliente, ordenándolas desde las de mayor prioridad hasta las de menor [33].

3.1.1.4 Metodología Kanban

Se denomina Metodología Kanban por la unión de sus palabras Kan y Ban, siendo el significado de Kan, plano visual, y Ban, tarjeta. Esta una metodología ágil que permite visualizar claramente lo que se termina, lo que se está realizando y lo que se va a realizar, por medio de las tarjetas; facilitando la implementación de la misma en proyectos pequeños y medianos de una empresa.

La forma de trabajo de esta metodología es mediante un tablero visible para todos, donde se expone la información clara y completa. Además, Kanban expone cierto fundamentos para su implementación [34]:

- **Visualizar el flujo de trabajo:** Se realiza mediante el tablero que presenta todas las tareas.
- **Limitar la cantidad de Trabajo en Proceso:** Determinar tareas según la prioridad del proyecto.
- **Lectura fácil de indicadores visuales:** Al trabajar con colores en cuanto a tarjetas, facilita su identificación y por ende la información que contiene.
- **Realizar un seguimiento del tiempo:** Se lleva un control constante de tiempos de entrega.

3.1.15 Metodología Extreme Programming (XP)

La Metodología XP es una metodología ágil destinada al desarrollo de software cuyo objetivo es asegurar la producción del software con alta calidad, cumpliendo con los requerimientos del usuario.

Es una metodología exitosa por la flexibilidad que brinda al implementar cambios en los requerimientos del cliente, en alguna fase del desarrollo del proyecto [35].

Valores de XP

- **Comunicación:** Es un valor fundamental en esta metodología por priorizar la comunicación.
- **Simplicidad:** Propone realizar lo necesario, evitando desperdiciar recursos.
- **Retroalimentación:** Con una retroalimentación constante es posible obtener un diseño simple y que se puede modificar a futuro.
- **Coraje:** Hace referencia al valor de cambiar o aceptar cambios sin mayor problema.
- **Respeto:** Valor importante al comunicar comentarios constructivos del proyecto.

Roles

Los roles definen a quienes participan en el proyecto en base a la Metodología XP y se detallan a continuación [35]:

- **El cliente:** es la persona que requiere la aplicación y expone las historias de usuario.
- **El programador:** responsable de la codificación completa del proyecto, en base a las historias de usuario ya recopiladas.
- **El entrenador:** es quien supervisa el trabajo del equipo o busca alternativas para mejorar el desarrollo del proyecto.
- **El rastreador:** es la persona que monitorea constantemente el avance del desarrollo del software.
- **El probador:** responsable de realizar las pruebas necesarias para verificar el funcionamiento del proyecto.
- **El pronosticador:** es quien monitorea y advierte de inconvenientes encontrados en el proyecto.

Fases de desarrollo

Las fases de desarrollo de la metodología XP son [36]:

- **Fase de planificación:** En esta fase se identificarán las historias de usuario, que son esquemas donde se detalla de manera general la funcionalidad del proyecto desde la perspectiva del cliente. Estas deben ser valoradas según la prioridad que tengan el desarrollo del proyecto.
- **Fase de diseño:** En la fase de diseño se trabajará de manera sencilla con un código simple para cumplir con la funcionalidad de la historia de usuario, también se hace uso de tarjetas CRC (Clase Responsabilidad Colaboración), donde se describe las clases utilizadas en la programación de una historia.
- **Fase de codificación:** En esta fase comienza la programación. La programación debe hacerse bajo estándares ya que mantiene el código consistente y facilita su comprensión y escalabilidad.
- **Fase de pruebas:** La fase de pruebas es muy importante ya que evalúa el funcionamiento en base a la historia de usuario, donde se puede modificar requerimientos o errores y continuar con el desarrollo del proyecto.

3.1.1.6 Comparación de Metodologías ágiles

Tabla 6. Comparación de Metodologías ágiles

Características	Scrum	XP	Kanban
Tamaño de equipo	Pequeños	Pequeños	Pequeños, Medianos
Roles	Scrum Máster, Product Owner, Scrum Team, Usuarios	Programador, Cliente, Coach, Tracker, Consultor	Roles no definidos
Entregas	Entregas cada cierre del sprint entre 2 y 4 semanas	Entregas en iteraciones basadas en historias de usuario y prioridades	Entregas continuas con prioridades
Trabajo	Dividido en presentaciones	Tareas en iteraciones	Dividido en tareas

	Sprints		
Cambios	Solo al finalizar el ciclo.	En el mantenimiento.	En cualquier momento
Uso	En modelos donde las prioridades son fijadas al inicio y no sufren cambios en el proceso.	En modelos donde se basa en la simplicidad comunicación y retroalimentación	En situaciones de trabajo en donde existan prioridad de tareas.

Elaborado por: El Investigador

En la tabla 6. Se analizó varios parámetros los cuales determinan la Metodología XP, como la metodología más óptima para el desarrollo de la aplicación por estar dedicada al desarrollo de software, para proyectos a corto plazo y para realizarlo en equipos de trabajo pequeño, y como aspecto fundamental tiene la comunicación e integración constante del cliente al desarrollo del proyecto.

3.1.2 Tecnologías de desarrollo de Back-end

El presente proyecto está enfocado a ser una aplicación multiplataforma, definido a trabajar en el front end con Ionic y angular, por lo que se pretende elegir una tecnología de desarrollo back end con cierta similitud en el lenguaje de desarrollo, se pone a consideración a Firebase y NodeJS, tecnologías de desarrollo back end con una gran acogida en el mercado.

3.1.2.1 Firebase

Se conoce a Firebase como una plataforma creada para el desarrollo de aplicaciones tanto móvil como web, facilitando el mismo al contar con herramientas que simplifican el desarrollo y mantenimiento. Es muy útil cuando el desarrollador no quiere involucrarse mucho en el back end [37].

3.1.2.1.1 Servicios de Firebase

Firestore incorpora los siguientes servicios [37]:

- **Realtime Database:** Es una base de datos NoSQL que trabaja en tiempo real.

- **Cloud Firestore:** Es una base de datos NoSQL alojada en la nube a la que pueden acceder las apps para iOS, Android y Web directamente desde los SDK nativos.
- **Autenticación:** Ofrece un método de registro por medio del correo electrónico y clave, además de otras plataformas de redes sociales.
- **Almacenamiento en la nube:** Es una herramienta fundamental que ofrece esta plataforma para guardar los ficheros.
- **Hosting:** La plataforma de Firebase incluye un servidor donde se puede desplegar las apps muy fácil y seguros, ya que al momento del despliegue automáticamente se instala los certificados de seguridad SSL y HTTP2.

3.1.2.2 Node Js

NodeJs es un marco de JavaScript y un entorno de ejecución, donde ejecuta código JavaScript fuera del navegador web. Además, es un lenguaje de programación que cambió drásticamente el desarrollo web y ha sido utilizado por especialistas de TI en todo el mundo desde 1995.

Node.js pertenece a la categoría de software que permite a los desarrolladores codificar aplicaciones y ejecutarlas simultáneamente.

Las principales características de Node.js son:

- Es un marco de servidor.
- Es de código abierto.
- Es adecuado para diferentes plataformas.

Cuenta con el apoyo de una gran comunidad de desarrolladores que trabajan constantemente para mejorarlo. Realiza transmisión de datos de alto nivel. También es adecuado para desarrollar software para diferentes plataformas independientemente del sistema operativo subyacente.

3.1.2.3 Comparación de tecnologías back-end

Tabla 7. Comparación de tecnologías back-end

Tecnología	Ventajas	Desventajas
NodeJs	Multiplataforma El desarrollo se realiza en JavaScript.	Cuello de botella en el rendimiento API inestable Paquetes independientes
Firebase	Multiplataforma El desarrollo se realiza en TypeScript Integración con Notificaciones Push Base de datos en tiempo real. (Functions) Funciones en la nube. Plan de prueba gratuita.	Pruebas limitadas. Servicios de pago.

Elaborado por: El Investigador

En la tabla 7. Se analizó las tecnologías en base a sus características, ventajas y desventajas los cuales determinan a firebase, como la tecnología back-end óptima para el desarrollo de la aplicación, debido a su integración con diversos servicios, contar con functions que permite integrar código propio y por su lenguaje de programación TypeScript, el mismo que se utiliza en el front-end.

Además, firebase proporciona bases de datos en tiempo real, un hosting y un plan gratuito para desplegar la aplicación, disponible para cualquier dispositivo, requerimiento fundamental de este proyecto.

3.1.3 Desarrollo de la propuesta

3.1.3.1 Fase I: Planificación

Es la primera fase, donde el cliente plantea los requerimientos para transformarlas en historias de usuario facultando el inicio al desarrollo del proyecto.

3.1.3.1.1 Resultado de la entrevista

Como resultado de la entrevista realizada en el punto 2.2.4.1, se obtiene la siguiente información.

Es imprescindible mencionar que la aplicación debe funcionar en cualquier dispositivo móvil, una aplicación donde el administrador pueda registrar a los motorizados, visualizar los pedidos, asignar a los motorizados y visualizar la ubicación de los mismos, que el cliente pueda registrar y consultar los pedidos o encomiendas, así como calificar la entrega de los motorizados; en cuanto a los motorizados puedan ver los pedidos que se le asignaron y los acepte, donde el estado del pedido cambiara para todos los usuarios; además que todos estos procesos sean notificados en momento de cada acción.

3.1.3.1.2 Definición de roles

Los roles permiten determinar a todos los involucrados en el desarrollo de la aplicación.

Tabla 8. Definición de roles

Encargado	Descripción	Rol Xp
Sylvia Guanoluisa Tesisista	Persona responsable de planificar, diseñar, codificar y probar la aplicación móvil.	Programador Tester
Ing. David Guevara Tutor del proyecto de investigación	Persona encargada de revisar el cumplimiento mensual de las actividades realizadas en base al cronograma.	Coach Consultor
Sr. Gabriel Romo Gerente propietario de la Empresa de encomiendas "Gabitos Express"	Persona que necesita el desarrollo de la aplicación y conoce los procesos del servicio.	Cliente

Elaborado por: El Investigador

3.1.3.1.3 Historias de Usuario

Las historias de usuario son indispensable para el desarrollo del proyecto, se basa en recoger y transmitir una lista de características por parte del cliente en su producto final, para ello se procede a dividir las en tareas, para así estimar el tiempo que tomará realizar o cumplir cada una de ellas.

Para el diseño de las historias de usuario, se ha utilizado el modelo estructurado en la Tabla 9, en base a la Metodología seleccionada.

Tabla 9. Plantilla de Historia de Usuario

HISTORIAS DE USUARIO			
Número:		Usuario:	
Nombre de la historia:			
Prioridad en el negocio:		Riesgo en el desarrollo:	
Puntos estimados:		Iteración asignada:	
Programador responsable:			
Descripción:			
Observaciones:			

Elaborado por: El Investigador

Información de cada elemento que comprende la plantilla de historia de usuario.

- **Numero:** Etiqueta única de la historia de usuario.
- **Usuario:** A quien está dirigida la historia de usuario.
- **Nombre de la historia:** Título definido la historia de usuario.
- **Prioridad en el negocio:** Lugar donde se asigna valores (Alta, media y Baja) según la necesidad de los usuarios.
- **Riesgo de desarrollo:** Lugar donde se asigna el valor (Alto, Medio, Bajo) según el riesgo que tenga el desarrollador dicha historia de usuario.
- **Programador responsable:** Persona responsable de llevar a cabo dicha historia de usuario.
- **Puntos estimados:** El número de los días que se tomara para el desarrollar la historia de usuario.
- **Iteración asignada:** La iteración asociada a la historia de usuario.
- **Descripción:** En este apartado en cliente puede expresar con sus propias palabras si se desea realizar alguna validación.
- **Observación:** Se detallan aquellas actividades con relación a la historia.

Tabla 10. Historia de usuario Pantalla de inicio

Historia de Usuario			
Numero: 01		Usuario: Todos	
Nombre de la historia: Pantalla de inicio			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	3	Iteración asignada:	1
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará 3 pantallas de bienvenida al usuario donde se expondrá la misión, los servicios y beneficios que ofrece la empresa.			
Observación: La pantalla de inicio será solo informativa.			

Elaborado por: El Investigador

Tabla 11. Historia de usuario Iniciar sesión

Historia de Usuario			
Numero: 02		Usuario: Todos	
Nombre de la historia: Iniciar Sesión			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	5	Iteración asignada:	1
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla donde se ingresa el correo y contraseña para acceder a la aplicación y proceder a utilizarla.			
Observación: Existen tres tipos de usuarios: Administrador, Motorizado y Cliente, con distintas funcionalidades.			

Elaborado por: El Investigador

Tabla 12. Historia de usuario Recuperar clave

Historia de Usuario			
Numero: 03		Usuario: Todos	
Nombre de la historia: Recuperar clave			
Prioridad en el negocio:	Medio	Riesgo en el desarrollo:	Medio
Puntos estimados:	2	Iteración asignada:	4
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla donde se debe ingresar el correo electrónico.			
Observación: Se envía un enlace al correo electrónico ingresado, donde se ingresará la nueva clave.			

Elaborado por: El Investigador

Tabla 13. Historia de usuario Registrar cliente

Historia de Usuario			
Numero: 04		Usuario: Cliente	
Nombre de la historia: Registrar cliente			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	7	Iteración asignada:	1
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla donde los clientes pueden registrar datos personales como nombre, apellido, correo electrónico, número de celular, ubicación y clave para acceder a la aplicación.			
Observación: La clave debe ser mayor a 7 dígitos, la ubicación se obtiene por google maps.			

Elaborado por: El Investigador

Tabla 14. Historia de usuario Actualizar perfil cliente

Historia de Usuario			
Numero: 05		Usuario: Cliente	
Nombre de la historia: Actualizar perfil cliente			
Prioridad en el negocio:	Medio	Riesgo en el desarrollo:	Medio
Puntos estimados:	2	Iteración asignada:	3
Programador responsable:		Sylvia Guanoluisa	
Descripción: Actualizar los datos personales y la foto delo usuario para acceder a la aplicación.			
Observación:			

Elaborado por: El Investigador

Tabla 15. Historia de usuario Registrar motorizado

Historia de Usuario			
Numero: 06		Usuario: Administrador	
Nombre de la historia: Registrar motorizado			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	5	Iteración asignada:	1
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla donde se ingresa datos personales como nombre, apellido, correo electrónico, número de celular, clave e imagen.			
Observación: Adicional a esta información se registrar el nuevo motorizado con estado disponible. El cambio de clave debe ser realizada por el motorizado.			

Elaborado por: El Investigador

Tabla 16. Historia de usuario Dar de baja motorizado

Historia de Usuario			
Numero: 07		Usuario: Administrador	
Nombre de la historia: Dar de baja motorizado			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	3	Iteración asignada:	4
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con la información del motorizado, donde podrá dar de baja.			
Observación:			

Elaborado por: El Investigador

Tabla 17. Historia de usuario Mostrar motorizados

Historia de Usuario			
Numero: 08		Usuario: Administrador	
Nombre de la historia: Mostrar motorizados			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	3
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con la ubicación del motorizado.			
Observación:			

Elaborado por: El Investigador

Tabla 18. Historia de usuario CRUD productos

Historia de Usuario			
Numero: 09		Usuario: Administrador	
Nombre de la historia: CRUD productos			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	4	Iteración asignada:	1
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla donde pueda registrar (producto como nombre, descripción e imagen), modificar de un producto y eliminar el producto.			
Observación: Debe cambiar la clave el motorizado.			

Elaborado por: El Investigador

Tabla 19. Historia de usuario Actualizar perfil motorizado

Historia de Usuario			
Numero: 10		Usuario: Motorizado	
Nombre de la historia: Actualizar perfil motorizado			
Prioridad en el negocio:	Medio	Riesgo en el desarrollo:	Medio
Puntos estimados:	2	Iteración asignada:	3
Programador responsable:		Sylvia Guanoluisa	
Descripción: Actualizar los datos personales para acceder a la aplicación.			
Observación:			

Elaborado por: El Investigador

Tabla 20. Historia de usuario Mostrar productos (Tienda)

Historia de Usuario			
Numero: 11		Usuario: Cliente	
Nombre de la historia: Mostrar productos (Tienda)			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	1
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla de productos con la descripción y un botón para pedir.			
Observación:			

Elaborado por: El Investigador

Tabla 21. Historia de usuario Mostrar productos seleccionados (carrito)

Historia de Usuario			
Numero: 12		Usuario: Cliente	
Nombre de la historia: Mostrar productos seleccionados (carrito)			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	6	Iteración asignada:	2
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con los productos seleccionados donde debe ingresar la dirección de destino.			
Observación: Se da la opción a ingresar las direcciones mediante Google Maps. El estado del pedido se registra como ENVIADO.			

Elaborado por: El Investigador

Tabla 22. Historia de usuario Ingresar descripción producto pedido

Historia de Usuario			
Numero: 13		Usuario: Cliente	
Nombre de la historia: Registro descripción producto pedido			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	2
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con producto donde se debe ingresar la descripción.			
Observación:			

Elaborado por: El Investigador

Tabla 23. Historia de usuario Mostrar pedidos

Historia de Usuario			
Numero: 14		Usuario: Administrador	
Nombre de la historia: Mostrar pedidos			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	3	Iteración asignada:	2
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con los pedidos por estado pendientes y asignados por orden de fecha y hora.			
Observación: Se visualiza los pedidos o encomiendas con estado pendiente.			

Elaborado por: El Investigador

Tabla 24. Historia de usuario Asignar motorizado

Historia de Usuario			
Numero: 15		Usuario: Administrador	
Nombre de la historia: Asignar motorizado			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	8	Iteración asignada:	2
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con los motorizados activos, detallando si mantiene un pedido en camino, pedidos asignados y el tiempo aproximado desde la ubicación del motorizado al pedido, donde permite seleccionar a uno de ellos.			
Observación: El estado del pedido pasa a ASIGNADO.			

Elaborado por: El Investigador

Tabla 25. Historia de usuario Mostrar pedidos asignados

Historia de Usuario			
Numero: 16		Usuario: Motorizado	
Nombre de la historia: Mostrar pedidos asignados			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	3	Iteración asignada:	3
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con los pedidos asignados por orden de fecha y hora.			
Observación: Todos los pedidos o encomiendas están con estado ASIGNADO.			

Elaborado por: El Investigador

Tabla 26. Historia de usuario Actualizar estado de pedido

Historia de Usuario			
Numero: 17		Usuario: Motorizado	
Nombre de la historia: Actualizar el estado de pedido			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	8	Iteración asignada:	3
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualiza una pantalla donde se podrá modificar el estado del pedido de CAMINO o ENTREGADO.			
Observación: Solo existe un pedido en camino.			

Elaborado por: El Investigador

Tabla 27. Historia de usuario Mostar ubicación motorizado

Historia de Usuario			
Numero: 18		Usuario: Administrador	
Nombre de la historia: Mostar ubicación motorizado			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	10	Iteración asignada:	2
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará la ubicación del motorizado.			
Observación: El motorizado debe conceder el permiso al iniciar sesión por primera vez en el dispositivo.			

Elaborado por: El Investigador

Tabla 28. Historia de usuario Recorrido diario

Historia de Usuario			
Numero: 19		Usuario: Administrador	
Nombre de la historia: Recorrido diario			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	4	Iteración asignada:	4
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con el recorrido diario de cada motorizado.			
Observación:			

Elaborado por: El Investigador

Tabla 29. Historia de usuario Calificar Motorizado

Historia de Usuario			
Numero: 20		Usuario: Cliente	
Nombre de la historia: Calificar Motorizado			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	3
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla donde podrá elegir la calificación del motorizado.			
Observación:			

Elaborado por: El Investigador

Tabla 30. Historia de usuario Reporte pedidos

Historia de Usuario			
Numero: 21		Usuario: Administrador	
Nombre de la historia: Reporte pedidos			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	4	Iteración asignada:	4
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con los pedidos diarios y se podrá visualizar pedidos mediante fechas.			
Observación:			

Elaborado por: El Investigador

Tabla 31. Historia de usuario Reporte motorizados

Historia de Usuario			
Numero: 22		Usuario: Administrador	
Nombre de la historia: Reporte motorizado			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	4	Iteración asignada:	4
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se visualizará una pantalla con los motorizados, el total de pedidos detallando los pedidos asignados y pedidos en camino.			
Observación:			

Elaborado por: El Investigador

Tabla 32. Historia de usuario Mostrar menú

Historia de Usuario			
Numero: 23		Usuario: Todos	
Nombre de la historia: Mostrar menú			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se diseñará un menú con todos los módulos de la aplicación y la foto, nombre y apellido del usuario.			
Observación:			

Elaborado por: El Investigador

Tabla 33. Historia de usuario Desplegar la aplicación en el hosting

Historia de Usuario			
Numero: 24		Usuario: Administrador	
Nombre de la historia: Desplegar la aplicación en el hosting			
Prioridad en el negocio:	Alta	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:		Sylvia Guanoluisa	
Descripción: Se podrá acceder desde la web para usuarios clientes y administrador.			
Observación:			

Elaborado por: El Investigador

3.1.3.1.4 Tareas

Historia de usuario: Pantalla de inicio

Tabla 34. Diseño de las interfaces para pantalla de inicio

Tarea	
Numero: T1	Número de Historia: H01
Nombre: Diseño de las interfaces de usuario para pantalla de inicio	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará 3 pantallas de bienvenida al usuario donde se expondrá la misión, los servicios y beneficios que ofrece la empresa.	

Elaborado por: El Investigador

Tabla 35. Codificar interfaces

Tarea	
Numero: T2	Número de Historia: H01
Nombre: Codificar interfaces	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se codificará interfaces para manipular la información en la base de datos.	

Elaborado por: El Investigador

Tabla 36. Codificar servicio firestore

Tarea	
Numero: T2	Número de Historia: H01
Nombre: Codificar servicio firestore	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se implementara las funcionalidades de firestore: crear, seleccionar, eliminar, actualizar documentos.	

Elaborado por: El Investigador

Historia de usuario: Iniciar sesión

Tabla 37. Diseño de interfaz de usuario para iniciar sesión

Tarea	
Numero: T2	Número de Historia: H02
Nombre: Diseño de interfaz de usuario para iniciar sesión	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla para ingresar el correo electrónico y la clave.	

Elaborado por: El Investigador

Tabla 38. Implementación de métodos de autenticación

Tarea	
Numero: T3	Número de Historia: H02
Nombre: Implementación del método de autenticación correo electrónico	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Codificar del servicio firebaseauth mediante las funcionalidades de firebase., Desarrollar métodos para ingreso y consumir el servicio método de autenticación.	

Elaborado por: El Investigador

Tabla 39. Implementación de métodos de seguridad

Tarea	
Numero: T4	Número de Historia: H03
Nombre: Implementación de métodos de seguridad	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se implementará tipos de seguridad de Firebase restringiendo el acceso a las vistas en función de los roles tales como: administrador, motorizado y cliente.	

Elaborado por: El Investigador

Historia de usuario: Recuperar clave

Tabla 40. Diseño interfaz de usuario para recuperar clave

Tarea	
Numero: T5	Número de Historia: H03
Nombre: Diseño de interfaz de usuario para recuperar clave	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla para registrar nombre, apellido, correo electrónico, número de celular y clave.	

Elaborado por: El Investigador

Tabla 41. Implementación del servicio para recuperar clave

Tarea	
Numero: T6	Número de Historia: H03
Nombre: Implementación del servicio para recuperar clave	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir el servicio firebaseaut para recuperar la clave del usuario.	

Elaborado por: El Investigador

Historia de usuario: Registrar cliente

Tabla 42. Diseño interfaz de usuario para registro cliente

Tarea	
Numero: T5	Número de Historia: H04
Nombre: Diseño de interfaz de usuario para registro cliente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla para registrar nombre, apellido, correo electrónico, número de celular, dirección y clave.	

Elaborado por: El Investigador

Tabla 43. Diseño interfaz de usuario para registro cliente

Tarea	
Numero: T6	Número de Historia: H03
Nombre: Diseño de interfaz para el google maps	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla para mostrar el mapa de google maps.	

Elaborado por: El Investigador

Tabla 44. Implementación del servicio de google maps

Tarea	
Numero: T7	Número de Historia: H03
Nombre: Implementación del servicio de google maps.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el servicio GooglemapsService para obtener la dirección por medio de google maps.	

Elaborado por: El Investigador

Tabla 45. Implementación del servicio de creación de clientes

Tarea	
Numero: T8	Número de Historia: H03
Nombre: Implementación del servicio de creación de clientes.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir servicio para crear nuevos clientes en la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Actualizar perfil cliente

Tabla 46. Diseño de interfaz para actualizar perfil

Tarea	
Numero: T9	Número de Historia: H05
Nombre: Diseño de interfaz para actualizar perfil cliente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde puede cambiar la información.	

Elaborado por: El Investigador

Tabla 47. Implementación del servicio para actualizar perfil cliente

Tarea	
Numero: T10	Número de Historia: H05
Nombre: Implementación del servicio para actualizar perfil cliente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021

Descripción: Desarrollar el método para consumir servicios para actualizar información en la base de datos.

Elaborado por: El Investigador

Historia de usuario: Registrar motorizado

Tabla 48. Diseño de interfaz de usuario para registrar motorizado

Tarea	
Numero: T11	Número de Historia: H06
Nombre: Diseño de interfaz de usuario para registrar motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla para ingresar nombre, apellido, correo electrónico, número de celular, estado, clave e imagen.	

Elaborado por: El Investigador

Tabla 49. Implementación del servicio de subida de imágenes al servidor

Tarea	
Numero: T12	Número de Historia: H06
Nombre: Implementación del servicio de subida de imágenes al servidor	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el servicio FirestoreService para subir archivos de imagen al servidor las funcionalidades de firebase.	

Elaborado por: El Investigador

Tabla 50. Implementación del servicio para registrar motorizados

Tarea	
Numero: T13	Número de Historia: H06
Nombre: Implementación del servicio para registrar motorizados.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir el servicio para crear nuevos motorizados en la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Dar de baja motorizado

Tabla 51. Diseño de interfaz de usuario para dar de baja motorizado

Tarea	
Numero: T14	Número de Historia: H07
Nombre: Diseño de interfaz de usuario para dar de baja motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde puede cambiar el estado del motorizado (Disponible a No disponible).	

Elaborado por: El Investigador

Tabla 52. Implementación del servicio para dar de baja motorizado

Tarea	
Numero: T15	Número de Historia: H07
Nombre: Implementación del servicio para dar de baja motorizado.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir el servicio para modificar motorizados en la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Mostrar motorizados

Tabla 53. Diseño de interfaz para mostrar motorizados

Tarea	
Numero: T16	Número de Historia: H08
Nombre: Diseño de interfaz para mostrar motorizados	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde muestra a lista de motorizados, con la imagen, nombre y apellido, además un icono para visualizar su ubicación.	

Elaborado por: El Investigador

Tabla 54. Implementación del servicio para mostrar motorizados

Tarea	
Numero: T15	Número de Historia: H07

Nombre: Implementación del servicio para mostrar motorizados	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir servicios para mostrar motorizados en la base de datos.	

Elaborado por: El Investigador

Historia de usuario: CRUD productos

Tabla 55. Diseño de interfaz para CRUD productos

Tarea	
Numero: T17	Número de Historia: H09
Nombre: Diseño de interfaz para CRUD productos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde puede registrar, actualizar, eliminar y consultar productos. Para el registro de nuevos productos debe ingresar, nombre, descripción, dirección y la imagen.	

Elaborado por: El Investigador

Tabla 56. Implementación de los servicios para CRUD productos

Tarea	
Numero: T18	Número de Historia: H09
Nombre: Implementación de los servicios para CRUD productos	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir servicios para CRUD de motorizados.	

Elaborado por: El Investigador

Historia de usuario: Actualizar perfil motorizado

Tabla 57. Diseño de interfaz para actualizar perfil motorizado

Tarea	
Numero: T19	Número de Historia: H10
Nombre: Diseño de interfaz para actualizar perfil motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1

Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde puede cambiar la información.	

Elaborado por: El Investigador

Tabla 58. Implementación del servicio para actualizar perfil motorizado

Tarea	
Numero: T20	Número de Historia: H10
Nombre: Implementación del servicio para actualizar perfil	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir servicios para actualizar información en la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Mostrar productos (Tienda)

Tabla 59. Diseño de interfaz para mostrar productos (Tienda)

Tarea	
Numero: T21	Número de Historia: H11
Nombre: Diseño de interfaz para mostrar productos (Tienda)	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde muestra la imagen del producto con la ubicación y un botón de pedir.	

Elaborado por: El Investigador

Tabla 60. Implementación del servicio para mostrar productos

Tarea	
Numero: T22	Número de Historia: H11
Nombre: Implementación del servicio para mostrar productos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar métodos para mostrar productos de la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Mostrar productos seleccionados (carrito).

Tabla 61. Diseño de interfaz para mostrar productos seleccionados (carrito)

Tarea	
Numero: T23	Número de Historia: H12
Nombre: Diseño de interfaz para mostrar productos seleccionados.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde muestra los producto seleccionados, adicional se pueda ingresar la ubicación del pedido e ingresar una referencia de ubicación.	

Elaborado por: El Investigador

Tabla 62. Implementación del servicio para guardar el pedido.

Tarea	
Numero: T24	Número de Historia: H12
Nombre: Implementación del servicio para guardar el pedido.	
Tipo de tarea: Desarrollo	Puntos estimados: 5
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir servicios para guardar el pedido en la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Registrar descripción producto pedido

Tabla 63. Diseño de interfaz para ingresar descripción pedido producto

Tarea	
Numero: T25	Número de Historia: H13
Nombre: Diseño de interfaz para ingresar descripción pedido producto.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde ingrese la descripción del pedido producto.	

Elaborado por: El Investigador

Tabla 64. Implementación del servicio para ingresar descripción pedido producto.

Tarea	
Numero: T26	Número de Historia: H13

Nombre: Implementación del servicio para ingresar descripción pedido producto.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el servicio CarritoService que guardará temporalmente en la base de datos los productos seleccionados.	

Elaborado por: El Investigador

Historia de usuario: Mostrar pedidos

Tabla 65. Diseño de interfaz para mostrar pedidos

Tarea	
Numero: T27	Número de Historia: H14
Nombre: Diseño de interfaz para mostrar pedidos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde se muestra los pedidos con estado enviado y asignado, lo cuales deben estar por orden de fecha y hora.	

Elaborado por: El Investigador

Tabla 66. Implementación del servicio para mostrar pedidos

Tarea	
Numero: T28	Número de Historia: H14
Nombre: Implementación del servicio para mostrar pedidos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método en el servicio firestore y consumirlo el servicio para mostrar pedidos.	

Elaborado por: El Investigador

Historia de usuario: Asignar motorizado

Tabla 67. Diseño de interfaz para asignar motorizado

Tarea	
Numero: T29	Número de Historia: H15
Nombre: Diseño de interfaz para asignar motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde se pueda seleccionar a un motorizado activo, el cual debe visualizar foto, nombre y apellido del motorizado, a la par el	

pedido en camino, pedidos asignados, la dirección donde se encuentra y el tiempo aproximado de entrega.

Elaborado por: El Investigador

Tabla 68. Implementación del servicio para asignar motorizado

Tarea	
Numero: T30	Número de Historia: H15
Nombre: Implementación del servicio para asignar motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 7
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el servicio motorizado. Desarrollar métodos para consumir servicios para la mostrar motorizados, pedidos asignados, pedidos en camino, actualizar el pedido, pedidos en camino y su ubicación.	

Elaborado por: El Investigador

Historia de usuario: Mostrar pedidos asignados

Tabla 69. Diseño de interfaz para mostrar pedidos asignados

Tarea	
Numero: T31	Número de Historia: H16
Nombre: Diseño de interfaz para mostrar pedidos asignados	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde se muestra los pedidos con estado asignado y terminado, lo cuales deben estar por orden de fecha y hora.	

Elaborado por: El Investigador

Tabla 70. Implementación del servicio para mostrar pedidos asignados

Tarea	
Numero: T32	Número de Historia: H16
Nombre: Implementación del servicio para mostrar pedidos asignados	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir el servicio para mostrar pedidos asignados y terminados de la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Actualizar estado de pedido

Tabla 71. Diseño de interfaz para actualizar estado pedido

Tarea	
Numero: T33	Número de Historia: H17
Nombre: Diseño de interfaz para actualizar estado pedido	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde pueda seleccionar el pedido y cambiar el estado a CAMINO o ENTREGADO.	

Elaborado por: El Investigador

Tabla 72. Implementación del servicio para actualizar estado pedido

Tarea	
Numero: T34	Número de Historia: H17
Nombre: Implementación del servicio para actualizar estado pedido	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir el servicio para actualizar pedidos en la base de datos.	

Elaborado por: El Investigador

Tabla 73. Implementación del servicio de notificaciones

Tarea	
Numero: T35	Número de Historia: H17
Nombre: Implementación del servicio de notificaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 6
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Configurar función de firebase para notificaciones push. Desarrollar el servicio NotificacionesService. Desarrollar el método para consumir el servicio.	

Elaborado por: El Investigador

Historia de usuario: Mostar ubicación motorizado

Tabla 74. Mostar ubicación motorizado

Tarea	
Numero: T36	Número de Historia: H18

Nombre: Mostar ubicación motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se visualizara la ubicación del motorizado en el mapa con un icono de ubicación y el nombre.	

Elaborado por: El Investigador

Tabla 75. Implementación del servicio para mostrar ubicación motorizado

Tarea	
Numero: T37	Número de Historia: H18
Nombre: Implementación del servicio para mostrar ubicación motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 9
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Configurar API'S de google maps y plugin Capacitor Geolocalización. Desarrollar métodos para obtener la ubicación actual del motorizado y para consumir servicios.	

Elaborado por: El Investigador

Historia de usuario: Recorrido diario del motorizado

Tabla 76. Diseño de interfaz para mostrar recorrido diario del motorizado

Tarea	
Numero: T38	Número de Historia: H19
Nombre: Diseño de interfaz para mostrar recorrido diario del motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde se visualice los pedidos realizados por ese día.	

Elaborado por: El Investigador

Tabla 77. Implementación del servicio para mostrar recorrido diario del motorizado

Tarea	
Numero: T39	Número de Historia: H19
Nombre: Implementación del servicio para mostrar recorrido diario del motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 2021	Fecha fin: 2021

Descripción: Desarrollar el método en firebase, consumir para mostrar los pedidos diarios de la base de datos.

Elaborado por: El Investigador

Historia de usuario: Calificar motorizado

Tabla 78. Diseño de interfaz para calificar motorizado

Tarea	
Numero: T40	Número de Historia: H20
Nombre: Diseño de interfaz para calificar motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde el cliente pueda calificar al motorizado de la entrega de pedido.	

Elaborado por: El Investigador

Tabla 79. Implementación del servicio para calificar motorizado

Tarea	
Numero: T41	Número de Historia: H20
Nombre: Implementación del servicio para calificar motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir el servicio para actualizar el pedido con la calificación del motorizado en la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Reporte pedidos

Tabla 80. Diseño de interfaz para mostrar reporte pedidos

Tarea	
Numero: T42	Número de Historia: H21
Nombre: Diseño de interfaz para mostrar pedidos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde se visualice los pedidos realizados por ese día y podrá elegir el rango de fechas para la consulta.	

Elaborado por: El Investigador

Tabla 81. Implementación del servicio para mostrar reporte pedidos

Tarea	
Numero: T43	Número de Historia: H21
Nombre: Implementación del servicio para mostrar reporte pedidos	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método en firebase, consumir para mostrar los pedidos por rango de fechas de la base de datos.	

Elaborado por: El Investigador

Historia de usuario: Reporte motorizados

Tabla 82. Diseño de interfaz para mostrar reporte motorizados

Tarea	
Numero: T44	Número de Historia: H22
Nombre: Diseño de interfaz para mostrar reporte motorizados	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará una pantalla donde se visualice motorizados con el detalle de números pedidos entregados, asignados, en camino.	

Elaborado por: El Investigador

Tabla 83. Implementación del servicio para mostrar reporte motorizados

Tarea	
Numero: T45	Número de Historia: H22
Nombre: Implementación del servicio para mostrar reporte motorizados	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para consumir el servicio para mostrar motorizados.	

Elaborado por: El Investigador

Historia de usuario: Crear menú para el acceso de módulos

Tabla 84. Diseño de interfaz para mostrar menú

Tarea	
Numero: T46	Número de Historia: H23

Nombre: Diseño de interfaz para mostrar recorrido diario del motorizado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se diseñará un menú con todos los módulos de la aplicación y la foto, nombre y apellido del usuario.	

Elaborado por: El Investigador

Tabla 85. Implementación del menú

Tarea	
Numero: T47	Número de Historia: H23
Nombre: Implementación del menú	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Desarrollar el método para mostrar y cerrar el menú. Consumir el servicio para mostrar información del usuario.	

Elaborado por: El Investigador

Historia de usuario: Subir el software a un hosting de firebase

Tabla 86. Desplegar la aplicación en el hosting de firebase

Tarea	
Numero: T48	Número de Historia: H24
Nombre: Desplegar la aplicación en el hosting de firebase	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 2021	Fecha fin: 2021
Descripción: Se desplegará la aplicación en un hosting gratuito de firebase, el cual tendrá un comportamiento PWA.	

Elaborado por: El Investigador

3.1.3.1.5 Estimación de Historias de Usuario

Después de identificar las historias de usuario con sus respectivas tareas, se procede con su estimación, determinando el tiempo aproximado en realizar las tareas, que permitan cumplir con los requerimientos planteados en las historias de usuario.

Considerando que se ha dedicado un total de 4 horas diarias dentro de los días laborables para la realización de las tareas definidas.

Tabla 87. Estimación de Historias de usuario

Código	Historia de usuario	Tiempo estimado	
		Días	Horas
H01	Pantalla de inicio	3	12
H02	Iniciar sesión	5	20
H03	Recuperar clave	2	8
H04	Registro cliente	7	28
H05	Actualizar perfil cliente	2	8
H06	Registrar motorizado	5	20
H07	Dar de baja motorizado	3	12
H08	Mostrar motorizados	2	8
H09	CRUD productos	4	16
H10	Actualizar perfil motorizado	2	8
H11	Mostrar productos (Tienda)	2	8
H12	Mostrar productos seleccionados (Carrito)	6	24
H13	Registro descripción producto pedido	1	4
H14	Mostar pedidos	3	12
H15	Asignar motorizado	8	32
H16	Mostrar pedidos asignados	3	12
H17	Actualizar estado de pedido	8	32
H18	Mostrar ubicación del motorizado	10	40
H19	Mostar recorrido diario	4	16
H20	Calificar motorizado	2	8
H21	Reporte pedidos	4	16
H22	Reporte motorizados	4	16
H23	Mostrar menú	1	4
Tiempo estimado		91	364

Elaborado por: El Investigador

3.1.3.1.6 Plan de Entrega

Para la elaboración del plan de entrega de la aplicación, se establece el cronograma considerando que cada iteración se realiza durante un periodo de cuatro semanas y la prioridad determinada para cada una de ella.

Tabla 88. Plan de Entrega

Código	Historia de usuario	Tiempo estimado		Iteraciones				
		Días	Horas	1	2	3	4	5
H01	Pantalla de inicio	3	12	X				
H02	Iniciar sesión	5	20	X				
H04	Registro cliente	7	28	X				
H06	Registrar motorizado	5	20	X				
H09	CRUD productos	4	16		X			
H11	Mostrar productos (Tienda)	2	8		X			
H12	Mostrar productos seleccionados (Carrito)	6	24		X			
H13	Registro descripción producto pedido	1	4		X			
H14	Mostar pedidos	3	12		X			
H18	Mostrar ubicación del motorizado	10	40			X		
H15	Asignar motorizado	8	32			X		
H16	Mostrar pedidos asignados	3	12			X		
H17	Actualizar estado de pedido	8	32			X		
H08	Mostrar motorizados	2	8				X	
H05	Actualizar perfil cliente	2	8				X	
H10	Actualizar perfil motorizado	2	8				X	
H20	Calificar motorizado	2	8				X	
H19	Mostar recorrido diario	4	16				X	
H21	Reporte pedidos	4	16					X
H22	Reporte motorizados	4	16					X
H23	Crear menú para el acceso de módulos	1	4					X
H07	Dar de baja motorizado	3	12					X

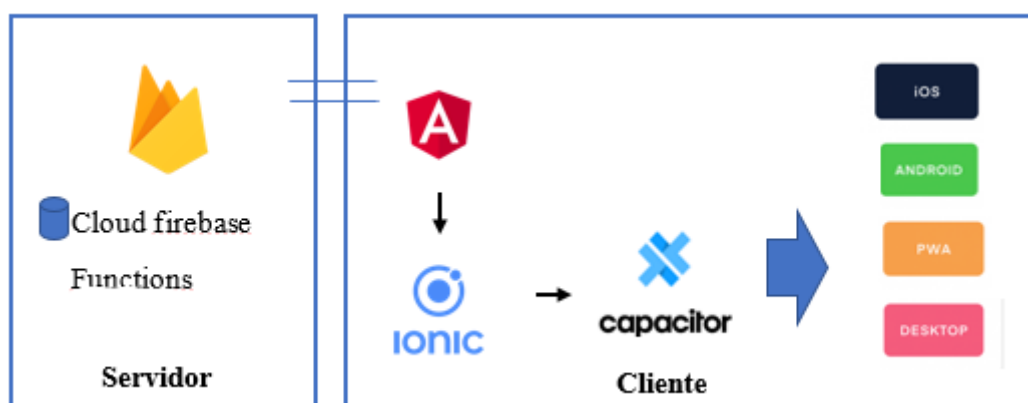
H03	Recuperar clave	2	8					X
-----	-----------------	---	---	--	--	--	--	---

Elaborado por: El Investigador

3.1.3.2 Fase II: Diseño

3.1.3.2.1 Arquitectura de la aplicación

La arquitectura de la aplicación se encuentra estructurada con la tecnología Ionic y Angular por parte de front end y firebase como back end, asegurando que la aplicación se ejecute en cualquier dispositivo móvil.



*Figura 1. Arquitectura de la aplicación
Elaborado por: El Investigador*

Se establece las herramientas que serán utilizadas para el desarrollo del proyecto.

Hardware

1 lapto HP, 6GB de memoria RAM, core i5 Sistema Operativo Windows 10.

1 Smartphone BISON Pro 8GB de RAM, Versión de android 11.0.

1 Smartphone Samsung J4 4GB de RAM, Version de android 9.0

Software

Para comenzar a trabajar con el framework Ionic es necesario instalar previamente Node y NPM. Node ya incluye NPM, el administrador de paquetes JavaScript. Además, se debe instalar un editor de código fuente en este caso utilizaremos Visual Studio Code.

Node.js

La instalación de Node.js es muy simple, se debe descargar de la siguiente página <https://nodejs.org/es/> dependiendo la arquitectura y sistema operativo del computador.



*Figura 2. Node.js
Elaborado por: El Investigador*

Visual Studio Code

La instalación de Visual Studio Code, se debe descargar de la siguiente página <https://code.visualstudio.com/>, dependiendo la arquitectura y sistema operativo del computador.



Visual Studio Code

*Elaborado por: El Investigador
Figura 3. Visual Studio Code*

Después de preparar el ambiente de trabajo se procede a crear el proyecto en Ionic 5.

Crear el proyecto

Instalar de forma global Ionic.

```
C:\Users\Sylvia>npm install -g @ionic/cli
```

*Figura 4. Instalar ionic global
Elaborado por: El Investigador*

En el transcurso de la creación del proyecto nos visualiza un menú para elegir el lenguaje de programación ente caso, se elegirá Angular, posteriormente nos preguntara si queremos incorporar al proyecto capacitor, para utilizar plugins nativos del

dispositivo móvil, una vez creado el proyecto debemos levantar el servidor donde, correrá el proyecto simultáneamente.

```
C:\Users\Sylvia\Desktop\Desarrollo>ionic start gabitos blank
```

*Figura 5. Crear el proyecto
Elaborado por: El Investigador*

```
C:\Users\Sylvia\Desktop\Desarrollo>ionic serve
```

*Figura 6. Correr el proyecto
Elaborado por: El Investigador*

Después de construir la aplicación se procederá a realizar un build para añadir a las plataformas Android y IOS.

```
PS C:\Users\Sylvia\Desktop\DESARROLLOFINAL> ionic build --prod
```

*Figura 7. Realizar un build a la aplicación
Elaborado por: El Investigador*

```
PS C:\Users\Sylvia\Desktop\DESARROLLOFINAL> npx cap add ios
```

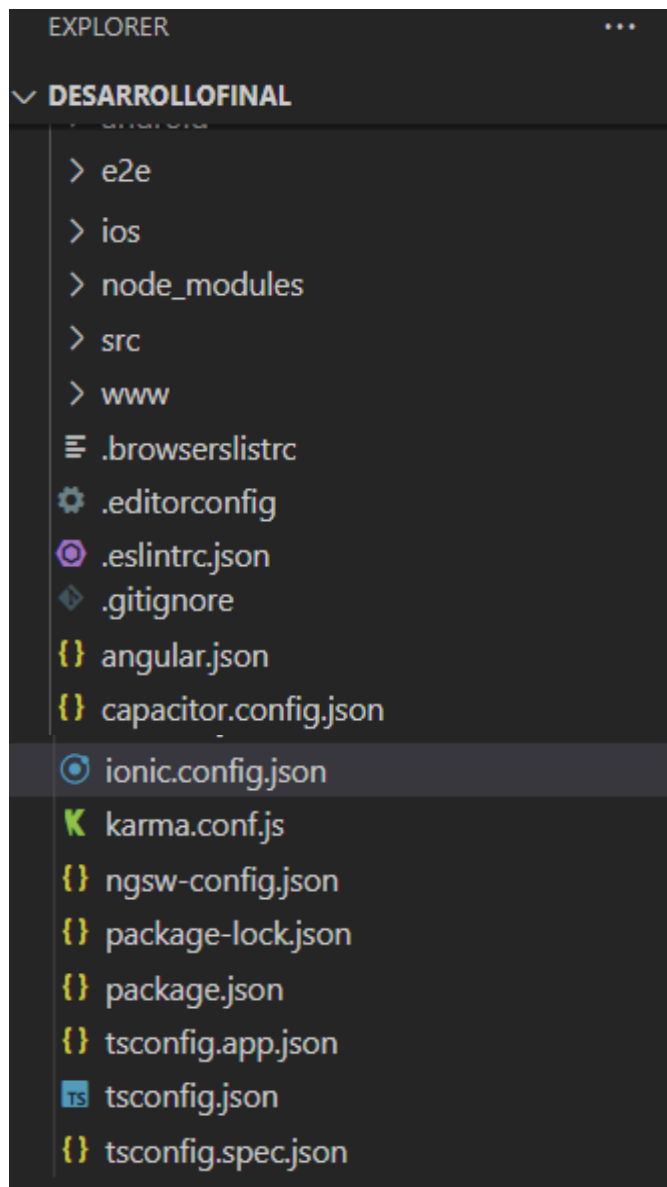
*Figura 8. Agregar a la plataforma Android
Elaborado por: El Investigador*

```
PS C:\Users\Sylvia\Desktop\DESARROLLOFINAL> npx cap add android
```

*Figura 9. Agregar a la plataforma IOS
Elaborado por: El Investigador*

Modelo y estructura de la aplicación

Ya iniciada la aplicación la estructura del proyecto la define el framework, la cual será detallada a continuación:



*Figura 10. Estructura del Proyecto Ionic
Elaborado por: El Investigador*

- **android:** Es un proyecto nativo de android totalmente independiente y el mismo puede ser editado usando una herramienta nativa.
- **e2e:** En esta carpeta se encuentra el código para escribir tests end to end que prueben la aplicación, no nos vamos a preocupar de esta carpeta.
- **IOS:** Es un proyecto nativo de IOS totalmente independiente y el mismo puede ser editado usando una herramienta nativa.
- **node_modules:** Esta carpeta se genera automáticamente al instalar las dependencias npm, que contiene las dependencias o librerías necesarias para el

funcionamiento de la aplicación, estas vienen definidas en el `package.json`. No se necesita tocar nada en esta carpeta.

- **src:** carpeta conformada por archivos fundamentales para el desarrollo del proyecto, mismo que son modelos, interfaces, servicios o imágenes.
- **www:** esta carpeta se genera cuando se realiza un build a la aplicación, contiene todos los archivos de una aplicación web, con Angular.

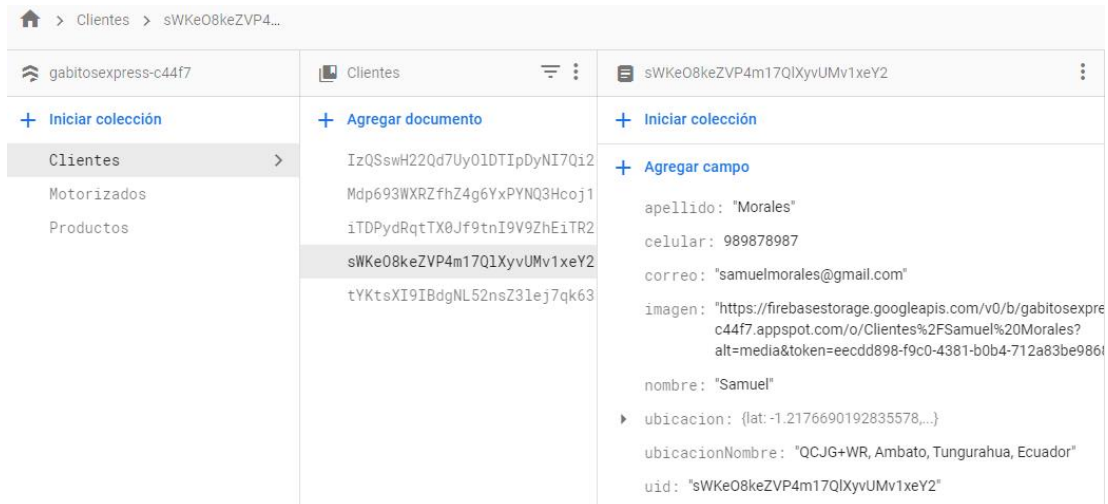
Se detalla otros archivos importantes para el desarrollo de la aplicación:

- **angular.json:** Archivo de configuración de la app.
- **Ionic.config.json:** Archivo que contiene información básica sobre la configuración nuestro proyecto.
- **karma.conf.js:** Es el archivo de configuración de karma, se utiliza para la realización de test unitarios.
- **package.json:** Contiene paquetes y dependencias de nodeJS.
- **tsconfig.json:** Es el archivo que contienen información necesaria a la hora de compilar TypeScript, no se necesita editar este archivo.

3.1.3.2.2 Diseño de la base de datos

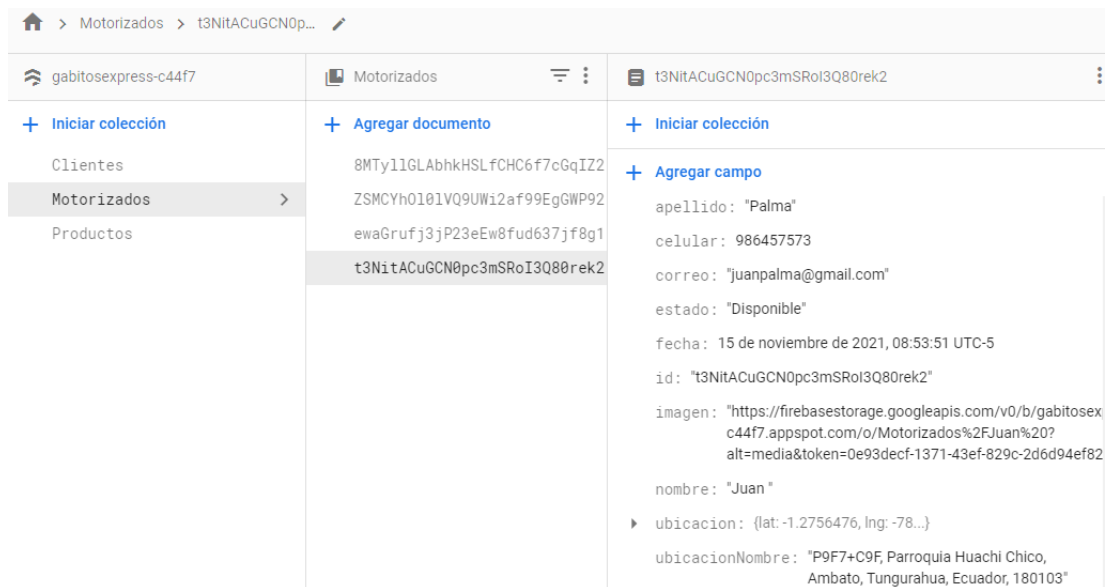
Para el desarrollo de la aplicación en cuanto a la base de datos se utilizó la base de datos NoSql de Firebase, ya por requerimiento de la aplicación es de tiempo real y lo misma nos ofrece el acceso desde cualquier dispositivo.

Cientes: Colección que almacena la información de clientes.



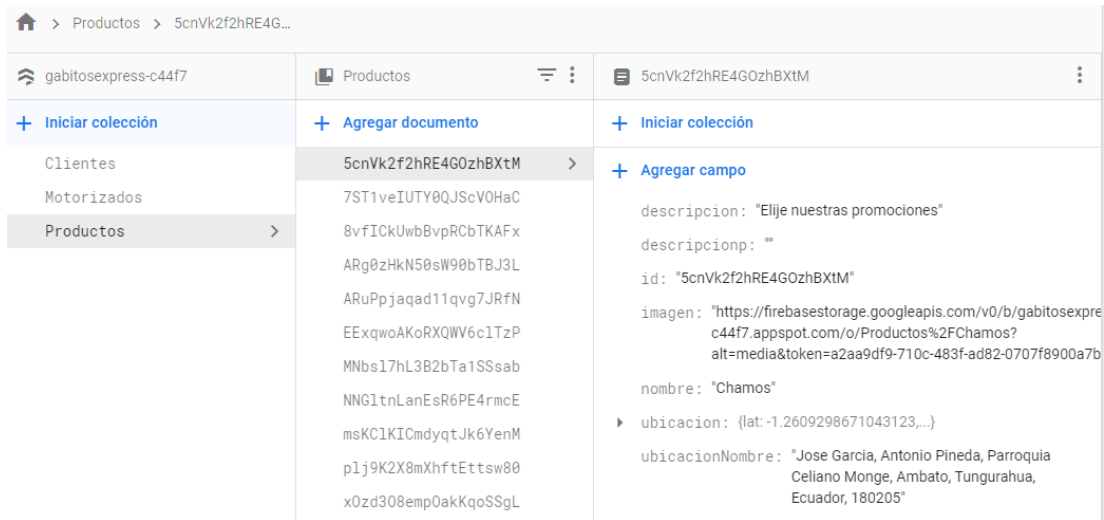
*Figura 11. Colección de clientes
Elaborado por: El Investigador*

Motorizados: Colección que almacena la información de motorizados.



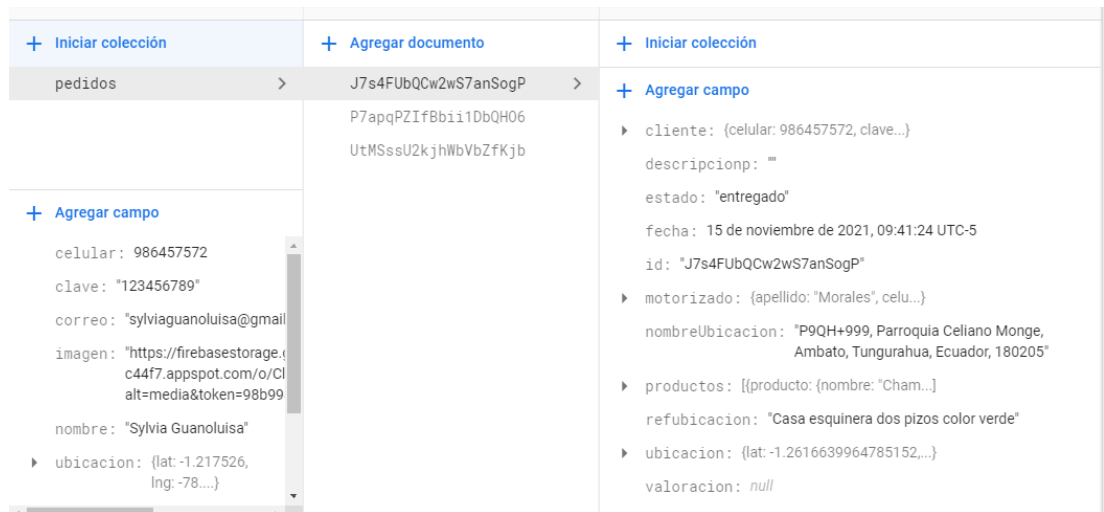
*Figura 12. Colección de motorizados
Elaborado por: El Investigador*

Productos: Colección que almacena la información de productos.



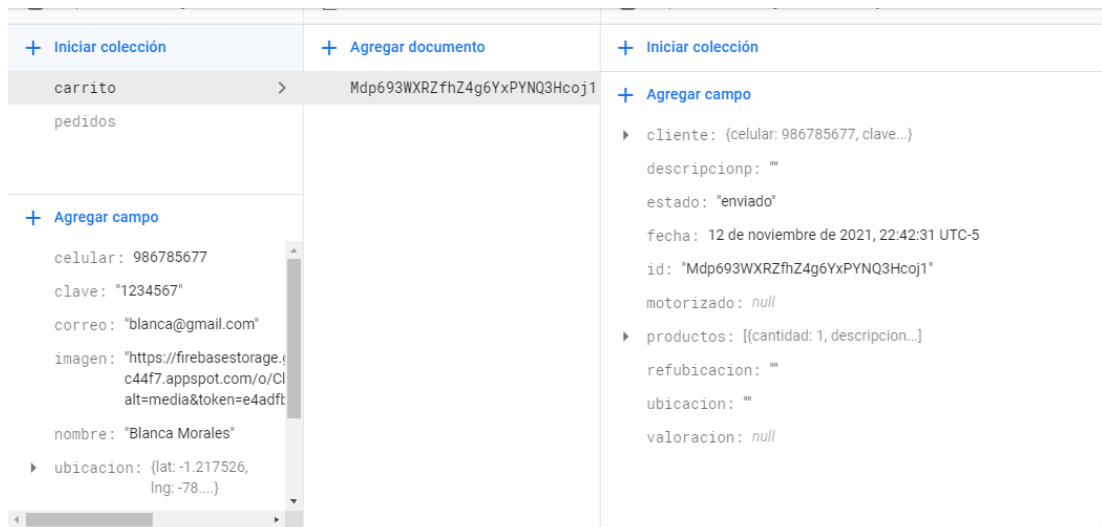
*Figura 13. Colección de productos
Elaborado por: El Investigador*

Pedidos: Es una sub colección que almacena la información de todos los pedidos del usuario.



*Figura 14. Sub-colección de pedidos
Elaborado por: El Investigador*

Carrito: Es una sub colección almacenada temporalmente, la misma que se borra cuando se confirma y el pedido es registrado.



*Figura 15. Sub-colección de carrito
Elaborado por: El Investigador*

Reglas de seguridad: Se codifican para controlar el ingreso a la aplicación. El administrador es cliente el cual tendrá el absoluto control de toda la aplicación.

```
1 rules_version = '2';
2 service cloud.firestore {
3
4     match /Productos/{documents=**} {
5         allow read;
6         allow write: if request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
7     }
8
9     match /{path=**}/pedidos/{pedidosId} {
10        allow read: if request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
11        allow write: if request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
12    }
13
14    match /Clientes/{userId}/carrito/{documents=**} {
15        allow read: if request.auth.uid == userId || request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
16        allow write: if request.auth.uid == userId || request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
17    }
18
19    match /Motorizados/{userId} {
20        allow read;
21        allow write: if request.auth.uid == userId || request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
22    }
23
24    match /Motorizados/{userId}/pedidos/{pedidosId} {
25        allow read: if request.auth.uid == userId || request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
26        allow write: if request.auth.uid == userId || request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
27    }
28
29    match /Clientes/{userId}/pedidos/{pedidosId} {
30        allow read: if request.auth.uid == userId || request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
31        allow write: if request.auth.uid == userId || request.auth.uid == 'tYKtsXI9IBdgNL52nsZ31ej7qk63'
32    }
33
34    match /Productos/{documents=**} {
35        allow read: if request.auth != null
36        allow write: if request.auth != null
37    }
38
39    match /Clientes/{userId} {
40        allow read: if request.auth != null
41        allow write: if request.auth != null
42    }
```

*Figura 16. Reglas firebase
Elaborado por: El Investigador*

3.1.3.2.3 Tarjetas CRC (Clase-Responsabilidad-Colaboración)

Las tarjetas CRC esta una herramienta fundamental en la Metodología Xp, para el diseño de software orientado por objetos, en la cual se describe las actividades y responsabilidades a cumplir.

En las siguientes tablas se muestran las diferentes clases mostrados en las tarjetas CRC:

Tabla 89. Tarjeta CRC - Ubicación

Clase: Ubicación	
Responsabilidades	Colaboradores
Agregar ubicación Modificar ubicación Listar ubicación	

Elaborado por: El Investigador

Tabla 90. Tarjetas CRC - Cliente

Clase: Cliente	
Responsabilidades	Colaboradores
Agregar cliente Actualizar cliente Listar cliente	Ubicación

Elaborado por: El Investigador

Tabla 91. Tarjeta CRC - Producto

Clase: Producto	
Responsabilidades	Colaboradores
Agregar producto Actualizar producto Listar producto Eliminar producto	Ubicación

Elaborado por: El Investigador

Tabla 92. Tarjeta CRC - Autenticación

Clase: Autenticación	
Responsabilidades	Colaboradores
Iniciar sesión Validar autenticación usuario Recuperar clave	Cliente Motorizado

Elaborado por: El Investigador

Tabla 93. Tarjeta CRC - Carrito

Clase: Carrito	
Responsabilidades	Colaboradores
Agregar carrito	Producto Cliente

Elaborado por: El Investigador

Tabla 94. Tarjeta CRC - Pedido

Clase: Pedido	
Responsabilidades	Colaboradores
Agregar pedido Actualizar pedido Listar pedido	Carrito Ubicación Motorizado

Elaborado por: El Investigador

Tabla 95. Tarjeta CRC - Motorizado

Clase: Ubicación - Motorizado	
Responsabilidades	Colaboradores
Agregar motorizado Modificar motorizado Listar motorizado	Ubicación

Elaborado por: El Investigador

Tabla 96. Tarjeta CRC – Calificación

Clase: Calificación	
Responsabilidades	Colaboradores
Agregar calificación Listar calificación	Pedido

Elaborado por: El Investigador

3.1.3.2.4 Diseño de Interfaces

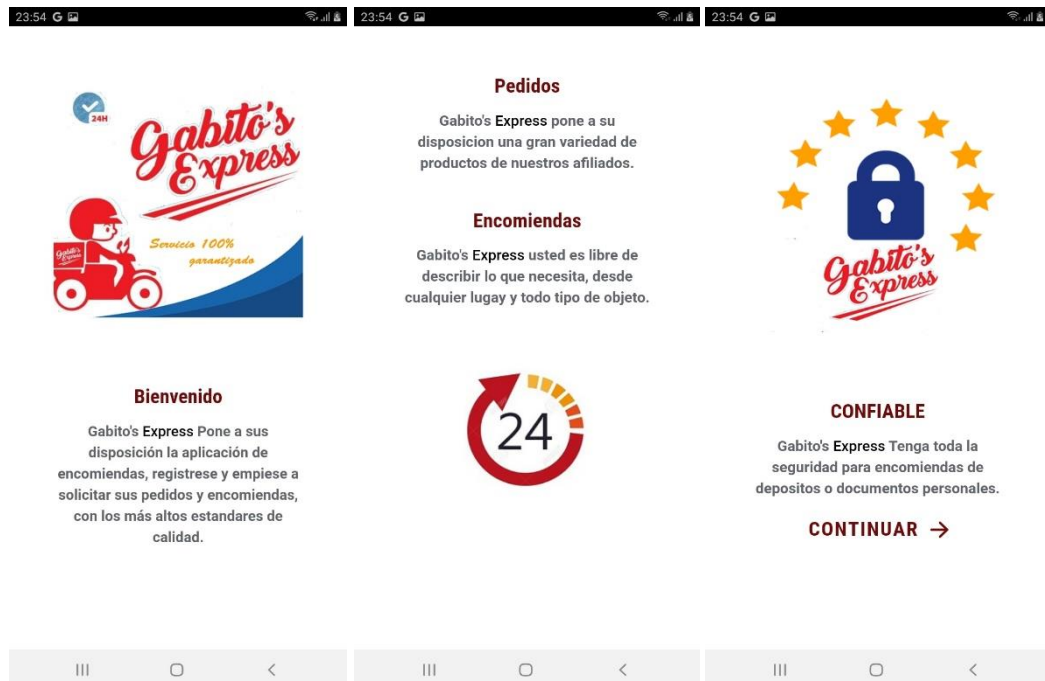


Figura 17. Interfaz de inicio de la aplicación
Elaborado por: El Investigador



Figura 18. Interfaz para iniciar sesión
Elaborado por: El Investigador

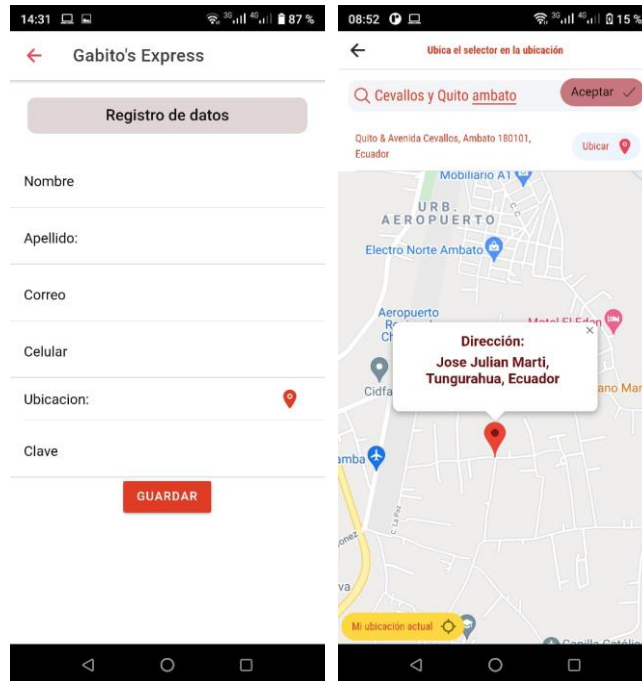


Figura 19. Interfaz para registrar el cliente
Elaborado por: El Investigador



Figura 20. Interfaz para registrar motorizado
Elaborado por: El Investigador



Figura 21. Interfaz para mostrar productos
Elaborado por: El Investigador



Figura 22. Interfaz del carro de compras
Elaborado por: El Investigador

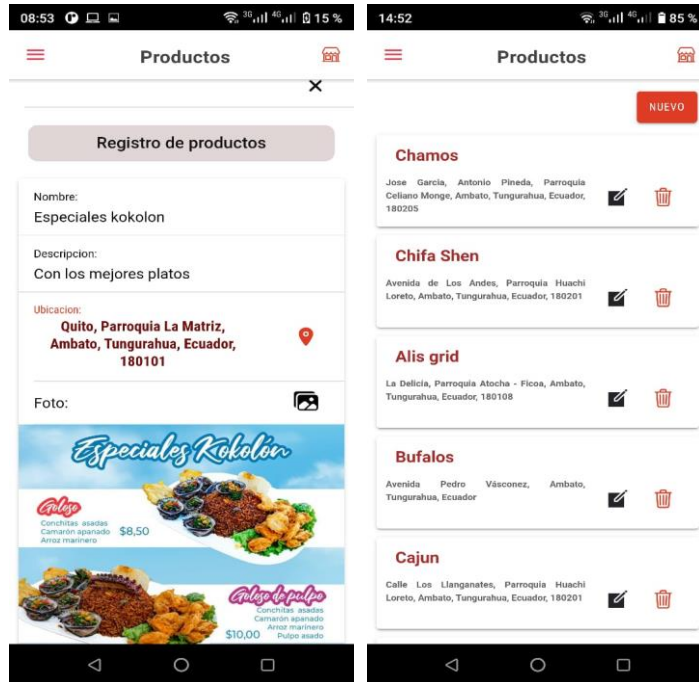


Figura 23. Interfaz para el CRUD Productos
Elaborado por: El Investigador



Figura 24. Interfaz para visualizar la ubicación del motorizado
Elaborado por: El Investigador

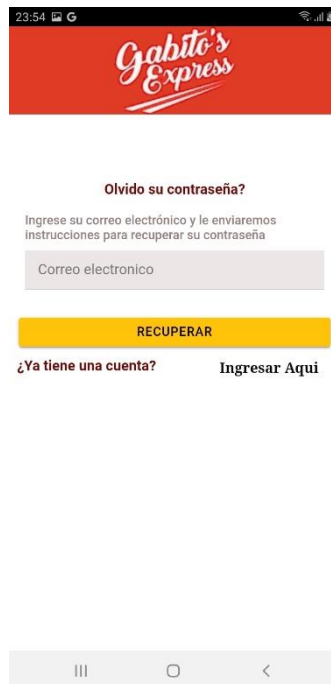


Figura 25. Interfaz para recuperar clave
Elaborado por: El Investigador

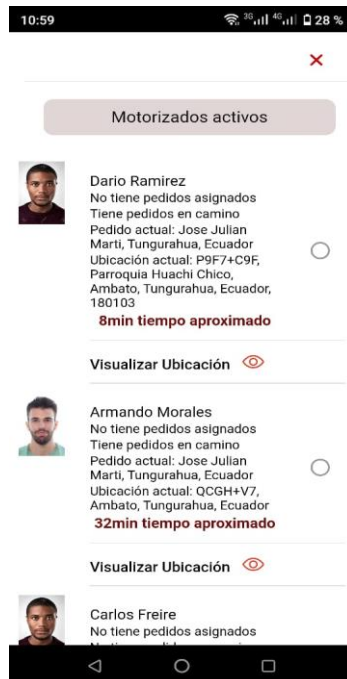


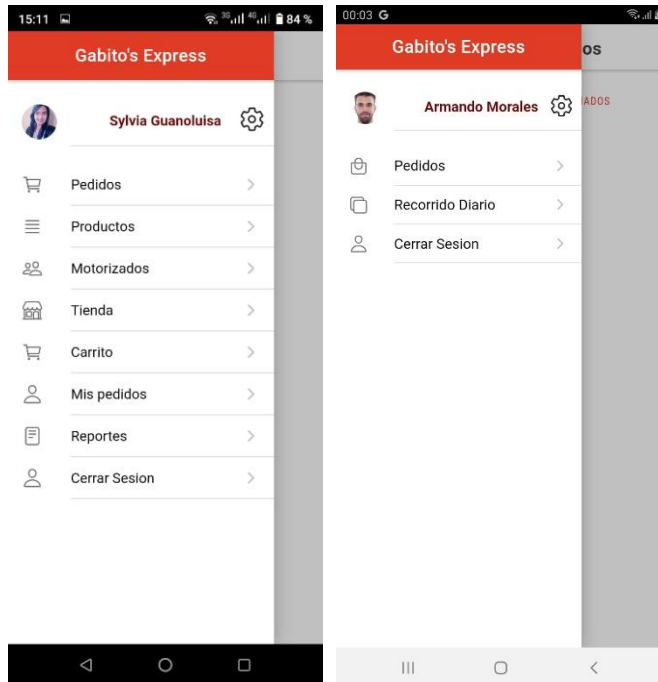
Figura 26. Interfaz para asignar el motorizado
Elaborado por: El Investigador



*Figura 27. Interfaz para mostrar los pedidos
Elaborado por: El Investigador*



*Figura 28. Interfaz de pedidos asignados
Elaborado por: El Investigador*



*Figura 29. Interfaz de Menú
Elaborado por: El Investigador*

3.1.3.3 Fase III: Codificación

A continuación se muestra la codificación de métodos e implementación de herramientas de firebase, en base a los requerimientos expuestos.

3.1.3.3.1 Modelos

Se creó modelos.ts para trabajar con la información de la base de datos.

```
export interface Cliente{
  uid: string;
  correo: string;
  nombre: string;
  apellido: string;
  celular: string;
  ubicacion?: any;
  ubicacionNombre?: string;
  imagen: string;
}

export interface Motorizado{
  nombre: string;
  apellido: string;
  correo: string;
  celular: number;
  estado: estadoMotorizado,
  id: string;
  fecha: Date;
  imagen: string;
  ubicacion?: any;
  ubicacionNombre?: string;
  distancia?: number;
  tiempo?: string;
}
```

```

export interface Producto{
  nombre: string;
  descripcion: string;
  ubicacion: any;
  ubicacionNombre?: string;
  id: string;
  imagen: string;
}

export interface Pedido{
  id: string;
  cliente: Cliente;
  productos: ProductoPedido[];
  motorizado: Motorizado;
  estado: estadoPedido;
  fecha: any;
  descripcionp: string;
  ubicacion: any;
  nombreUbicacion?: string;
  refubicacion: string;
  valoracion: number;
}

export interface ProductoPedido{
  producto: Producto;
  descripcionp: string;
  cantidad: number;
}

export type estadoPedido = 'enviado' | 'asignado' | 'camino' | 'entregado';
export type estadoMotorizado = 'Disponible' | 'No disponible';

```

*Figura 30. Código modelos
Elaborado por: El Investigador*

3.1.3.3.2 Servicios

FirestoreService

En el servicio se codifica métodos para utilizar las funcionalidades de firebase para manipular la información de la base de datos.

```

//Crear un documento
crearDoc(data: any, path:string, id:string){
  const coleccion = this.database.collection(path);
  return coleccion.doc(id).set(data);
}

//Leer el documento
selectDoc<tipo>(path: string, id:string){
  const coleccion = this.database.collection<tipo>(path);
  return coleccion.doc(id).valueChanges();
}

//Eliminar documento
eliminarDoc<tipo>(path: string, id: string){
  const coleccion = this.database.collection(path);
  return coleccion.doc(id).delete();
}

```

```

//Actualizar documento
editarDoc (data: any, path: string, id: string){
  const coleccion = this.database.collection(path);
  return coleccion.doc(id).update(data);
}

//Crear id automatico
getId(){
  return this.database.createId();
}

//Seleccionar todos los documentos
getColeccion<tipo>(path: string){
  const coleccion = this.database.collection<tipo>(path);
  return coleccion.valueChanges();
}

//Seleccionar una coleccion
getColeccionQuery<tipo>(path: string, parametro: string, condicion: any, busqu
  const coleccion = this.database.collection<tipo>(path,
    ref => ref.where(parametro, condicion, busqueda));
  return coleccion.valueChanges();
}

//Selecciona un grupo de colecciones
getColeccionAll<tipo>(path, parametro: string, condicion: any, busqueda: string, s
  if (startAt == null) {
    startAt = new Date();
  }
  const collection = this.database.collectionGroup<tipo>(path,
    ref => ref.where( parametro, condicion, busqueda)
    .orderBy('fecha', 'desc')
    .limit(3)
    .startAfter(startAt)
  );
  return collection.valueChanges();
}

getColeccionFecha<tipo>(path: string, fechaInicio:Date, fechaFin:Date ){
  const coleccion = this.database.collection<tipo>(path, ref => ref
    .where('fecha', '>=', fechaInicio)
    .where('fecha', '<=', fechaFin));
  console.log("despues del where", fechaInicio, fechaFin);
  return coleccion.valueChanges();
}

```

*Figura 31. Servicio firestore
Elaborado por: El Investigador*

FirestoreService

En el servicio se codifica el método para subir imágenes a Storage de firebase.

```

subirImagen(file:any, path: string, nombre: string): Promise<string> {
  return new Promise( resolve => {
    const filePath= path + '/' + nombre;
    const ref = this.firestorestorage.ref(filePath);
    const task = ref.put(file);
    task.snapshotChanges().pipe(
      finalize (() => {
        ref.getDownloadURL().subscribe(res => {
          const downloadUrl = res;
          resolve(downloadUrl);
          return;
        });
      })
    ).subscribe();
  });
}
}

```

*Figura 32. Servicio Firestorestorage
Elaborado por: El Investigador*

FirestoreauthService

En el servicio se codifica métodos para la autentificación con funciones de firebase.

```

//Login por medio de correo y clave
login(email: string, password: string){
  return this.auth.signInWithEmailAndPassword(email, password);
}

//Cerrar sección
logout(){
  this.auth.signOut();
}

//Registrar por medio de correo electrónico
registrar(email: string, password: string){
  return this.auth.createUserWithEmailAndPassword(email, password);
}

//Recuperar clave por email
recuperarClave(email: string,){
  return this.auth.sendPasswordResetEmail(email);
}

//Obtener el Uid del usuario
async getUid(){
  const user = await this.auth.currentUser;
  if(user === null){
    return null;
  }else{
    console.log("getUid",user.uid);
    return user.uid;
  }
}

//Obtener estado del usuario
estadoAuth(){
  console.log(this.auth.authState);
  return this.auth.authState;
}
}

```

*Figura 33. Servicio Firestoreauth
Elaborado por: El Investigador*

CarritoService

En el servicio se codifica el método para la gestión de productos – carrito de la base de datos.

```
loadcarrito(){
  const path = 'Clientes/' + this.uid + '/' + this.path;
  this.carritoSuscriber = this.firestore.selectDoc<Pedido>(path, this.uid)
  .subscribe( res => {
    if(res){
      this.pedido = res;
      this.pedido$.next(this.pedido);
    }else{
      this.initCarrito();
    }
  });
}

getCarrito(): Observable<Pedido>{
  setTimeout(() => {
    this.pedido$.next(this.pedido);
  }, 100);
  return this.pedido$.asObservable();
}

addProducto(producto: Producto, des:string){
  this.presentLoading();
  const path = 'Clientes/' + this.uid + '/' + this.path;
  console.log("Path del cliente",path);
  const add: ProductoPedido ={
    cantidad: 1,
    descripcion: des,
    producto,
  };

  this.pedido.productos.push(add);

  console.log(this.pedido);
  this.firestore.crearDoc(this.pedido, path, this.pedido.id).then(res =>{
    this.loading.dismiss();
    this.presentToast('Guardado con éxito, Dirijase al carrito');
  }).catch(error => {
    this.presentToast('No se pudo guardar');
  });
}
```

*Figura 34. Servicio Carrito
Elaborado por: El Investigador*

GooglemapsService

En el servicio se codifica el método para la iniciar google maps.

```

init(renderer: any, document: any): Promise<any> {
  return new Promise((resolve) => {
    if (this.mapsLoaded) {
      console.log('google cargado')
      resolve(true);
      return;
    }
    const script = renderer.createElement('script');
    script.id = 'googleMaps';

    window['mapInit'] = () => {
      this.mapsLoaded = true;
      if (google) {
        console.log('google is cargado')
        this.directionsServiceTmp = new google.maps.DirectionsService();
        this.Geocoder = new google.maps.Geocoder();
      } else {
        console.log('google no se pude cargar')
      }
      resolve(true);
      return;
    }
    if(this.apiKey){
      script.src = 'https://maps.googleapis.com/maps/api/js?key=' + this.apiKey
    } else {
      script.src = 'https://maps.googleapis.com/maps/api/js?callback=mapInit';
    }
    renderer.appendChild(document.body, script);
  });
}

```

*Figura 35. Servicio Googlemaps
Elaborado por: El Investigador*

MotorizadoService

En el servicio se codifica métodos para iniciar la geolocalizacion delo motorizado.

```

async init(uid: string) {
  if (this.initialiced) { return; };
  console.log('Inicio servicio motorizado');
  this.initialiced = true;
  this.uid = uid;
  this.Geocoder = new google.maps.Geocoder();
  const options = {
    enableHighAccuracy: true,
    timeout: this.tiempo * 1000,
    maximumAge: 0
  }

  this.watchID = await Geolocation.watchPosition(options, (res) => {
    console.log('posicion actual motorizado ->', res);
    if (res) {
      const position = {
        lat: res.coords.latitude,
        lng: res.coords.longitude
      }
      this.updatePosition(position);
    }
  })
}

```



```

async updatePosition(position: any) {
  console.log("posicion actual del motorizado",position);
  const path = 'Motorizados/';
  const ubicacionNombre = await this.getUbicacionNombre(position);
  const updateDoc = {
    ubicacion: position,
    ubicacionNombre,
  };
  console.log('actualizar posicion-> ', updateDoc);
  this.firestoreService.editarDoc(updateDoc, path, this.uid).then( () => {
    console.log('posición del motorizado actualizada con éxito');
  });
}

destroy() {
  if (this.watchID) {
    Geolocation.clearWatch({id: this.watchID});
    this.initialiced = false;
  }
}
}

```

*Figura 36. Servicio motorizados
Elaborado por: El Investigador*

NotificacionesService

En el servicio se codifica métodos para iniciar las notificaciones push.

```

// Se otorga permiso para push notifications
inicializar(){
  if(this.platform.is('capacitor')){
    PushNotifications.requestPermissions().then(result => {
      if (result.receive === 'granted') {
        console.log('Concedido permiso');
        PushNotifications.register();
        this.addListeners();
      } else {
        console.log(result.receive)
      }
    });
  }else{
    console.log('No es movil');
  }
}

addListeners(){
  PushNotifications.addListener('registration', (token: Token) => {
    // alert('Push registration success, token: ' + token.value);
    this.guardarToken(token.value);
  });

  PushNotifications.addListener('registrationError', (error: any) => {
    alert('Error on registration: ' + JSON.stringify(error));
  });

  PushNotifications.addListener(
    'pushNotificationReceived',
    (notification: PushNotificationSchema) => {
      alert('Push received: ' + JSON.stringify(notification));
    },
  );
}

```

```

PushNotifications.addListener(
  'pushNotificationActionPerformed',
  (notification: ActionPerformed) => {
    alert('Push action performed: ' + JSON.stringify(notification));
    this.router.navigate(['/pedidos']);
    // this.router.navigate([notification.notification.data.enlace]);
  },
);

```

*Figura 37. Servicio Notificaciones
Elaborado por: El Investigador*

3.1.3.3.3 Métodos aplicación móvil

Se expone los métodos utilizados para consumir los servicios entre otros métodos que contribuyen para el desarrollo y buen funcionamiento de la aplicación.

Método para mostrar pedidos

Los siguientes métodos se consultan los pedidos en base al estado.

```

async getpedidosPendientes(){
  const path = 'Clientes/' + this.uid + '/pedidos/';
  console.log("getpedidosPendientes", path);
  this.nuevosSuscriber = this.firestore.getColeccionQuery<Pedido>
    (path, 'estado', '!=', 'entregado')
    .subscribe( res => {
      if(res.length){
        this.pedidos = res;
        console.log(this.pedidos);
      }
    });
}

async getpedidosTerminados(){
  const path = 'Clientes/' + this.uid + '/pedidos/';
  console.log("getpedidosTerminados", path);
  this.culmidadosSuscriber = this.firestore.getColeccionQuery<Pedido>
    (path, 'estado', '=', 'entregado')
    .subscribe( res => {
      if(res.length){
        this.pedidos = res;
      }
    });
}

```

*Figura 38. Método para mostrar pedidos
Elaborado por: El Investigador*

Método de autenticación

El método de autenticación podrá realizar por medio del correo electrónico y contraseña previamente realizados.

```

login(){
  this.isSubmitted = true;
  if (!this.usuarioFrom.valid) {
    console.log('Es necesario llenar los campos');
    return false;
  } else {
    this.presentLoading();
    const path = "Motorizados";
    const credenciales = {
      correo: this.usuarioFrom.controls.correo.value,
      clave: this.usuarioFrom.controls.clave.value,
    };

    this.firebaseauthservice.login(credenciales.correo, credenciales.clave)
    .then(res => {
      this.loading.dismiss();
      this.usuarioId = res.user.uid;
      this.subscripcionM = this.firestore.selectDoc<Motorizado>(path, this.usuarioId)
      .subscribe(resM => {
        console.log(resM);
        if(resM === undefined){
          if(this.usuarioId === 'tYKtsXI9IBdgNL52nsZ3lej7qk63'){
            this.router.navigate(["/pedidos"]);
          }else{
            this.router.navigate(["/mispedidos"]);
          }
        }
        }else if(resM !== null){
          if(this.motorizados.estado === "No disponible"){
            this.mensajeError = 'Usted no esta habilitado, comuniquese con el adminis';
            this.presentAlert(this.mensajeError);
          }else{
            this.router.navigate(["/pedidos-asignados"]);
          }
        }
      });
    }).catch(error => {
      this.loading.dismiss();
      const errorMensaje = error.code;
      switch(errorMensaje){
        case 'auth/user-not-found':
          this.mensajeError = "No hay ningún registro de usuario que corresponda a";
          break;
        case 'auth/wrong-password':
          this.mensajeError = "La contraseña no es válida o el usuario no tiene cont";
          break;
        default:
          this.mensajeError = "Es necesario ingresar todos los campos.";
          break;
      }
      this.presentAlert(this.mensajeError);
    });
  }
}

```

*Figura 39. Método de autenticación
Elaborado por: El Investigador*

Método de obtener la ubicación

Obtener la ubicación mediante el plugins de capacitor geolocalización.

```

async mylocation() {
  console.log('mylocation() click');
  Geolocation.getCurrentPosition().then((res) => {
    console.log('mylocation() -> get ', res);

    const position = {
      lat: res.coords.latitude,
      lng: res.coords.longitude,
    }
    this.addMarker(position);
  });
}

addMarker(position: any): void {
  let latLng = new google.maps.LatLng(position.lat, position.lng);
  this.marker.setPosition(latLng);
  this.map.panTo(position);
  this.positionSet = position;

  this.geocoder(position, this.marker);
}

```

*Figura 40. Método de obtener ubicación
Elaborado por: El Investigador*

3.1.3.3.4 Functions (Funciones de firebase)

Se implementa métodos para enviar notificaciones según el proceso del pedido utilizando Functions de firebase.

```

exports.newPedido = functions.firestore
  .document("Clientes/{userId}/pedidos/{pedidoId}")
  .onCreate(async (event) => {
    const pedido = event.data();
    const dataFcm = {
      enlace: "/pedidos",
    };
    const path = "/Clientes/"+uidAdmin;
    const docInfo = await firestore.doc(path).get();
    const dataUser = docInfo.data() as any;
    const token = dataUser.token;
    const registrationTokens = [token];
    const notificacion: INotification = {
      data: dataFcm,
      tokens: registrationTokens,
      notificacion: {
        title: pedido.cliente.nombre,
        body: "Nuevo pedido",
      },
    };
    return sendNotificacion(notificacion);
  });

```

```

});
console.log("List of tokens send", notificacion.tokens);
admin.messaging().sendMulticast[menssage]
.then((response) => {
  if (response.failureCount > 0) {
    const faildTokens: any[] = [];
    response.responses.forEach((resp, idx)=> {
      if (!resp.success) {
        faildTokens.push(notificacion.tokens[idx]);
      }
    });
    console.log("Lista de tokens fallidos", faildTokens);
  } else {
    console.log("Se envio con exito la notificaion");
  }
  resolve(true);
  return;
}).catch((error) => {
  console.log("Envio fallo", error);
  resolve(false);
  return;
});
});
});
};

const sendNotificacion = (notificacion: INotificacion) => {
  console.log("List of tokens send", notificacion.notificacion);
  return new Promise((resolve) => {
    const menssage: admin.messaging.MulticastMessage = {
      data: notificacion.data,
      tokens: notificacion.tokens,
      notification: notificacion.notificacion,
      android: {
        notification: {
          icon: "ic_stat_name",
          color: "#EB9234",
        },
      },
      apns: {
        payload: {
          aps: {
            sound: {
              critical: true,
              name: "default",
              volume: 1,
            },
          },
        },
      },
    },
  });
};

```

*Figura 41. Funciones de firebase
Elaborado por: El Investigador*

3.1.3.3.5 Desplegar el código en hosting firebase.

Antes de desplegar el código, se debe instalar los siguientes componentes.

```

PS C:\Users\Sylvia\Desktop\DESARROLLOFINAL\gabitosapp> ng add @angular/pwa

```

*Figura 42. Instalar herramienta angular PWA
Elaborado por: El Investigador*

```
PS C:\Users\Sylvia\Desktop\DESARROLLOFINAL\gabitosapp> firebase login
```

*Figura 43. Login en Firebase
Elaborado por: El Investigador*

```
PS C:\Users\Sylvia\Desktop\DESARROLLOFINAL\gabitosapp> firebase init
```

*Figura 44. Ejecutar firebase init
Elaborado por: El Investigador*

A continuación podemos visualizar un menú de opciones, las cuales seleccionaremos functions y hosting, para posteriormente desplegarlos.

```
PS C:\Users\Sylvia\Desktop\DESARROLLOFINAL\gabitosapp> firebase deploy --only hosting
```

*Figura 45. Despliegue hosting firebase
Elaborado por: El Investigador*



*Figura 46. Gabitos Express publicado
Elaborado por: El Investigador*

La siguiente línea de código se lo realiza para habilitar la persistencia, con la finalidad de utilizar la aplicación en el caso que se pierda la conexión a internet, la aplicación trabajara de manera normal hasta restablecer la conexión y sincronizar la información.

```
AngularFirestoreModule.enablePersistence(),
```

*Figura 47. Habilitar la persistencia
Elaborado por: El Investigador*

3.1.3.4 Fase IV: Pruebas

3.1.3.4.1 Pruebas de Aceptación

La etapa de pruebas es una de las más importantes en la metodología XP, ya que permite validar el correcto funcionamiento, identificando posibles fallos de parte del cliente.

Tabla 97. Prueba de inicio de sección

Prueba de aceptación	
Numero: 01	Número de Historia: 02
Nombre: Prueba de inicio de sección.	
Condiciones de ejecución: Iniciar sección.	
Proceso de ejecución: El usuario abre la aplicación y visualiza las pantallas informativas, debe dar clic en continuar. A continuación, el usuario debe ingresar el correo electrónico y contraseña. El usuario dará clic en el botón Ingresar.	
Resultado esperado: Se ejecuta la aplicación y posteriormente se visualiza la interfaz según el rol del usuario.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 98. Prueba de registrar cliente

Prueba de aceptación	
Numero: 02	Número de Historia: 04
Nombre: Prueba de registrar cliente	
Condiciones de ejecución: Llenar los campos expuestos en la interfaz.	
Proceso de ejecución: El usuario cliente podrá llenar los campos expuestos en la interfaz registro como nombre, email, celular, dirección y contraseña. El usuario dará click en el icono de ubicación donde visualizará el mapa de google maps. Posteriormente el usuario podrá dar click en cualquier parte del mapa o podrá dar click en el botón mí su ubicación actual, adicional puede buscar la dirección digitado el nombre del lugar a buscar. El usuario dará clic en el botón Guardar.	
Resultado esperado: Ingresa a la aplicación, dirigiéndose a la interfaz mis pedidos.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 99. Prueba de registrar motorizado

Prueba de aceptación	
Numero: 03	Número de Historia: 4
Nombre: Prueba de registrar motorizado	
Condiciones de ejecución: Iniciar sección con privilegios de administrador.	
<p>Proceso de ejecución: El usuario podrá debe llenar los campos expuestos en la interfaz registro como nombre, apellido, email, celular, foto y contraseña. El usuario podrá dar click en el icono de imagen donde debe elegir la foto del motorizado. Posteriormente el usuario dará click en la foto seleccionada. El usuario dará clic en el botón Guardar.</p>	
Resultado esperado: Se mostrará un mensaje “Guardado con éxito”.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 100. Prueba de CRUD productos

Prueba de aceptación	
Numero: 04	Número de Historia: 09
Nombre: Prueba de CRUD productos	
Condiciones de ejecución: Iniciar sección con privilegios de administrador.	
<p>Proceso de ejecución: El usuario ingresará y podrá visualizar los productos ya registrados. El usuario podrá dar click en el botón Nuevo para llenar los campos del nuevo producto. Posteriormente el usuario debe llenar los campos expuestos en la interfaz registro como nombre, descripción, dirección y foto; A continuación el usuario dará click en el botón Guardar. El usuario podrá dar click en botón editar ubicado en cada producto donde podrá cambiar los datos de ese producto. Posteriormente dará click en Guardar. El usuario podrá dar click en botón eliminar ubicado en cada producto, posteriormente podrá click en aceptar o cancelar esta acción.</p>	
Resultado esperado: Se mostrará un mensaje de éxito al guardar, actualizar o eliminar el producto.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 101. Prueba de mostrar productos. (Tienda)

Prueba de aceptación	
Numero: 05	Número de Historia: 11
Nombre: Prueba de mostrar productos. (Tienda)	
Condiciones de ejecución: Iniciar sección como cliente.	

<p>Proceso de ejecución: El usuario podrá visualizar todos los productos registrados. La pantalla visualizará la foto del producto y la ubicación del producto. Al dar click en el botón pedir, ubicado en la parte inferior de cada producto. El usuario debe ingresar la descripción del pedido producto. A continuación, el cliente dará click en guardar.</p>
<p>Resultado esperado: Se muestra un mensaje Guardado con éxito, Diríjase al carrito.</p>
<p>Evaluación de la prueba: La prueba se realizó de manera exitosa.</p>

Elaborado por: El Investigador

Tabla 102. Prueba de mostrar productos seleccionados (carrito)

Prueba de aceptación	
Numero: 06	Número de Historia: 12
Nombre: Prueba de mostrar productos seleccionados (carrito)	
Condiciones de ejecución: Iniciar sección como cliente.	
<p>Proceso de ejecución: El usuario cliente debe llenar los campos expuestos en la interfaz registro como dirección y referencia de ubicación. Posteriormente dará clic en el icono de ubicación donde se visualiza el mapa de google maps y seleccionar la ubicación. El usuario dará clic en el botón Pedir.</p>	
Resultado esperado: Se muestra un mensaje de Guardado con éxito. Se visualiza una notificación al usuario administrador del nuevo pedido.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 103. Prueba de registrar descripción producto pedido

Prueba de aceptación	
Numero: 07	Número de Historia: 13
Nombre: Registrar de descripción producto pedido	
Condiciones de ejecución: Iniciar sección como cliente.	
<p>Proceso de ejecución: El usuario cliente debe llenar el campo de descripción del producto a pedir. El usuario dará clic en el botón Guardar.</p>	
Resultado esperado: Se muestra un mensaje de Guardado con éxito.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 104. Prueba de mostrar pedidos

Prueba de aceptación	
Numero: 08	Número de Historia: 14
Nombre: Prueba de mostrar pedidos	
Condiciones de ejecución: Iniciar sección como administrador.	
Proceso de ejecución: El usuario ingresará y podrá visualizar los pedidos detallando la imagen, la descripción producto pedido y la dirección del pedido ya registrados. El usuario podrá dar click en la imagen donde se expande la imagen.	
Resultado esperado: Visualizar todos los pedidos registrados por orden.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 105. Prueba de mostrar ubicación motorizado

Prueba de aceptación	
Numero: 09	Número de Historia: 18
Nombre: Prueba de mostrar ubicación motorizado.	
Condiciones de ejecución: Iniciar sección como administrador.	
Proceso de ejecución: El usuario ingresará y podrá visualizar la ubicación del motorizado.	
Resultado esperado: Se visualiza el nombre y ubicación del motorizado.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 106. Prueba de asignar motorizado

Prueba de aceptación	
Numero: 10	Número de Historia: 15
Nombre: Prueba de asignar motorizado.	
Condiciones de ejecución: Iniciar sección como administrador, estar en la pantalla de pedidos pendientes.	
Proceso de ejecución: El usuario podrá visualizar la información del motorizado: Si mantiene pedidos pendientes, pedidos en camino y la ubicación actual del motorizado, además del tiempo aproximado de la entrega en el caso de ser asignado. El usuario dará click en el motorizado.	
Resultado esperado: Al dar clic el usuario se guardará automáticamente el motorizado y se cambiará el estado del pedido o encomienda. Se visualizará una notificación al motorizado asignado del nuevo pedido.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 107. Prueba de mostrar pedidos asignados

Prueba de aceptación	
Numero: 11	Número de Historia: 16
Nombre: Prueba de mostrar pedidos asignados	
Condiciones de ejecución: Iniciar sección como motorizado.	
Proceso de ejecución: El usuario ingresará y podrá visualizar los pedidos detallando la imagen, la descripción del pedido, así como la ubicación del pedido ya asignados. El usuario podrá dar click en la imagen donde se expande la imagen.	
Resultado esperado: Visualizar todos los pedidos asignados.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 108. Prueba de actualizar estado de pedido

Prueba de aceptación	
Numero: 12	Número de Historia: 17
Nombre: Prueba de actualizar estado de pedido.	
Condiciones de ejecución: Iniciar sección como motorizado.	
Proceso de ejecución: El usuario podrá dar click para cambiar el estado con opción a camino o entregado.	
Resultado esperado: Al dar clic en cualquiera de las opciones se guardará automáticamente. Se visualizará una notificación al cliente del estado del pedido.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 109. Prueba de mostrar motorizados

Prueba de aceptación	
Numero: 13	Número de Historia: 08
Nombre: Prueba de mostrar motorizados	
Condiciones de ejecución: Iniciar sección como administrador.	
Proceso de ejecución: El usuario ingresará y podrá visualizar los motorizados detallando la imagen, nombre y apellido.	
Resultado esperado: Visualizar todos los motorizados.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 110. Prueba de actualizar perfil cliente

Prueba de aceptación	
Numero: 14	Número de Historia: 05
Nombre: Prueba de actualizar perfil cliente	
Condiciones de ejecución: Iniciar sección como cliente.	
Proceso de ejecución: El usuario podrá cambiar los datos en cada campo expuesto en la interfaz. El usuario dará clic en el botón Guardar.	
Resultado esperado: Se mostrará un mensaje de Guardado con éxito.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 111. Prueba de actualizar perfil motorizado

Prueba de aceptación	
Numero: 15	Número de Historia: 10
Nombre: Prueba de actualizar perfil motorizado	
Condiciones de ejecución: Iniciar sección como motorizado.	
Proceso de ejecución: El usuario podrá cambiar los datos en cada campo expuesto en la interfaz. El usuario dará clic en el botón Guardar.	
Resultado esperado: Se mostrará un mensaje de Guardado con éxito.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 112. Prueba de calificar motorizado

Prueba de aceptación	
Numero: 16	Número de Historia: 20
Nombre: Prueba de calificar motorizado	
Condiciones de ejecución: Iniciar sección como cliente, el pedido se encuentre con estado terminado.	
Proceso de ejecución: El usuario ingresará y podrá dar click en calificar. A continuación, podrá elegir una opción entre excelente, bueno, regular y malo.	
Resultado esperado: al dar click el usuario en las opciones se guardará automáticamente.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 113. Prueba de recorrido diario

Prueba de aceptación	
Numero: 17	Número de Historia: 19
Nombre: Prueba de recorrido diario	
Condiciones de ejecución: Iniciar sección como motorizado.	
Proceso de ejecución: El usuario ingresará y podrá visualizar el recorrido realizado por ese día. Donde se detalla nombre, descripción, dirección y la calificación del pedido.	
Resultado esperado: Visualizar los pedidos diarios.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 114. Prueba de reporte pedidos

Prueba de aceptación	
Numero: 18	Número de Historia: 21
Nombre: Prueba de reporte pedidos	
Condiciones de ejecución: Iniciar sección como administrador.	
Proceso de ejecución: El usuario ingresará y podrá visualizar los pedidos del día actual, donde se detalla nombre, descripción, dirección y la calificación del pedido. Posteriormente el usuario dará clic en las fechas para cambiarlas y después clic en el botón buscar,	
Resultado esperado: Visualizar los pedidos diarios y por rango de fechas.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 115. Prueba de reporte motorizados

Prueba de aceptación	
Numero: 19	Número de Historia: 22
Nombre: Prueba de reporte motorizados	
Condiciones de ejecución: Iniciar sección como administrador.	
Proceso de ejecución: El usuario ingresará y podrá visualizar los motorizados y total de pedidos; donde se detalla pedidos asignados, terminados y en camino, Al dar clic en el icono se muestra los pedidos del motorizado.	
Resultado esperado: Visualizar los motorizados y sus pedidos.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 116. Prueba Mostrar menú

Prueba de aceptación	
Numero: 20	Número de Historia: 23
Nombre: Prueba Mostrar menú	
Condiciones de ejecución: Iniciar sección.	
Proceso de ejecución: El usuario podrá dar click en el icono superior izquierdo. Posteriormente el usuario visualizara una lista de módulos o actividades. El usuario dará clic en cualquier modulo o actividad dependiendo el rol.	
Resultado esperado: Se muestra el menú con la lista de módulos o actividades.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 117. Prueba dar de baja motorizado

Prueba de aceptación	
Numero: 21	Número de Historia: 5
Nombre: Prueba dar de baja motorizado.	
Condiciones de ejecución: Iniciar sección con privilegios de administrador.	
Proceso de ejecución: El usuario podrá dar click en estado. Posteriormente el usuario podrá seleccionar entre disponible o No disponible. El usuario dará clic en el botón Guardar.	
Resultado esperado: Se mostrará un mensaje “Guardado con éxito”.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

Tabla 118. Prueba recuperar clave

Prueba de aceptación	
Numero: 22	Número de Historia: 03
Nombre: Prueba recuperar clave	
Condiciones de ejecución: Ejecutar la aplicación.	
Proceso de ejecución: El usuario podrá dar click en recuperar clave. Posteriormente el usuario podrá ingresar el correo electrónico con el que se registró. El usuario dará clic en el botón Recuperar.	
Resultado esperado: Se mostrará un mensaje. Se enviara un enlace al correo electrónico para su recuperación. Este proceso es propio de firebase.	
Evaluación de la prueba: La prueba se realizó de manera exitosa.	

Elaborado por: El Investigador

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Del diagnóstico realizado se identificó que el proceso era manual, lo que conlleva a retrasos en todo el proceso, por no identificar la disponibilidad del motorizado de manera inmediata, al aceptar o entregar un pedido, al existir cuellos de botella en los pedidos entre otros, dando paso a definir las características fundamentales de la aplicación a desarrollar.

La investigación y fundamentación teoría aplicadas a este proyecto conjuntamente con la Metodología XP, facilitaron la construcción del proyecto por la constante comunicación con el cliente y las fases que ofrece la metodología. Mencionando que los cambios solicitados durante el desarrollo del proyecto se incluyeron de manera simultánea, de tal forma que el proyecto no fue afectado significativamente, gracias a la flexibilidad de la metodología.

El desarrollo de la aplicación móvil multiplataforma con el framework Ionic versión 5, al igual que las funciones de firebase y capacitor para utilizar los plugins que interactúan con los componentes propios de los dispositivos, permitió la automatización de los procesos de manera óptima. Adicional se desplegó una PWA en el hosting de firebase para los clientes con ciertas funcionalidades, el cual incorpora un Certificado SSL de manera automática, de tal forma que se entregue el contenido de manera segura.

La geolocalización como requerimiento de la aplicación, fue prioridad en el desarrollo del proyecto, implementando varias API's de Google Maps Platform, donde se visualiza el mapa con opción a elegir o buscar una dirección, cumpliendo así con los requerimientos de la aplicación y los objetivos propuestos en el proyecto.

4.2 Recomendaciones

Investigar nuevas tecnológicas de desarrollo o Frameworks, de tal forma que faciliten y minimicen tiempos de desarrollo, en la actualidad se puede mencionar Ionic, React, Vue, Xamarin, con documentación amplia para su implementación. De igual manera se recomienda verificar y actualizar versiones de las librerías, para evitar así problemas al momento de ejecutar la aplicación.

Se recomienda adquirir un hosting propio, con el fin de visualizar el nombre de la empresa para darse a conocer mediante redes sociales u otros medios, además de obtener almacenamiento ilimitado de Storage para guardar archivos multimedia en firebase.

Mantenerse informado sobre las reglas y normativas establecidas por Google en cuanto a las API's utilizadas, y de la misma manera con los servicios de firebase, con el fin de evitar problemas de funcionamiento o posibles restricciones.

Bibliografía

- [1] I. N. de E. y Censos, “de la Información y Comunicación Encuesta Multipropósito-TIC 2019,” 2019.
- [2] R. Cedeño Luna, K. Alcívar Vaca, and D. Ponce Vásquez, “Observaciones acerca de los dispositivos móviles,” *Dominio las Ciencias*, vol. 3, no. 4, pp. 89–103, 2017, doi: 10.23857/dom.cien.pocaip.2017.3.4.oct.89-103.
- [3] J. Stoldt, T. Uwe Trapp, and Toussai, “APP MOVIL PARA SUPERVISAR PERSONAS CON ALZHEIMER REGISTRADOS EN EL CENTRO DE SALUD SAN AGUSTIN, PARROQUIA DR. CAMILO PONCE – BABAHOYO.,” vol. 22, pp. 1–8, 2019.
- [4] J. Cuello, *Diseñando apps para móviles*. 2013.
- [5] R. Vicente Ruano Valenzuela, “UNIVERSIDAD TÉCNICA DEL NORTE ‘ESTUDIO COMPARATIVO DE LOS FRAMEWORKS IONIC Y REACT NATIVE.’”
- [6] Strapp Inc, “¿Qué es una Aplicación Web? – Desarrollo de Software y Consultoría Tecnológica,” Jul. 30, 2019. <https://www.strappinc.com/blog/strapp-datos/que-es-una-aplicacion-web> (accessed Jun. 01, 2021).
- [7] D. Triguero, “Aplicaciones móviles híbridas: la solución más eficiente para el desarrollo multiplataforma,” 2017. <https://profile.es/blog/aplicaciones-moviles-hbridas-la-solucion-mas-eficiente-para-el-desarrollo-multiplataforma/> (accessed Jun. 01, 2021).
- [8] Ortega Dinarle;, Guevara Maria;, and Benavides John, “ELEMENTARY: UN FRAMEWORK DE PROGRAMACIÓN WEB,” *Télématique*, Maracaibo, Dec. 02, 2016.
- [9] C. A. Herrera Díaz and G. G. Ochoa Guayanay, “Desarrollo de aplicaciones móviles híbridas con el framework Ionic, utilizando plataformas gratuitas y opensource,” pp. 1–8, 2015.
- [10] José María Aguero Atmitim, “Qué es Ionic: ventajas y desventajas de usarlo en apps móviles híbridas,” Feb. 22, 2021. <https://profile.es/blog/que-es-ionic/> (accessed Jun. 01, 2021).
- [11] R. Neftali *et al.*, “React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma,” *Rev. Digit. Univ.*, vol. 20, p. 2019, doi: 10.22201/codeic.16076079e.2019.v20n5.a5.
- [12] Jesús Lucas, “Qué es NodeJS y para qué sirve | OpenWebinars,” 2019. <https://openwebinars.net/blog/que-es-nodejs/> (accessed Oct. 25, 2020).
- [13] T. Diez, M. J. Domínguez, J. J. Martínez, and J. J. Sáenz De Navarrete, “Creación de páginas Web accesibles con HTML5.”
- [14] I. Buiol, “Apache Cordova vs Capacitor,” 2019. <https://www.moldeointeractive.com.ar/blog/moldeo-interactive-1/post/apache->

cordova-vs-capacitor-737 (accessed Jun. 01, 2021).

- [15] Campus MVP, “Las 5 principales ventajas de usar Angular para crear aplicaciones web | campusMVP.es,” Mar. 16, 2018. <https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx> (accessed Jun. 01, 2021).
- [16] I. de Souza, “API Rest: ¿qué es y cómo funciona ese recurso? [con ejemplos],” 2020. <https://rockcontent.com/es/blog/api-rest/> (accessed Oct. 25, 2020).
- [17] J. Melgar, “Creación de un API REST en Drupal e integración con Angular/IONIC - Blog de Hiberus Tecnología,” 2019. <https://www.hiberus.com/crecemos-contigo/creacion-de-un-api-rest-en-drupal-e-integracion-con-angular-ionic/> (accessed Jun. 01, 2021).
- [18] Visual Code, “Visual Studio Code - Code Editing. Redefined.” <https://code.visualstudio.com/> (accessed Jun. 01, 2021).
- [19] Sergio Vergara, “Tendencias tecnológicas web en 2019,” Jan. 24, 2019. <https://www.itdo.com/blog/tendencias-tecnologicas-web-en-2019/> (accessed Jun. 01, 2021).
- [20] F. Del Medico, “¿Qué es la geolocalización en dispositivos móviles?,” 2021. <https://maplink.global/es/blog/geolocalizacion-movil-en-los-celulares/> (accessed Jun. 01, 2021).
- [21] Luis Guzman, “La importancia de la Geolocalización en las Aplicaciones Móviles - Geek Bucket,” Jan. 30, 2019. <http://geekbucket.com.mx/blog/2019/01/30/la-importancia-de-la-geolocalizacion-en-las-aplicaciones-moviles/> (accessed Jun. 01, 2021).
- [22] Florencia del Medico, “¿Qué son las API de Google? Conoce ahora sus funciones principales,” May 20, 2020. <https://maplink.global/es/blog/que-son-google-apis/> (accessed Jun. 01, 2021).
- [23] Florencia del Medico, “Google Maps Platform: Coloca tu empresa en el mapa,” Apr. 29, 2020. <https://maplink.global/es/blog/que-es-google-maps-platform/> (accessed Jun. 01, 2021).
- [24] Ivan de Souza, “¿Qué tipos de hosting existen y cuáles son sus características?,” 2019. <https://rockcontent.com/es/blog/tipos-de-hosting/> (accessed Nov. 17, 2021).
- [25] C. I. Degollo Amaya *et al.*, “Manual SQL Server – Transact SQL Básico /Avanzado,” 2019. Accessed: Jun. 01, 2021. [Online]. Available: http://www.utsc.edu.mx/vidaEstudiantil/pdf/pdf_pades/manual_sql_server_2019.pdf.
- [26] Yecid Alexis Rendón, “Bases de datos relacionales vs. no relacionales,” May 28, 2019. <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales> (accessed Jun. 01, 2021).
- [27] M. Z. D. García-saiz, “Gestores NoSQL – MongoDB,” 2017.
- [28] Stephen J. Bigelow, “Comparación de Google Cloud Firestore y Firebase

- Realtime Database,” Oct. 09, 2019. <https://www.computerweekly.com/es/consejo/Comparacion-de-Google-Cloud-Firestore-y-Firebase-Realtime-Database> (accessed Nov. 17, 2021).
- [29] J. Luis Ventura-León, “Population or sample? A necessary difference,” 2017. Accessed: Jan. 06, 2021. [Online]. Available: <http://scielo.sld.cu>.
- [30] P. L. Ló, “POBLACIÓN MUESTRA Y MUESTREO.”
- [31] B. Montero, H. Cevallos, and J. Dávila, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales Rev. Multidiscip. Investig. ISSN 2550-6862*, vol. 2, no. 17, pp. 114–121, 2018, [Online]. Available: https://www.researchgate.net/publication/327537074_Metodologias_agiles_frente_a_las_tradicionales_en_el_proceso_de_desarrollo_de_software.
- [32] F. B. A. SILVANA and H. S. L. FERNANDO, “ANÁLISIS DEL IMPACTO DE LAS METODOLOGÍAS AGILES EN EL DESARROLLO DE SOFTWARE,” 217AD.
- [33] Encarna Abellán, “Metodología Scrum: qué es y cómo funciona,” Mar. 05, 2020. <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html> (accessed Jun. 01, 2021).
- [34] Manuel Maldonado, “Las mejores metodologías ágiles para la creación de software,” Apr. 12, 2018. <https://www.digital55.com/desarrollo-tecnologia/mejores-metodologias-agiles-creacion-software/> (accessed Jun. 01, 2021).
- [35] Marvin López Mendoza, “Extreme Programming: Qué es y cómo aplicarlo | OpenWebinars,” Sep. 18, 2020. <https://openwebinars.net/blog/extreme-programming-que-es-y-como-aplicarlo/> (accessed Jun. 01, 2021).
- [36] Gabriel Mancuzo, “▷ Metodología XP: La Mejor Vía para el Desarrollo de Software,” Aug. 08, 2020. <https://blog.comparasoftware.com/metodologia-xp/> (accessed Jun. 01, 2021).
- [37] Sara López, “Firebase: qué es, para qué sirve, funcionalidades y ventajas,” May 17, 2020. <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/> (accessed Jun. 01, 2021).

Anexos

Anexo A: Encuesta

A.1 Cuestionario elaborado con el objetivo de recopilar información, misma que fue realizada al Sr. Gabriel Alberto Romo Acurio gerente de la empresa de encomiendas “GABITO EXPRESS”.

N°	Preguntas
1	¿A qué tipo de clientes está enfocada la empresa?
2	¿La empresa cuenta con una página web o aplicación para gestionar los pedidos o motorizados?
3	¿Describa cuáles son las expectativas que usted tiene sobre la aplicación?
4	¿Cómo se realiza el registro de un pedido y su asignación a los motorizados?
5	¿Cuál es el tiempo de respuesta y entrega de un pedido?
6	¿Cómo los motorizados notifican la entrega del pedido?
7	¿De qué manera los clientes se informan del estado del pedido solicitado?
8	¿Cómo se gestiona la disponibilidad de los motorizados?
9	¿Indique la importancia de visualizar la ubicación de los motorizados?
10	¿Considera usted que la aplicación le ayudara a mejorar el proceso de gestión de motorizados?
11	¿Estaría dispuesto a invertir para llevar la aplicación a producción?