



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

**APLICACIÓN MÓVIL HÍBRIDA USANDO REACT NATIVE, NODE.JS
PARA LA GESTIÓN Y SEGUIMIENTO DE DELIVERY EN EL SISTEMA
DE CONTROL DE RESTAURANTES DE LA EMPRESA
CONNECTASERVICES.**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Tecnologías de la Información.

ÁREA: Base de datos y Sistemas informáticos.

LÍNEA DE INVESTIGACIÓN: Tecnologías de la información y Sistemas de
control

AUTOR: Francisco Israel Ruales Flores

TUTOR: Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

Ambato - Ecuador

agosto – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN MÓVIL HÍBRIDA USANDO REACT NATIVE, NODE.JS PARA LA GESTIÓN Y SEGUIMIENTO DE DELIVERY EN EL SISTEMA DE CONTROL DE RESTAURANTES DE LA EMPRESA CONNECTASERVICES, desarrollado bajo la modalidad Proyecto de Investigación por el señor Francisco Israel Ruales Flores, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2024.



Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: APLICACIÓN MÓVIL HÍBRIDA USANDO REACT NATIVE, NODE.JS PARA LA GESTIÓN Y SEGUIMIENTO DE DELIVERY EN EL SISTEMA DE CONTROL DE RESTAURANTES DE LA EMPRESA CONNECTASERVICES es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2024.



Francisco Israel Ruales Flores

C.C. 1850093566

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2024.



Francisco Israel Ruales Flores

C.C. 1850093566

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Francisco Israel Ruales Flores, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN MÓVIL HÍBRIDA USANDO REACT NATIVE, NODE.JS PARA LA GESTIÓN Y SEGUIMIENTO DE DELIVERY EN EL SISTEMA DE CONTROL DE RESTAURANTES DE LA EMPRESA CONNECTASERVICES**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2024.



Firmado electrónicamente por:
**ELSA PILAR URRUTIA
URRUTIA**

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL



Firmado electrónicamente por:
**EDISON HOMERO
ALVAREZ MAYORGA**

Ing. Edison Álvarez Mayorga, Mg.
PROFESOR CALIFICADOR



Firmado electrónicamente por:
**RUBEN EDUARDO
NOGALES PORTERO**

Ing. Rubén Nogales Portero, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Este proyecto de investigación lo dedico primero a mi buen Dios por que su Gracia a sido mayor y es mayor en cada paso que he dado en este increíble episodio de mi vida.

A mi padre Carlos Ruales por quien todos los días me enseña el gran valor de la responsabilidad, el sacrificio, y la humildad. A mi Mamita Linda Vanessa Ruales por siempre ser mi apoyo incondicional en todo este proceso y en mis más locos deseos, no dejarme solo, por su gran ejemplo de perseverancia y honestidad, los amo papis y este es el fruto de su amor, paciencia y apoyo sembrado en mi vida. A mis hermanos Melitos y David por su ejemplo, por su apoyo, por ser mi lugar seguro, mis amigos y mis guardaespaldas, los amo hermanos.

A mi esposa Melanie que demostrándome su amor decidió quedarse todos mis desvelos junto a mí, porque era fácil rendirse, pero con su gran apoyo salimos adelante. A mi hija Annie quien es mi motor mi mayor motivación y la niña de mis ojos.

Finalmente lo dedico a mi abuelita (mi angelito) que siempre me decía: “La carrera mijito es padre y madre” que de seguro debe estar gozándose junto a mi tío y mi primo allá en la gloria.

AGRADECIMIENTO

Agradezco a Dios por su eterno amor, su fidelidad en los momentos mas oscuros y los más brillantes de esta carrera, a Él sea la Gloria.

Agradezco a mis padres por su gran esfuerzo y trabajo para que no me falte nada, por levantarme en todas mis caídas no habrá suficientes palabras para poder expresar mi gratitud. Gracias a mis hermanos por que aun en mis caídas siempre estuvieron para mí, para ayudarme, para soportarme, para regañarme cuando no quería hacer mis tareas. Sin duda mi familia en mi mayor equipo y tesoro.

Gracias a mi Esposa por ser ese empuje para poder cumplir esta gran meta, por nunca dejarme solo y motivarme a seguir adelante.

A mi hija por que con tan solo una mirada y un tatata me llena de alegría y fuerza.

Agradezco a mi mejor amigo Jorge Ibarra por creer en mí, estoy seguro de que, sin sus ocurrencias, y sus locuras la carrera no sería la misma.

A mi buen amigo Josué Córdova por ser un excelente ser humano, y un verdadero crack

Agradezco a mi tutor Ing. Franklin Mayorga porque, aunque le mataba de iras no dejó de ayudarme y apoyarme en este difícil proceso.

ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE ANEXOS	xvi
RESUMEN EJECUTIVO	xvii
ABSTRACT	xviii
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	6

1.3.1 Tecnologías de la información.	6
1.3.2 Ingeniería en software.	6
1.3.3 Desarrollo de la Aplicación con móviles.	7
1.3.4 Framework React Native	7
1.3.5 Aplicación móvil híbrida usando react native, Node.js.	7
1.3.6 Gestión y seguimiento de delivery.	8
1.3.7 Gestión de ventas.	8
1.3.8 Gestión de Reparto.	9
1.4 Objetivos	9
1.4.1 Objetivo general	9
1.4.2 Objetivos específicos	9
CAPÍTULO II. METODOLOGÍA.....	10
2.1 Materiales.....	10
2.2 Métodos.....	10
2.2.1 Modalidad de la investigación	10
2.2.2 Población y muestra	11
2.2.3 Recolección de información.....	11
2.2.4 Procesamiento y análisis de datos	28
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	30
3.1 Análisis y Discusión.....	30

3.1.1 Análisis comparativo de versiones de react native para el desarrollo de aplicaciones móviles	31
3.1.2 Análisis comparativo de versiones de Node.js para el desarrollo de backend en aplicaciones móviles	34
3.1.3 Análisis de gestor de base de datos para el desarrollo aplicaciones móviles....	35
3.1.4 Metodologías de desarrollo de aplicaciones móviles.....	37
3.2 Desarrollo de la Propuesta	39
3.2.1 Fase 1: Exploración.....	40
3.2.2 Fase 2: Inicialización.....	43
3.2.3 Fase 3: Producción	63
3.2.4 Fase 4: Estabilización.....	69
3.2.5 Fase 5: Pruebas.....	69
3.2.6 Integración Aplicación móvil.....	79
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	81
4.1 Conclusiones	81
4.2 Recomendaciones.....	81
REFERENCIAS BIBLIOGRÁFICAS.....	82
ANEXOS.....	84

ÍNDICE DE TABLAS

Tabla 1. Cadenas de Búsqueda.....	3
Tabla 2. Criterios de inclusión y exclusión	4
Tabla 3 Población de Estudio.....	11
Tabla 4. Niveles de confianza – Alfa de Cronbach.....	12
Tabla 5. Validación de Instrumento Alfa de Cronbach.....	12
Tabla 6. Resultado de confiabilidad del instrumento.....	13
Tabla 7. Kuder-Richardson	13
Tabla 8. Cálculos de variables en el coeficiente de Kuder Richardson	14
Tabla 9. Resultado de confiabilidad Kuder-Richardson	14
Tabla 10. Descripción de Procesos	30
Tabla 11. Cuadro comparativo de versiones de react native.....	32
Tabla 12. Empresas que utilizan React native en aplicaciones es móviles	33
Tabla 13. Características de MySQL	36
Tabla 14. Aplicaciones móviles que utilizan MySQL	36
Tabla 15. Estudio comparativo de metodología de desarrollo de aplicaciones móviles	37
Tabla 16. Requisitos funcionales	41
Tabla 17. Planificación de fases.....	44
Tabla 18. Storycard 01 Inicio de Sesión	65
Tabla 19. Storycard 02 Registro de usuario	65

Tabla 20. Storycard 03 Pantalla de Inicio	66
Tabla 21. Storycard 04 interfaz Nuevo Pedido	66
Tabla 22. Storycard 04 Sincronizacion Woocommerce.....	67
Tabla 23. Storycard 05 interfaz inicio sesión para repartidor	67
Tabla 24. Storycard 06 Interfaz de seguimiento al delivery	68
Tabla 25. Storycard 07 notificaciones delivery.....	68
Tabla 26. Prueba de aceptación 1	75
Tabla 27. Prueba de aceptación 2.....	76
Tabla 28. Prueba de aceptación 3	76
Tabla 29. Prueba de aceptación 4.....	77
Tabla 30. Prueba de aceptación 5.....	77
Tabla 31. Prueba de aceptación 6.....	78
Tabla 32. Prueba de aceptación 7	78
Tabla 33. Prueba de aceptación 8.....	79

ÍNDICE DE FIGURAS

Figura 1. Los medios acogidos para le realización de pedidos.....	15
Figura 2. Conocimiento de la frecuencia de realización de pedidos.....	16
Figura 3. Grado de satisfacción al realizar un pedido con el sistema actual.....	17
Figura 4. Conocimiento del proceso de recepción de pedidos.....	18
Figura 5. Precisión en el seguimiento de los pedidos con el sistema actual	19
Figura 6. Consideración de eficiencia en el proceso actual de seguimiento de entregas a domicilio.	20
Figura 7. Conocimiento de utilización de una plataforma digital en la toma de pedidos.	21
Figura 8. Consideración en la falta de herramientas digitales causante de la insatisfacción de los clientes.	22
Figura 9. Presencia de errores y confusiones en la comunicación de los clientes sobre su pedido.....	23
Figura 10. Satisfacción en el proceso de gestión de pedidos del cliente mediante una aplicación móvil.	24
Figura 11. Estadística de respuestas de la encuesta	29
Figura 12. Diagrama de procesos en la gestión de delivery	30
Figura 13. Tablas Principales.....	45
Figura 14. Estructura tabla Clientes.....	46
Figura 15. Estructura tabla pedidos.....	46
Figura 16. Estructura tabla repartidor	47
Figura 17. Diagrama de casos de uso de cliente del restaurante.....	48

Figura 18. Pantalla de Carga	49
Figura 19. Mackup Pantalla de Inicio	50
Figura 20. Mackup Inicio de Sesión	51
Figura 21. Mackup Registro de Usuario	52
Figura 22. Pantalla mi Perfil	53
Figura 23. Mackup Página de Inicio	54
Figura 24. Mackup Pantalla Mis Órdenes.....	55
Figura 25. Pantalla Direcciones	56
Figura 26. Métodos de Pago	57
Figura 27. Pantalla Contáctanos	58
Figura 28. Pantallas de Configuración.....	59
Figura 29. Modal de Cerrar Sesión	60
Figura 30. Pantalla Realizar Pedidos	61
Figura 31. Pantalla desplegable del carrito de compras.....	62
Figura 32. Pantalla Orden Confirmada	63
Figura 33. Iniciar sesión cliente, repartidor	70
Figura 34. Registrar Usuario.....	71
Figura 35. Pantalla de Inicio	71
Figura 36. Editar Usuario.....	72
Figura 37. Realizar Pedido.....	73
Figura 38. Pantalla Contacto.....	73

Figura 39. Mis Órdenes.....	74
Figura 40. Pantalla de seguimiento de pedidos.....	75
Figura 41. Integración pedidos.....	80

ÍNDICE DE ANEXOS

Anexo A. Encuesta.....	84
Anexo B. Entrevista	86
Anexo C. Dependencias descargadas	87
Anexo D. Código Implementación de Login.....	88
Anexo E. Código de implementación para el registro de un nuevo usuario.....	90
Anexo F. Código de implementación para la página principal	96
Anexo G. Código de implementación de un nuevo pedido.....	99
Anexo H. Código de sincronización con la API de woocommerce	104
Anexo I. Código de implementación de inicio de sesión del repartidor.....	106
Anexo J. Código de implementación para seguimiento de pedidos.....	107
Anexo K. Código de implementación de notificaciones utilizando expo	110
Anexo L. Manual de usuario	112

RESUMEN EJECUTIVO

En el mercado empresarial de servicios de desarrollo de software, en particular en la empresa Connectaservices, enfrentan desafíos significativos en el sistema de restaurantes, siendo uno de ellos la gestión y seguimiento de pedidos en línea ante lo cual el presente proyecto propone una solución basada en el desarrollo de una aplicación móvil híbrida utilizando el framework React Native como herramienta de desarrollo del lado del cliente y Node.js como herramienta de manejo con la base de datos para mejorar la gestión, juntamente con la aplicación de la metodología de desarrollo en Mobile D para un desarrollo ordenado.

Dando como resultado un aplicativo móvil que optimiza la gestión y seguimiento de pedidos en línea, utilizando el framework react native, combinado con una metodología de desarrollo como mobile D, adicionalmente se realizó el análisis de requerimientos mediante encuestas y entrevista con el fin de establecer una efectiva comunicación entre el sistema de restaurantes de la empresa y el aplicativo móvil.

Esta solución ofrece una respuesta efectiva a los desafíos de gestión de pedidos en línea, mejorando la eficiencia operativa del sistema de restaurantes de la empresa Connectaservices ya que permite acelerar el proceso de pedido generado.

Palabras clave: Aplicación móvil híbrida, react native, node.js, gestión de pedidos, mobile d.

ABSTRACT

In the business market for software development services, particularly in the company Connectaservices, they face significant challenges, one of them being the management and tracking of online orders to which this project proposes a solution based on the development of a hybrid mobile application using the React Native framework as a development tool on the client side and Node.js as a management tool with the database to improve management, along with the application of the development methodology in Mobile D for an orderly development.

Resulting in a mobile application that optimizes the management and tracking of online orders, using the react native framework, combined with a development methodology such as mobile D, additionally the requirements analysis was conducted through surveys and interviews to establish an effective communication between the company's restaurant system and the mobile application.

This solution offers an effective response to the challenges of online order management, improving the operational efficiency of Connectaservices' restaurant system by speeding up the order process.

Keywords: Hybrid mobile application, React Native, Node.js, Order management, Mobile D.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

APLICACIÓN MÓVIL HÍBRIDA USANDO REACT NATIVE, NODE.JS PARA LA GESTIÓN Y SEGUIMIENTO DE DELIVERY EN EL SISTEMA DE CONTROL DE RESTAURANTES DE LA EMPRESA CONNECTASERVICES.

1.1.1 Planteamiento del problema

En un mundo cada vez más digitalizado y orientado hacia la comodidad del usuario, las aplicaciones móviles y las plataformas en línea permiten a los usuarios realizar pedidos de manera sencilla, realizar un seguimiento en tiempo real de sus entregas y brindar retroalimentación, lo que mejora la satisfacción del cliente, por lo que, el internet se ha vuelto una herramienta omnipresente e indispensable en la vida cotidiana, transformando la forma en que las personas se comunican, trabajan, se informan y se entretienen. La demanda de servicios de entrega a domicilio y la gestión eficiente de pedidos en restaurantes ha experimentado un crecimiento significativo en los últimos años [1].

Ecuador ha experimentado un crecimiento constante en la industria de alimentos en los últimos años. El aumento en el poder adquisitivo de la población, la urbanización y la creciente tendencia hacia el consumo de alimentos a través de servicios de entrega han impulsado la demanda de soluciones tecnológicas que faciliten la experiencia del cliente. La pandemia de COVID-19 también ha acelerado la necesidad de opciones de entrega seguras y convenientes [2].

El sector alimenticio en la ciudad de Ambato está altamente competitivo, con una variedad de restaurantes que ofrecen comida gourmet y experiencias culinarias únicas. Connectaservices, una empresa de desarrollo que busca mantener y expandir su base de clientes a través de la digitalización de sus servicios mediante un sistema para restaurantes que gestione de manera óptima la entrega de pedidos y el seguimiento de delivery.

El sistema para restaurante que proporciona la empresa Connectaservices, se encuentra en una etapa crucial de su desarrollo, el mercado de servicios para restaurantes es altamente competitivo, lo que requiere que el sistema esté a la altura de las expectativas y necesidades de los restaurantes y sus clientes para destacar y ganar una participación significativa en el mercado. Para llevar a cabo con éxito la creación de la aplicación móvil, se deben considerar aspectos como la optimización de los procesos de delivery, la gestión en la existencia en la carta del restaurante, además, los beneficiarios será directamente la empresa, portadora del sistema y el restaurante capaz de tener agilidad en su gestión.

1.2 Antecedentes investigativos

En esta sección, se llevó a cabo un análisis de la literatura, focalizándonos en la identificación de documentos relevantes para respaldar y enriquecer el marco de investigación. Se han explorado diversas fuentes, entre las que se incluyen, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE Xplore), Asociación de Maquinaria Computacional (ACM Digital Library), Willey, Springer y Scielo, con el propósito de localizar estudios significativos. Además, se ha considerado el repositorio de la Universidad Técnica de Ambato como una fuente adicional de información.

En el proceso, se definió criterios de búsqueda utilizando términos clave como "aplicación móvil", "entrega", "react native" y "node.js", estableciendo límites temporales para la inclusión de artículos publicados entre los años 2019 y 2023. La Tabla 1. presenta de manera detallada las cadenas de búsqueda utilizadas y el número correspondiente de artículos obtenidos durante este proceso de investigación.

Tabla 1. Cadenas de Búsqueda

	Revista	Cadena	N° Artículos
1ra. Cadena de búsqueda	Repositorio de la Universidad Técnica de Ambato	"mobile application" AND "delivery" AND ("react native" OR "node.js") AND fecha publicado 2022	1
	IEEE Xplore	("All Metadata":"mobile application") AND ("All Metadata":"delivery") OR (("All Metadata":"react native") OR ("All Metadata":"node.js")) AND year 2020 - 2023	189
	ACM Digital Library	[[All: "delivery application"] AND [All: "restaurant"]] OR [All: "react native"] OR [All: "node.js"] AND [E-Publication Date: Past 5 year	135
	Willey	"mobile application" AND "delivery" AND "restaurant" OR ("react native" AND "node.js") within Article 2019 - 2023	11
	Springer	mobile application AND "delivery" AND "restaurant" OR ("react AND native" AND "node.js") within Article 2019 - 2023	48
	scielo	((mobile application) AND (delivery)) OR ((react native) OR (node.js)) within Article 2019 - 2023	4
TOTAL			388

El propósito de llevar a cabo una revisión literaria radica en identificar la viabilidad generada por una aplicación móvil en la administración de pedidos. De este modo, en el presente análisis y proceso de aprendizaje, se abordan las diversas tecnologías que son fundamentales para la implementación exitosa de una aplicación móvil.

En esta perspectiva, el producto de las búsquedas en las distintas bases de datos arrojó un conjunto de 388 artículos relacionados con las palabras clave especificadas. Por lo tanto, en la Tabla 2. se exponen los criterios utilizados para determinar la inclusión o exclusión de los artículos pertinentes para el estudio.

Tabla 2. Criterios de inclusión y exclusión

Inclusión	Artículos que implementen aplicaciones móviles para delivery
	Artículos que expongan aplicaciones móviles híbridas usando React Native y Node.js
	Artículos que sean de Ciencia Computacional e Ingeniería.
Exclusión	Artículos que no estén escritos en español o inglés
	Los artículos que no sean relevantes para el tema de investigación.
	Artículos que se refieran a la implementación de otras tecnologías en una aplicación móvil.

Analizando estos criterios se tomó en cuenta 4 artículos y una tesis que cumplen con la totalidad de los criterios revisados.

L. Fabricio E. Montenegro [3], [4] Destaca el notable aumento experimentado por las aplicaciones de entrega a domicilio en diversos ámbitos comerciales, como el sector restaurantero y farmacéutico, especialmente impulsado por la pandemia de COVID-19 y las medidas de distanciamiento social. El proyecto de investigación propuesto tiene como meta el desarrollo de aplicaciones móviles multiplataforma con un enfoque centrado en la atención al cliente. Esto permitiría a los usuarios explorar supermercados cercanos desde la comodidad de sus hogares. El objetivo principal es simplificar la gestión de ventas para los propietarios de estos establecimientos al brindar la opción de realizar compras de productos esenciales a través de la aplicación.

A. Luthfia, S. Ramadanty, T. L. Anita, A. Zulkarnain and A. R. Ridzuan[5] Explican la importancia de las aplicaciones de entrega de alimentos como solución para

satisfacer las necesidades alimentarias de la comunidad, especialmente durante restricciones de movilidad. Se enfoca en la influencia de las habilidades digitales de los usuarios en la aceptación de estas aplicaciones. La investigación revela que las habilidades digitales afectan la aceptación en un 27,3%, siendo las habilidades de comunicación las más influyentes. Además, se señala que las habilidades técnicas y de seguridad de los usuarios son menos desarrolladas, sugiriendo que los desarrolladores deben mejorar la seguridad y usabilidad de las aplicaciones para aumentar su aceptación.

Suri B., Taneja S., Bhanot I., Sharma H., Raj A. [6] Propinan la importancia en donde a la actualidad, la habilidad de desarrollar aplicaciones móviles se considera esencial debido al crecimiento en el uso de teléfonos inteligentes y tabletas. La complejidad de esta tarea radica en la diversidad de dispositivos y plataformas disponibles. El desarrollo multiplataforma ha ganado popularidad, pero la elección entre los principales marcos de desarrollo para Android, como React Native, Flutter y Kotlin, sigue siendo un tema debatido.

Famula J, Pittman D, Haring K [3] abordan la percepción del público sobre la privacidad en una aplicación de teléfono móvil destinada a drones de reparto comerciales. El objetivo es identificar las medidas necesarias para generar confianza en las plataformas de reparto no tripuladas. Se concluye que, a pesar del interés en los sistemas de entrega con drones, los riesgos de seguridad y privacidad son obstáculos significativos para la confianza y aceptación públicas. Se sugiere que una aplicación móvil centrada en el usuario podría mejorar la confianza en los vuelos comerciales con drones.

Xervaxius Naphan G, Setyawan A, Edmund Friry J, Nugroho Perwiro Atmojo R [7] examinan la aplicación de entrega en línea como una tecnología en constante desarrollo, destacando su eficacia durante la pandemia de Covid-19 al facilitar la movilidad de la entrega en comunidades con limitaciones espaciales para reducir la transmisión del virus. Utilizando la Teoría unificada de la aceptación y uso de tecnología (UTAUT), se exploran variables como expectativas de desempeño, esfuerzo, influencia social, confianza, satisfacción y continuidad en la intención de

uso. El objetivo principal es evaluar el impacto de la pandemia en la adopción de servicios de entrega en línea. Aunque se encuentra que estas aplicaciones fueron muy útiles, especialmente para alimentos, bienes, abarrotes y medicinas, aún existen factores que generan reticencia en las personas para utilizarlas de manera continua.

1.3 Fundamentación teórica

1.3.1 Tecnologías de la información.

Las tecnologías de la información y la comunicación (TIC) se generan a partir de progresos científicos realizados en el ámbito de la informática y las telecomunicaciones. En la actualidad, estas tecnologías están omnipresentes en la sociedad, posibilitando el acceso, la producción, el procesamiento y la comunicación mediante códigos como texto, imagen, sonido y video. Las TIC comprenden tres medios fundamentales que operan de manera interactiva: la microelectrónica, la informática y las telecomunicaciones [3].

El objetivo principal de la informática es automatizar la manipulación de información con cualquier dispositivo o herramienta tecnológica. Su auge se manifestó en las computadoras de la Segunda Guerra Mundial debido a la necesidad de disponer de tablas de tiro precisas. John von Neumann, un matemático húngaro-estadounidense, propuso diversas soluciones, como la eliminación de la necesidad de volver a cablear los circuitos internos para desarrollar un nuevo programa y la representación binaria de los programas, entre otras[8].

1.3.2 Ingeniería en software.

Su definición se fundamenta en la aplicación de los sólidos principios de la ingeniería, los cuales están dirigidos a lograr un software confiable y con un rendimiento eficaz en cualquier entorno laboral.[2]

Dentro de los procedimientos de software se busca incorporar una porción de la vida diaria, ajustándola según ciertos criterios y, finalmente, materializándola en la creación de un sistema de software que respalde las actividades de ese contexto [2].

1.3.3 Desarrollo de la Aplicación con móviles.

Una aplicación móvil, también conocida como app, se configura como un software diseñado para operar en dispositivos móviles. Estos programas están disponibles en plataformas de distribución conocidas como tiendas, entre las cuales se incluyen Google Play Store y AppStore. Se identifican tres categorías de aplicaciones móviles: aplicaciones nativas y aplicaciones híbridas [3].

En el caso de las aplicaciones nativas, su desarrollo implica la consideración de que utilizan un Kit de Desarrollo de Software (SDK) específico para el sistema operativo al que están destinadas [3].

1.3.4 Framework React Native

React Native es un framework de código abierto para crear aplicaciones móviles nativas multiplataforma con una única base de código en JavaScript y React para la creación de componentes visuales, en lugar de ser ejecutados en navegador, se ejecute directamente sobre las plataformas móviles nativas, en este caso iOS y Android.[9]

1.3.5 Aplicación móvil híbrida usando react native, Node.js.

Una aplicación móvil híbrida, construida mediante el uso de tecnologías como React Native y Node.js, representa una solución versátil y eficaz para satisfacer las demandas del entorno móvil actual. Este enfoque híbrido permite aprovechar las ventajas de ambas plataformas, combinando la eficiencia de desarrollo de React Native con la potencia de Node.js en el backend [1].

La aplicación móvil híbrida basada en React Native y Node.js tiene la capacidad de ser distribuida a través de plataformas reconocidas, como Google Play Store y AppStore. Al adoptar este enfoque, se puede lograr un rendimiento eficiente y confiable en diversos entornos laborales [10].

1.3.6 Gestión y seguimiento de delivery.

La gestión y seguimiento de delivery se ha convertido en un componente esencial en el contexto actual, donde el comercio electrónico y las entregas a domicilio han adquirido una relevancia significativa. Este proceso se enfoca en garantizar que los productos o servicios solicitados en línea lleguen de manera eficiente y satisfactoria al destino final.[10]

Más allá de la mera logística, la gestión de delivery se preocupa por la experiencia del cliente. La comunicación transparente sobre el estado de la entrega y la resolución eficiente de cualquier problema contribuyen a construir la confianza del cliente en el servicio proporcionado [4].

Enfrentando desafíos operativos diarios, desde el tráfico hasta las condiciones climáticas impredecibles, la gestión de delivery requiere estrategias flexibles y una capacidad adaptativa para garantizar entregas exitosas. Además, en un contexto donde la sostenibilidad es una consideración creciente, se exploran prácticas más eco amigables, desde la elección de vehículos sostenibles hasta la optimización de rutas para reducir la huella ambiental [4].

1.3.7 Gestión de ventas.

La gestión de ventas y el seguimiento de delivery son componentes clave en el ciclo comercial moderno. Comenzando con la interacción en línea, la gestión de ventas influye directamente en la experiencia del cliente. La comunicación eficiente sobre productos y tiempos de entrega contribuye a expectativas claras.

La tecnología desempeña un papel crucial, permitiendo un monitoreo en tiempo real de ventas y entregas. La coordinación efectiva entre la gestión de ventas y el delivery es esencial para construir relaciones sólidas y resolver problemas rápidamente [11].

La sostenibilidad también se integra en estas operaciones, desde opciones de envío ecológicas hasta prácticas sostenibles en el embalaje. En resumen, la gestión de ventas y el seguimiento de delivery trabajan en conjunto para ofrecer una experiencia

completa, desde la compra hasta la entrega, impactando directamente en la satisfacción del cliente y el éxito comercial [11].

1.3.8 Gestión de Reparto.

La gestión de reparto se suma de manera crucial a la coordinación entre la gestión de ventas y el seguimiento de delivery. Este proceso se centra en optimizar la asignación de tareas y recursos para garantizar entregas eficientes y puntuales [11].

Una gestión efectiva de reparto implica la asignación inteligente de rutas, considerando factores como la proximidad de los destinos y la cantidad de entregas. La integración de tecnologías de mapeo y seguimiento en tiempo real facilita esta optimización, permitiendo ajustes en tiempo real para superar obstáculos y mejorar la eficiencia operativa [11].

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar una aplicación móvil aplicando react native, Node.js para la gestión de pedidos (delivery) en el sistema de restaurantes para la empresa Connectaservice.

1.4.2 Objetivos específicos

- Determinar los inconvenientes que enfrenta la empresa Connectaservices en la toma de pedidos y el seguimiento de entregas a domicilio.
- Investigar los beneficios que presenta react native, Node.js en el desarrollo de aplicaciones móviles híbridas.
- Integrar la aplicación móvil de delivery aplicando react native. Node.js para la gestión de pedidos y seguimiento delivery en el sistema de restaurantes para la empresa Connectaservices.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

En el presente proyecto de investigación el material empleado para la recolección de información se basó en la aplicación de encuestas, en donde está formado por 10 preguntas repartidas entre escalas de Likert 5 y también preguntas dicotómicas que serán de selección entre sí o no. Este cuestionario se aplicó al personal encargado de la recepción de pedidos, encargadas de realizar los pedidos y Personas encargadas del motorizado con la finalidad de recolectar información detallada de los inconvenientes que presentan los restaurantes al momento de realizar el proceso de gestión de pedidos mediante el cuestionario reflejado en el **Anexo A**.

Además, se realizó una entrevista al gerente administrador de la empresa que consta de 10 preguntas abiertas con la finalidad de obtener la opinión importante y relevante de parte del administrador de la empresa a fin de que pueda ayudar con el desarrollo del proyecto reflejada en el **Anexo B**.

2.2 Métodos

2.2.1 Modalidad de la investigación

Investigación de campo

Investigación de campo ya que el investigador examinó los desafíos directamente en el lugar de ocurrencia, en este caso, la empresa. Este enfoque tiene como objetivo identificar las razones y efectos asociados con la problemática en cuestión.

Investigación bibliográfica-documental

En el desarrollo de este proyecto, fue imperativo adquirir un entendimiento exhaustivo al conocer, analizar y evaluar tanto trabajos previos como investigaciones realizadas en el área pertinente. Además, se respaldará el proyecto mediante la consulta de

diversas fuentes de información, tales como artículos, textos, libros, documentos, y otros recursos relevantes.

2.2.2 Población y muestra

La población con el que se trabajó es el personal encargado de la recepción de pedidos, las personas encargadas de realizar los pedidos y el gerente administrador del restaurante.

Tabla 3 Población de Estudio

Población	Número	Porcentaje
Personal encargado de la recepción de pedidos	1	12.5%
Personas encargadas de realizar los pedidos	1	12.5%
Gerente administrador de la empresa	1	12.5%
Personas encargadas del motorizado	5	62.5%
Total	8	100%

La población en la cual se enfocó la investigación se lo consideró a medida de usabilidad por pedido. Debido a que la población no sobrepasa de cien elementos. Por lo tanto, no fué necesario realizar muestras representativas, ya que se trabajó con la totalidad de la población en consideración.

2.2.3 Recolección de información

Validación del Instrumento

Para garantizar la confiabilidad y consistencia interna de los datos recolectados en la encuesta, se aplicó el coeficiente Alfa de Cronbach. Este método estadístico produce un valor dentro de una escala específica, el cual evalúa la precisión y homogeneidad del instrumento de medición utilizado, que permite identificar si existen sesgos o inconsistencias en alguno de los ítems que componen la encuesta, además se realizó la validación de la confiabilidad mediante el coeficiente Kuder Richardson para las preguntas de Si o no.

Tabla 4. Niveles de confianza – Alfa de Cronbach

RANGO	CONFIABILIDAD
0.53 a menos	Confiabilidad nula
0.54 a 0.59	Confiabilidad baja
0.60 a 0.65	Confiable
0.66 a 0.71	Muy confiable
0.72 a 0.99	Excelente confiabilidad
1	Confiabilidad perfecta

Para el cálculo del coeficiente Alfa de Cronbach se requieren de 3 variables expresadas en la siguiente formula:

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum S_i^2}{S_T^2} \right] \quad (1)$$

Donde:

α es el coeficiente de confiabilidad del cuestionario.

K es el número de ítems del instrumento.

$\sum S_i^2$ es la sumatoria de las varianzas de los ítems.

S_T^2 es la varianza total del instrumento.

Para proceder con los cálculos respectivos se dispone el resultado de los ítems de la siguiente manera:

Tabla 5. Validación de Instrumento Alfa de Cronbach

ENCUESTADOS	P1	P2	P3	P5	P10	SUMA
E1	4	2	3	1	2	12
E2	1	3	4	2	3	13
E3	2	4	3	5	5	19
E4	3	5	4	4	4	20
E5	1	2	3	2	1	9
E6	4	4	3	5	4	20
E7	4	5	4	4	5	22

VARIANZA	1,63265306	1,3877551	0,24489796	2,204081633	1,959183673	
SUMATORIA DE VARIANZAS	7,428571429					
VARIANZA DE LA SUMA DE LOS ÍTEMS	21,3877551					

Mediante la **Tabla 5**. Se calculan las variables correspondientes a la ecuación del alfa de Cronbach obteniendo los siguientes resultados:

Tabla 6. Resultado de confiabilidad del instrumento

α :	Coeficiente de confiabilidad del cuestionario	0,82
k:	Número de ítems del instrumento	5
$\sum_{i=1}^k S_i^2$:	Sumatoria de las varianzas de los ítems.	7,429
S_t^2 :	Varianza total del instrumento.	21,388

Una vez calculado el coeficiente de confiabilidad del cuestionario según la **Tabla 4**. Califica mediante a los rangos el coeficiente resultante de excelente confiabilidad.

Tabla 7. Kuder-Richardson

ENCUESTADOS	P4	P6	P7	P8	P9	
E1	1	1	1	1	1	5
E2	1	1	1	1	1	5
E3	0	0	1	1	1	3
E4	1	0	0	0	0	1
E5	0	0	1	1	0	2
E6	1	0	0	0	0	1
E7	1	0	0	0	0	1
	5	2	4	4	3	

Para el cálculo del coeficiente Kuder-Richardson se requieren de 3 variables expresadas en la siguiente formula:

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum pq}{\sigma^2} \right] \quad (2)$$

Donde:

α es el coeficiente de confiabilidad del cuestionario.

K es el número de ítems del instrumento.

q es el porcentaje de personas que respondieron por el sí.

p es el porcentaje de personas que respondieron por el no.

σ^2 es la varianza total del instrumento

Para proceder con los cálculos respectivos se dispone el resultado de los ítems de la siguiente manera:

Tabla 8. Cálculos de variables en el coeficiente de Kuder Richardson

p	0,71428571	0,28571429	0,57142857	0,57142857	0,42857143
q	0,28571429	0,71428571	0,42857143	0,42857143	0,57142857
p*q	0,20408163	0,20408163	0,24489796	0,24489796	0,24489796

Tabla 9. Resultado de confiabilidad Kuder-Richardson

k	5
$\sum p*q$	1,14285714
σ^2	2,81632653
α	0,74275362

Resultados de la encuesta

Pregunta 1: ¿Cuáles de los siguientes medios es la más escogida para realizar pedidos a domicilio?

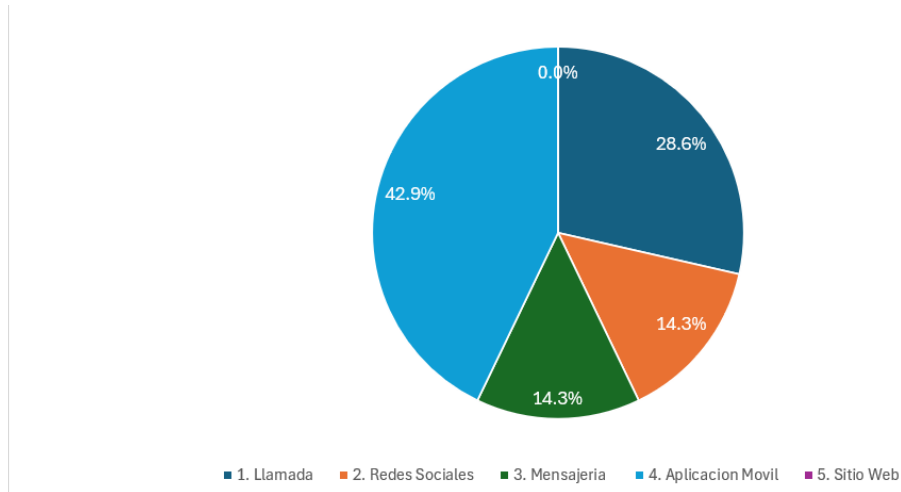


Figura 1. Los medios acogidos para le realización de pedidos

Análisis e Interpretación

Al analizar los datos adquiridos de la pregunta 1, el 42.9% de los encuestados afirman que los clientes prefieren realizar pedidos a domicilio por medio de un aplicativo móvil, el 28.6% de los clientes se inclina por llamada telefónica, el 14,3% de los encuestados prefieren las redes sociales y el otro 14,3% la mensajería. Los resultados denotan que los clientes prefieren un aplicativo móvil para poder realizar pedidos a domicilio, y como segunda opción se inclinan en realizar los pedidos por medio de llamada telefónica

Pregunta2. ¿Con que frecuencia se realizan pedidos en línea?

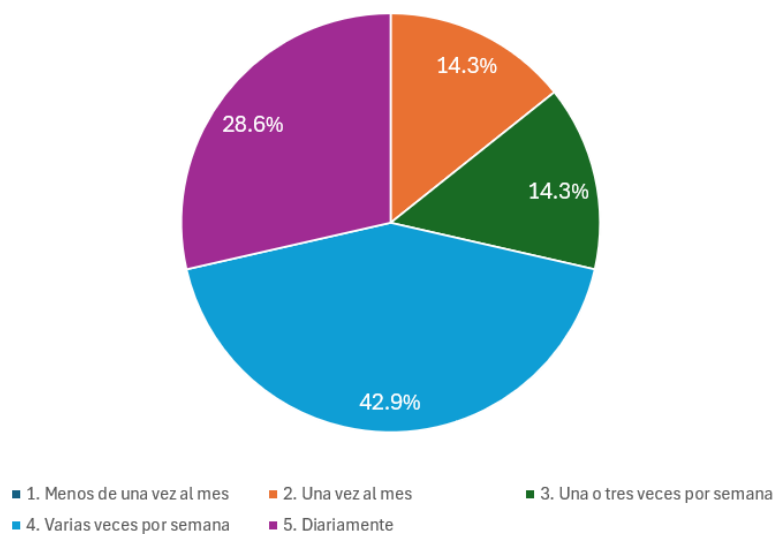


Figura 2. Conocimiento de la frecuencia de realización de pedidos.

Análisis e Interpretación

Al analizar los datos adquiridos de la pregunta 2, el 42.9% de las personas encuestadas afirman que los clientes realizan varias veces por semana, el 28.6 afirma que realizan pedidos a domicilio diariamente, el 14,3% realiza pedidos a domicilio una o tres veces por semanas y 14,3% realiza pedidos una vez al mes. Esto quiere decir que los clientes realizan pedidos a domicilio con mucha frecuencia entre semana.

Pregunta 3. ¿Con qué grado de satisfacción se realiza un pedido con el sistema actual?

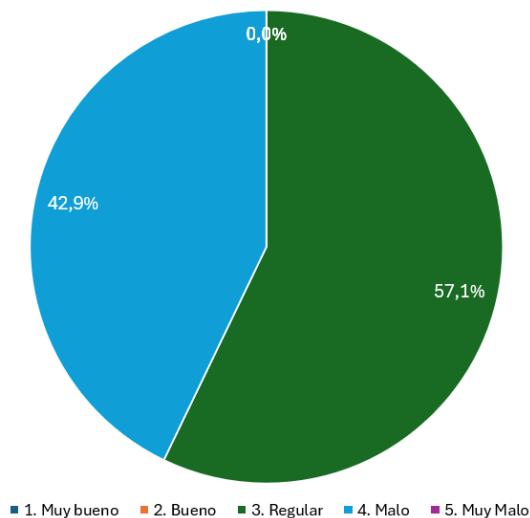


Figura 3. Grado de satisfacción al realizar un pedido con el sistema actual.

Análisis e interpretación

Recapitulando los resultados obtenidos en la pregunta 3, el 57.1% de los encuestados afirman que los clientes presentan un nivel regular de satisfacción al momento de entregar pedidos a domicilio, el 42.9% afirma que presentan un grado de satisfacción malo. Los resultados confirman que el grado de satisfacción de los clientes al momento de realizar pedidos a domicilio es regular.

Pregunta 4. ¿Conoce el proceso de recepción de pedidos?

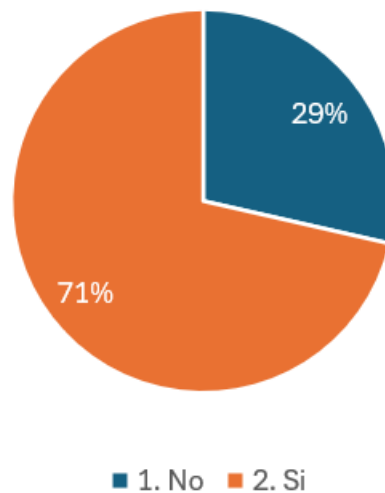


Figura 4. Conocimiento del proceso de recepción de pedidos

Análisis e interpretación

En el análisis de la pregunta 4, el 71% de las personas encuestadas afirman que, si se tienen conocimiento del proceso de recepción de pedidos, pero el 29% de los encuestados menciona que no conoce el proceso. Los resultados demuestran que si se tiene el conocimiento del proceso de recepción de pedidos por medio de los encuestados.

Pregunta 5. ¿Qué tan preciso es el seguimiento de los pedidos en curso con el sistema actual?

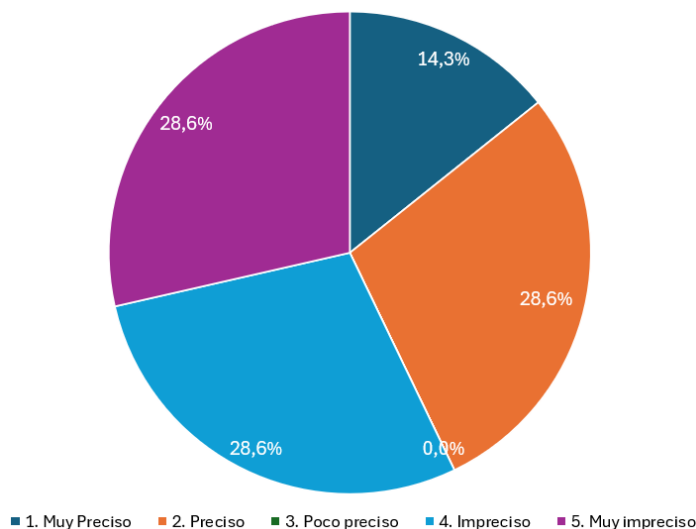


Figura 5. Precisión en el seguimiento de los pedidos con el sistema actual

Análisis e interpretación

Los resultados de la pregunta 5 reflejan que el 28.6% de las personas encuestadas argumentan que el seguimiento de los pedidos en el sistema actual es muy impreciso, el otro 28.6% argumentan que es impreciso, el otro 28.6% que es preciso además el 14.3% argumenta que el seguimiento de pedidos es muy preciso. en el análisis de los resultados se puede afirmar que la mayor cantidad de encuestados argumentan que el seguimiento de los pedidos en curso del sistema actual es muy impreciso.

Pregunta 6. ¿Considera que el proceso actual de seguimiento de entregas a domicilio es eficiente?

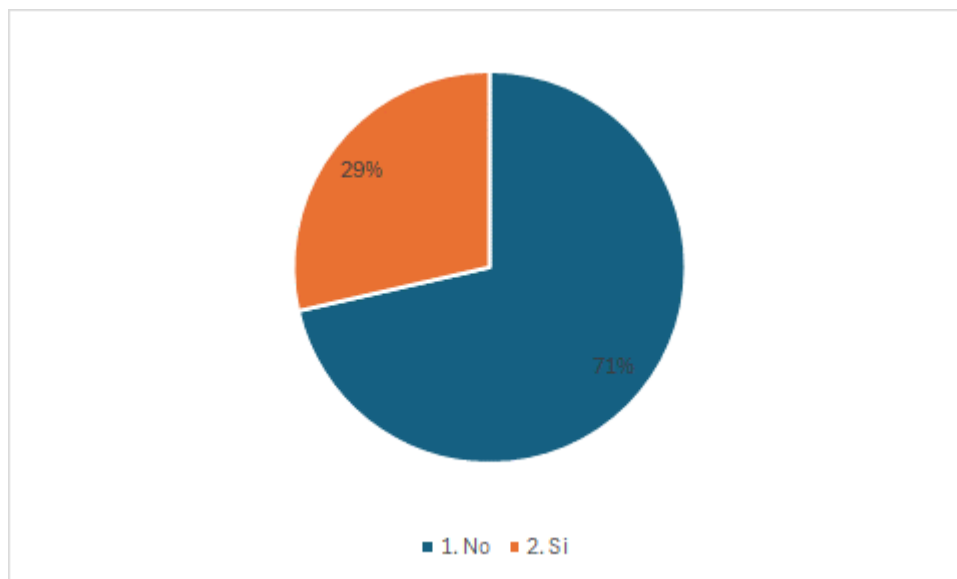


Figura 6. Consideración de eficiencia en el proceso actual de seguimiento de entregas a domicilio.

Análisis e interpretación

En el análisis de la pregunta 6, el 71% de las personas encuestadas argumentan que no es eficiente el proceso actual de seguimiento de entregas, el 29% afirma que es eficiente. Mediante los resultados obtenidos se interpreta que el sistema actual de seguimiento de entrega a domicilio no es eficiente.

Pregunta 7. ¿Estaría de acuerdo en utilizar una plataforma digital a fin de superar las dificultades en la toma de pedidos?

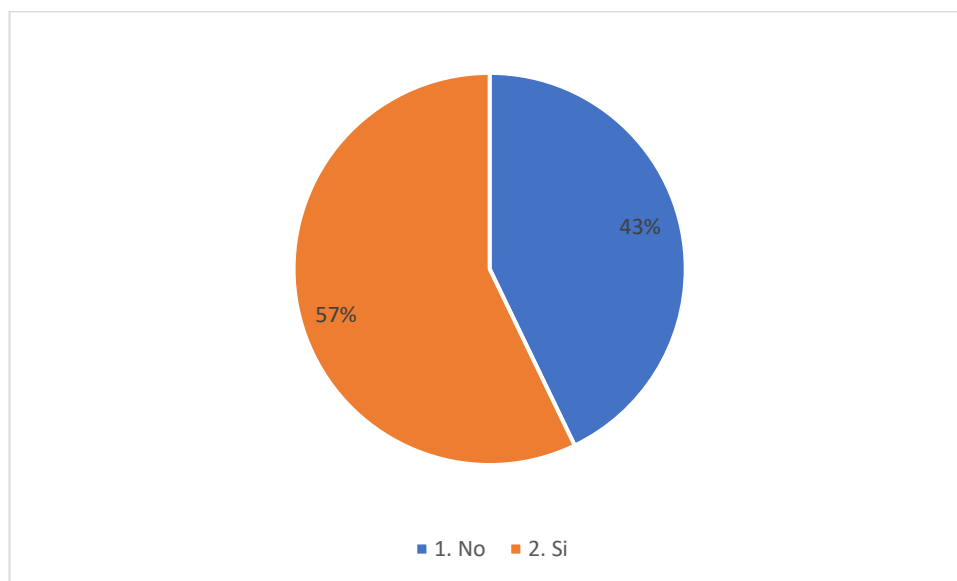


Figura 7. Conocimiento de utilización de una plataforma digital en la toma de pedidos.

Análisis e Interpretación

En el análisis de la pregunta 7, el 57% de la población encuestada argumenta que, si está de acuerdo en utilizar una plataforma digital para la toma de pedidos, el 43% argumenta que no que no está de acuerdo en utilizar una plataforma digital. De acuerdo con los datos resultantes se puede mencionar que el personal está de acuerdo en utilizar una plataforma digital para la toma de pedidos.

Pregunta 8. ¿Está de acuerdo que la falta de herramientas digitales para coordinar horarios de entrega es una causa importante de las dificultades recurrentes en la satisfacción de los clientes?

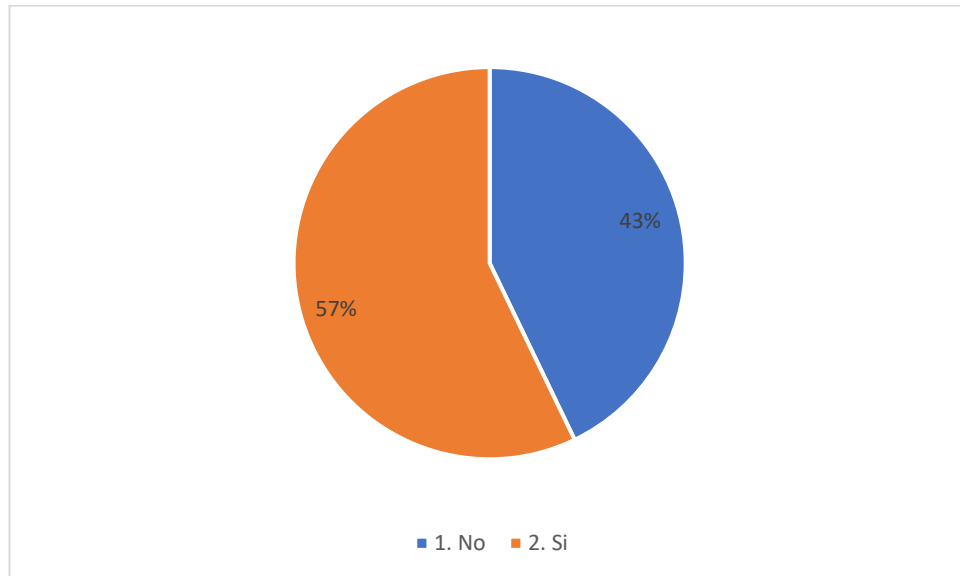


Figura 8. Consideración en la falta de herramientas digitales causante de la insatisfacción de los clientes.

Análisis e Interpretación

de acuerdo con los datos obtenidos en la pregunta 8, el 57% de los encuestados están de acuerdo que la falta de una herramienta digital es la causa de la insatisfacción en la entrega de pedidos, el 43% argumenta que no está de acuerdo que sea por la falta de una herramienta digital. Mediante los resultados obtenidos se menciona que las dificultades de los restaurantes al realizar pedidos son por la falta de una herramienta digital

Pregunta 9. ¿Ha experimentado errores o confusiones en la comunicación de información a los clientes sobre el estado de sus pedidos?

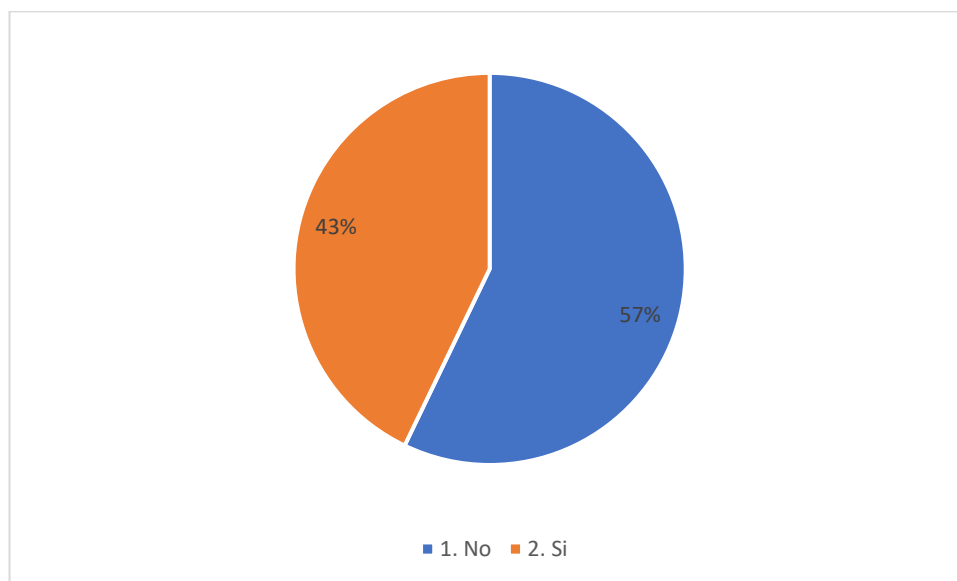


Figura 9. Presencia de errores y confusiones en la comunicación de los clientes sobre su pedido

Análisis e Interpretación

mediante el análisis de la pregunta 9, el 57% afirma que no ha tenido confusiones o errores en el momento de informar a los clientes sobre el estado de sus pedidos, el 43% afirma que sí ha tenido confusiones o errores. se puede definir que en su mayoría no se ha tenido errores o confusiones al momento de informar al cliente sobre el estado de sus pedidos.

Pregunta 10. ¿Qué tan satisfactorio es el proceso de gestión de pedidos del cliente mediante un aplicativo móvil?

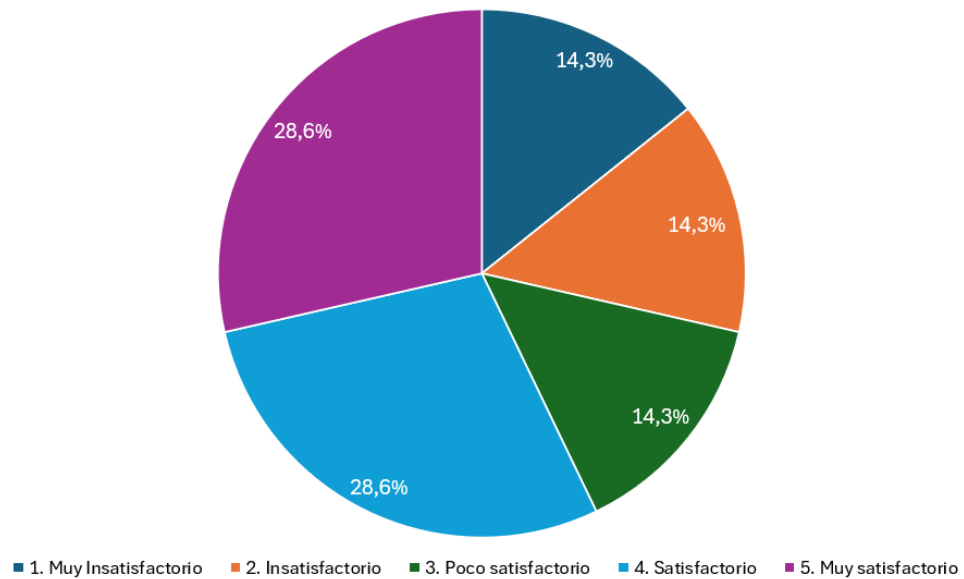


Figura 10. Satisfacción en el proceso de gestión de pedidos del cliente mediante una aplicación móvil.

Análisis e interpretación

En el análisis de la pregunta 10, el 28,6% afirma que es muy satisfactorio el proceso de gestión de pedidos mediante un aplicativo móvil, el otro 28,6% argumenta que es satisfactoria el proceso de gestión de pedidos con una aplicación móvil, el 14,3% afirman que es muy insatisfactorio, el 14,3% insatisfactorio y el otro 14,3% poco insatisfactorio al momento de realizar la gestión de pedidos mediante una aplicación móvil. Mediante los resultados analizados la mayor parte del personal encuestado argumenta que es muy satisfactorio y satisfactorio realizar el proceso de gestión de pedidos con una aplicación móvil puesto que agiliza y reduce el tiempo de respuesta a cualquier pedido que se requiera realizar.

Validación de la entrevista

Para una mejor confiabilidad del instrumento de entrevista se logró validó por criterio de expertos.

Resultados de la entrevista aplicada al administrador de la empresa

<p>Datos personales</p> <p>Apellido y Nombre entrevistado: Ing. Patricio Constante</p> <p>Institución: Connectaservices</p> <p>Cargo: Gerente general</p> <p>Objetivo: Determinar los inconvenientes que enfrenta la empresa Connectaservices en la toma de pedidos y el seguimiento de entregas a domicilio.</p>			
Nº	Preguntas	Respuesta	Observaciones
1	¿Cómo se lleva el registro de pedidos a domicilio?	Actualmente por medio de WhatsApp	Gracias a una base de datos definida se procede a tener un registro de pedidos realizado por cliente
2	¿Cuáles son los inconvenientes que presentan los empleados del restaurante al receptor los pedidos?	Que no es con información muy precisa por ejemplo al ser un mensaje a veces la gente quiere el almuerzo completo, pero en el mensaje solo ingresa el plato fuerte sea la opción de fuerte que toca escribir de nuevo y decirle que eso parece entonces se pierde tiempo eso es un problema de recepción acá que tenemos	En la integración que compete a uno de los objetivos del proyecto se da solución mediante la comunicación del sistema de restaurantes con la aplicación móvil, para guardar un registro tanto en el sistema de restaurantes como en la aplicación móvil.
3	¿Cuáles son las quejas del personal y de los clientes que recibe al momento de receptor los pedidos a domicilio?	La mayor queja del personal es que se manda información imprecisa o incompleta, en	La comunicación precisa y eficiente es crucial en el momento de gestionar los pedidos, como tal

		cuanto al cliente las quejas generalmente es el tiempo que se demora en hacer el pedido por ejemplo el cliente pide a las 12pm le está llegando casi 1 hora después por la ineficiencia de la gestión	se implementó mediante el servicio de woocommerce el ingreso de nuevos pedidos y el rápido manejo de la información desde el restaurante mediante la cajera del restaurante.
4	¿Cómo describiría la eficiencia del registro de los pedidos en curso en la empresa?	Mala ahorita es mala porque pues todo se lo hace describiendo en una hoja, no es nada digitalizada	Se implementó una solución digitalizada, eficiente y rápida en la toma de pedidos.
5	¿Qué dificultades ha experimentado al comunicar información sobre el estado de los pedidos a los clientes?	Generalmente como el restaurante está un poquito lleno no se los notifica, simplemente nos da el pedido nos dice que quiere y pues simplemente hasta a veces ni le avisamos que ya se fue motorizado	La comunicación del cliente con el restaurante es importante para eso se implementó notificaciones para saber si su pedido está preparándose y cuando se lo despachó
6	¿Considera que el seguimiento de entregas a domicilio en la empresa es lo suficientemente eficiente? ¿Por qué sí o por qué no?	no, es muy ineficiente por qué todo es manual, el hecho de que escriba por WhatsApp o llamen a veces ni se contesta y que se manejan papeles	La implementación de notificaciones facilita la llegada de información del cliente con el restaurante en la gestión de pedidos.
7	¿En qué medida cree que la ausencia de un sistema de delivery afecta la capacidad de Connectaservices para gestionar eficazmente los pedidos y entregas?	Realmente la ausencia de un sistema afecta bastante porque primero no hay un	Este proceso se facilita en tiempo y forma, ya que el cliente tiene conocimiento del

		correcto seguimiento y no se puede gestionar de una manera muy rápida y sobre todo lo que sepa el cliente que ya el pedido está en camino	manejo de su pedido y el restaurante tiene el conocimiento y el registro para la elaboración y despacho del delivery
8	¿Cómo describiría la precisión de los registros de pedidos en comparación con las expectativas de los clientes?	Las expectativas del cliente es que piensan que todo es digital y que el pedido entra ya, pero realmente es ineficiente	El cliente del restaurante puede experimentar un cambio en la toma de pedidos ya que todo el proceso de delivery se convierte en un proceso digitalizado
9	¿Qué problemas adicionales ha experimentado al gestionar la disponibilidad de productos y el seguimiento de entrega sin una solución digitalizada?	Los problemas son que, a veces hay falta de comunicación con la cocina, saber si es que está disponible ese plato todavía, cuando hacen un delivery o sea digamos que piden que el plato no hay	El cliente tendrá el conocimiento la disponibilidad del menú al momento de generar un nuevo pedido
10	¿En qué medida afecta a la empresa no contar con una óptima gestión de delivery?	Afecta económicamente porque se pudiera a un incrementar las ventas ofreciendo un delivery eficiente si pudiera haber más gente que pida	Esta solución incrementa económicamente los ingresos del restaurante ya que permite tener mas confianza al cliente y así poder atraer mas clientela al restaurante

Conclusión: La entrevista realizada al gerente de la empresa Conectaservices demuestra que existe una gestión de pedidos deficiente, se pudo evidenciar la necesidad de una solución digital en la toma de pedidos junto a su respectivo registro y notificación del cliente del restaurante con la gestión de pedidos.

2.2.4 Procesamiento y análisis de datos

Una vez obtenidos los datos recopilados con los instrumentos aplicados para las personas encargadas del motorizado, personas encargadas de realizar los pedidos, personal encargado de la recepción de pedidos y una encuesta para el gerente de la empresa se llegó a las siguientes conclusiones:

- Los datos sugieren que existe una necesidad de mejorar la gestión de pedidos y entregas a domicilio, ya que los clientes experimentan insatisfacción con el sistema actual.
- Se recomienda implementar una plataforma digital que permita a los clientes realizar pedidos y recibir notificaciones sobre las entregas. Esta plataforma debería integrarse con el sistema de gestión del restaurante para optimizar la eficiencia y la comunicación.
- La implementación de una plataforma y otras herramientas digitales puede tener un impacto positivo en la satisfacción del cliente, aumentando la eficiencia, la transparencia y la comunicación.
- Una gestión de pedidos más eficiente puede conducir a un aumento de las ventas y la rentabilidad para el restaurante.

En la **Figura 11** se puede describir de mejor manera el análisis de los instrumentos aplicados.

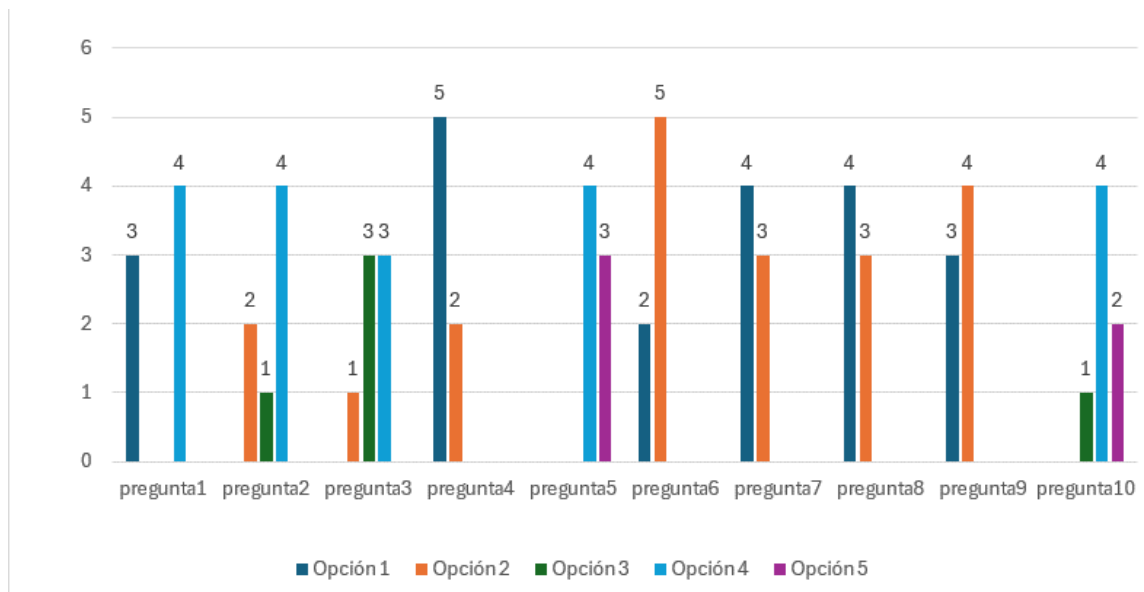


Figura 11. Estadística de respuestas de la encuesta

Mediante la **Figura 11** se puede visualizar la frecuencia en las respuestas agrupadas por las preguntas que tienen varios ítems (preguntas 1, 2, 3, 5, 10) con las preguntas de si y no (preguntas 4, 6, 7, 8, 9) de esta manera podemos visualizar mediante las preguntas anteriormente respondidas la variación de y consistencia en las respuestas que se imponen como tema de estudio dando por denotación la importancia de una solución digital a los inconvenientes que tiene la empresa Connectaservices con el sistema de restaurantes trasladado a un restaurante ya operando en la ciudad de Ambato. Puesto que no se tiene una buena comunicación con el usuario, repartidor, y el recepcionista se pretende generar una solución en donde visualmente el usuario pueda realizar el seguimiento en tiempo real de cual es el estado de su pedido, así también como el repartidor poder realizar la actualización necesaria de cada pedido para garantizar si el pedido esta siendo atendido o no, adicionalmente se denota la necesidad de poder generar una solución a el registro de pedidos ya que en la actualidad el restaurante mediante el sistema proporcionado por la empresa Connectaservices no tiene la función para poder guardar un registro de los pedidos que los usuarios realizan en ese restaurante.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y Discusión

La empresa “Connectaservices” cuenta con un diagrama de procesos en la gestión de pedidos lo cual en la **Figura 12** se implementa una tarea más que es la notificación y la comprobación de la entrega de pedidos.

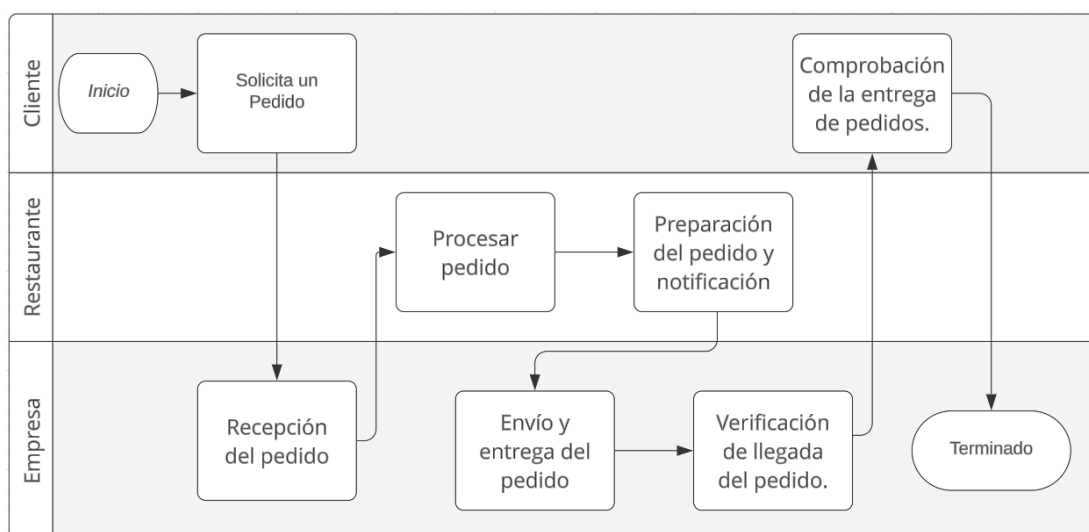


Figura 12. Diagrama de procesos en la gestión de delivery

Descripción de procesos

Tabla 10. Descripción de Procesos

Nº	Proceso	Descripción
1	Solicita un pedido.	El cliente escoge lo que el menú muestre a su disposición e ingresa la información necesaria al momento de realizar la entrega.
2	Recepción del pedido.	La empresa recibe información detallada del producto solicitado por el cliente. Se recaban datos importantes como cantidad y características del producto en cuestión.

3	Procesar pedido.	Se verifica que el pedido sea el correcto y se concreta una forma de pago mediante la cajera.
4	Preparación del pedido y notificación.	En el sistema de la empresa se procede a preparar el pedido en el restaurante y se procede a notificar al cliente que el pedido se está preparando.
5	Envío y entrega del pedido.	La cajera del restaurante procede a contratar un repartidor externo el cual se encargará de la entrega del pedido, siempre antes de notificar al cliente la salida de su pedido.
6	Verificación de llegada del pedido.	El cajero se pone en contacto con el cliente para verificar la llegada de su pedido.
7	Comprobación de la entrega de pedidos.	Una vez hecha la entrega de mercancía, se deja constancia de esta a través de un estado, foto u otras opciones.

3.1.1 Análisis comparativo de versiones de react native para el desarrollo de aplicaciones móviles

En el desarrollo de aplicaciones móviles para gestión de delivery, es importante y crucial seleccionar la versión adecuada y se debe tomar en cuenta factores como rendimiento, lenguaje de programación, renderización, entre otros, los cuales son temas importantes al momento de poder realizar la respectiva selección.

Para determinar cuál es la versión más idónea para el proyecto, se desarrolló un cuadro comparativo donde se analiza de manera profunda las características y beneficios que tiene cada una de las versiones de react native como se puede notar en la siguiente tabla:

Tabla 11. Cuadro comparativo de versiones de react native

Versión	Características principales	Ventajas	Desventajas
0.41	Lanzamiento inicial de React Native	<ul style="list-style-type: none"> - Facilidad de aprendizaje para desarrolladores de JavaScript - Desarrollo de aplicaciones multiplataforma - Rendimiento nativo 	Falta de algunas características nativas de las plataformas móviles
0.47	Mejoras en el rendimiento y la estabilidad	<ul style="list-style-type: none"> - Mejoras en la carga de imágenes - Mejoras en la navegación - Mejoras en el manejo de eventos 	- Todavía faltan algunas características nativas de las plataformas móviles
0.57	Implementación de componentes de navegación nativos	<ul style="list-style-type: none"> - Mejoras en la experiencia del usuario - Mayor rendimiento - Mayor estabilidad 	- Aumento de la complejidad del código
0.59	Implementación de hooks	<ul style="list-style-type: none"> - Simplificación del código - Mayor reutilización de componentes - Mayor facilidad de mantenimiento 	- Curva de aprendizaje para los hooks
0.63	Implementación de la arquitectura Jest	<ul style="list-style-type: none"> - Mejoras en las pruebas unitarias - Mayor confiabilidad del código 	- Aumento del tamaño del bundle de la aplicación
0.64	Implementación de la arquitectura Hermes	<ul style="list-style-type: none"> - Mayor rendimiento de JavaScript - Reducción del tamaño del bundle de la aplicación 	- Incompatibilidad con algunas bibliotecas de terceros
0.65	Implementación de la API de depuración remota	<ul style="list-style-type: none"> - Facilidad de depuración de aplicaciones en dispositivos reales 	- Requiere la instalación de herramientas adicionales
0.68	Implementación de la API de reanimación	<ul style="list-style-type: none"> - Mejoras en las animaciones - Mayor fluidez de las interfaces de usuario 	- Aumento de la complejidad del código

0.7	Implementación de la API de composición de estilos	- Simplificación de la creación de estilos CSS - Mayor flexibilidad en el diseño de interfaces de usuario	- Curva de aprendizaje para la nueva API de estilos
0.71	Implementación de la API de suspensión	- Mejoras en el rendimiento de las aplicaciones con muchos componentes - Mayor eficiencia en el uso de la memoria	- Requiere la actualización de las dependencias de la aplicación

La mejor versión de React Native para el desarrollo de aplicaciones móviles depende de las necesidades específicas del proyecto. Si se necesita una solución rápida y sencilla, la versión 0.41 puede ser una buena opción. Sin embargo, se requiere una aplicación con alto rendimiento y características avanzadas, las versiones recientes como 0.71 es la más recomendable.

Muchas empresas han utilizado React native en sus aplicaciones móviles, en la **Tabla 12** se presentan las más representativas de está tomando los criterios necesarios, se comprueba que React Native es un framework óptimo para poder realizar aplicaciones móviles.

Tabla 12. Empresas que utilizan React native en aplicaciones es móviles

Aplicación	Descripción	Categoría
Facebook	Red social	Redes sociales
Instagram	Red social para compartir fotos y videos	Redes sociales
Skype	Aplicación de mensajería instantánea y videollamadas	Comunicación
Uber Eats	Servicio de entrega de comida a domicilio	Comida y bebida
Tesla	Aplicación para controlar tu vehículo Tesla	Transporte
Pinterest	Red social para descubrir ideas	Redes sociales

Soundcloud	Plataforma para compartir y escuchar música	Música
Bloomberg	Aplicación de noticias y finanzas	Noticias
Duolingo	Aplicación para aprender idiomas	Educación
Maybelline	Aplicación de belleza y maquillaje	Belleza

3.1.2 Análisis comparativo de versiones de Node.js para el desarrollo de backend en aplicaciones móviles

Node.js se ha convertido en una de las opciones más populares para el desarrollo de backend de aplicaciones móviles debido a su velocidad, escalabilidad y facilidad de uso. Sin embargo, con el lanzamiento de nuevas versiones, es importante realizar un análisis comparativo para determinar cuál es la mejor opción para el proyecto de desarrollo.

En este análisis, se comparó las versiones más recientes de Node.js (v20.12.2 LTS, v18.20.2 LTS y v21.12.2 Current) en términos de rendimiento, características y estabilidad.

Rendimiento

Según la página oficial de Node.js argumenta que en [1] términos de rendimiento, Node.js v20.12.2 LTS ha demostrado ser la versión más rápida, con mejoras significativas en la velocidad de ejecución de JavaScript y la utilización de la memoria. Node.js v18.20.2 LTS también ofrece mejoras de rendimiento, aunque no son tan significativas como las de Node.js v20.12.2 LTS. Node.js v21.12.2 Current sigue siendo una versión sólida, pero es probable que experimente un rendimiento inferior en comparación con las versiones más recientes agregando a que esta versión no lleva un tiempo en el mercado y no contiene con todo el soporte obligatorio.

Características

Node.js v20.12.2 LTS introduce varias características nuevas, incluyendo:

Nuevas APIs de diagnóstico: Estas APIs proporcionan información más detallada sobre el rendimiento de la aplicación, lo que puede ayudar a identificar y solucionar problemas.

Soporte mejorado para WebAssembly: WebAssembly es un formato de compilación que permite ejecutar código C++ en JavaScript. Node.js v20.12.2 LTS ofrece un soporte mejorado para WebAssembly, lo que puede mejorar el rendimiento de las aplicaciones que utilizan código C++.

Nuevas características de seguridad: Node.js v20.12.2 LTS incluye varias nuevas características de seguridad, como la capacidad de deshabilitar la ejecución de código arbitrario (ASLR) y la capacidad de restringir el acceso a los recursos del sistema.

Node.js v18.20.2 LTS y 14 también incluyen nuevas características, pero no tantas como Node.js v20.12.2 LTS.

Estabilidad

Todas las versiones de Node.js son estables y han sido ampliamente probadas. Sin embargo, Node.js v20.12.2 LTS es la versión más reciente y, por lo tanto, es probable que tenga menos errores y problemas de estabilidad que las versiones anteriores.

3.1.3 Análisis de gestor de base de datos para el desarrollo aplicaciones móviles

En el desarrollo de aplicaciones móviles para la gestión de delivery es importante determinar que gestor de base de datos para el manejo de la información de la aplicación móvil, cabe recalcar que la empresa cuenta con su base de datos y su gestor de base de datos ya establecida utilizando MySQL, puesto que el proyecto se acopla para el desarrollo mediante el gestor de bases de datos MySQL.

Características de MySQL

Tabla 13. Características de MySQL

Característica	Descripción
Tipo de base de datos	Relacional
Lenguaje de consulta	SQL (Structured Query Language)
Licencia	Open-source y gratuita bajo la licencia GNU
Plataformas compatibles	Windows, macOS, Linux, Unix, Solaris
Almacenamiento	Soporta grandes bases de datos con hasta 64 índices por tabla
Motores de almacenamiento	Ofrece una variedad de motores de almacenamiento, incluyendo InnoDB, MyISAM, NDB, etc.
Funciones	Soporta funciones de cadena, matemáticas, fecha/hora, etc.
Escalabilidad	Altamente escalable, capaz de manejar grandes cargas de trabajo
Seguridad	Ofrece un conjunto de características de seguridad, como autenticación de usuario, cifrado de datos y control de acceso
Herramientas	Amplia gama de herramientas disponibles, incluyendo MySQL Workbench, MySQL Connector/ODBC, MySQL Shell
Popularidad	Uno de los sistemas de gestión de bases de datos relacionales más populares del mundo

Muchas empresas han utilizado MySQL en sus aplicaciones móviles, en la **Tabla 14** se presentan las más representativas.

Tabla 14. Aplicaciones móviles que utilizan MySQL

Aplicación	Descripción	Categoría
Gmail	Cliente de correo electrónico	Productividad
Facebook	Red social	Redes sociales
Twitter	Red social de microblogging	Redes sociales
Instagram	Red social para compartir fotos y videos	Redes sociales
Pinterest	Red social para compartir imágenes	Redes sociales

Uber	Aplicación para solicitar viajes en coche	Transporte
Airbnb	Plataforma para reservar alojamientos	Viajes
Booking.com	Plataforma para reservar hoteles y alojamientos	Viajes
PayPal	Sistema de pago en línea	Finanzas
eBay	Plataforma de comercio electrónico	Compras
Amazon	Tienda minorista en línea	Compras
Spotify	Servicio de streaming de música	Música
Netflix	Servicio de streaming de video	Entretenimiento
YouTube	Plataforma para compartir videos	Entretenimiento
Evernote	Aplicación para tomar notas y organizar tareas	Productividad
Dropbox	Servicio de almacenamiento en la nube	Productividad
Slack	Herramienta de comunicación empresarial	Productividad

3.1.4 Metodologías de desarrollo de aplicaciones móviles

La implementación de una metodología en el desarrollo de software trae consigo diversos beneficios, entre ellos la optimización del proceso y la mejora en la calidad del producto final. Al elegir una metodología ágil, se gana la flexibilidad necesaria para adaptarse rápidamente a los requerimientos del sistema y del usuario. Esto se traduce en una gestión más eficiente de los recursos y el tiempo, permitiendo la entrega oportuna de prototipos funcionales.[10]

Tabla 15. Estudio comparativo de metodología de desarrollo de aplicaciones móviles

	XP (Programación Extrema)	Mobile D	FDD (Feature Driven Development)
Enfoque	Se centra en la comunicación, la colaboración y la adaptación a los cambios.	Adaptación de XP para el desarrollo móvil, con énfasis en la simplicidad y la entrega rápida de valor.	Se centra en la entrega incremental de funcionalidades valiosas, con una fuerte planificación y seguimiento.

Ciclo de desarrollo	Ciclos cortos de desarrollo ("iteraciones") con entregas frecuentes.	Ciclos de desarrollo de 2 a 4 semanas, con lanzamientos semanales o quincenales.	Ciclos de desarrollo más largos (6 a 12 semanas), divididos en fases de planificación, diseño, desarrollo y pruebas.
Planificación	Planificación flexible y adaptativa, con priorización continua de tareas.	Planificación inicial detallada, con actualizaciones regulares a medida que avanza el proyecto.	Planificación detallada a nivel de funcionalidad, con flexibilidad para ajustar el alcance según sea necesario.
Diseño	Diseño emergente, con énfasis en la simplicidad y la refactorización.	Diseño por contrato, con interfaces bien definidas entre componentes.	Modelado de objetos detallado, con énfasis en la reutilización y la extensibilidad.
Pruebas	Pruebas unitarias y de aceptación exhaustivas a lo largo del ciclo de desarrollo.	Pruebas unitarias y de integración continuas, con pruebas de aceptación al final de cada ciclo.	Pruebas unitarias, de integración y de sistema exhaustivas, con pruebas de aceptación al final de cada fase.
Herramientas	Herramientas de colaboración y seguimiento de tareas, herramientas de pruebas unitarias, herramientas de integración continua.	Entorno de desarrollo integrado (IDE) móvil, herramientas de pruebas unitarias y de integración, herramientas de gestión de lanzamientos.	Herramientas de modelado de objetos, herramientas de gestión de requisitos, herramientas de seguimiento de errores, herramientas de pruebas unitarias, de integración y de sistema.
Ventajas	Entregas rápidas, alta calidad, adaptación a cambios, fuerte colaboración.	Simplicidad, enfoque en el valor, entrega rápida de aplicaciones móviles.	Planificación sólida, seguimiento del progreso, entrega de funcionalidades valiosas.
Desventajas	Requiere un equipo experimentado y disciplinado, puede ser caótico para equipos grandes.	Puede ser demasiado rígido para proyectos con requisitos cambiantes, requiere herramientas específicas.	Ciclos de desarrollo más largos, mayor tiempo de inicio de tarea de planificación, puede ser menos adaptable a cambios.

Se seleccionó la metodología ágil Mobile D ya que en el estudio realizado en la **Tabla 15** sobresale en el desarrollo de proyectos móviles gracias a su enfoque centrado en el usuario y su flexibilidad para adaptarse a los cambios. Para el proyecto, esto se traduce en una colaboración directa con los usuarios, ciclos de desarrollo ágiles que permiten ajustes rápidos y pruebas continuas para asegurar una experiencia móvil de alta calidad. [12]

Fases de la metodología Mobile-D

1. **Exploración:** En esta etapa, el equipo de desarrollo se dedica a entender qué es lo que los usuarios necesitan y esperan. Para lograrlo, llevan a cabo acciones como identificar quiénes son los interesados, delimitar qué incluirá el proyecto y establecer las metas específicas del mismo.
2. **Inicialización:** Durante este período, el equipo de desarrollo delinea claramente qué abarcará el proyecto, establece los objetivos clave de la aplicación y elabora un plan detallado de cómo llevarlo a cabo. Además, se identifican y preparan los elementos necesarios para el proyecto, como los prototipos que mostrarán el funcionamiento de la aplicación y los casos de prueba que se usarán para garantizar su calidad.
3. **Fase de producto:** Durante esta etapa, se lleva a cabo el desarrollo de la aplicación móvil. Este proceso se fragmenta en ciclos de desarrollo iterativos, en los que se trabaja en una sección específica de la aplicación y se somete a pruebas exhaustivas para garantizar su funcionamiento adecuado.
4. **Fase de estabilización:** Durante esta etapa, se ejecutan pruebas exhaustivas en la aplicación móvil para asegurar su calidad y correcto funcionamiento. Además, se llevan a cabo ajustes y mejoras necesarias en la aplicación basados en los resultados de las pruebas realizadas.
5. **Fase de pruebas:** En este período, la aplicación móvil se somete a pruebas con usuarios reales con el fin de recabar sus opiniones y comentarios. Estos aportes son fundamentales para realizar mejoras y ajustes finales en la aplicación antes de su lanzamiento oficial.

3.2 Desarrollo de la Propuesta

A continuación, se presentará las fases que conforma la aplicación de la metodología anteriormente seleccionada.

3.2.1 Fase 1: Exploración

En esta fase se determinó qué es lo que los usuarios necesitan y esperan.

a. Definición de Stakeholders

- Usuarios de la aplicación móvil: Clientes que formen parte del sistema de restaurantes.
- Desarrollador: Autor del presente proyecto de investigación.

b. Definición del Alcance

- Las funcionales del aplicativo móvil se tomaron en consideración mediante las respuestas generadas en los instrumentos de recolección de la información realizadas al personal del restaurante y también al gerente de la empresa.
- La aplicación presentará funcionalidades para el usuario identificado como: el Restaurante ya que se lo considera como un cliente con respecto a la empresa Connectaservices mediante su sistema de restaurantes. También es factible mencionar que la interfaz y funcionalidad está impartida para el cliente del restaurante.
 - Las funcionalidades para el cliente de restaurante son: Inicio de Sesión, registro de usuarios, Cambio de contraseña en caso de olvido, editar la información del usuario, Acceso a la información de menús de comidas dependiendo la sucursal del establecimiento, generación de pedido, agregar adicionales a la reservación de Menús.
- Limitaciones:
 - La Aplicación necesita que el dispositivo se encuentre con conexión a internet.
 - La aplicación tiene como objetivo principal la generación y gestión de pedidos en el sistema de restaurantes de la empresa Connectaservices.

c. Establecimiento de dependencias

- La aplicación está destinada para las personas clientes del restaurante Gourmendy.

d. Requisitos funcionales de la aplicación móvil

En la **Tabla 16** se especifican los requisitos funcionales presentados para el usuario.

Tabla 16. Requisitos funcionales

Número de requerimiento	Requerimiento	Descripción
RQ1	Registro del usuario	El usuario deberá registrarse antes de poder ingresar y usuario los servicios del aplicativo móvil.
RQ2	Inicio de sesión	El usuario podrá ingresar al aplicativo móvil con el correo y contraseña registrados en el listado de usuarios.
RQ3	Recuperar contraseña	El usuario puede solicitar la contraseña mediante un enlace enviado por correo el cual podrá usar para cambiar la contraseña olvidada.
RQ4	Información de menús de sucursales	El usuario puede visualizar los menús que proporciona el restaurante dependiendo de cada sucursal.
RQ5	Selección y gestión de pedido	El usuario puede generar un pedido a la sucursal seleccionada, también opcionalmente puede solicitar extras.
RQ6	Notificaciones al cliente y cajero de restaurante	El usuario podrá recibir notificaciones a su dispositivo móvil comunicándole el estado de su pedido. El cajero de restaurante deberá ser informado mediante notificaciones la solicitud de pedido de parte del usuario.
RQ7	Editar perfil	El usuario puede editar la información de su perfil anteriormente registrado.

e. Establecimiento del proyecto

A continuación, se detallan las herramientas que se usaron en el desarrollo del proyecto, cabe recalcar que cada una de las herramientas planteadas aportaron para el rápido desarrollo mediante un código fuente estable:

- Visual Studio Code

Es un editor de código fuente disponible para Windows y ios para el desarrollo de aplicaciones móviles.

- React native

Es un Framework que permite la creación de aplicaciones móviles híbridas de manera personalizada.

- Expo Cli.

Es una herramienta de línea de comandos que permite el desarrollo de aplicaciones móviles y facilita el debug. Gracias a sus extensas funcionalidades, expo nos permite poder generar código responsivo para móviles Android e IOS, de esta manera expo permite poder depurar en tiempo real una aplicación híbrida.

- Node.js

Es un entorno de código abierto para el desarrollo de la capa del servidor y la comunicación de la base de datos.

- MySQL

Es un sistema de gestión de bases de datos que permite almacenar y obtener datos de forma rápida y sencilla.

3.2.2 Fase 2: Inicialización

Se Procede a identificar los recursos para el desarrollo y el diseño de la aplicación móvil.

a. Herramientas del ambiente de desarrollo

A continuación, se detallan las herramientas utilizadas para el desarrollo del proyecto.

- Tipo de aplicación: Aplicación nativa
- Framework: React native
- Lenguaje de programación: typeScript
- Servicio de autenticación: webToken, expo authentication
- Gestor de base de datos: MySql

b. Preparación del ambiente

Las herramientas instaladas para el desarrollo del proyecto son las siguientes:

- Visual Studio Code
- Xampp
- MySql
- React native
- Expo Cli
- Node js

Estas herramientas permitieron que la aplicación pueda ser efectivamente rápida en la navegación, obtención de datos y mucho mas optima a poder realizar callbacks de manera rápida y generar una comunicación estable entre el frontend y el backend.

c. Planificación de fases

Tabla 17. Planificación de fases

Fase	Ítem	Descripción
Exploración	Item 0	Establecer stakeholders Definir alcances y limitaciones Establecimiento del proyecto
Iniciación	Item 1	Configuración del entorno o ambiente de desarrollo Preparación de entorno
Producción	Item 2	Desarrollo de la interfaz de inicio Elaboración de storycard Pruebas de funcionamiento
	Item 3	Desarrollo de interfaz de registro de usuario Elaboración de storycard Pruebas de funcionamiento
	Item 4	Desarrollo de interfaz de recuperación de contraseña Elaboración de storycard Pruebas de funcionamiento
	Item 5	Desarrollo de interfaz principal Elaboración de storycard Pruebas de funcionamiento
	Item 6	Desarrollo de interfaz de nuevo pedido Elaboración de storycard Pruebas de funcionamiento
	Item 7	Implementación de servicios que sincronicen los pedidos mediante woocommerce Elaboración de storycard Pruebas de funcionalidad
	Item 8	Desarrollo de la interfaz de inicio para repartidor Elaboración de storycard Pruebas de funcionamiento
	Item 9	Desarrollo de la interfaz de seguimiento al delivery para repartidor Elaboración de storycard Pruebas de funcionamiento

	Item 10	Implementación de notificaciones para la creación de pedidos, preparación y despacho de pedidos Elaboración de storycard Pruebas de funcionalidad
	Item 11	Implementación y revisión de controles de interfaz Elaboración de storycard Pruebas de funcionalidad
Estabilización	Item 12	Corrección de errores de funcionalidad Elaboración de storycard Pruebas de funcionalidad
Pruebas	Item 13	Pruebas de la aplicación móvil conforme a sus funcionalidades

d. Diseño de la base de datos

El gestor de base de datos mySql es de tipo relacional, por el cual nos permite poder obtener la información y alojar información que va a guardar y obtener el aplicativo móvil. Para este proyecto la empresa Connectaservices proporcionó la base de datos a usar puesto que la información se manejará mediante el sistema de restaurantes que se encuentra en uso. Además, la empresa brinda si servicio woocommerce conectado directamente a su sistema para la conexión de peticiones.

- Tablas principales

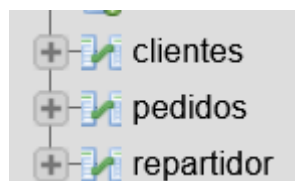


Figura 13. Tablas Principales

- Estructura tabla clientes

<i>Cientes</i>	
<i>id</i>	<i>int</i>
<i>ruc</i>	<i>varchar</i>
<i>nombres</i>	<i>varchar</i>
<i>direccion</i>	<i>varchar</i>
<i>telefono</i>	<i>varchar</i>
<i>nacimiento</i>	<i>date</i>
<i>email</i>	<i>varchar</i>
<i>clave</i>	<i>varchar</i>
<i>imagen</i>	<i>longblob</i>
<i>puntos</i>	<i>int</i>

Figura 14. Estructura tabla Clientes

- Estructura tabla pedidos

<i>Pedidos</i>	
<i>id</i>	<i>int</i>
<i>id_fkcliente</i>	<i>int</i>
<i>id_fkmenusucursal</i>	<i>int</i>
<i>tipo_variacion</i>	<i>varchar</i>
<i>flag_estado_pedido</i>	<i>varchar</i>
<i>created_at</i>	<i>date</i>
<i>updated_at</i>	<i>date</i>
<i>detalle_pedido</i>	<i>longtext</i>
<i>id_fkrepartidor</i>	<i>int</i>
<i>direccion</i>	<i>varchar</i>
<i>required</i>	<i>tinyint</i>

Figura 15. Estructura tabla pedidos

- Estructura tabla repartidor

<i>Repartidor</i>	
<i>id</i>	<i>int</i>
<i>ruc</i>	<i>varchar</i>
<i>nombres</i>	<i>varchar</i>
<i>direccion</i>	<i>varchar</i>
<i>telefono</i>	<i>varchar</i>
<i>nacimiento</i>	<i>date</i>
<i>email</i>	<i>varchar</i>
<i>clave</i>	<i>varchar</i>
<i>imagen</i>	<i>longblob</i>
<i>puntos</i>	<i>int</i>
<i>rol</i>	<i>int</i>

Figura 16. Estructura tabla repartidor

e. Diagrama de caso de uso

Este diagrama nos permite visualizar las acciones que los usuarios de la aplicación móvil pueden realizar, en la **Figura 17** se podrá visualizar las acciones que el usuario puede realizar.

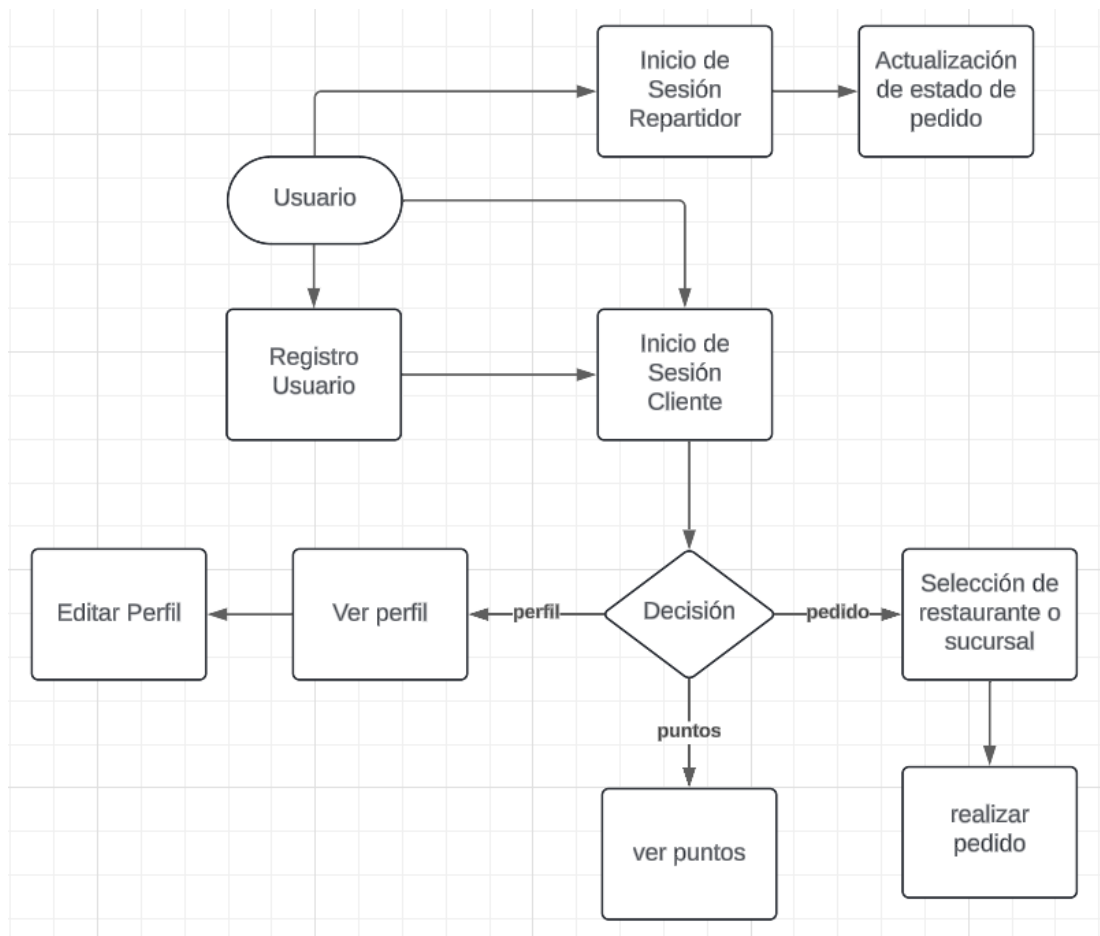


Figura 17. Diagrama de casos de uso de cliente del restaurante

f. Prototipo de la aplicación

Los prototipos son importantes al momento de desarrollar una aplicación móvil, ya que facilitan la visualización del diseño y la visualización de funcionalidad antes de su creación, esto permitirá visualizar la funcionalidad de manera garantizada los requerimientos del usuario.

En ese caso la empresa proporcionó el diseño, modelos puestos para la interfaz de usuario.

- Pantalla de carga

Al momento de ingresar a la aplicación se visualiza una página de carga el cual hará la carga de todos los procesos.



Figura 18. Pantalla de Carga

- Pantalla de Inicio

En esta pantalla le permitirá al usuario elegir entre iniciar sesión o registrarse dependiendo del usuario.



Figura 19. Mackup Pantalla de Inicio

- Iniciar Sesión

El usuario en la pantalla de inicio de sesión podrá realizar diferentes acciones como iniciar sesión, cambiar la contraseña por olvido, registrar nuevo usuario, iniciar sesión con servicios de autenticación en Gmail o Facebook.

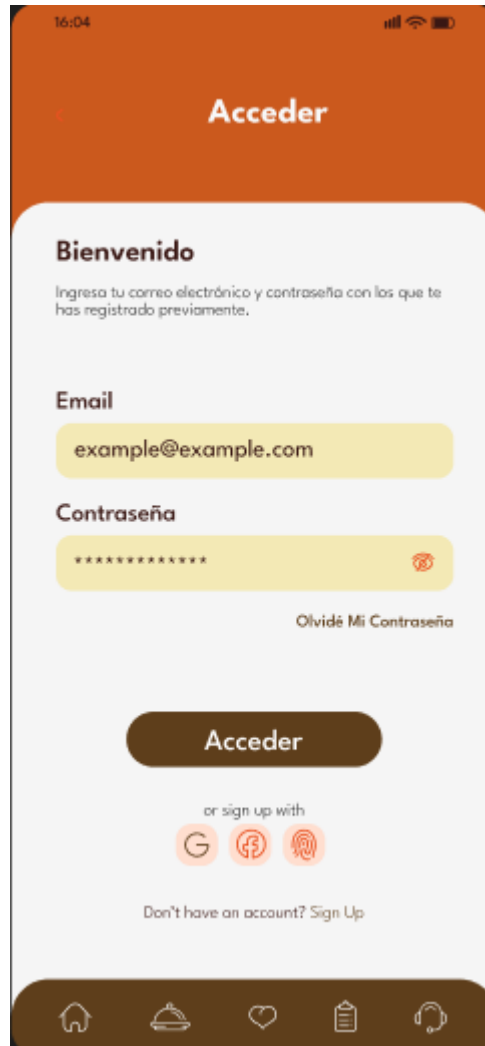


Figura 20. Mockup Inicio de Sesión

- Registro de usuario

En la pantalla de registro el usuario podrá registrar una nueva cuenta para poder ingresar al sistema y obtener los servicios de la aplicación.



The image shows a mobile app interface for user registration. At the top, there's an orange header with the text "Crea tu cuenta". Below this, the form is divided into several sections, each with a label and a corresponding input field:

- Nombre y Apellido:** A yellow input field containing "example@example.com".
- Contraseña:** A yellow input field with masked characters "*****" and a red eye icon to toggle visibility.
- Email:** A yellow input field containing "example@example.com".
- Whatsapp:** A yellow input field containing "0995054979".
- Fecha de Nacimiento:** A yellow input field with the placeholder "DD / MM / YYYY".

Below the input fields, there is a small text line: "Para continuar, debes aceptar nuestros Terminos de Use y Política de Privacidad." followed by a large, dark brown button labeled "Registrarme".

Under the button, it says "O registrarme con" followed by two circular icons: a Google 'G' logo and a Facebook 'f' logo.

At the bottom of the form area, it says "¿Ya tienes una cuenta? Accede".

The bottom of the screen features a dark brown navigation bar with five white icons: a home icon, a shopping cart icon, a heart icon, a document icon, and a speech bubble icon.

Figura 21. Mackup Registro de Usuario

- Pagina mi perfil

La página de mi perfil permite actualizar perfil y visualizar el perfil.



Figura 22. Pantalla mi Perfil

- Página de Inicio

La página de inicio está conformada con la información de las sucursales al seleccionar debe dirigirme a la pantalla de generar un pedido.



Figura 23. Mackup Página de Inicio

- Pantalla Mis Órdenes

En esta pantalla se podrán visualizar los pedidos que el usuario tiene realizadas o las que están en proceso o ya terminaron esta pantalla tiene dos variaciones, si el usuario no tiene órdenes y si el cliente tiene órdenes.

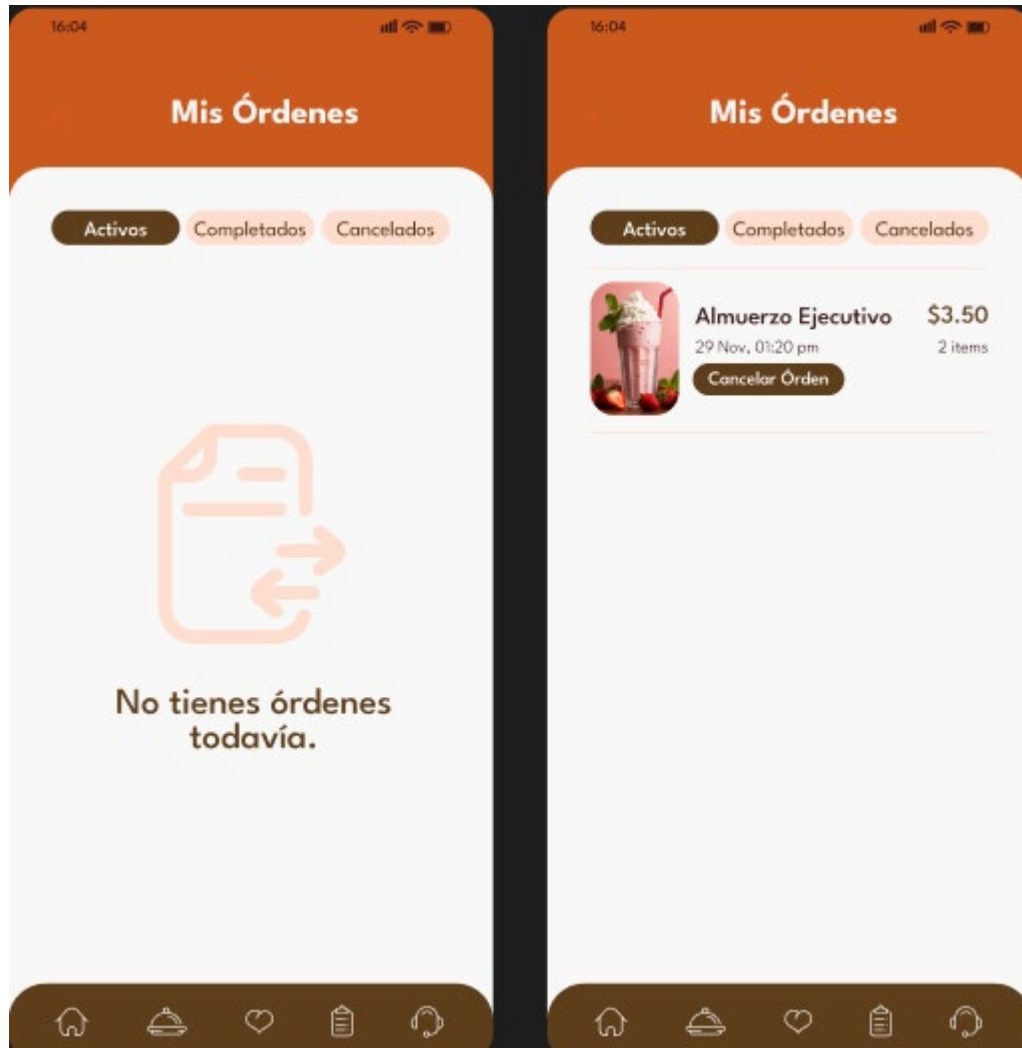


Figura 24. Mackup Pantalla Mis Órdenes

- Pantalla mis Direcciones

En esta pantalla se muestra las direcciones en donde se encuentra cada restaurante

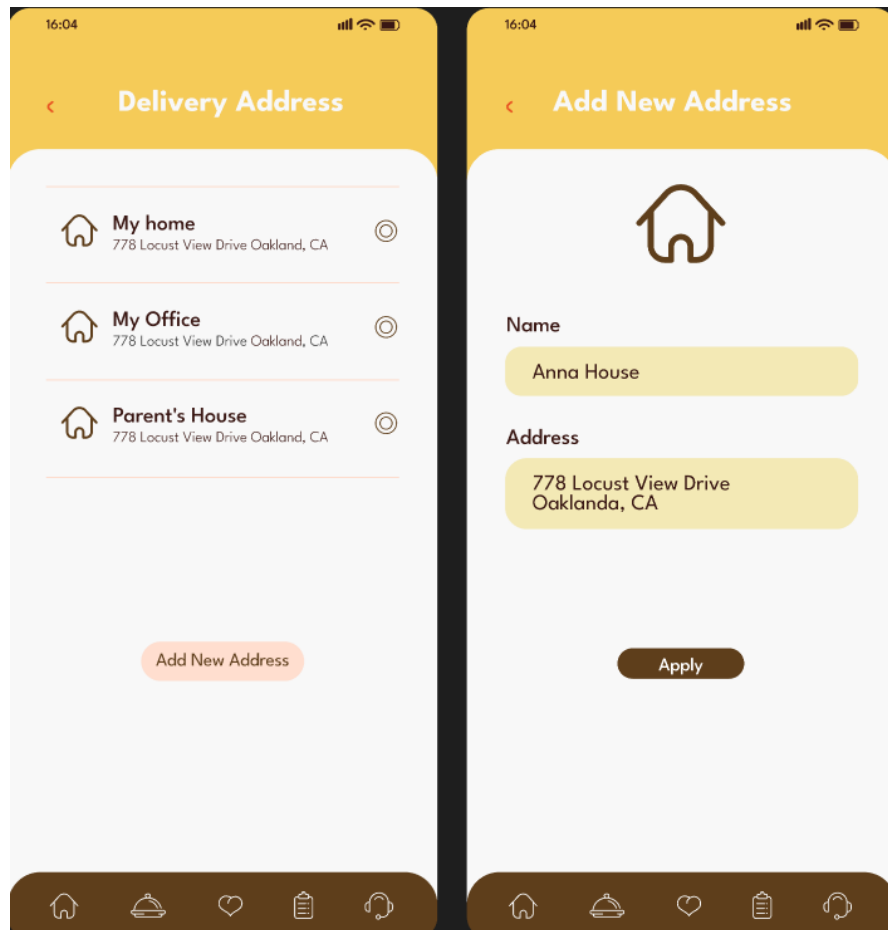


Figura 25. Pantalla Direcciones

- Pantalla Métodos de Pago

Esta pantalla será una pantalla informativa en donde se muestre la información de pago del restaurante.

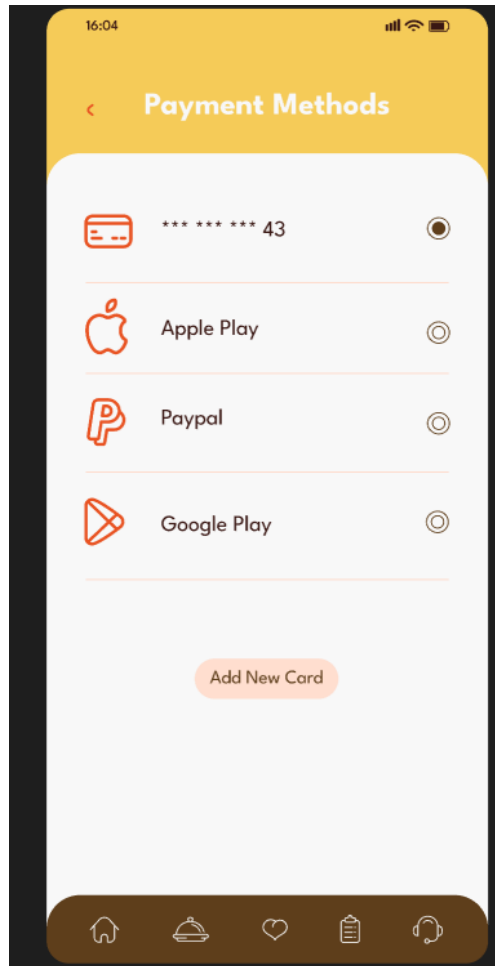


Figura 26. Métodos de Pago

- Pantalla Contáctanos

Esta pantalla será informativa en donde contenga la información de contacto del restaurante



Figura 27. Pantalla Contáctanos

- Pantalla Ajustes

Esta pantalla por requerimientos de la empresa quien es la que genera los mackups argumenta que será estática sin funcionalidad puesto que tienen nuevos planes para esa pantalla.

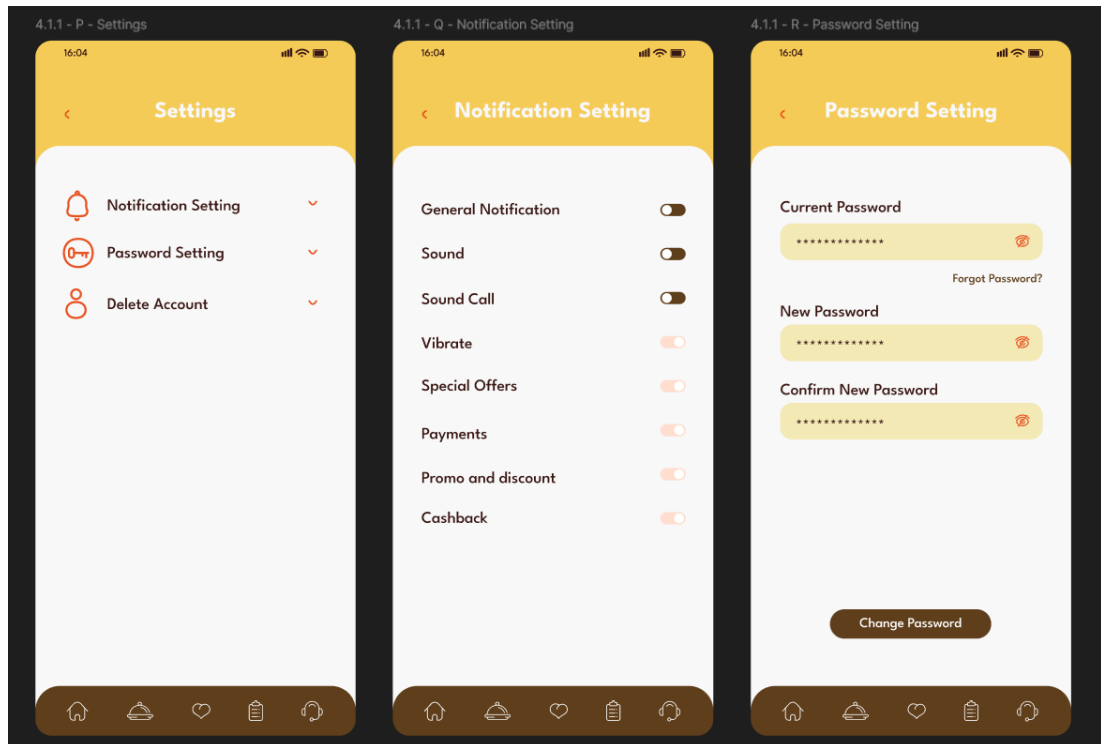


Figura 28. Pantallas de Configuración

- Modal de cerrar sesión

En este modal se opaca toda la interfaz dando más relevancia a la pregunta de salir o no sesión.

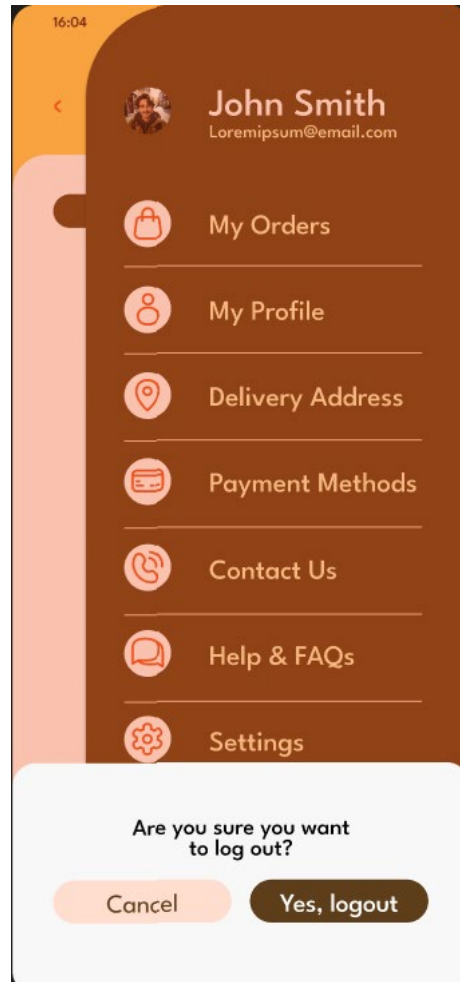


Figura 29. Modal de Cerrar Sesión

- Pantalla Realizar Pedido

Una vez seleccionado en la página principal la sucursal se visualiza la pantalla en este caso nos permitirá seleccionar la variación que queramos como sea en la selección de sopa, plato fuerte y el método de envío.

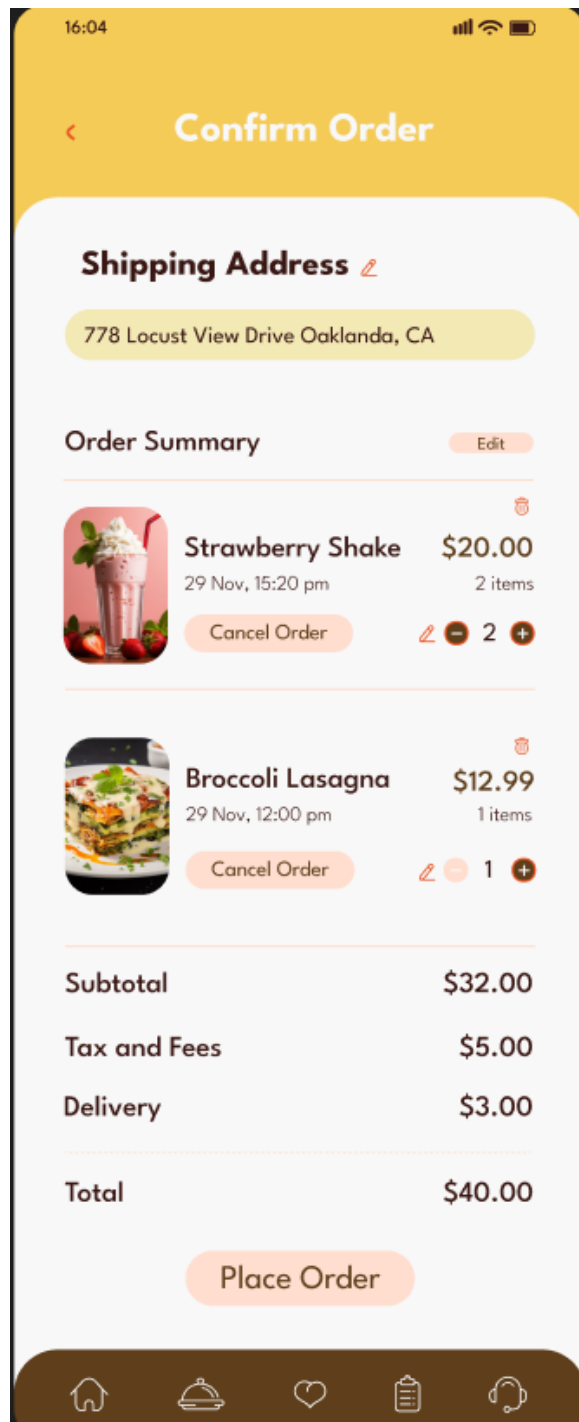


Figura 30. Pantalla Realizar Pedidos

- Pantalla desplegable del carrito de compras

En esta pantalla podremos ver el carrito de compras con el menú que se seleccionó si no tiene el menú en el carrito le pedirá realizar una nueva orden.

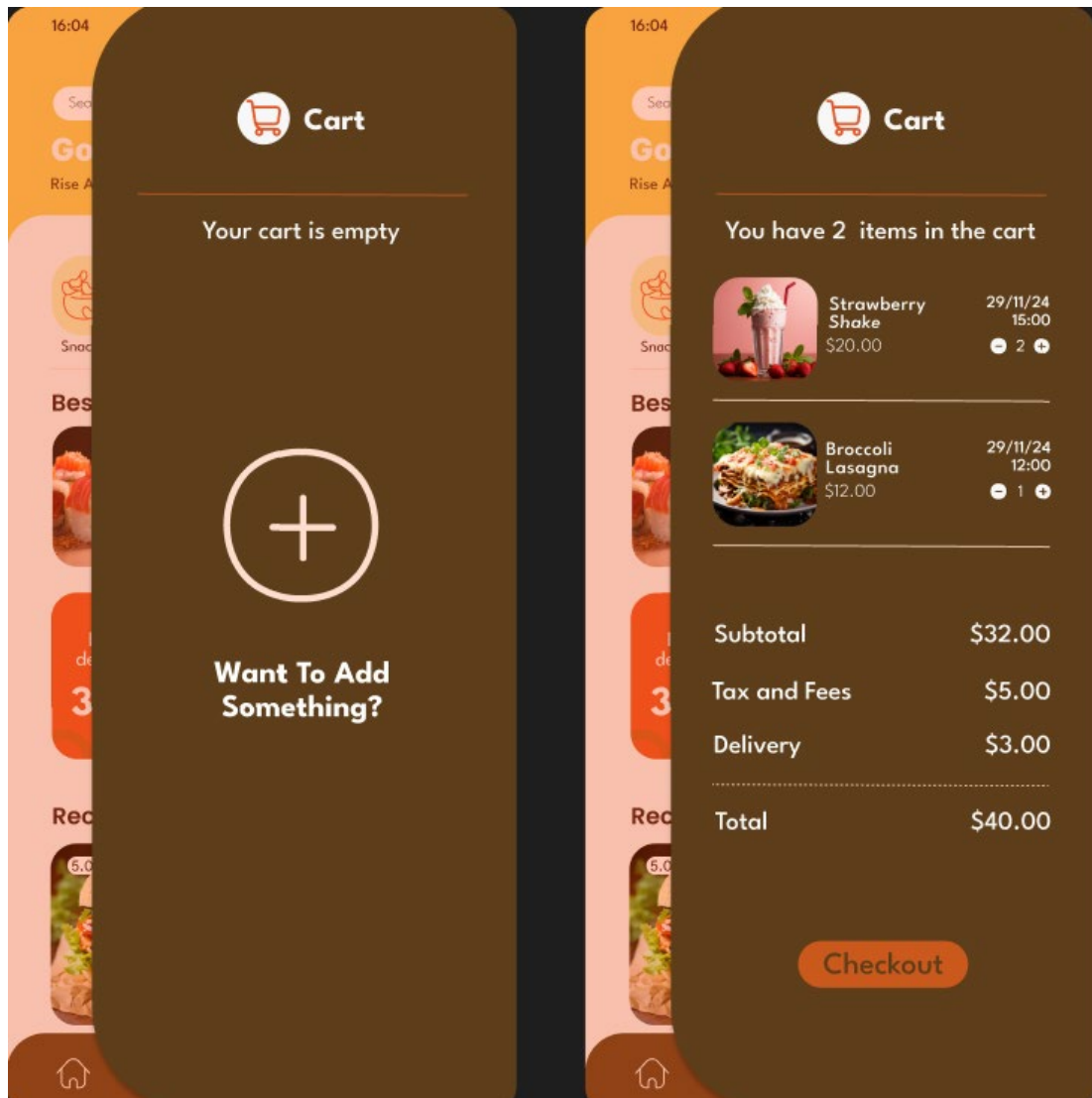


Figura 31. Pantalla desplegable del carrito de compras

- Pantalla Pedido Confirmado

Esta pantalla es una pantalla de confirmación que se realizó correctamente

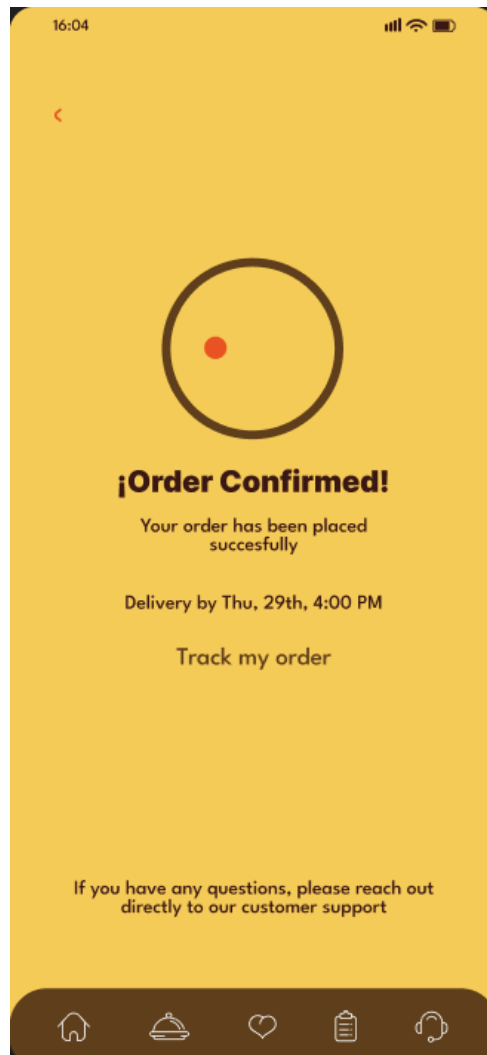


Figura 32. Pantalla Orden Confirmada

3.2.3 Fase 3: Producción

En esta fase se procedió con el desarrollo de la aplicación móvil tomando en cuenta los requisitos e interacciones que se mencionan en la fase 1 y 2.

a. Backend

Al crear un backend integrando un gestor de bases de datos como es mysql y un framework de desarrollo de backend como es Node.js nos permite poder desplegar las

apis en un servidor y obtener la información en cualquier momento. La empresa Connectaservices consta con un servidor en la nube en donde se podrá subir el backend y consumir desde el dominio que este servidor nos devuelva.

React Native como framework de desarrollo de aplicaciones usa una arquitectura “game changer” ya que es una arquitectura flexible, y visiblemente ordenada enfocada en el desarrollo de módulos.

- Módulos

En la visualización de las pantallas se presenta mediante módulos de desarrollo donde se evidencia la pantalla a intención. A continuación, se presenta una lista de los módulos de la aplicación móvil.

- RegisterScree.tsx
- ProfileScreen.tsx
- OrdersScreen.tsx
- LoginScreen.tsx
- initScreen.tsx
- HomeScreen.tsx
- ChangePasswordScreen.tsx
- ContactScreen.tsx
- LoginScreenDelivery.tsx
- OrderScreenDelivery.tsx

- Dependencias instalas en la aplicación móvil

Las dependencias utilizadas el proyecto nos permiten la implementación de funcionalidades de forma rápida y robusta. En el **Anexo C** se puede observar todas las dependencias utilizadas para el desarrollo de la aplicación, así garantizando una aplicación robusta y efectiva.

- Storycards

Es una herramienta utilizada para describir y organizar las funcionalidades de una aplicación móvil de manera ágil y visual.

a. Storycard del Usuario

- Storycard Inicio de sesión

Tabla 18. Storycard 01 Inicio de Sesión

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
01	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Pantalla de Inicio de sesión						
Fecha		Estado		Comentario		
13/05/2024		Definido		Sin comentarios		
16/05/2024		Implementado		Sin comentarios		
16/05/2024		Ejecutado		Sin comentarios		
16/05/2024		verificado		Sin comentarios		

En el siguiente **Anexo D** se muestra el código de cómo se implementó las tecnologías para el Inicio de sesión, utilizando los diferentes componentes propuestos tanto para iOS como para Android.

- Storycard Registro usuario

Tabla 19. Storycard 02 Registro de usuario

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
01	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Pantalla de Registro usuario						
Fecha		Estado		Comentario		

13/05/2024	Definido	Sin comentarios
17/05/2024	Implementado	Sin comentarios
17/05/2024	Ejecutado	Sin comentarios
17/05/2024	verificado	Sin comentarios

En el **Anexo E** se muestra el código de implementación en el cual se procede a tipiar los datos que se ingresan para poder generar un arreglo en el cual estará guardado en un objeto para después utilizarlo en la llamada del procedimiento para ingresar la información del nuevo cliente en la tabla clientes.

- StoryCard Pantalla de Inicio

Tabla 20. Storycard 03 Pantalla de Inicio

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
02	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Pantalla de Inicio						
Fecha		Estado		Comentario		
20/05/2024		Definido		Sin comentarios		
21/05/2024		Implementado		Sin comentarios		
21/05/2024		Ejecutado		Sin comentarios		
21/05/2024		verificado		Sin comentarios		

En el siguiente anexo **Anexo F** se puede observar la implementación del código para crear la página en el que el usuario podrá poder visualizar las diferentes sucursales así también como se puede observar las diferentes acciones que el usuario puede realizar para la navegación del usuario.

- StoryCard interfaz Nuevo Pedido

Tabla 21. Storycard 04 interfaz Nuevo Pedido

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
01	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Pantalla de Registro usuario						
Fecha		Estado		Comentario		
24/05/2024		Definido		Sin comentarios		
26/05/2024		Implementado		Sin comentarios		
26/05/2024		Ejecutado		Sin comentarios		
26/05/2024		verificado		Sin comentarios		

En la implementación para poder ingresar un nuevo pedido se necesitó realizar un modal en el que permitirá obtener los datos propuestos por el sistema de restaurantes previamente de esta manera garantiza el consumo y la comunicación del aplicativo móvil con el sistema de restaurantes de la empresa Connectaservices, así como se puede visualizar en el **Anexo G**

- StoryCard Sincronizar los pedidos mediante woocommerce

Tabla 22. Storycard 04 Sincronizacion Woocommerce

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
01	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Sincronizar los pedidos mediante woocommerce						
Fecha		Estado		Comentario		
26/05/2024		Definido		Sin comentarios		
28/05/2024		Implementado		Sin comentarios		
28/05/2024		Ejecutado		Sin comentarios		
28/05/2024		verificado		Sin comentarios		

Para poder sincronizar los pedidos que se ingresan al servicio de woocommerce en el cual está conectado el sistema de restaurantes de la empresa Connectaservices se necesitó hacer un consumo del a api que provenía de la aplicación web del restaurante ya puesto en curso para que mediante una conexión ya anteriormente proporcionada se gestione mediante el sistema de restaurantes de la empresa, de esa manera visualizándose en el **Anexo H**.

- StoryCard Desarrollo de la interfaz inicio sesión para repartidor

Tabla 23. Storycard 05 interfaz inicio sesión para repartidor

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
01	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Iniciar sesión para repartidor						
Fecha		Estado		Comentario		
29/05/2024		Definido		Sin comentarios		
31/05/2024		Implementado		Sin comentarios		
31/05/2024		Ejecutado		Sin comentarios		
31/05/2024		verificado		Sin comentarios		

En el código presentado por el **Anexo I** se puede visualizar como se implementó mediante la misma lógica del usuario el inicio de sesión del repartidor.

- Storycard desarrollo de la interfaz de seguimiento al delivery

Tabla 24. Storycard 06 Interfaz de seguimiento al delivery

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
01	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Interfaz de seguimiento al delivery						
Fecha		Estado		Comentario		
2/05/2024		Definido		Sin comentarios		
6/05/2024		Implementado		Sin comentarios		
6/05/2024		Ejecutado		Sin comentarios		
6/05/2024		verificado		Sin comentarios		

A continuación, se muestra mediante el **Anexo J** la implementación de la interfaz para que el repartidor pueda realizar el seguimiento mediante los estados de los pedidos en un carrusel de pedidos.

- Storycard implementación de notificaciones para la creación, preparación y despacho de pedidos

Tabla 25. Storycard 07 notificaciones delivery

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
01	Nuevo	Moderado	Moderado	3	5	Alta
Descripción: Notificaciones delivery						
Fecha		Estado		Comentario		
7/06/2024		Definido		Sin comentarios		
13/06/2024		Implementado		Sin comentarios		
13/06/2024		Ejecutado		Sin comentarios		
13/06/2024		verificado		Sin comentarios		

En el siguiente **Anexo K** se puede visualizar la implementación de las notificaciones utilizando la clase propia de expo para la comunicación mediante el repartidor y también el usuario que realiza el pedido.

3.2.4 Fase 4: Estabilización

Cambios y actualizaciones que se realizaron en la aplicación en el proceso de desarrollo.

- **Cambios en base de datos**

Se implemento nuevos campos en la tabla clientes para poder tener más información.

- **Cambios en diseño de la aplicación**

Se modificó la pantalla de solicitar pedido a una más interactiva y rápida visualmente para el cliente, se modificó la pantalla de métodos de pago, configuraciones a vistas informativas como requerimiento de la empresa.

3.2.5 Fase 5: Pruebas

En la fase 5 de la metodología Mobile-D se realizó las pruebas que corroboren la funcionalidad de la aplicación, considerando las necesidades de la empresa, de esta manera entregando una versión estable y funcional del aplicativo móvil donde brinda al restaurante una mejor gestión de pedidos.

a. Pruebas de funcionalidad

- **Inicio de Sesión**

En esta pantalla el usuario podrá ingresar sesión mediante su correo y contraseña y utilizar las funcionalidades de la aplicación.

También si el usuario es un usuario repartidor podrá ingresar dando tap en el botón repartidor

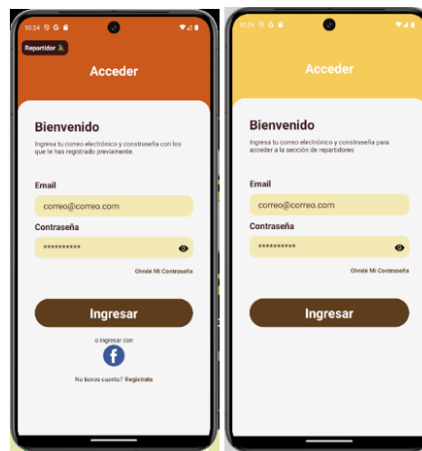


Figura 33. Iniciar sesión cliente, repartidor

Registrar Usuario

En esta pantalla el usuario podrá registrar su cuenta ingresando correctamente lo que le pide la aplicación.

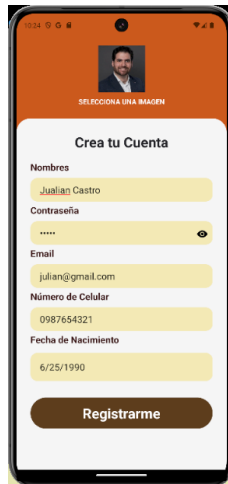


Figura 34. Registrar Usuario

- **Pantalla de inicio**

En esta pantalla se podrá visualizar las sucursales que el restaurante posee, así también como poder realizar el pedido del menú con respecto al restaurante que el usuario quiera realizar.



Figura 35. Pantalla de Inicio

- **Pantalla actualizar perfil**

En esta pantalla el usuario podrá actualizar la información del perfil.

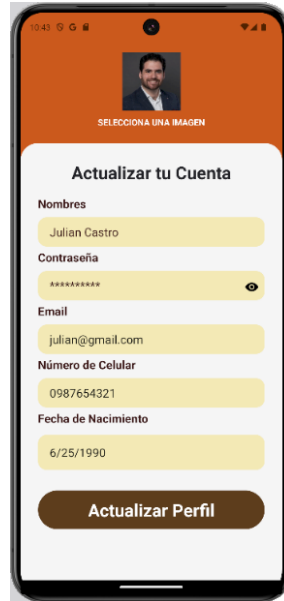


Figura 36. Editar Usuario

- **Realizar pedido**

Para realizar el pedido el usuario debe seleccionar un menú y se visualizará un modal que le permitirá generar el pedido

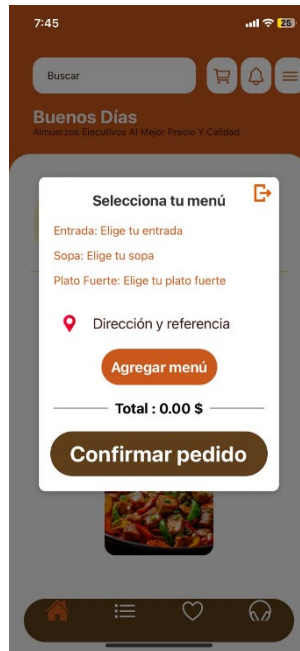


Figura 37. Realizar Pedido

- **Pantalla de Contacto**

En esta pantalla se podrá visualizar la información que cada sucursal tiene

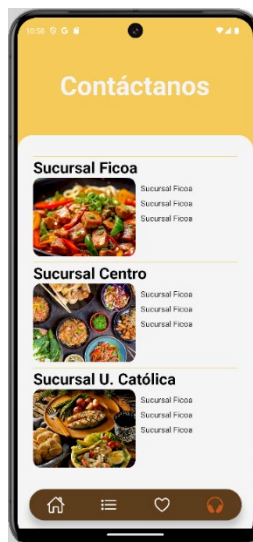


Figura 38. Pantalla Contacto

- **Pantalla Mis Ordenes**

En esta pantalla se podrá ver los pedidos realizados enlistados con su información correspondiente

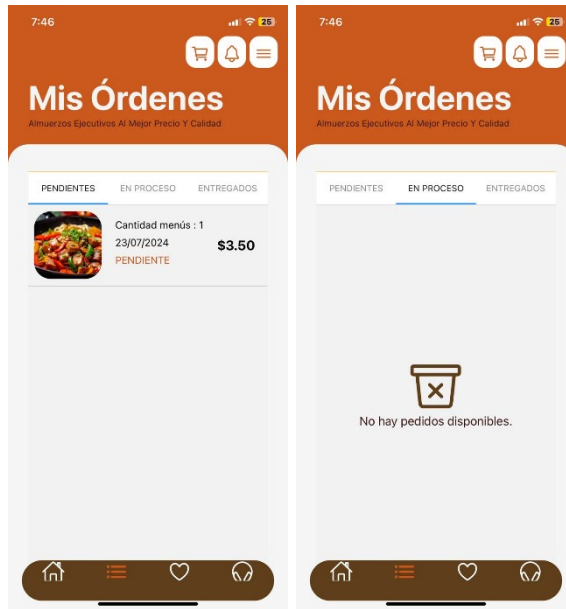


Figura 39. Mis Órdenes

- **Pantalla Seguimiento de Pedidos**

En esta pantalla el repartidor podrá visualizar los pedidos ordenados por estado en un carrusel de estados como también poder actualizar el estado con la finalidad de dar seguimiento al pedido que se requiera realizar.

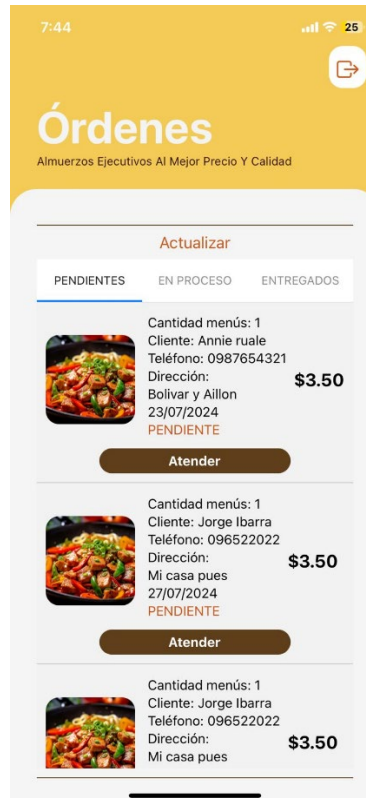


Figura 40. Pantalla de seguimiento de pedidos

b. Pruebas de aceptación

Tabla 26. Prueba de aceptación 1

Prueba de aceptación	Número: 1
N° de storycard: 1	
Nombre: Inicio de sesión	
Descripción: Los usuarios pueden ingresar a la aplicación siempre y cuando exista una cuenta ya creada.	
Condiciones de ejecución de interfaz: En la interfaz, el usuario deberá ingresar su correo y contraseña y deberá presionar el botón de ingresar en el cual el aplicativo se encargará de poder validar las credenciales ingresadas.	
Resultado esperado: Una vez ingresado el correo y la contraseña correctamente la aplicación le autoriza el pase a la pantalla principal	
Evaluación de prueba: Satisfactoria	

Tabla 27. Prueba de aceptación 2

Prueba de aceptación	Número: 2
N° de storycard: 2	
Nombre: Registro de usuario	
Descripción: Los usuarios pueden registrar un nuevo usuario para el ingreso de la aplicación	
Condiciones de ejecución de interfaz: En la interfaz, el usuario deberá presionar el texto registrarse, en donde le direccionará a la pantalla de ingreso al nuevo usuario.	
Resultado esperado: Una vez ingresado el correo y la contraseña correctamente la aplicación le autoriza el pase a la pantalla principal	
Evaluación de prueba: Satisfactoria	

Tabla 28. Prueba de aceptación 3

Prueba de aceptación	Número: 3
N° de storycard: 3	
Nombre: Pantalla de Inicio	
Descripción: Los usuarios pueden navegar entre diferentes pantallas	
Condiciones de ejecución de interfaz: En la interfaz, el usuario podrá visualizar una interfaz resumida entre las sucursales que el restaurante gourmendy posee, también el usuario puede navegar entre ventanas y diferentes opciones	
Resultado esperado: La página principal le permite navegar entre las diferentes pantallas y realizar un pedido	
Evaluación de prueba: Satisfactoria	

Tabla 29. Prueba de aceptación 4

Prueba de aceptación	Número: 4
N° de storycard: 4	
Nombre: Nuevo Pedido	
Descripción: Los usuarios pueden realizar un pedido	
Condiciones de ejecución de interfaz: En la interfaz, de la pantalla de inicio el usuario podrá escoger la sucursal de dónde quiera hacer el pedido, presionando en cualquiera de las imágenes, se abrirá un modal dónde podrás seleccionar el menú que prefiera el usuario	
Resultado esperado: Una vez ingresado el menú que quiera servirse se guardará en la tabla pedidos	
Evaluación de prueba: Satisfactoria	

Tabla 30. Prueba de aceptación 5

Prueba de aceptación	Número: 5
N° de storycard: 5	
Nombre: Sincronización Woocommerce	
Descripción: El pedido se integra con el sistema woocommerce de la empresa	
Condiciones de ejecución de interfaz: Después de haber generado el pedido se debe sincronizar en el servicio woocommerce de la empresa	
Resultado esperado: el pedido se sincronizó efectivamente en el sistema woocommerce	
Evaluación de prueba: Satisfactoria	

Tabla 31. Prueba de aceptación 6

Prueba de aceptación	Número: 6
Nº de storycard: 6	
Nombre: Inicio sesión para repartidor	
Descripción: La persona que sea repartidor podrá ingresar al aplicativo con sus credenciales creadas anteriormente	
Condiciones de ejecución de interfaz: En la pantalla de inicio de sesión, si el usuario es un repartidor podrá presionar el botón repartidor el cual navegará a la interfaz de inicio de sesión del repartidor el cual el repartidor deberá ingresar correctamente sus credenciales de acceso, como correo y contraseña.	
Resultado esperado: El usuario repartidor podrá ingresar a su interfaz de seguimiento de delivery	
Evaluación de prueba: Satisfactoria	

Tabla 32. Prueba de aceptación 7

Prueba de aceptación	Número: 7
Nº de storycard: 7	
Nombre: Seguimiento al delivery	
Descripción: La persona que sea repartidor editar el estado del delivery	
Condiciones de ejecución de interfaz: En la pantalla principal del repartidor se podrán ver todos los pedidos realizados el cual el repartidor tendrá que seleccionar un pedido y actualizar el estado.	
Resultado esperado: el repartidor podrá dar seguimiento a los pedidos que la cajera administradora del restaurante le asigne.	
Evaluación de prueba: Satisfactoria	

Tabla 33. Prueba de aceptación 8

Prueba de aceptación	Número: 8
Nº de storycard: 8	
Nombre: Notificaciones delivery	
Descripción: El usuario que realiza el pedido deberá llegarle una notificación acerca del estado del pedido	
Condiciones de ejecución de interfaz: en el momento que el usuario repartidor actualiza el estado del pedido, deberá llegarle una notificación al cliente informando el estado del pedido	
Resultado esperado: el cliente podrá visualizar el estado del pedido.	
Evaluación de prueba: Satisfactoria	

3.2.6 Integración Aplicación móvil

Se integró la aplicación móvil mediante la comunicación del backend y las apis de woocommerce, enviando toda la información del pedido generado desde el aplicativo móvil hacia el api de woocommerce, ya que al generar un pedido desde woocommerce el sistema de restaurantes de la empresa Connectaservices realiza peticiones cada 5 segundos para obtener la información de nuevos pedidos. En la **Figura 41** se observa la integración y la información que el aplicativo móvil genera hacia el servicio de woocommerce.

Detalles de Pedido #1716

Pago a través de Transferencia bancaria. Pagado el 18 de junio de 2024 @ 00:47

General

Fecha de creación:

2024-06-18

@

00

:

47

Estado:

Procesando

Cliente:

Invitado

Facturación

Pako Ruales
Ambato
Ambato
Tungurahua
Ecuador

Dirección de correo electrónico:
john.doe@example.com

Teléfono:
[\(593\) 555-5555](tel:(593)555-5555)

Envío

Dirección:
Sin dirección de envío configurada.

Artículo	Coste	Cantidad	Total
 MENÚ COMPLETO FICOA ID de la variación: 1148 PARA LLEVAR: DESECHABLES ENTRADA: EMPANADA DE QUESO SOPA: CONSOMÉ PLATO FUERTE: LASAÑA DE CARNE PARA LLEVAR: SERVIRSE EN EL LOCAL	\$3.50	× 1	\$3.50
Subtotal de artículos:			\$3.50
Total del pedido:			\$3.50
Pagado:			\$3.50

18 de junio de 2024 a través de Direct Bank Transfer

Figura 41. Integración pedidos

Además, se creó un repositorio en GitHub con el fin de acceder a la aplicación móvil mediante el siguiente enlace: https://github.com/pakoruales27/tesis_app.git con el objetivo que la empresa pueda generar nuevos cambios y actualizaciones que lleven a poder subirlas en las tiendas de app store y play store.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La implementación de una aplicación móvil híbrida utilizando React Native y Node.js para la gestión y seguimiento de delivery ofrece una notable mejora en la eficiencia operativa de los restaurantes conectados a ConnectaServices.
- Al analizar los inconvenientes en la toma de pedidos y el seguimiento de entregas que enfrenta Connectaservices, se pudo diseñar una solución efectiva utilizando tecnologías modernas como React Native y Nodejs permitiendo mejorar la eficiencia operativa y la experiencia de usuario.
- Al integrar la aplicación móvil en el sistema de restaurantes de Connectaservices no solo se pudo mejorar la gestión de pedidos, sino que también facilitó una comunicación más efectiva entre el cliente y el restaurante. Esto permite que haya una mayor satisfacción del cliente y tener una operación más efectiva en el servicio del restaurante.
- La utilización de tecnologías modernas como React Native y Node.js demostró ser clave para la transformación digital efectiva de los procesos de delivery en el sector restaurante al proporcionar una solución escalable y adaptable que puede crecer con las necesidades futuras de Connectaservices.

4.2 Recomendaciones

- Se recomienda la implementación de un personal exclusivo de reparto de pedidos en el restaurante gourmendy ya que permitiría la mejora de procesos y la integración de nuevas funciones promoviendo la escalabilidad en el aplicativo móvil.
- Se recomienda programar actualizaciones para agregar nuevas características al aplicativo móvil como también brindar soporte técnico y realizar un monitoreo en tiempo real de los procesos que realiza el aplicativo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Leonel Fabricio Escobar Montenegro, “APLICACIÓN MÓVIL MULTIPLATAFORMA PARA LA GESTIÓN DE VENTAS EN TIEMPO DE PANDEMIA EN LOS SUPERMERCADOS DE AMBATO”, Ambato, mar. 2022. Consultado: el 15 de noviembre de 2023. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/34831/1/t1966si.pdf>
- [2] O. L. Roa, “Ingeniería de Software”, 2019.
- [3] J. Famula, D. E. Pittman, y K. S. Haring, “Building Trust with a Mobile Application for Last-Mile Commercial Drone Delivery”, en *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 462–467. doi: 10.1109/ICUAS54217.2022.9836198.
- [4] D. J. Poza García Fernando Acebes Senovilla, “Retos para la Transformación Digital de las PYMES: Competencia Organizacional para la Transformación Digital”. Consultado: el 18 de octubre de 2023. [En línea]. Disponible en: <https://uvadoc.uva.es/bitstream/handle/10324/47767/Tesis1874-210729.pdf?sequence=1>
- [5] A. Luthfia, S. Ramadanty, T. L. Anita, A. Zulkarnain, y A. R. Ridzuan, “The Effect of Digital Skills on Technology Acceptance of Food Delivery Mobile Application”, en *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, 2021, pp. 1–5. doi: 10.1109/ICORIS52787.2021.9649587.
- [6] B. Suri, S. Taneja, I. Bhanot, H. Sharma, y A. Raj, “Cross-Platform Empirical Analysis of Mobile Application Development Frameworks: Kotlin, React Native and Flutter”, en *Proceedings of the 4th International Conference on Information Management & Machine Intelligence*, en ICIMMI '22. New York, NY, USA: Association for Computing Machinery, 2023. doi: 10.1145/3590837.3590897.
- [7] G. Xervaxius Naphan, A. Setyawan, J. Edmund Friry, y R. Nugroho Perwiro Atmojo, “The Impact of Covid-19 Pandemic on Online Delivery Application User”, en *Proceedings of the 2021 International Conference on Computer, Control, Informatics and Its Applications*, en IC3INA '21. New York, NY, USA: Association for Computing Machinery, 2022, pp. 194–199. doi: 10.1145/3489088.3489108.
- [8] Ayala Evelyn - Gonzales Santiago, “T E C N O L O G Í A S D E L A I N F O R M A C I Ó N Y L A C O M U N I C A C I Ó N”. [En línea]. Disponible en: www.uigv.edu.pe
- [9] R. N. L. C. Lobato Tapia, “React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma”, *Revista Digital Universitaria*, vol. 20, núm. 5, ago. 2019, doi: 10.22201/codeic.16076079e.2019.v20n5.a5.
- [10] J. A. Polanco Gonzales y E. Y. Quispe Andrade, “Aplicación móvil multiplataforma utilizando la metodología Mobile-D para mejorar la gestión de pedidos en la Empresa D’fruto S.A.C. - Trujillo”.
- [11] I. E. Javier y S. Freire, “Análisis del comportamiento del consumidor en tiempos de Covid en el sector de alimentos de la ciudad de Ambato”, AMBATO, feb. 2022.

Consultado: el 18 de octubre de 2023. [En línea]. Disponible en:
<https://repositorio.uta.edu.ec/bitstream/123456789/34275/1/823%20MKT.pdf>

- [12] Sangama Oñate Abel Fernando, “Metodologías ágiles Scrum, XP, SLeSS, Scrumban, HME, Mobile-D y MASAN empleadas en la industria de dispositivos móviles: Un contraste en favor de la industria del desarrollo móvil.”, 2020.

ANEXOS

Anexo A. Encuesta

Cuestionario para encuesta en la Implementación de una Aplicación Móvil Híbrida de Delivery

Objetivo: *Determinar los inconvenientes que enfrenta la empresa Connectaservices en la toma de pedidos y el seguimiento de entregas a domicilio.*

Población: *personal encargado de la recepción de pedidos, encargadas de realizar los pedidos y Personas encargadas del motorizado.*

1. ¿Cuáles de los siguientes medios es la más escogida para realizar pedidos a domicilio?
 - a. Aplicación móvil
 - b. Redes sociales
 - c. Mensajería
 - d. Llamada
 - e. Sitio Web
2. ¿Qué tan seguido los clientes realizan pedidos en línea?
 - a. Diariamente
 - b. Varias veces por semana
 - c. Una o tres veces por semana
 - d. Una vez al mes
 - e. Menos de una vez al mes
3. ¿Cuál es el grado de satisfacción al momento de realizar un pedido con el sistema actual?
 - a. Muy bueno
 - b. Bueno
 - c. Regular
 - d. Malo
 - e. Muy malo
4. ¿Conoce el proceso de recepción de pedidos?
 - a. Si
 - b. No
5. ¿qué tan preciso es el seguimiento de los pedidos en curso con el sistema actual?
 - a. Muy Preciso
 - b. Preciso
 - c. Poco preciso
 - d. impreciso
 - e. Muy impreciso
6. ¿Considera que el proceso actual de seguimiento de entregas a domicilio es eficiente?
 - a. Si

b. No

7. ¿Estaría de acuerdo en utilizar una plataforma digital a fin de superar las dificultades en la toma de pedidos?
 - a. Sí
 - b. No
8. ¿está de acuerdo que la falta de herramientas digitales para coordinar horarios de entrega es una causa importante de las dificultades recurrentes en la satisfacción de los clientes?
 - a. Si
 - b. No
9. ¿Ha experimentado errores o confusiones en la comunicación de información a los clientes sobre el estado de sus pedidos?
 - a. Si
 - b. No
10. ¿Qué tan satisfactorio es el proceso de gestión de pedidos del cliente mediante un aplicativo móvil?
 - a. Muy insatisfactorio
 - b. Insatisfactorio
 - c. Poco satisfactorio
 - d. Satisfactorio
 - e. Muy satisfactorio

Anexo B. Entrevista

Entrevista para la Implementación de una Aplicación Móvil Híbrida de Delivery

Objetivo: Determinar los inconvenientes que enfrenta la empresa Connectaservices en la toma de pedidos y el seguimiento de entregas a domicilio.

Población: Gerente de la empresa connectaservices

Datos personales			
Apellido y Nombre entrevistado:			
Institución:			
Cargo:			
Objetivo:			
N°	Preguntas	Respuesta	Observaciones
1	¿Cómo se lleva el registro de pedidos a domicilio?		
2	¿Cuáles son los inconvenientes que presentan los empleados del restaurante al receptar los pedidos?		
3	¿Cuáles son las quejas del personal y de los clientes que recibe al momento de receptar los pedidos a domicilio?		
4	¿Cómo describiría la eficiencia del registro de los pedidos en curso en la empresa?		
5	¿Qué dificultades ha experimentado al comunicar información sobre el estado de los pedidos a los clientes?		
6	¿Considera que el seguimiento de entregas a domicilio en la empresa es lo suficientemente eficiente? ¿Por qué sí o por qué no?		
7	¿En qué medida cree que la ausencia de un sistema de delivery afecta la capacidad de Connectaservices para gestionar eficazmente los pedidos y entregas?		

8	¿Cómo describiría la precisión de los registros de pedidos en comparación con las expectativas de los clientes?		
9	¿Qué problemas adicionales ha experimentado al gestionar la disponibilidad de productos y el seguimiento de entrega sin una solución digitalizada?		
10	¿En qué medida afecta a la empresa no contar con una óptima gestión de delivery?		
Conclusión:			

Anexo C. Dependencias descargadas

```
"dependencies": {
  "@react-native-async-storage/async-storage": "^1.23.1",
  "@react-native-community/datetimepicker": "8.0.1",
  "@react-native-picker/picker": "2.7.5",
  "@react-navigation/bottom-tabs": "^6.5.20",
  "@react-navigation/drawer": "^6.6.15",
  "@react-navigation/material-top-tabs": "^6.6.13",
  "@react-navigation/native": "^6.1.17",
  "@react-navigation/native-stack": "^6.9.26",
  "@react-navigation/stack": "^6.3.29",
  "axios": "^1.7.2",
  "date-fns": "^3.6.0",
  "expo": "~51.0.22",
  "expo-auth-session": "~5.5.2",
  "expo-crypto": "~13.0.2",
  "expo-dev-client": "~4.0.20",
  "expo-device": "~6.0.2",
  "expo-image-picker": "~15.0.6",
  "expo-notifications": "~0.28.9",
  "expo-random": "~14.0.1",
  "expo-status-bar": "~1.12.1",
  "expo-tracking-transparency": "~4.0.2",
  "expo-updates": "~0.25.21",
  "expo-web-browser": "~13.0.3",
  "firebase": "^10.12.2",
  "google-auth-library": "^9.10.0",
```

```

    "i": "^0.3.7",
    "jwt-decode": "^4.0.0",
    "react": "18.2.0",
    "react-native": "0.74.3",
    "react-native-fbsdk-next": "^13.0.0",
    "react-native-gesture-handler": "~2.16.1",
    "react-native-pager-view": "6.3.0",
    "react-native-reanimated": "~3.10.1",
    "react-native-safe-area-context": "4.10.5",
    "react-native-screens": "3.31.1",
    "react-native-tab-view": "^3.5.2",
    "socket.io-client": "^4.7.5",
    "expo-system-ui": "~3.0.7"
  },

```

Anexo D. Código Implementación de Login

```

export const LoginScreen = () => {
  const navigation =
useNavigation<StackNavigationProp<RootStackParamList>>();
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const dismissKeyboard = () => {
    Keyboard.dismiss();
  };

  // Metodo de logeo
  const handleLogin = async () => {
    try {
      const resUser = await loginUser(email, password);
      // console.log('DATA LOGIN:', resUser);
      // showLoginMessage(email, password);
      console.log(resUser.success);
      if (resUser.success) {
        Alert.alert("Bienvenido", resUser.message);
        navigation.navigate('MyDrawer'); // Navegar a la pantalla de
inicio
      } else {
        Alert.alert("No se pudo iniciar sesión", resUser.message);
      }
    } catch (error) {
      console.error('Error logging in:', error);
      // Muestra un mensaje de error al usuario
      if (Platform.OS === "android") {
        ToastAndroid.show("Error al iniciar sesión",
ToastAndroid.LONG);

```

```

    } else {
      console.log("Error al iniciar sesión");
    }
  }
};

return (
  <KeyboardAvoidingView
    behavior={Platform.OS === "ios" ? "padding" : "height"}
    style={styles.container}
  >
    <StatusBar style="light" />
    <View style={styles.container}>

      <View style={{marginStart: 10, backgroundColor:
textColors.secondary, borderRadius: 10,
        marginTop: 55, position: "absolute",}}>
        <TouchableOpacity onPress={() =>
navigation.navigate('LoginScreenDelivery')}>
          <Text style={styles.repartidortext}>Repartidor </Text>
        </TouchableOpacity>

      </View>

      <View style={styles.logoContenedor}>
        <Text style={styles.logotext}>Acceder</Text>
      </View>

      <TouchableWithoutFeedback onPress={dismissKeyboard}>
        <View style={styles.form}>
          <Text style={styles.formText}>Bienvenido</Text>
          <Text style={styles.formText2}>Ingresa tu correo
electrónico y contraseña con los que te has registrado
previamente.</Text>

          <View style={styles.spaceViewInput}></View>
          <Text style={styles.formText3}>Email</Text>
          <TextInputComponent
icon={require("../assets/icons/usuario.png")}
title="correo@correo.com" secureTextEntry={false} onChangeText={(text)
=> setEmail(text)} />
          <View style={styles.spaceView}></View>
          <Text style={styles.formText3}>Contraseña</Text>

```

```

        <PasswdInputComponent
icon={require("../assets/icons/password.png")} title=" *****"
secureTextEntry={true} onChangeText={(text) => setpassword(text)} />
        <View style={styles.registrarform}>
            <TouchableOpacity onPress={() =>
navigation.navigate('ChangePassword')}>
                <Text style={styles.registrartext}>Olvidé Mi
Contraseña</Text>
            </TouchableOpacity>
        </View>
        <View style={styles.spaceViewImput}></View>
        <View>
            <MyButton2 title="Ingresar" onPress={handleLogin} />
            { /* <MyButton2 title="Ingresar" onPress={() =>
navigation.navigate('HomeScreen')} /> */}
        </View>
        <View style={styles.ingresarConform}>
            <Text>o ingresar con</Text>
        </View>
        <View style={styles.ingresarConform}>

            <Image
                style={styles.logoImage}
                source={require("../assets/icons/facebook.png")}
            ></Image>
        </View>
        <View style={styles.ingresarConform}>
            <Text>No tienes cuenta?</Text>
            <TouchableOpacity onPress={() =>
navigation.navigate('Register')}>
                <Text style={styles.registrartext}>Registrate</Text>
            </TouchableOpacity>
        </View>
        </View>
    </TouchableWithoutFeedback>
</View>
</KeyboardAvoidingView>
);
};

```

Anexo E. Código de implementación para el registro de un nuevo usuario

```

export const Register = () => {
    const navigation =
useNavigation<StackNavigationProp<RootStackParamList>>();
    const [date, setDate] = useState(new Date());
    const [showDatePicker, setShowDatePicker] = useState(false);

```

```

    const [formattedDate, setFormattedDate] =
useState(date.toLocaleDateString());
    const [loading, setLoading] = useState(false);
    const [file, setFile] = useState<ImagePicker.ImagePickerAsset>();
    const [dataUser, setDataUser] = useState<User>({
        id: 0,
        email: '',
        nacimiento: new Date(),
        nombres: '',
        telefono: '',
        clave: '',
        image: ''
    });
    const [modalVisible, setModalVisible] = useState(false);

    const pickImage = async () => {
        let result = await ImagePicker.launchImageLibraryAsync({
            mediaTypes: ImagePicker.MediaTypeOptions.All,
            allowsEditing: true,
            quality: 1
        });

        if (!result.canceled) {
            onChange('image', result.assets[0].uri);
            setFile(result.assets[0]);
        }
    };

    const takePhoto = async () => {
        const { status } = await
ImagePicker.requestCameraPermissionsAsync();
        if (status !== 'granted') {
            Alert.alert('Permiso denegado', 'Se necesita acceso a la cámara
para tomar fotos.');
```

```

const onChange = (property: string, value: any) => {
  setDataUser({ ...dataUser, [property]: value });
};

const openDatePicker = () => {
  setShowDatePicker(true);
};

const dismissKeyboard = () => {
  Keyboard.dismiss();
};

const showLoginMessage = () => {
  if (Platform.OS === "android") {
    ToastAndroid.show("Iniciando Sesión", ToastAndroid.LONG);
  } else {
    Alert.alert("Iniciando Sesión");
  }
};

const handleRegister = async () => {
  setLoading(true);
  try {
    const userResult = await getUserByEmail(dataUser.email);
    console.log(userResult);
    if (!userResult.success) {
      // Verificar si file está definido
      if (!file) {
        Alert.alert('Error', 'Por favor seleccione una imagen.');
```

setLoading(false);

return;

}

// Convertir imagen a blob

const response = await fetch(file.uri);

const blob = await response.blob();

// Crear FormData y agregar los campos necesarios

const formData = new FormData();

formData.append('image', {

name: 'photo.jpg',

type: 'image/jpeg',

uri: file.uri

} as any);

formData.append('id', dataUser.id ? dataUser.id.toString() :

'0'); // Verificar que id no sea null

formData.append('email', dataUser.email);

```

        formData.append('nacimiento',
dataUser.nacimiento.toISOString());
        formData.append('nombres', dataUser.nombres);
        formData.append('telefono', dataUser.telefono);
        formData.append('clave', dataUser.clave ?
dataUser.clave.toString() : "");

        // Registrar nuevo usuario
        const usuarioRegistrado = await registerUser(formData);
        if (usuarioRegistrado.success) {
            console.log(usuarioRegistrado.data);
            Alert.alert('Registro exitoso', usuarioRegistrado.message);
            // TODO: CONFIGURAR RENDER DEL HOMESCREEN CON EL LOGEO
            navigation.navigate('HomeScreen'); // Navegar a la pantalla
de inicio
        } else {
            Alert.alert('Hubo un problema!', usuarioRegistrado.message);
        }

        } else {
            // El usuario ya existe
            Alert.alert('Hey!', userResult.message);
        }
    } catch (error) {
        console.error('Error en el registro:', error);
        Alert.alert('Error', 'Hubo un error al registrar el usuario.');
```

de inicio

```

    } finally {
        setLoading(false);
    }
};

const handleDateChange = (event: any, selectedDate?: Date) => {
    const currentDate = selectedDate || date;
    setShowDatePicker(false);
    setDate(currentDate);
    setFormattedDate(currentDate.toLocaleDateString());
    onChange('nacimiento', currentDate);
};

return (
    <KeyboardAvoidingView
        behavior={Platform.OS === "ios" ? "padding" : "height"}
        style={styles.container}
    >
        <StatusBar barStyle="dark-content" />
        <TouchableWithoutFeedback onPress={dismissKeyboard}>
            <View style={styles.container}>
                <View style={styles.logoContenedor}>
```

```

        <TouchableOpacity onPress={() => { setModalVisible(true);
console.log(dataUser.image) }}>
        {
            dataUser.image == ''
            ? <Image
                source={require('../assets/icons/usuario(3).png')}
                style={styles.logoImage}
            />
            : <Image
                source={{ uri: dataUser.image }}
                style={styles.logoImage}
            />
        }
    </TouchableOpacity>
    <Text style={styles.logotext}>SELECCIONA UNA IMAGEN</Text>
</View>

<View style={styles.form}>
    <Text style={styles.formText}>Crea tu Cuenta</Text>
    <ScrollView>
        <View style={styles.spaceViewInput}></View>
        <Text style={styles.formText3}>Nombres</Text>
        <TextInputComponent
            icon={require("../assets/icons/usuario.png")}
            title="Nombre y Apellido"
            secureTextEntry={false}
            onChangeText={(text) => setDataUser(prevState => ({
...prevState, nombres: text })))}
        />
        <View style={styles.spaceView}></View>
        <Text style={styles.formText3}>Contraseña</Text>
        <PasswdInputComponent
            icon={require("../assets/icons/password.png")}
            title="*****"
            secureTextEntry={true}
            onChangeText={(text) => setDataUser(prevState => ({
...prevState, clave: text })))}
        />
        <View style={styles.spaceView}></View>
        <Text style={styles.formText3}>Email</Text>
        <TextInputComponent
            icon={require("../assets/icons/usuario.png")}
            title="correo@dominio.com"
            secureTextEntry={false}
            onChangeText={(text) => setDataUser(prevState => ({
...prevState, email: text })))}
        />
    </ScrollView>
</View>

```

```

<View style={styles.spaceView}></View>
<Text style={styles.formText3}>Número de Celular</Text>
<TextInputComponent
  icon={require("../assets/icons/password.png")}
  title="09999999999"
  secureTextEntry={false}
  onChangeText={(text) => setDataUser(prevState => ({
...prevState, telefono: text })))}
/>
<View style={styles.spaceView}></View>
<Text style={styles.formText3}>Fecha de Nacimiento</Text>
<View style={styles.formboxInputView}>
  <TextInput
    // style ={{ height: 40, borderColor: "gray",
borderWidth: 1 }}
    style = {styles.formTextInpuDateTimet}
    value={formattedDate}
    onFocus={openDatePicker}
    onBlur={() => setShowDatePicker(false)}
    placeholder="DD/MM/YY"
  />
  {showDatePicker && (
    <DateTimePicker
      value={date}
      mode="date"
      display="default"
      onChange={handleDateChange}
    />
  )}
</View>
</ScrollView>
<View>
  <MyButton2 title="Registrarme" onPress={handleRegister}
/>

</View>
<ModalPickImage
  openGallery={pickImage}
  openCamera={takePhoto}
  modalUseState={modalVisible}
  setModalUseState={setModalVisible}
/>

{
  loading &&
  <ActivityIndicator
    style={styles.loading}
    size="large"
    color={backgroundColors.tertiary}

```

```

        />
      }
    </View>
  </View>
</TouchableWithoutFeedback>
</KeyboardAvoidingView>
);
};

```

Anexo F. Código de implementación para la página principal

```

export const HomeScreen = () => {
  const [modalVisible, setModalVisible] = useState(false);
  const [modalVisibleInfo, setModalVisibleInfo] = useState(false);

  const navigation =
useNavigation<StackNavigationProp<RootStackParamList>>();
  const openDrawer = () => {
    navigation.dispatch(DrawerActions.openDrawer());
  };

  const dismissKeyboard = () => {
    Keyboard.dismiss();
  };

  const handleConfirm = (soup: string, mainCourse: string, orderType:
string) => {
    Alert.alert('Pedido Confirmado', `Sopa: ${soup}\nPlato Fuerte:
${mainCourse}\nTipo de Pedido: ${orderType}`);
    setModalVisible(false);
  };

  return (
    <KeyboardAvoidingView
      behavior={Platform.OS === 'ios' ? 'padding' : 'height'}
      style={styles.container}
    >
      <StatusBar style="light" />
      <TouchableWithoutFeedback onPress={dismissKeyboard}>
        <View style={styles.container}>
          <View style={styles.logoContenedor}>
            <View style={styles.inputContenedor}>
              <TextInputBuscador
                icon={require('../assets/icons/usuario.png')}
                title="Buscar"
                secureTextEntry={false}

```

```

        onChangeText={(text) => console.log(text)}
      />
      <View style={styles.iconos}>
        <View style={styles.iconoback}>
          <Icons name="cart-outline" size={28}
color={backgroundColors.secondary} style={styles.icono}/>
        </View>
        <View style={styles.iconoback}>
          <TouchableOpacity onPress={openDrawer}>
            <Icons name="notifications-outline" size={28}
color={backgroundColors.secondary} style={styles.icono} />
          </TouchableOpacity>
        </View>
        <View style={styles.iconoback}>
          <TouchableOpacity onPress={openDrawer}>
            <Icons name="menu-outline" size={28}
color={backgroundColors.secondary} style={styles.icono}/>
          </TouchableOpacity>
        </View>
      </View>
    </View>
    <View>
      <Text style={styles.formText}>Buenos Días</Text>
      <Text style={styles.formText2}>Almuerzos Ejecutivos Al
Mejor Precio Y Calidad</Text>
    </View>
  </View>

  <View style={styles.form}>
    <View style={styles.formbox}>
      <TouchableOpacity onPress={() =>
setModalVisibleInfo(true)}>
        <View style={styles.imageboton}>
          <Icons name="restaurant-outline" size={65}
color={backgroundColors.secondary} />
          <Text>Almuerzos</Text>
        </View>
      </TouchableOpacity>
      <TouchableOpacity onPress={() => console.log('hola')}>
        <View style={styles.imageboton}>
          <Icons name="ice-cream-outline" size={65}
color={backgroundColors.secondary} />
          <Text>Postres</Text>
        </View>
      </TouchableOpacity>

      <TouchableOpacity onPress={() => console.log('hola')}>
        <View style={styles.imageboton}>

```

```

        <Icons name="beer-outline" size={65}
color={backgroundColors.secondary} />
        <Text>Bebidas</Text>
    </View>
</TouchableOpacity>

</View>
<View style={styles.divider} />
<Text style={styles.texthome}> PEDIR EN </Text>

<ScrollView>
    <View style={styles.homecenter}>
        <View style={styles.distribucionrow}>
            <TouchableOpacity onPress={() =>
setModalVisible(true)} style={styles.registrarform}>
                <Text style={styles.texthome}>Ficoa</Text>
                <Image
                    style={styles.logoImage}
                    source={{
                        uri: 'https://gourmendy.com/wp-
content/uploads/2022/03/menu-completo-ficoa.jpg',
                    }} />
            </TouchableOpacity>

            <View style={styles.registrarform}>
                <Text style={styles.texthome}>Centro</Text>
                <Image
                    style={styles.logoImage}
                    source={{uri: 'https://gourmendy.com/wp-
content/uploads/2022/03/menu-completo-centro.jpg'}}
                />
            </View>

        </View>
        <View style={styles.distribucionrow}>

            <View style={styles.registrarform}>
                <Text style={styles.texthome}>Católica</Text>
                <Image
                    style={styles.logoImage}
                    source={require("../assets/icons/almuerzo1.jpe
g")}
                />
            </View>

        </View>
    </View>
</View>

```

```

        </ScrollView>

        </View>
    </View>
</TouchableWithoutFeedback>

<OrderModal
    visible={modalVisible}
    onClose={() => setModalVisible(false)}
    onConfirm={handleConfirm}
    onAddMore={() => Alert.alert('Agregar Más', 'Función para
agregar más')}
/>
<ModalInfo
    visible={modalVisibleInfo}
    onClose={() => setModalVisibleInfo(false)}
/>
</KeyboardAvoidingView>
);
};

```

Anexo G. Código de implementación de un nuevo pedido

```

const handleConfirm = async () => {
    if (selectedItems.length === 0) {
        setErrorMessage('Por favor, seleccione al menos un menú.');
```

```
        return;
    }
    if (direccion.trim() === '') {
        setErrorMessage('Por favor, ingrese una dirección.');
```

```
        return;
    }

    const pedido: Pedido = {
        menus: selectedItems,
        tipoPedido: "PARA LLEVAR",
        direccion: direccion
    };

    try {
        await guardar('menuSeleccionado', JSON.stringify(pedido));
        await insertPedido(pedido, direccion);
        onConfirm(soup, mainCourse, "PARA LLEVAR");
    } catch (error) {
        console.error('Error saving to local storage or inserting
order:', error);
    }
}

```

```

    }
  };

  const handleAddSelection = () => {
    if (soup && mainCourse && entrada) {
      setSelectedItems([...selectedItems, { sopa: soup,
platoFuerte: mainCourse, entrada }]);
      setSoup('');
      setMainCourse('');
      setEntrada('');
    }
  };

  const handleEditItem = (index: number) => {
    const item = selectedItems[index];
    setSoup(item.sopa);
    setMainCourse(item.platoFuerte);
    setEntrada(item.entrada);
    setSelectedItems(selectedItems.filter((_, i) => i !== index));
  };

  const handleDeleteItem = (index: number) => {
    setSelectedItems(selectedItems.filter((_, i) => i !== index));
  };

  const renderSelectedItem = ({ item, index }: { item: Menu, index:
number }) => (
    <Pressable>
      <View style={styles.selectedItem}>
        <Text>`Entrada: ${item.entrada}\nSopa:
${item.sopa}\nPlato Fuerte: ${item.platoFuerte}`</Text>
        <View style={styles.actionButtons}>
          <TouchableOpacity onPress={() =>
handleEditItem(index)}>
            <Image
source={require('../assets/icons/edit.png')}
style={styles.actionIcon} />
          </TouchableOpacity>
          <TouchableOpacity onPress={() =>
handleDeleteItem(index)}>
            <Image
source={require('../assets/icons/delete.png')}
style={styles.actionIcon} />
          </TouchableOpacity>
        </View>
      </View>
    </Pressable>
  );

```

```

return (
  <Modal
    animationType="fade"
    transparent={true}
    visible={visible}
    onRequestClose={onClose}
  >
    <TouchableWithoutFeedback onPress={onClose}>
      <SafeAreaView style={styles.centeredView}>
        <TouchableWithoutFeedback>
          <View style={styles.modalView}>
            <TouchableOpacity
style={styles.closeButton} onPress={onClose}>
              <Image
                source={require('../assets/icons/exitModal.png')}
                style={styles.closeButtonImage}
              />
            </TouchableOpacity>

            <View style={styles.header}>
              <Text
style={styles.headerText}>Selecciona tu menú</Text>
            </View>

            {errorMessage ? (
              <Text
style={styles.errorMessage}>{errorMessage}</Text>
            ) : null}

            {loading ? (
              <ActivityIndicator size="large"
color={textColors.sexto} />
            ) : (
              <>
                <TouchableOpacity onPress={() =>
setShowEntradaPicker(true)} style={styles.floatingPicker}>
                  <Text
style={styles.pickerTitle}>Entrada: {entrada || 'Elige tu
entrada'}</Text>

                  {showEntradaPicker && (
                    <View
style={styles.pickerContainer}>
                      <Picker
                        selectedValue={entrada}

```

```

200, width: '100%' }}
mValue) => {
Value);
icker(false);

style={{ height:
onValueChange={(ite
setEntrada(item
setShowEntradaP
}}}
>
{getPickerItems('EN
TRADA').map((plato, index) => (
<Picker.Item
key={index} label={plato || 'Elige tu entrada'} value={plato} />
))}
</Picker>
</View>
)}
</TouchableOpacity>

<TouchableOpacity onPress={() =>
setShowSoupPicker(true)} style={styles.floatingPicker}>
<Text
style={styles.pickerTitle}>Sopa: {soup || 'Elige tu sopa'}</Text>
{showSoupPicker && (
<View
style={styles.pickerContainer}>
<Picker
selectedValue={soup
}
style={{ height:
200, width: '100%' }}
onValueChange={(ite
mValue) => {
setSoup(itemVal
ue);
setShowSoupPick
er(false);
}}}
>
{getPickerItems('SO
PA').map((plato, index) => (
<Picker.Item
key={index} label={plato || 'Elige tu sopa'} value={plato} />
))}
</Picker>
</View>
)}

```

```

        </TouchableOpacity>

        <TouchableOpacity onPress={() =>
setShowMainCoursePicker(true)} style={styles.floatingPicker}>
            <Text
style={styles.pickerTitle}>Plato Fuerte: {mainCourse || 'Elige tu plato
fuerte'}</Text>

            {showMainCoursePicker && (
                <View
style={styles.pickerContainer}>
                    <Picker
                        selectedValue={main
Course}
                        style={{ height:
200, width: '100%' }}
                        onChange={(ite
mValue) => {
                            setMainCourse(i
temValue);
                            setShowMainCour
sePicker(false);
                        }}
                    >
                        {getPickerItems('PL
ATO FUERTE').map((plato, index) => (
                            <Picker.Item
key={index} label={plato || 'Elige tu plato fuerte'} value={plato} />
                        ))}
                    </Picker>
                </View>
            )}
        </TouchableOpacity>

        <TextInputComponentModalPedido
icon={require("../../assets/icons/direccion.png")} title="Dirección
y referencia" secureTextEntry={false} onChangeText={(text) =>
setDireccion(text)} />

        <MyButton3 title='Agregar menú'
onPress={handleAddSelection} />

        <View style={styles.divider} />
        {selectedItems.length > 0 && (
            <FlatList
                data={selectedItems}
                renderItem={renderSelectedI
tem}

```

```

                                keyExtractor={({item, index})
=> index.toString()})
                                style={styles.selectedList}
                                />
                                ))
                                <View style={styles.divider} />

                                <View style={{ flexDirection:
'row', alignItems: 'center' }}>
                                <View style={{ flex: 1, height:
1, backgroundColor: 'black' }} />
                                <View>
                                    <Text style={{ width:
140, textAlign: 'center', fontWeight: 'bold', fontSize: 20 }}>
                                        Total :
                                        {(selectedItems.length * precioPedido).toFixed(2)} $
                                    </Text>
                                </View>
                                <View style={{ flex: 1, height:
1, backgroundColor: 'black' }} />
                                </View>

                                <View style={styles.divider} />
                                <MyButton2 title='Confirmar pedido'
onPress={handleConfirm} />
                                </>
                                ))
                                </View>
                                </TouchableWithoutFeedback>
                                </SafeAreaView>
                                </TouchableWithoutFeedback>
                                </Modal>
                            );
    };

```

Anexo H. Código de sincronización con la API de woocommerce

```

const insertarOrden = async (req, res) => {
  const { token, pedidodMap, direccion } = req.body;
  let selectedMenu = pedidodMap;
  try {
    let usuario = { ...decodeToken(token) }; // Decodificar token y
obtener usuario
    console.log(usuario);

    let calidNombres = usuario.data.nombres.trim().includes(" ")
      ? usuario.data.nombres.split(" ")

```

```

        : [usuario.data.nombres];
usuario.data.nombres = calidNombres;
console.log('Nombre a ingresar');
console.log(calidNombres);
const variation_id = 1148;
const product_id = 1096;

const detallePedidoMenu = selectedMenu.map((menuIteration) => {
    return {
        product_id: product_id,
        variation_id: variation_id,
        meta_data: menuIteration,
    };
});

let dataUser = {
    first_name: usuario.data.nombres[0],
    last_name:
        usuario.data.nombres.length > 1 && usuario.data.nombres.length
< 3
        ? usuario.data.nombres[1]
        : usuario.data.nombres[0],
    address_1: direccion,
    city: "Ambato",
    state: "Tungurahua",
    country: "Ecuador",
    email: usuario.data.email,
    phone: usuario.data.telefono,
};

let menu = {
    payment_method: "bacs",
    payment_method_title: "Direct Bank Transfer",
    set_paid: true,
    billing: { ...dataUser },
    line_items: detallePedidoMenu,
};

console.log(JSON.stringify(menu));
// Pedidos?
if (selectedMenu.length > 0) {
    try {
        const response = await woocommerce.post("orders", menu);
        if (response.status === 201 || response.status === 200) {
            // Verifica el código de estado
            console.log("Order created:", response.data);

            let sqlInsertarpedido = `

```

```

        INSERT INTO pedidos
            (id_fkcliente, id_fkmenuscursal, tipo_variacion,
flag_estado_pedido, created_at, updated_at, detalle_pedido,
id_fkrepartidor, direccion, required)
            VALUES (?, ?, ?, ?, NOW(), NOW(), ?, 0, ?, 1)
        `;
        // Ejecución de la sentencia SQL aquí, usando tu cliente
MySQL2
        await DB.query(sqlInsertarpedido, [usuario.data.id,
product_id, variation_id, "PENDIENTE", JSON.stringify({total:
response.data.total, menus: response.data.line_items}), direccion] );
        console.log("SE PROCESA LA INSERCIÓN A LA BASE DE DATOS");
        res.json({ data: response.data });
    } else {
        console.error("Failed to create order:", response.data);
        res.json({ error: "Failed to create order", details:
response.data });
    }
    } catch (error) {
        console.error(error.message);
        res.json({ error: 'Failed to create order-> '+error.message,
success: false });
    }
    } else {
        res.json({ success: false, error:"No hay pedidos ingresados" });
    }
    } catch (error) {
        res.json({ error: error.message, success: false });
    }
    }
};

```

Anexo I. Código de implementación de inicio de sesión del repartidor

```

return (
    <KeyboardAvoidingView
        behavior={Platform.OS === "ios" ? "padding" : "height"}
        style={styles.container}
    >
        <StatusBar style="light" />
        <View style={styles.container}>

            <View style={styles.logoContenedor}>
                <Text style={styles.logotext}>Acceder</Text>
            </View>

            <TouchableWithoutFeedback onPress={dismissKeyboard}>

```

```

        <View style={styles.form}>
            <Text style={styles.formText}>Bienvenido</Text>
            <Text style={styles.formText2}>Ingresa tu correo
electrónico y contraseña para acceder a la sección de
repartidores</Text>

            <View style={styles.spaceViewInput}></View>
            <Text style={styles.formText3}>Email</Text>
            <TextInputComponent
icon={require(".././../assets/icons/usuario.png")}
title="correo@correo.com" secureTextEntry={false} onChangeText={({text})
=> setEmail(text)} />
            <View style={styles.spaceView}></View>
            <Text style={styles.formText3}>Contraseña</Text>
            <PasswdInputComponent
icon={require(".././../assets/icons/password.png")}
title=" *****" secureTextEntry={true} onChangeText={({text}) =>
setpassword(text)} />
            <View style={styles.registrarform}>
                <TouchableOpacity onPress={() =>
navigation.navigate('ChangePassword')}>
                    <Text style={styles.registrartext}>Olvidé Mi
Contraseña</Text>
                </TouchableOpacity>
            </View>
            <View style={styles.spaceViewInput}></View>
            <View>
                <MyButton2 title="Ingresar" onPress={handleLogin} />
                { /* <MyButton2 title="Ingresar" onPress={() =>
navigation.navigate('HomeScreen')} /> */}
            </View>

        </View>
    </TouchableWithoutFeedback>
</View>
</KeyboardAvoidingView>
);
};

```

Anexo J. Código de implementación para seguimiento de pedidos

```

const ScreenPedidoCocinando: React.FC<ScreenPedidoPendienteProps> = ({
pedidos, onPedidoSelect, onPedidoConfirm }) => {
    const [modalVisible, setModalVisible] = useState(false);

```

```

    const [selectedPedido, setSelectedPedido] = useState<{ id: number;
newEstado: string } | null>(null);

    const handleConfirm = (id: number, newEstado: string) => {
        setSelectedPedido({ id, newEstado });
        setModalVisible(true);
    };

    const confirmAction = () => {
        if (selectedPedido !== null) {
            onPedidoConfirm(selectedPedido.id,
selectedPedido.newEstado);
        }
        setModalVisible(false);
    };

    const cancelAction = () => {
        setModalVisible(false);
    };

    const prepararPedidos = pedidos.filter(pedido =>
pedido.flagEstadoPedido === 'PREPARANDO');
    const enCaminoPedidos = pedidos.filter(pedido =>
pedido.flagEstadoPedido === 'EN CAMINO');

    if (pedidos.length === 0) {
        return (
            <View style={styles.container}>
                <Text style={styles.errorText}>No hay pedidos
disponibles.</Text>
            </View>
        );
    }
    interface PedidoItemProps {
        pedido: any;
    }

    const PedidoItem: React.FC<PedidoItemProps> = ({ pedido }) => {
        const [expanded, setExpanded] = useState(false);

        const toggleExpand = () => {
            setExpanded(!expanded);
        };

        return (<Pressable onPress={() => { onPedidoSelect(pedido.id);
toggleExpand() }}>
            <View style={styles.pedidoItem}>
                <View style={styles.container}>
                    <View style={styles.direccion}>

```

```

        <Image
            style={styles.logoImage}
            source={require('../assets/icons/almuerzo1.jpeg')}
        />
        <View style={styles.sucursal}>
            <Text style={styles.textsucursal}>Cantidad
menús: {(JSON.parse(pedido.detallePedido).menus).length}</Text>
            <Text style={styles.textsucursal}>Cliente:
{pedido.nombresCli}</Text>
            <Text style={styles.textsucursal}>Teléfono:
{pedido.telefonoCli}</Text>
            <Text
style={styles.textsucursal}>Dirección: </Text>
            <Text
style={styles.textsucursal}>{pedido.direccion}</Text>
            <Text
style={styles.textsucursal}>{format(new Date(pedido.creadoEl),
'dd/MM/yyyy')}</Text>
            <Text
style={styles.textEstado}>{pedido.flagEstadoPedido}</Text>
        </View>
        <View style={styles.sucursal}>
            <Text
style={styles.price}>${JSON.parse(pedido.detallePedido).total}</Text>
        </View>
    </View>
    {expanded && (
        <View style={styles.expandableContent}>
            <MenuTable
menus={JSON.parse(pedido.detallePedido).menus} />
        </View>
    )}
</View>
    {pedido.flagEstadoPedido === 'PREPARANDO' && (
        <MyButtonMini title="Marcar en Camino" onPress={()
=> handleConfirm(pedido.id, 'EN CAMINO')} />
    )}
    {pedido.flagEstadoPedido === 'EN CAMINO' && (
        <MyButtonMini title="Marcar como entregado"
onPress={() => handleConfirm(pedido.id, 'ENTREGADO')} />
    )}
</View>
</Pressable>
);

const renderItem = ({ item }: { item: Pedido }) => (
    <PedidoItem pedido={item} />

```

```

    );

    return (
      <View>
        <FlatList
          data={[...prepararPedidos, ...enCaminoPedidos]}
          renderItem={renderItem}
          keyExtractor={(item) => item.id.toString()}
        />
        <Modal
          animationType="slide"
          transparent={true}
          visible={modalVisible}
          onRequestClose={() => setModalVisible(false)}
        >
          <View style={styles.modalView}>
            <Text style={styles.modalText}>¿Está seguro de
cambiar el estado del pedido?</Text>
            <View style={styles.buttonContainer}>
              <MyButton3 title="Sí" onPress={confirmAction}
            />
              <MyButton3 title="No" onPress={cancelAction} />
            </View>
          </View>
        </Modal>
      </View>
    );
  };
};

```

Anexo K. Código de implementación de notificaciones utilizando expo

```

useEffect(() => {
  registerForPushNotificationsAsync().then(token => {
    console.log("Token de notificación push:", token);
    if (token) {
      setExpoPushToken(token);
    }
  });

  if (Platform.OS === 'android') {
    Notifications.getNotificationChannelsAsync().then(value =>
setChannels(value ?? []));
  }

  notificationListener.current =
Notifications.addNotificationReceivedListener(notification => {
  setNotification(notification);
});

```

```

    });

    responselister.current =
Notifications.addNotificationResponseReceivedListener(response => {
    console.log(response);
});

    return () => {
        notificationListener.current &&
Notifications.removeNotificationSubscription(notificationListener.curre
nt);
        responselister.current &&
Notifications.removeNotificationSubscription(responselister.current);
    };
}, []);

useEffect(() => {
    const checkAuth = async () => {
        const token = await LocalStorage().obtenerItem('authToken');
        setIsAuthenticated(token != null);
    };

    checkAuth();
}, []);
// Configuración de notificaciones
async function registerForPushNotificationsAsync() {
    let token;

    if (Platform.OS === 'android') {
        await Notifications.setNotificationChannelAsync('default', {
            name: 'default',
            importance: Notifications.AndroidImportance.MAX,
            vibrationPattern: [0, 250, 250, 250],
            lightColor: '#FF231F7C',
        });
    }

    if (Device.isDevice) {
        const { status: existingStatus } = await
Notifications.getPermissionsAsync();
        let finalStatus = existingStatus;
        if (existingStatus !== 'granted') {
            const { status } = await Notifications.requestPermissionsAsync();
            finalStatus = status;
        }
        if (finalStatus !== 'granted') {
            alert('Failed to get push token for push notification!');
            return;
        }
    }
}

```

```

    }
    try {
        const projectId = Constants?.expoConfig?.extra?.eas?.projectId ??
        Constants?.easConfig?.projectId;
        if (!projectId) throw new Error('Project ID not found');
        token = (await Notifications.getExpoPushTokenAsync({ projectId
    })).data;
        console.log(token);
    } catch (e) {
        console.error('Error obteniendo el token de notificaciones
push:', e);
    }
    } else {
        alert('Debe usar un dispositivo físico para las notificaciones
push');
    }

    return token;
}

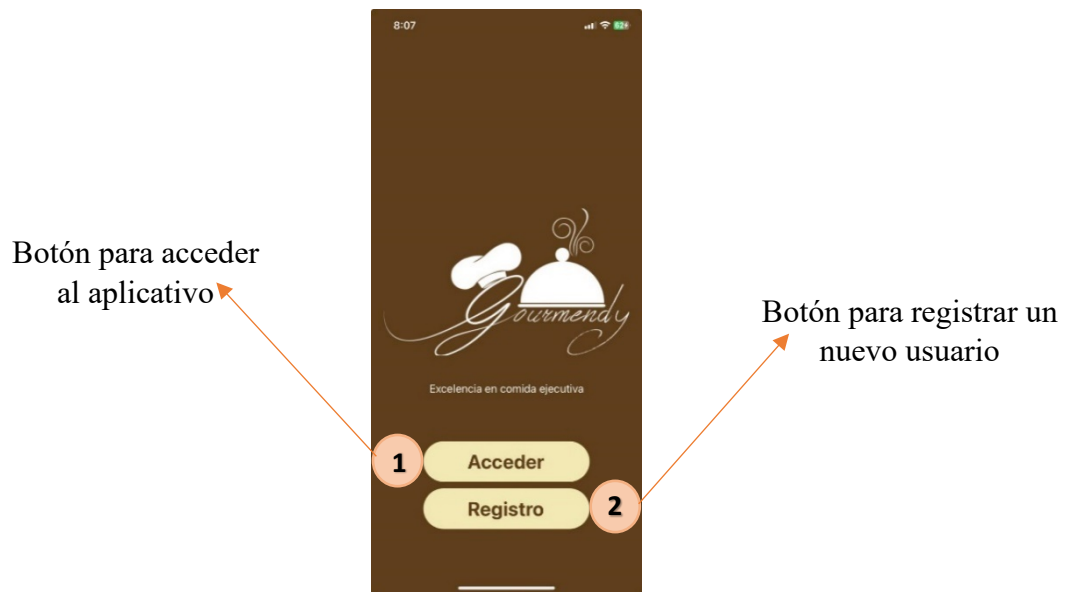
async function schedulePushNotification() {
    await Notifications.scheduleNotificationAsync({
        content: {
            title: "¡Tienes un nuevo mensaje! 📬",
            body: 'Aquí está el cuerpo de la notificación',
            data: { data: 'data goes here' },
        },
        trigger: { seconds: 2 },
    });
}

```

Anexo L. Manual de usuario

Manual de Usuario para aplicación móvil de gestión de pedidos y seguimiento de delivery en el sistema de control de restaurantes de la empresa Connectaservices

1. Pantalla de Inicio



En la pantalla de inicio encontramos el botón de acceder para poder ingresar a nuestra cuenta o el botón de registro donde podemos crear un nuevo usuario

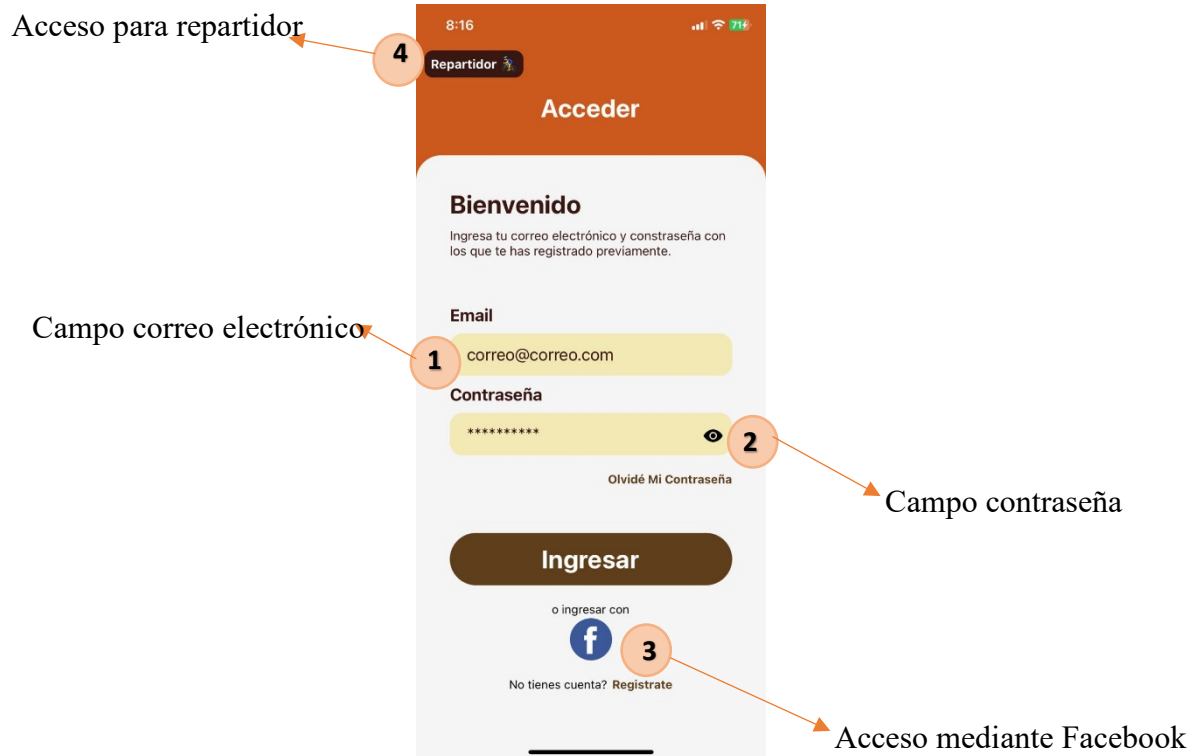
2. Página de registro de usuario

En esta página podemos crear la cuenta para ingresar a la aplicación móvil ingresando la siguiente información:

1. Ingresamos nuestro nombre y apellido

2. Introducimos una contraseña con mínimos de 6 caracteres que tenga al menos una mayúscula y un número
3. Registramos nuestro correo electrónico, el cual se usa en el inicio de sesión
4. Introducimos nuestro número de celular que nos servirá en caso de olvidarnos la contraseña
5. Ingresamos nuestra fecha de nacimiento con la cual verificamos si tenemos la edad suficiente para manejar la aplicación

3. Pagina de acceso



En esta pagina ingresamos el correo electrónico y la contraseña con la que nos registramos previamente para poder ingresar a la aplicación, también encontramos el botón para poder ingresar desde nuestro Facebook y además se encuentra en la parte superior izquierda el botón de acceso para repartidor.

4. Pantalla de acceso repartidor

8:17

Acceder

Bienvenido

Ingresa tu correo electrónico y contraseña para acceder a la sección de repartidores

Email

1 correo@correo.com

Contraseña

***** 2

Olvidé Mi Contraseña

Ingresar

Esta página es exclusiva para los repartidores, ya que ellos ingresaran el correo y la contraseña que cada repartidor ya debe poseer

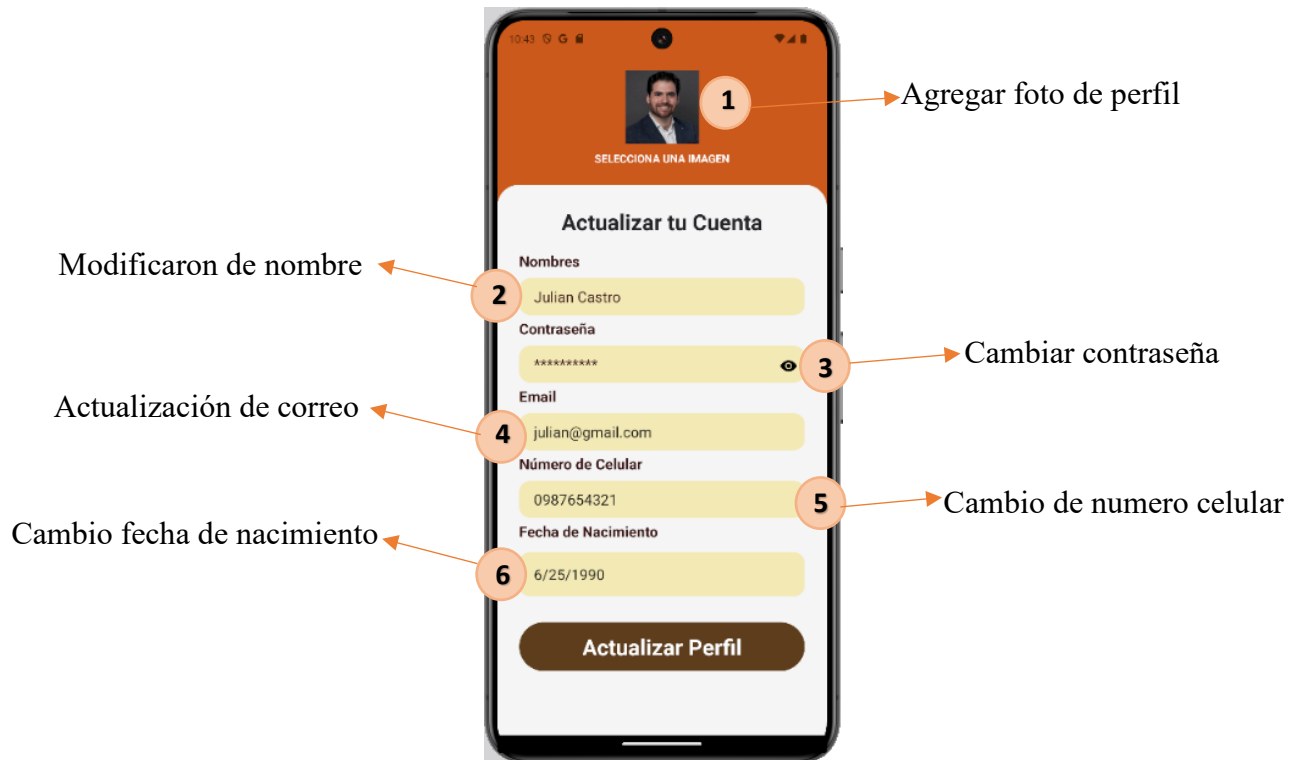
5. Pantalla de inicio



En la pantalla de inicio se encuentran una variedad de acciones como:

1. Buscador para las diferentes sucursales que posee el restaurante
2. Carrito de pedidos pendientes
3. Campanita de configuracion notificaciones
4. Boton para acceso al perfil de usuario
5. Pestañas informativas de los diferentes productos de los puntos de venta
6. Seccion para realizar pedido por sucursal
7. Navegador de pantallas

6. Pantalla de actualización de perfil



En esta pantalla podemos actualizar los diferentes campos de nuestro perfil de usuario como:

1. Agregar una foto para nuestro perfil de usuario
2. Modificación de Nombre y Apellido
3. Cambio de contraseña
4. Actualización de correo electrónico
5. Cambio de numero celular
6. Ajuste de fecha de nacimiento

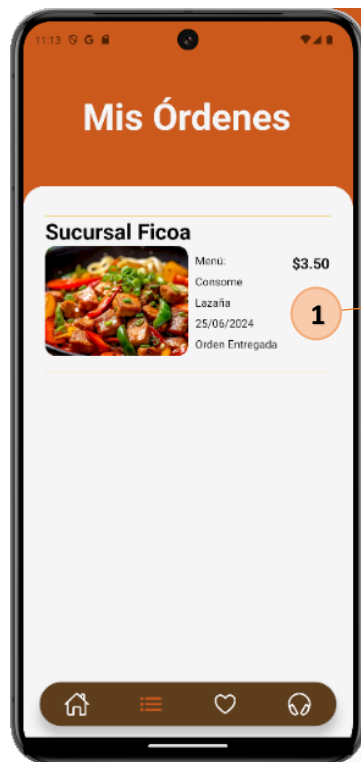
7. Pantalla para realizar pedido



Esta pantalla se abre después de haber seleccionado la sucursal a la que deseamos hacer el pedido, aquí encontramos diferentes opciones como:

1. Botón para poder agregar otro pedido
2. Botón para escoger el tipo de entrada que desea seleccionar
3. Botón de selección de sopa
4. Botón para poder elegir el plato fuerte que queremos
5. Botón para confirmar nuestro pedido

8. Pantalla de mis ordenes



Ordenes realizadas con estado de seguimiento

En esta pantalla podemos visualizar todas nuestras órdenes pasadas y de igual manera nuestra orden actual con su estado de seguimiento para su correcta entrega

9. Pantalla de contacto



En esta pantalla encontramos la información de contacto de cada una de las sucursales con las que el restaurante cuenta como:

1. Sucursal de Ficoa
2. Sucursal del Centro de Ambato
3. Sucursal Universidad Católica Ambato

10. Pantalla de Listado de pedidos para el resepcionista



En esta pantalla se podrá visualizar el estado del pedido, como también navegar entre pantallas ancladas a los pedidos