

**Desarrollo de un aplicativo móvil como estrategia de ventas para apoyar el
servicio al cliente: Caso de estudio UUZAJI en Aguachica**

Jhon Jairo Suarez Florez

Director

Msc. Luis Manuel Palmera Quintero

Monografía para optar por el título de Ingeniero de Sistemas

Universidad Popular del Cesar – Seccional Aguachica

Facultad Ingenierías y Tecnologías

Programa ingeniería de Sistemas

Línea de investigación – Ingeniería del Software

Aguachica, Cesar

2024

Nota de aceptación

Nota de aceptación:

Luis Manuel Palmera Quintero
Director

Msc. Lizeth Badillo Duran
Evaluador 1

Esp. María Fernanda Toscano
Evaluador 2

Aguachica, Día_____ Mes_____ Año_____

Dedicatoria

Primeramente, a Dios que me concedió la disciplina necesaria para lograr este objetivo, segundo una total gratitud a mis padres que estuvieron durante toda mi etapa de formación académica apoyándome con sus deseos de superación, brindando la capacidad de ser perseverante y tener la sabiduría en cada objetivo que me propuse. Un agradecimiento especial a mis padres por su constante apoyo y motivación.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas e institución que han hecho posible la realización de este proyecto de grado:

En primer lugar, al Ingeniero Luis Palmera, director de este proyecto, por su guía, sus asesorías y su constante apoyo durante todo el proceso de investigación y redacción.

A la Universidad Popular del Cesar, Seccional Aguachica, por brindarme la oportunidad de formarme como profesional y proporcionarme los recursos necesarios para desarrollar esta investigación.

A todas las personas que colaboraron directa o indirectamente en el proceso de investigación, por su tiempo, sus aportes y su disposición para compartir sus conocimientos y experiencias.

Finalmente, a mi familia y amigos, por su apoyo incondicional, su paciencia y su constante motivación durante todo mi trayecto universitario.

Tabla de contenido

1. Planteamiento del problema.....	9
1.1 Descripción del problema.....	9
1.1.1 Formulación del problema.....	12
2. Justificación.....	13
3. Objetivos.....	16
3.1 Objetivo general.....	16
3.2 Objetivos específicos.....	16
4. Marco referencial.....	17
4.1 Marco de Antecedentes.....	17
4.1.1 Antecedentes Internacionales.....	17
4.1.2 Antecedentes nacionales.....	20
4.1.3 Antecedentes regionales.....	21
4.2 Marco teórico.....	22
4.2.1 Reactive Architecture.....	22
4.2.2 Node.JS.....	23
4.2.3 Metodología de desarrollo Agíl XP.....	24
4.2.4 Teoría de Diseño Centrado en el Usuario para Aplicaciones Móviles.....	25
4.3 Marco conceptual.....	25
4.4 Marco legal.....	27
5. Metodología.....	29
5.1 Tipo de investigación.....	29
5.2 Metodología de desarrollo.....	29
6. Cronograma de actividades.....	31
7. Resultados.....	32

7.1 Cumplimiento objetivo 1: Requisitos funcionales para el desarrollo del Frontend que permitan el funcionamiento del aplicativo móvil.....	32
7.1.1 Requerimientos Funcionales.....	32
7.1.2 Historias de usuario.....	33
7.2 Cumplimiento objetivo 2: Diseño de los módulos para Frontend que permitan el funcionamiento del aplicativo móvil empleando el entorno de ejecución de ReactNative.....	35
7.2.1 Codificación de los módulos.....	35
7.3 Cumplimiento objetivo 3: Establecer la ejecución del Frontend mediante una herramienta de simulación.....	44
Conclusiones.....	54
Recomendaciones.....	55
Referencias.....	56

Lista de figuras

Figura 1. Servidor Node.JS.....	24
Figura 2. Ciclo de liberación de la Programación Extrema (XP).....	30
Figura 3. CartScreen UUZAJI.....	36
Figura 4. HomeScreen UUZAJI.....	38
Figura 5. PlaceOrdering UUZAJI.....	39
Figura 6. ProductScreen UUZAJI.....	40
Figura 7. Welcome UUZAJI.....	41
Figura 8. Profile UUZAJI.....	42
Figura 9. Login UUZAJI.....	43
Figura 10. Bienvenida UUZAJI.....	44
Figura 12. Registrarse UUZAJI.....	45
Figura 15. Añadir al carrito - UUZAJI.....	46
Figura 17. Ubicación del domicilio - UUZAJI.....	47
Figura 19. Cerrar sesión - UUZAJI.....	48

Lista de Tablas

Tabla 1. Cronograma de actividades	30
Tabla 2. Requisitos Funcionales.....	31
Tabla 3. Historias de usuario.....	33

Organismo: Uuzaji Agencia de Marketing y Comunicación SAS

Línea de investigación: Ingeniería del Software

Campo de aplicación: Desarrollo de aplicativos móviles

1. Planteamiento del problema

1.1 Descripción del problema

Los clientes de hoy se preocupan por la forma como son tratados. No se conformarán con menos de lo que creen que se merecen. ¿Está cumpliendo con sus expectativas? Si no, ¿sabe cómo está afectando eso a su marca hoy? Las implicaciones del servicio al cliente son de mayor alcance de lo que se cree. Solo en los últimos años, han cambiado las actitudes y expectativas de los clientes. Un mal servicio prestado puede tener efectos devastadores. A nivel mundial, el 96% de los consumidores dice que el servicio al cliente es un factor importante en su elección de lealtad a una marca. Al mismo tiempo, expresa que casi el 80% de los consumidores perdonarían una mala experiencia si califican al equipo de servicio como muy bueno, (Ega Futura, 2022). Sin embargo, Inuk, (2023), habla que al navegar en el ámbito del servicio al cliente puede resultar complicado ya que existen errores comunes (respuestas tardías, falta de conocimiento, entre otras) en los que suele caer la empresa.

Con lo mencionado anteriormente, una de las necesidades primordiales de los clientes es que esperan soluciones rápidas a sus problemas. Sin embargo, muchos se enfrentan a largos tiempos de espera para obtener ayuda o son transferidos entre diferentes agentes sin resolver su consulta. Esto es particularmente problemático cuando las aplicaciones móviles no ofrecen un sistema de seguimiento adecuado para las solicitudes de los usuarios. Zendesk, (2023), explica que la resolución de problemas es uno de los mayores desafíos del servicio al cliente, ya que muchas empresas no se enfocan en el origen del problema, esto significa que, probablemente, situaciones similares vuelvan a surgir en el camino. Asimismo, Martínez (2024), expresan que otro problema clave es el

desconocimiento de estrategias efectivas de atención al cliente por parte del personal de la empresa, y estos se manifiestan por falta de procedimiento claros.

Como afirma, Mines y Mines (2024), la gestión de consultas al cliente no solo abarca la venta, sino también la garantía de la satisfacción del cliente. Igualmente, Solís y Valenzuela (2024), indica que las empresas han logrado mantener sus actividades usando herramientas digitales gratuitas, pero son pocas las que cuentan con aplicaciones móviles como desarrollos propios, por lo cual no es posible determinar de qué manera una aplicación móvil mejora un proceso de venta eficazmente. También es relevante mencionar que a través del tiempo el mundo ha cambiado la manera en las cuales se realizan acciones cotidianas, lo cual ha generado que varias empresas transformen su estrategia de negocios para lograr impactar en la sociedad, (Nogales y Salazar, 2024).

La ausencia de estas plataformas ha llevado a que los residentes de estos pueblos adquieran la costumbre de realizar pedidos por teléfono o, en última instancia, desplazándose físicamente hasta los establecimientos, (Cobo y Barrios, 2024). En consecuencia, el uso de la tecnología generó un cambio en la forma como se concebía y se establecían las estrategias de marketing, las cuales debieron adaptarse a las variaciones que trajo el avance tecnológico, (Peña y Figueroa, 2024). De acuerdo con Quiroga y Jaimes (2023), en la actualidad existen aplicaciones de delivery en el mercado que permiten al cliente comprar productos desde una plataforma digital y recibirlo directamente en su domicilio, sin embargo, estas se centran en la atención del consumidor final y no en la distribución de productos para los comerciantes minoristas.

El acelerado desarrollo de las tecnologías de la información y la comunicación (TIC) ha transformado significativamente la forma en que las empresas interactúan con sus clientes y gestionan sus operaciones comerciales, (Donayre, 2024). En este contexto, las aplicaciones móviles se han convertido en herramientas esenciales para optimizar procesos, mejorar la experiencia del usuario y, en última instancia, impulsar las ventas y la fidelización de clientes, (Vivar y Mendro, 2024). Sin

embargo, la adopción de estas tecnologías en las pequeñas y medianas empresas (PYMES), especialmente en contextos local como Aguachica, ha sido un desafío considerable, marcado por limitaciones en el acceso a recursos tecnológicos, resistencia al cambio y falta de conocimiento sobre el desarrollo y uso de herramientas digitales avanzadas.

UUZAJI, una empresa emergente en Aguachica, ha reconocido la necesidad de adaptarse a este nuevo entorno digital para mantenerse competitiva en el mercado local y expandir su base de clientes. Sin embargo, a pesar de su esfuerzo por mejorar sus operaciones, enfrenta varios problemas críticos en la implementación efectiva de estrategias digitales, particularmente en lo que respecta al servicio al cliente y las ventas. La empresa carece de un sistema centralizado y eficiente para manejar las interacciones con los clientes, procesar pedidos y gestionar las ventas, lo que resulta en ineficiencias operativas, insatisfacción del cliente y, en última instancia, una disminución en las ventas.

La falta de un enfoque estratégico en la implementación de tecnologías móviles ha impedido que la empresa aproveche plenamente el potencial de las TIC para mejorar sus operaciones. La empresa actualmente depende de métodos tradicionales de comunicación y gestión de ventas, como llamadas telefónicas, visitas físicas a las instalaciones y la gestión manual de inventarios y pedidos. Estos métodos no solo son ineficientes, sino que también limitan la capacidad de la empresa para responder de manera rápida y efectiva a las necesidades cambiantes de los clientes, especialmente en un entorno de mercado cada vez más dinámico y competitivo.

La creciente demanda de los consumidores por soluciones rápidas y accesibles desde dispositivos móviles, como smartphones y tablets, subraya la urgencia de que las organizaciones implementen estrategias digitales robustas que incluya el desarrollo de un aplicativo móvil, (Quiroga y Jaimes, 2023). En un mercado donde la satisfacción del cliente es un factor clave para el éxito empresarial, la falta de un sistema eficiente para manejar consultas, quejas y sugerencias, así como para

realizar seguimientos postventa, ha resultado en una baja retención de clientes y en una reputación de servicio que no logra destacar frente a la competencia.

1.1.1 Formulación del problema

¿Cómo el desarrollo de un aplicativo móvil como estrategia de ventas puede apoyar el servicio al cliente en la empresa UUZAJI en la ciudad de Aguachica, Cesar?

2. Justificación

En la actualidad, el entorno empresarial global está caracterizado por la rápida evolución de las tecnologías de la información y la comunicación (TIC), las cuales han transformado radicalmente las dinámicas de las ventas y el servicio al cliente. Las empresas, independientemente de su tamaño o sector, se enfrentan al reto de adaptarse a este entorno digital para mantenerse competitivas. Las TIC han revolucionado la manera en que las empresas se comunican con sus clientes, gestionan sus procesos internos y expanden su presencia en el mercado. Sin embargo, muchas empresas, especialmente las pequeñas y medianas (PYMES), aún enfrentan desafíos significativos en la adopción de estas tecnologías.

El desarrollo de un aplicativo móvil no solo responde a la demanda creciente de los consumidores por soluciones rápidas y accesibles, sino que también representa una estrategia integral para mejorar la eficiencia operativa, optimizar los recursos y, en última instancia, incrementar la competitividad. La investigación justifica su relevancia en el hecho de que un aplicativo móvil puede convertirse en una herramienta esencial para cualquier empresa que desee modernizar su proceso de ventas y mejorar su servicio al cliente.

Esta herramienta facilita la automatización de tareas repetitivas, permite un seguimiento más preciso de las interacciones con los clientes y proporciona una plataforma centralizada para gestionar pedidos, inventarios y comunicaciones. De este modo, las empresas pueden reducir significativamente los errores, mejorar los tiempos de respuesta y ofrecer una experiencia de usuario más fluida y satisfactoria, lo que es fundamental en un entorno donde la fidelización del cliente es cada vez más difícil de lograr.

La satisfacción del cliente es un factor determinante para el éxito a largo plazo de cualquier empresa. En un mercado donde los consumidores tienen acceso a una amplia gama de opciones y son cada vez más exigentes, ofrecer un servicio al cliente excepcional se ha convertido en un diferenciador clave. Un aplicativo móvil permite a las empresas ofrecer un servicio personalizado y accesible, adaptado a

las necesidades y preferencias individuales de cada cliente. Esto no solo mejora la experiencia del cliente, sino que también fortalece la relación empresa-cliente, aumentando las probabilidades de retención y lealtad.

Con lo mencionado anteriormente, cabe resaltar que el desarrollo del aplicativo móvil permitirá a los clientes interactuar con la empresa en cualquier momento y desde cualquier lugar, lo que es crucial en un mundo donde la inmediatez es altamente valorada. La posibilidad de realizar compras, hacer consultas o gestionar pedidos desde un dispositivo móvil proporciona a los clientes una mayor comodidad y flexibilidad, lo que puede traducirse en un aumento de las ventas y una mayor satisfacción general.

La capacidad de innovar y adaptarse a los cambios tecnológicos es esencial para la supervivencia de las empresas en el entorno actual. La investigación se justifica no solo por su impacto directo en la operatividad y la satisfacción del cliente, sino también por su potencial para impulsar la innovación dentro de la empresa. El desarrollo de un aplicativo móvil requiere que la empresa reevalúe y optimice sus procesos internos, lo que puede llevar a la identificación de nuevas oportunidades de mejora y a la creación de valor añadido. Una cultura de innovación dentro de la empresa, motivando a los empleados a explorar nuevas ideas y soluciones tecnológicas que mejoren aún más las operaciones y la experiencia del cliente. Esta mentalidad innovadora es crucial para mantenerse relevante en un mercado que evoluciona rápidamente.

Desde una perspectiva académica, esta investigación contribuye significativamente al cuerpo de conocimiento existente sobre el uso de aplicaciones móviles como estrategias de ventas y apoyo al servicio al cliente. Aunque existe una cantidad considerable de estudios sobre la implementación de tecnologías móviles en grandes empresas, hay una brecha en la investigación sobre cómo estas herramientas pueden ser aplicadas de manera efectiva en PYMES y otros contextos menos explorados.

Este estudio no solo aporta conocimientos prácticos sobre el desarrollo e implementación de aplicaciones móviles, sino que también proporciona insights valiosos sobre los desafíos y oportunidades específicos asociados con la adopción de estas tecnologías en diferentes contextos empresariales. Los hallazgos de esta investigación podrían ser utilizados como base para estudios futuros que exploren la adopción de otras tecnologías emergentes, como la inteligencia artificial o el análisis de datos, en las estrategias de ventas y servicio al cliente. La sostenibilidad empresarial no solo se refiere a la capacidad de una empresa para mantenerse en el mercado, sino también a su capacidad para crecer y adaptarse a las condiciones cambiantes. Un aplicativo móvil puede ser una herramienta clave para lograr esta sostenibilidad, ya que permite a las empresas ser más ágiles y receptivas a las necesidades del mercado.

3. Objetivos

3.1 Objetivo general

Desarrollar un aplicativo móvil como estrategia de ventas para apoyar el servicio al cliente en la empresa UUZAJI de la ciudad de Aguachica, Cesar.

3.2 Objetivos específicos

- Definir los requerimientos funcionales para el desarrollo del Frontend que permitan el funcionamiento del aplicativo móvil.
- Diseñar los módulos del Frontend para el funcionamiento del aplicativo móvil empleando el entorno de ejecución de ReactNative.
- Establecer el conjunto de peticiones para la ejecución de pruebas del Frontend mediante una herramienta de simulación.

4. Marco referencial

4.1 Marco de Antecedentes

El análisis de trabajos académicos o científicos permite al investigador tener bases suficientes para seleccionar la mejor alternativa respecto a la metodología, los instrumentos o herramientas a utilizar, como una forma de complementar su conocimiento a fin de dar respuesta al problema, la pregunta de investigación y los objetivos planteados.

4.1.1 Antecedentes Internacionales

En el ámbito internacional se encuentra el trabajo realizado por Vallejo (2022), titulado "Estudio comparativo de las Tecnologías para el desarrollo del Back-end "NodeJS y PHP"", el cual tuvo como objetivo principal analizar dos tecnologías back-end de amplio uso en el desarrollo de proyectos, por un lado, NodeJS, entorno de ejecución de JavaScript, de código abierto, multiplataforma, versátil, rápido y eficiente que trabaja en el lado del servidor por lo que puede utilizarse como intermediario entre la interfaz de usuario y la base de datos facilitando así la construcción de programas de red escalables, hace posible compilar JavaScript cuando no existe cálculo intensivo de CPU lo que le brinda mayor velocidad, caso contrario sucede cuando existe dicho cálculo.

Por otra parte, PHP es un lenguaje del lado del servidor y de propósito general que permite conectarse a un número significativo de motores de búsqueda de datos utilizados hasta la actualidad, es libre y abierto, de fácil aprendizaje, accesibilidad y que pese a la gran variedad y alternativas existentes presenta gran demanda.

La comparación realizada a través de Google Trends comprendió aspectos relacionados con la seguridad, accesibilidad y rendimiento de estas tecnologías, ambas con ventajas significativas, tanto NodeJS como PHP pueden trabajarse en diferentes plataformas, servicios y gestores, sin embargo, a este último posiblemente se le dejará de dar soporte a futuro, dado que se observa una curva descendiente respecto a su uso. La autora concluye que NodeJS aunque presenta una demanda creciente por las ventajas que ofrece, su uso requiere un buen nivel

de conocimiento en relación con su metodología, sintaxis y funcionalidad en backend.

Por otro lado, Mora (2021), presenta su trabajo de grado titulado “Construir una aplicación móvil para Caarry Soluciones S.A. que se encargue de juntar los servicios de personas independientes y que tengan conocimientos en una determinada área”, quien desarrolla una aplicación móvil que permite ofrecer servicios profesionales de personas independientes, aunque se realiza para Android tiene la posibilidad de ser multiplataforma, eligiendo SCRUM como metodología de desarrollo ágil por las ventajas que ofrece. El interés en este trabajo es la finalidad de la aplicación desarrollada, se puede decir que es similar, dado que se trata de emprendedores de la región buscan a través del aplicativo llegar a los cientos de hogares a nivel local y proyección nacional para vender productos agrícolas propios de la localidad; por otro lado, las fases que comprendió: planificación, análisis y diseño, implementación y pruebas.

Por otra parte, la metodología empleada la cual es de tipo cuantitativo que busca recopilar y analizar información sobre el uso de aplicaciones para la búsqueda de profesionales, diseñándose dos (2) encuestas con preguntas abiertas y aplicadas a una muestra de 18 personas.

El trabajo realizado por Bangare, Gupta, Dalal & Inamdar (2016), artículo titulado “Using Node.JS to Build High Speed and Scalable Backend Database Server”, el cual tuvo como objetivo describir el uso de Node.JS para la construcción de un servidor de base de datos escalable y de alta velocidad, debido a que una de las principales dificultades para los desarrolladores es el contar con un backend flexible y permanente para sus aplicaciones, por lo que hacen uso de servidores en la nube de terceros o implementan un servidor backend, contando entre sus alternativas para la codificación Python y PHP para el acceso a los datos.

Cabe resaltar que la diferencia con Node.JS, es que, al realizar peticiones a la base de datos, este evita que se presenten sobrecargas y se bloquen debido a que se

realizan en forma paralela junto con otras funciones como esperar datos de un archivo, atender una solicitud HTTP entrante, etc.; efectuando subprocesos múltiples en un solo subproceso por eventos en una única CPU, esta es la ventaja más significativa de este entorno, esto gracias a que permite integrar diversas tecnologías para crear un servidor para el sitio web y la aplicación.

El framework incluye paquetes de software como NodeJS y MongoDB, para crear API que les permita conectarse a la base de datos sin tener que preocuparse por la codificación del lado del servidor, al igual que crear controladores para distintas plataformas (Android, .Net, iOS, entre otras) y realizar pruebas de API, sin dejar de lado la seguridad de los datos, para verificar su rendimiento los autores crearon una API con todas las operaciones CRUD empleando una funcionalidad de NodeJS para efectuar las peticiones, recibiendo la solicitud de HTTP de un cliente con datos en JSON y ejecutándolo en MongoDB.

Una última investigación que evidencia el rendimiento y velocidad de NodeJS es la realizada por Subramani, Hemapriya, Anto & Yazhini (2016), descrita en su artículo titulado "Node JS: Building an High Performance Event Manager in Android Platform", cuyo objetivo fue crear un administrador de eventos de alto rendimiento en Android, empleando para ello tecnologías móviles inalámbricas, mensajería en la nube de Google (GCM) y WebSocket.

Se crea un sitio web y la publicidad de este, se debe contar con una comunicación eficiente entre organizador y participantes de un evento, la cual es monitoreada, sin actualización de estados en tiempo real, efectuando la publicación de un determinado evento (simposios, conferencias, etc.), se evita el alojamiento del sitio web y el costo de publicidad en dichos sitios. El organizador ingresa al sistema que verifica los datos de quien inicia sesión en el sitio, tanto usuario como eventos se almacenan en la base de datos, la información o publicidad del evento se envía luego a los móviles.

Las pruebas de rendimiento de NodeJS se comparan con los resultados del

rendimiento de .Net, la cual es superada significativa por el primero, que se evidencia con el número de solicitudes generadas por segundo con un máximo de 420 en .Net y de 509 con NodeJS; respecto al número de solicitudes concurrentes igualmente es mayor con NodeJS así como es menor el tiempo requerido (en segundos), con esta prueba se concluye que este entorno es un 20% más rápido que JavaEE, sin embargo, ambos escalan más allá de lo requerido por un servidor normal (400-500 solicitudes por segundo).

4.1.2 Antecedentes nacionales

En el contexto nacional, el trabajo titulado “Desarrollo del Backend de una aplicación para vinculación de clientes en una entidad bancaria”, realizado por Lagos (2017), cuyo objetivo fue el modelamiento, análisis, desarrollo e implementación de un backend para el módulo de una aplicación de vinculación de clientes de BBVA fiduciaria, proceso que se realizó bajo la metodología ágil XP teniendo en cuenta el nivel de seguridad requerido y acorde con el tipo de datos.

Las operaciones a realizar, la empresa y los clientes. Se determinan las historias de usuario en la etapa de planeación, para proceder a un análisis y definición de los requerimientos funcionales que sirven de base para establecer los casos de uso, la arquitectura y el desarrollo del módulo; así mismo la realización de pruebas que permitió corroborar la integración de este con el sistema ya existente. Respecto a las tecnologías utilizadas se encuentra PHP como lenguaje del lado del servidor y la interfaz REST la cual conecta sistemas basados en HTTP, y en el que se ejecutaron las pruebas de funcionamiento de los diferentes formularios, no se hace referencia a tiempos de respuesta, dado que se trabajó con una arquitectura preexistente.

Asimismo, Sánchez y Molinares (2020), presentan su trabajo titulado “Desarrollo de aplicación móvil para el mejoramiento de los procesos de recolección de datos para inventarios forestales de la Fundación BIMA”, para lo cual utilizan herramientas *Open Source* multiplataforma aplicación que puede ejecutarse en sistemas operativos iOS y Android, accesible desde móviles, Ipad y tabletas.

Respecto a la metodología de investigación es de tipo proyectivo y el enfoque es sistemático basado en la metodología ágil SCRUM y el diseño no experimental, a partir de lo cual se establecen seis (6) fases o etapas de desarrollo.

Los frameworks utilizados fueron Ionic Framework, Ionic Angular, en el diseño de la arquitectura se utilizaron PHP, MySQL, NodeJS, Apache Córdoba, Visual Studio Code y plugins para SQLite, Toast y Córdoba (Diagnóstico, Geolocalización, SQLite, Cámara, Archivos), en este caso en particular, se utilizó NodeJS para construir la CLI de Ionic, como también para la ejecución de JavaScript en el lado del servidor.

Los resultados de las pruebas evidencian el nivel de eficiencia, portabilidad, rendimiento, funcionalidad, mantenibilidad, usabilidad y fiabilidad de la aplicación móvil a instalar en el sistema operativo Android; en el backend se desarrolló un segundo componente para la sincronizar y visualizar los datos, la actualización de las herramientas Open Source y la refactorización le darán mejor velocidad de carga y en el peso de la aplicación.

4.1.3 Antecedentes regionales

En el contexto regional, Pérez et al., (2021) en el artículo titulado “Aplicación para promover la cultura y el turismo de la comunidad de Atanquez”, exponen el desarrollo de una aplicación multiplataforma para promover el turismo cultural. En relación con la metodología de investigación es de tipo cuantitativo y diseño experimental teniendo en cuenta las variables tecnología y servicio de información; en cuanto a la metodología de desarrollo ágil se empleó XP en combinación con el método de trabajo de SCRUM, comprendida por cinco (5) etapas: análisis del sistema, diseño, codificación, pruebas y lanzamiento. En la primera etapa se analizó la información referente al turismo y cultura del corregimiento de Atanquez, lo que sirvió de base para determinar los requerimientos del sistema. Igualmente, realizaron un trabajo de campo donde se aplicó una encuesta para analizar la viabilidad en el uso de la aplicación, encontrándose un buen nivel de aceptación entre la población. No se mencionaron las tecnologías utilizadas en el desarrollo de la aplicación.

De la misma manera, se encuentra el artículo titulado “Aplicación móvil para la ubicación de buses y usuarios de transporte público colectivo”, en el cual Aristizábal et al., (2020), desarrollaron un prototipo de sistema embebido que cuenta con un sensor de movimiento y una placa de Arduino, incluyendo también el desarrollo de una aplicación móvil para Android bajo la metodología ágil SCRUM, empleando como herramientas Ionic, Angular y Firebase a fin de obtener y gestionar datos en tiempo real, complementario a estas tecnologías se encuentra Google Maps que permite una mejor gestión de la información de las rutas. El aporte de este trabajo es la metodología y los sprints utilizados, esto es, los ciclos de ejecución para la creación del aplicativo.

Los autores concluyen que es posible mejorar el servicio de transporte público colectivo al optimizar el tiempo de espera de los usuarios porque se mantienen informados respecto en qué punto del trayecto se encuentran los buses, de modo que puedan establecer un tiempo de llegada a la estación y una forma más segura de abordarlo, al estar un menor tiempo de espera en la estación o paradero; no se hace referencia alguna sobre velocidad y rendimiento del aplicativo, aunque se debe suponer un tiempo de respuesta favorable para el usuario al utilizar Google Maps para obtener información sobre la ruta.

4.2 Marco teórico

4.2.1 Reactive Architecture

Esta teoría se basa en la creación de aplicaciones que responden de manera eficiente a los eventos, son resilientes a fallas, y escalables. En el contexto de React Native, una librería de JavaScript utilizada para desarrollar aplicaciones móviles, la teoría de la arquitectura reactiva se enfoca en la creación de interfaces de usuario que respondan rápidamente a las interacciones del usuario, manteniendo la consistencia en la experiencia a través de diferentes plataformas móviles.

El proyecto explora cómo mediante React Native, puede mejorar la experiencia del cliente al proporcionar una interfaz de usuario fluida y receptiva. Esto es crucial

para garantizar que el aplicativo móvil sea efectivo en su rol de apoyar el servicio al cliente y fomentar las ventas.

4.2.2 Node.JS

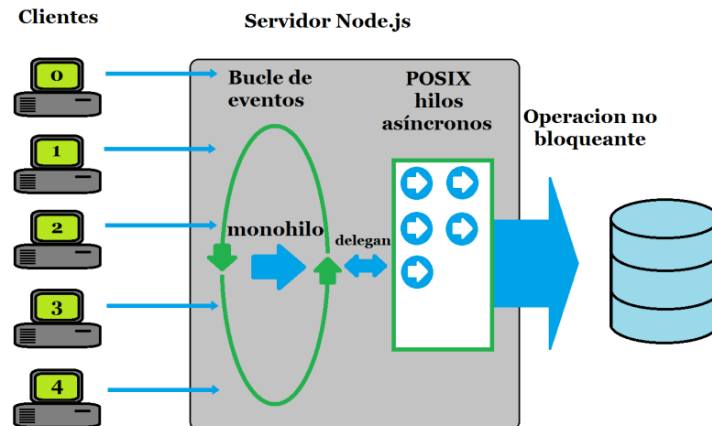
Node.JS es un entorno de ejecución para JavaScript del lado del servidor sin bloqueos y por eventos, construido con el motor de JavaScript V8 de Chrome (Blancarte, 2017), de acuerdo con Íscar (2015), fue creado por Ryan Dahl en 2009 con el fin de realizar “un entorno de programación asíncrona que le permitiese ejecutar tareas concurrentes sin que se bloquearan entre sí” (pág. 13), programado en lenguaje C++, está dirigido a la construcción de servidores principalmente, aunque en la actualidad se conocen otros usos, beneficios, ventajas y desventajas, caracterizado por:

Es un programa monohilo que puede ser multihilo con la librería Clúster, se pueden crear programas con su único hilo, no obstante, es escalable, porque es posible agregar más hilos y máquinas, aunque esto no implica necesariamente grandes cambios. Su arquitectura se halla comprendida por cinco (5) capas y está orientada a eventos, la primera y tercera capas se encargan funcionamiento básico y avanzado, mientras que las capas de interfaz y de abstracción binaria facilitan la comunicación, el acceso entre módulos y aplicaciones con las dependencias, y gestionar los cambios en la interfaz; la última capa contiene las aplicaciones desarrolladas por los usuarios. Esta estructura favorece la compatibilidad entre versiones.

La ejecución es asíncrona, es decir que las llamadas y funciones se ejecutan de forma secuencial, incluso si las anteriores aún no han terminado, lo que reduce el tiempo de ejecución y, por tanto, el tiempo de espera, lo permite gestionar entradas y salidas (E/S), evitando así cuellos de botella en el servidor, al atender varias peticiones como se aprecia en la figura 1. Aunque se observa como una ventaja, esto genera un obstáculo, porque no es capaz de devolver valores, lo cual requiere de distintas técnicas y cambios en el estilo de programación para adaptarlo al nuevo sistema, por lo que se emplea para desarrollar determinadas

aplicaciones: Chat, Gestión de base de datos, datos en streaming, servidor proxy y balanceador de carga; servidor SMTP, servidor DNS y juegos online multijugador (Íscar, 2015).

Figura 1. Servidor Node.JS



Tomada del trabajo de grado titulado "NODE.JS Do's and Don'ts", de Íscar (2015).

La investigación tomo como base Node.js para desarrollar un backend eficiente que respalde las funciones críticas del aplicativo móvil, como la gestión de usuarios, el procesamiento de transacciones y la integración con otros servicios. Esto sería clave para garantizar que la aplicación pueda manejar la carga operativa requerida por la empresa, manteniendo al mismo tiempo un rendimiento óptimo.

4.2.3 Metodología de desarrollo Agíl XP

Se centra en mejorar la calidad del software y la capacidad de adaptación a los cambios a través de prácticas como la programación en pareja, la integración continua, y la retroalimentación rápida. XP promueve la entrega de software funcional en iteraciones cortas y frecuentes, lo que permite a los equipos de desarrollo responder rápidamente a las necesidades del cliente.

En el desarrollo del aplicativo móvil, la adopción de XP es fundamental para garantizar que la aplicación evolucione de manera rápida y flexible según las necesidades cambiantes del negocio y del mercado. La investigación explora cómo la implementación de XP puede facilitar la creación de un aplicativo móvil que esté alineado con los objetivos estratégicos de ventas y servicio al cliente, asegurando al mismo tiempo la calidad y usabilidad del producto.

4.2.4 Teoría de Diseño Centrado en el Usuario para Aplicaciones Móviles

Esta teoría postula que el diseño de aplicaciones debe centrarse en las necesidades, preferencias y limitaciones de los usuarios finales. En el desarrollo de aplicaciones móviles, esto implica un enfoque iterativo donde los prototipos son continuamente probados y mejorados con base en la retroalimentación de los usuarios. El diseño centrado en el usuario se aplica en el desarrollo del aplicativo móvil para mejorar la satisfacción del cliente. Esto incluye la creación de una interfaz, la simplificación de procesos de ventas y atención al cliente, y la personalización de la experiencia del usuario en función de sus interacciones previas con la empresa.

4.3 Marco conceptual

Aplicación móvil, es una aplicación diseñada para ejecutar en dispositivos móviles (tableta o teléfono inteligente), con diferentes fines como el comercio móvil que facilitan el acceso a productos y servicios por parte de los clientes, quienes pueden elegir entre la variedad de métodos de pago (Herazo, 2022). De acuerdo con Florido (2016), las aplicaciones tienen buena aceptación por parte de la población, porque es posible acceder de forma más rápida debido a que el celular hoy día es una tecnología en constante uso por su fácil transporte, que permite la conexión a internet de forma inalámbrica, las primeras aplicaciones fueron introducidas en la tienda de iTunes en 2007, el éxito de estas fue tal que dio lugar a la aparición de nuevas empresas de desarrollo contándose entre ellas Android.

Frontend (ReactNative): se refiere al desarrollo de la interfaz de usuario (UI) de aplicaciones móviles utilizando la librería React Native. React Native es un

framework de código abierto creado por Facebook que permite a los desarrolladores construir aplicaciones móviles para iOS y Android utilizando JavaScript y React, una biblioteca de JavaScript para construir interfaces de usuario. El frontend con React Native incluye la creación de elementos como botones, listas, formularios, y la navegación dentro de la aplicación. Además, permite manejar la interacción del usuario con la aplicación, como gestos táctiles, desplazamientos, y transiciones entre diferentes pantallas.

Backend, es la parte de la aplicación móvil que el usuario no puede ver ni acceder, se encuentra en el lado del servidor y se encarga de la interacción de los datos a través de distintas acciones que ejerce el usuario desde las vistas creadas por el desarrollador de frontend (UOC, 2021). De acuerdo con Owius (como se cita en Tecnova, 2020), un backend móvil es un motor que recopila todas las peticiones efectuadas por los usuarios desde sus dispositivos móviles, su creación puede realizarse a nivel local o interno dentro y por la empresa; o en la nube con conectividad completa al servidor de aplicaciones y los usuarios; para lo cual debe tenerse en cuenta la autenticación y seguridad, escalabilidad y velocidad, estabilidad y a prueba de futuro (*future-proofing*).

Desarrollo ágil de software, es una metodología flexible, enfocada en el trabajo en equipo para proporcionar sistemas de software funcionales y rápidos en sus iteraciones, la cual nació en 2001 debido a los problemas presentados con los modelos en cascada; su objetivo es priorizar cuatro aspectos fundamentales: personas e interacciones; software en funcionamiento; colaboración con el cliente y la respuesta ante el cambio (Red Hat, 2022).

NODE JS, es un entorno de ejecución multiplataforma de código abierto, basado en JavaScript, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor V8, a partir del cual pueden construir aplicaciones web, de escritorio multiplataforma compatibles con los diferentes sistemas operativos (Desarrollo Web, 2022).

Postman, es un cliente HTTP mediante el cual se gestionan las peticiones realizadas a la aplicación, su uso se orienta mayormente a la realización y automatización de pruebas, además proporciona información sobre el comportamiento de la aplicación, indicando que debe ajustarse o mejorarse (Izquierdo, 2018).

XP (acrónimo de *Extreme Programming*), metodología ágil enfocada en la comunicación constante y la retroalimentación, flexible y de trabajo en equipo, que comprende cuatro etapas: Plantificación, diseño, codificación y pruebas (Orientación Universia, 2020).

4.4 Marco legal

Ley 1581 de 2012 (Ley de Protección de Datos Personales): Esta ley establece las disposiciones generales para la protección de datos personales en Colombia. Define los principios que deben seguirse en el tratamiento de datos personales y establece los derechos de los titulares de la información. En el contexto del desarrollo de un aplicativo móvil, es fundamental garantizar que todos los datos personales recopilados, procesados y almacenados estén protegidos conforme a la Ley 1581 de 2012. Esto incluye implementar medidas de seguridad adecuadas y obtener el consentimiento informado de los usuarios para el uso de sus datos. La investigación debe asegurarse de que el aplicativo móvil cumpla con esta ley para proteger la privacidad de los usuarios y evitar sanciones legales.

Ley 527 de 1999 (Ley de Comercio Electrónico): Esta ley regula el comercio electrónico en Colombia, proporcionando un marco legal para las transacciones realizadas a través de medios electrónicos, incluyendo la validación y el reconocimiento de la firma digital y la equivalencia funcional entre documentos electrónicos y documentos físicos. Para un aplicativo móvil enfocado en la venta y servicio al cliente, la Ley 527 de 1999 es crucial, ya que regula las transacciones comerciales electrónicas que se realizarán a través de la aplicación. Esto incluye la validez de los contratos y las transacciones, así como el reconocimiento de las firmas electrónicas utilizadas para validar las compras y otras operaciones. La

investigación debe explorar cómo el aplicativo móvil puede integrarse plenamente dentro del marco legal del comercio electrónico en Colombia.

Ley 1341 de 2009 (Ley de TIC): Esta ley establece el marco legal para el desarrollo y promoción de las Tecnologías de la Información y las Comunicaciones (TIC) en Colombia. Su objetivo es garantizar el acceso universal y eficiente a las TIC y promover su uso para el desarrollo social y económico del país. La Ley 1341 de 2009 es relevante para la creación de un aplicativo móvil, ya que promueve el acceso y uso de las TIC para impulsar el desarrollo económico. Esta ley también fomenta la innovación tecnológica y el emprendimiento digital, aspectos clave para la investigación, al considerar cómo el aplicativo móvil puede contribuir al crecimiento del comercio digital y la inclusión tecnológica en el contexto empresarial.

Ley 1480 de 2011 (Estatuto del Consumidor): Esta ley establece los derechos y deberes de los consumidores en Colombia, así como las obligaciones de los proveedores y expendedores de bienes y servicios. Incluye disposiciones sobre publicidad, contratos, garantías, y protección contra prácticas comerciales abusivas. La Ley 1480 de 2011 es fundamental para el desarrollo de un aplicativo móvil, ya que establece las obligaciones que la empresa debe cumplir para garantizar la protección de los derechos del consumidor. Esto incluye asegurar la transparencia en la información proporcionada, respetar los términos y condiciones de las transacciones, y ofrecer garantías adecuadas para los productos y servicios vendidos a través de la aplicación.

5. Metodología

5.1 Tipo de investigación

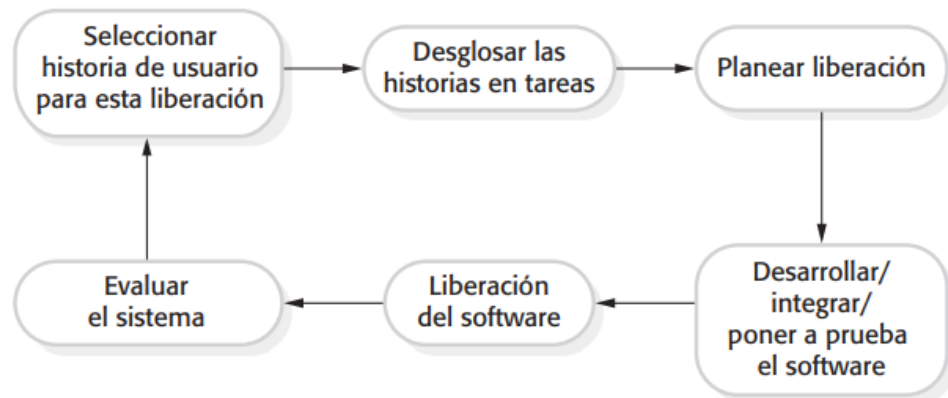
El proyecto se llevará a cabo bajo un tipo de investigación aplicada con un enfoque descriptivo, según lo expuesto por Obando (2024), donde explica que “se orienta a resolver un problema práctico y específico dentro del contexto empresarial, buscando desarrollar una solución tecnológica que optimice los procesos de ventas y mejore significativamente la experiencia del cliente”.

El enfoque descriptivo de la investigación permitirá identificar y detallar las características, funcionalidades y requerimientos necesarios para la implementación exitosa del aplicativo móvil. A través de la recolección y análisis de datos relevantes, se describirá cómo el aplicativo puede integrarse en las operaciones existentes de la empresa, facilitando la automatización de procesos, mejorando la comunicación con los clientes, y generando un impacto positivo en las ventas y la fidelización de clientes.

5.2 Metodología de desarrollo

Teniendo en cuenta lo anterior, el tamaño del proyecto y la finalidad del aplicativo móvil en el presente trabajo se abordará la metodología de desarrollo ágil XP, por un lado, porque permite la integración de las partes del aplicativo (Frontend y Backend) tras haber sido desarrollado por diferentes programadores y ponerlo a prueba (Sommerville, 2011), también porque son los usuarios (la empresa y sus clientes) quienes proporcionarán información relevante sobre las funcionalidad requerido por la aplicación, esto comprenderían las historias de usuario que se desglosa en las tareas a realizar por parte del desarrollador o grupo de desarrolladores como se aprecia en la figura 2.

Figura 2. Ciclo de liberación de la Programación Extrema (XP)



Fuente: Tomada del libro “Ingeniería de Software”, de Sommerville (2011).

Como se puede ver en la figura 2, el cliente (la empresa Uuzaji y sus clientes) participa en el desarrollo “y es responsable de definir las pruebas de aceptación para el sistema”, respecto a las pruebas estas se realizan con el primer prototipo de la aplicación y en el desarrollo de pruebas incrementales a partir de escenarios, teniendo en cuenta estas definiciones y los objetivos del proyecto, se determinan las fases para su ejecución:

1. **Fase de Planificación**, en la cual se determina un cronograma, equipo y forma de trabajo, elegir las herramientas y tecnologías que apoyarán el trabajo con ReactNative para el desarrollo del aplicativo y conocer los requerimientos del aplicativo.
2. **Fase de Gestión**, definir un lugar de encuentro con el cliente a fin de mantener una comunicación y participación constante, gestionar la carga de trabajo, revisar tanto los procesos realizados en la Programación Extrema como los avances efectuados en el aplicativo, por ejemplo, estudiar las

historias de usuario y definir sobre cuál de ellas se trabajará en un tiempo determinado, etc. (Raebum, 2022)

3. **Fase de Diseño**, se definirá en esta fase la herramienta de simulación, así como el conjunto de peticiones para la ejecución de pruebas del Frontend.
4. **Fase de Codificación**, se inicia el diseño de los módulos del Frontend trabajando con ReactNative.
5. **Fase de Pruebas**, se realiza la validación de los resultados obtenidos en las pruebas, se aplican las mejoras, se analiza la información final y se describen resultados claros respecto a funcionalidad, mantenimiento, usabilidad, etc.

6. Cronograma de actividades

Tabla 1. Cronograma de actividades

Tiempo	Mes 1	Mes 2	Mes 3	Mes 4
Definir los requerimientos funcionales para el desarrollo del Frontend que permitan el funcionamiento del aplicativo móvil.				
Diseñar los módulos del Frontend para el funcionamiento del aplicativo móvil empleando el entorno de ejecución de ReactNative.				
Establecer la ejecución del Frontend mediante una herramienta de simulación.				

Fuente: elaboración propia.

7. Resultados

7.1 Cumplimiento objetivo 1: Requisitos funcionales para el desarrollo del Frontend que permitan el funcionamiento del aplicativo móvil.

7.1.1 Requerimientos Funcionales

Se detallan los requerimientos funcionales que fueron identificados como esenciales para la construcción del Frontend del aplicativo, asegurando que cada uno de ellos esté alineado con las necesidades operativas y de experiencia del usuario establecidas en la fase inicial del proyecto.

Este análisis comprende la especificación de las características visuales, la interacción del usuario, la usabilidad, y los componentes clave que integran la interfaz de usuario, garantizando que el aplicativo móvil sea intuitivo, eficiente y eficaz en su propósito de apoyar las estrategias de ventas y mejorar el servicio al cliente. Cada requerimiento ha sido cuidadosamente evaluado para asegurar su viabilidad técnica y su impacto positivo en la funcionalidad general del aplicativo, proporcionando una base sólida para el desarrollo e implementación exitosa del Frontend.

Tabla 2. Requisitos Funcionales

Requisito	Requerimiento	Descripción
RFA1	Autenticación de Usuarios	El sistema debe permitir a los usuarios registrarse e iniciar sesión de manera segura mediante credenciales.
RFA2	Navegación intuitiva	El aplicativo debe contar con una barra de navegación que permita a los usuarios

		acceder fácilmente a todas las secciones principales.
RFA3	Gestión de pedidos	Los usuarios deben poder crear, revisar, modificar y cancelar pedidos directamente desde la aplicación.
RFA4	Carrito de compras	El sistema debe permitir agregar productos al carrito, visualizarlo y proceder al pago desde el mismo.
RFA5	Notificaciones Push	El aplicativo debe enviar notificaciones push para alertar a los usuarios sobre promociones, actualizaciones de pedidos, y otros eventos relevantes.

Fuente: elaboración propia.

El cuadro de requerimientos funcionales como se muestra en la Tabla 2, proporciona una base estructurada y clara para guiar el desarrollo del Frontend del aplicativo móvil. Estos requerimientos son fundamentales para asegurar que la aplicación cumpla con las expectativas y necesidades tanto de la empresa como de sus usuarios finales. Al definir y documentar estos requerimientos de manera detallada, se garantiza que el equipo de desarrollo cuente con un marco de referencia sólido que permita la construcción de una interfaz de usuario intuitiva, segura y eficiente, capaz de optimizar las ventas y mejorar significativamente el servicio al cliente. Este enfoque detallado y planificado es esencial para el éxito del proyecto, asegurando que cada aspecto del desarrollo esté alineado con los objetivos estratégicos y operativos de la empresa.

7.1.2 Historias de usuario

Estas historias de usuario se derivan directamente de los requerimientos funcionales previamente establecidos y representan la perspectiva del usuario final. Cada historia describe de manera clara lo que el usuario quiere lograr dentro de la aplicación y proporciona un contexto para el desarrollo y la validación de las funcionalidades correspondientes. Estas historias de usuario son esenciales para guiar el proceso de desarrollo ágil, asegurando que el producto final cumpla con las expectativas de los usuarios y satisfaga sus necesidades de manera efectiva.

Tabla 3. Historias de usuario

ID de Usuario	Historia de Usuario	Descripción	Requerimiento Asociado
HU-01	Como usuario, quiero registrarme e iniciar sesión de manera segura, para poder acceder a mi cuenta personal en la aplicación.	Esta historia de usuario cubre la necesidad de autenticación, permitiendo que los usuarios creen una cuenta y accedan a sus datos personales de forma segura.	(Autenticación de Usuarios)
HU-02	Como usuario, quiero navegar fácilmente entre las diferentes secciones de la aplicación, para encontrar rápidamente lo que necesito.	Esta historia de usuario se enfoca en la navegación intuitiva, garantizando que los usuarios puedan acceder fácilmente a todas las secciones principales de la aplicación.	(Navegación Intuitiva)
HU-03	Como usuario, quiero poder crear, revisar, modificar y cancelar pedidos, para gestionar mis compras de manera eficiente.	Permite a los usuarios realizar todas las acciones necesarias relacionadas con la gestión de pedidos dentro de la aplicación, proporcionando flexibilidad y control.	(Gestión de Pedidos)
HU-04	Como usuario, quiero agregar productos a un carrito de compras, para poder revisar y proceder al pago cuando esté listo.	Esta historia cubre la funcionalidad del carrito de compras, permitiendo a los usuarios gestionar sus selecciones de productos antes de realizar una compra.	(Carrito de Compras)
HU-05	Como usuario, quiero recibir notificaciones push sobre promociones y actualizaciones de mis pedidos, para mantenerme informado.	Asegura que los usuarios estén informados en tiempo real sobre eventos importantes, como promociones y el estado de sus pedidos.	(Notificaciones Push)

Fuente: elaboración propia.

Como se muestra en la tabla 3, a partir de los requerimientos funcionales proporciona un enfoque centrado en el usuario para guiar el diseño y desarrollo del

Frontend del aplicativo móvil. Estas historias de usuario traducen las necesidades y expectativas de los usuarios en descripciones claras y accionables, lo que permite al equipo de desarrollo construir una aplicación que no solo cumple con los objetivos técnicos, sino que también ofrece una experiencia de usuario óptima. Al articular claramente lo que los usuarios desean lograr, estas historias sirven como una herramienta fundamental para priorizar funcionalidades, facilitar la comunicación entre los miembros del equipo, y asegurar que el producto final esté alineado con las metas estratégicas de la empresa y las demandas del mercado. Este enfoque orientado al usuario es clave para el éxito del proyecto y para la creación de una aplicación que realmente responda a las necesidades de su audiencia.

7.2 Cumplimiento objetivo 2: Diseño de los módulos para Frontend que permitan el funcionamiento del aplicativo móvil empleando el entorno de ejecución de ReactNative.

7.2.1 Codificación de los módulos

Los módulos clave diseñados para el Frontend de la aplicación, destacando cómo cada uno de ellos ha sido conceptualizado y estructurado utilizando React Native, una tecnología que permite la creación de interfaces de usuario eficientes y nativas para dispositivos móviles. Este enfoque modular asegura que cada componente esté optimizado para la funcionalidad específica que soporta, facilitando no solo la implementación inicial, sino también la escalabilidad y mantenimiento futuro del aplicativo.

Se abordan las decisiones de diseño que guían la interacción del usuario con la aplicación, garantizando una experiencia fluida y coherente en todas las plataformas móviles. Este diseño modular es fundamental para asegurar que el aplicativo móvil cumpla con los requisitos operativos y de usabilidad necesarios para apoyar eficazmente las estrategias de ventas y mejorar el servicio al cliente, de acuerdo a los requerimientos funcionales establecidos.

Figura 3. CartScreen UZAJI

```
CartScreen.js X
screens > CartScreen.js > @styles > testcounted
1 import { View, Text, StyleSheet, TouchableOpacity, ScrollView, } from "react-native";
2 import React, { useEffect, useState } from "react";
3 import ArrowGoBack from "../components/arrowGoBack";
4 import { Icon, Image } from "@rneui/themed";
5 import { useNavigation } from "@react-navigation/native";
6 import { useDispatch, useSelector } from "react-redux";
7 import { selectProducts } from "../slice/productSlice";
8 import { removeFromCart, selectCartItems, selectCartTotal, } from "../screens/slice/cartSlice";
9
10 export default function CartScreen() {
11   const navigation = useNavigation();
12   const cartItems = useSelector(selectCartItems);
13   const cartTotal = useSelector(selectCartTotal);
14   const products = useSelector(selectProducts);
15   const [groupedItems, setGroupedItems] = useState({});
16   const dispatch = useDispatch();
17
18   useEffect(() => {
19     const items = cartItems.reduce((group, item) => {
20       if (group[item.id]) {
21         group[item.id].push(item);
22       } else {
23         group[item.id] = [item];
24       }
25       return group;
26     }, {});
27     setGroupedItems(items);
28   }, [cartItems]);
29
30   return (
31     <View style={styles.viewcontainer}>
32       <ArrowGoBack />
33       <Text>CartScreen</Text>
34       <View>
35         <Text style={styles.textyourcart}>Your Cart </Text>
36         <Text style={styles.textyourcart}>{products.name} </Text>
37       </View>
38       <View style={styles.viewcontainerdelivery}>
39         <Image
40           source={require("../assets/motorcycle-delivery2.jpeg")}
41           style={styles.imgdelivery}
42         />
43         <Text style={styles.textimedeliver}>Deliver in 28-30 minutes</Text>
44         <TouchableOpacity>
45           <Text style={styles.textchange}>Change</Text>
46         </TouchableOpacity>
47       </View>
48       <ScrollView
49         contentContainerStyle={styles.scrollviewcontentcontainer}
50         style={styles.stylescrollviewcontentcontainer}
51       >
52         {Object.entries(groupedItems).map(([key, items]) => {
53           let itemsp = items[0];
```

La importancia de realizar el código fuente de CartScreen radica en su capacidad para proporcionar una base sólida y flexible para una aplicación que, presumiblemente, desempeñará un papel crucial en un ámbito específico, ya sea en la gestión de datos, la visualización de información o la interacción con los usuarios.

```
54     return (  
55       <View key={key} style={styles.viewitemproducts}>  
56         <Text style={styles.textcounted}>{items.length} x</Text>  
57         <Image source={itemsp.image} style={styles.imgproductscounted} />  
58         <Text style={styles.textname}>{itemsp.name}</Text>  
59  
60         <Text style={styles.textprice}>{itemsp.price}</Text>  
61  
62         <TouchableOpacity  
63           style={styles.touchminus}  
64           onPress={() => dispatch(removeFromCart({ id: itemsp.id })))  
65         >  
66           <Icon  
67             name="minus-circle"  
68             type="material-community"  
69             color="orange"  
70           />  
71         </TouchableOpacity>  
72       </View>  
73     );  
74   ]]  
75 </ScrollView>  
76  
77 /* Total */  
78 <View style={styles.viewtotalcontainer}>  
79   <View style={styles.viewtext}>  
80     <Text style={styles.textsubtotalfee}>Subtotal</Text>  
81     <Text style={styles.textsubtotalfee}>${cartTotal}</Text>  
82   </View>  
83   <View style={styles.viewtext}>  
84     <Text style={styles.textsubtotalfee}>Delivery fee</Text>  
85     <Text style={styles.textsubtotalfee}>${2}</Text>  
86   </View>  
87   <View style={styles.viewtext}>  
88     <Text style={styles.texttotal}>Orden Total</Text>  
89     <Text style={styles.texttotal}>${2*cartTotal}</Text>  
90   </View>  
91   <TouchableOpacity  
92     style={styles.touchplaceorder}  
93     onPress={() => navigation.navigate("orderpreparing")}  
94   >  
95     <Text style={styles.textplaceorder}>Place Order</Text>  
96   </TouchableOpacity>  
97 </View>  
98 </View>  
99 );  
100 }  
101 const styles = StyleSheet.create({  
102   viewcontainer: {  
103     flex: 1,  
104   },  
105   textyourcart: {  
106     fontFamily: "bold"
```

Fuente: elaboración propia.

Una de las principales razones para realizar el código fuente de CartScreen es la personalización. Las aplicaciones genéricas rara vez satisfacen todas las necesidades específicas de una organización o usuario. Al desarrollar un código fuente propio, se tiene la libertad de crear funcionalidades a medida que respondan a las demandas particulares del proyecto.

Figura 4. HomeScreen UUAZI

```
HomeScreenjs X
screens > HomeScreenjs > ...
1 import { StyleSheet, Text, View } from "react-native";
2 import React from "react";
3 import { SafeAreaView } from "react-native-safe-area-context";
4 import { Icon, Input } from "@rneui/themed";
5
6 //mis imports
7 import { Frutas, Verduras } from "../components/constants/ListCategories";
8 import PresentationRow from "../components/presentationRow";
9
10 export default function HomeScreen() {
11   return (
12     <SafeAreaView>
13       { /* menu categorias */}
14       <View style={styles.viewmenucategories}>
15         { /* tarjeta de categorias */}
16         {[Frutas, Verduras].map((item, ListCategories) => {
17           return (
18             <PresentationRow
19               key={ListCategories}
20               title={item.title}
21               products={item.products}
22             />
23           );
24         })}
25       </View>
26     </SafeAreaView>
27   );
28 }
29
30 const styles = StyleSheet.create({
31   safeareaview: {
32     backgroundColor: "white",
33     flex: 1,
34   },
35   viewcontainerbar: {
36     paddingHorizontal: 30,
37     paddingVertical: 10,
38   },
39   viewsearchbar: {
40     paddingVertical: 20,
41     borderWidth: 1,
42     borderColor: "gray",
43     borderRadius: 20,
44     flexDirection: "row",
45     width: "100%",
46     flex: 1,
47   },
48   input: {
49     paddingLeft: 10,
50     textAlign: "center",
51     right: 11,
52   },
53   inputcontainr: {
54     borderColor: "transparent",
55   },
56   viewmenucategories: {
57     height: 650,
58     display: "flex",
59     flexDirection: "column",
```

Fuente: elaboración propia.

El diseño del código fuente de HomeScreen, es un proceso meticuloso y estratégico que busca optimizar la funcionalidad, escalabilidad, mantenimiento y seguridad de la aplicación. Que garantiza que la aplicación no solo cumpla con los requerimientos actuales, sino que también esté preparada para adaptarse a futuras demandas y tecnologías emergentes.

Figura 5. PlaceOrdering UUZAJI

```
PlaceOrdering.js X
screens > PlaceOrdering.js > ...
1 import { StyleSheet, Text, View } from "react-native";
2 import React, { useEffect } from "react";
3 import { useNavigation } from "@react-navigation/native";
4 import { Image } from "@rneui/themed";
5 import ArrowGoBack from "../components/arrowGoBack";
6
7 export default function PlaceOrdering() {
8   const navigation = useNavigation();
9   useEffect(() => {
10     setTimeout(() => {
11       //move to delivery screen
12       navigation.navigate("delivery");
13     }, 3000);
14   }, []);
15   return (
16     <>
17     <ArrowGoBack />
18
19     <View style={styles.viewdeliverygif}>
20       <Image
21         source={require("../assets/delivery.gif")}
22         style={styles.imgdelivery}
23       />
24     </View>
25   </>
26 );
27 }
28 const styles = StyleSheet.create({
29   viewdeliverygif: {
30     flex: 1,
31     display: "flex",
32     justifyContent: "center",
33     alignItems: "center",
34   },
35   imgdelivery: {
36     height: 390,
37     width: 380,
38   },
39 });
40
```

Fuente: elaboración propia.

PlaceOrdering, como sistema destinado a procesar órdenes de manera rápida y precisa, debe ser diseñado para manejar múltiples aspectos críticos, tales como la entrada de datos del usuario, la validación de las órdenes, la integración con sistemas de pago, la gestión de inventarios, y la notificación al usuario.

Figura 6. ProductScreen UZAJI

```
ProductsScreenjs X
screens > ProductsScreenjs > ProductsScreen
1 import { ScrollView, StyleSheet, Text, View } from "react-native";
2 import React, { useEffect } from "react";
3 import { useRoute } from "@react-navigation/native";
4 import { Image } from "@rneui/themed";
5 import ProductsRow from "../components/ProductsRow";
6 import ShopCart from "../components/ShopCart";
7 import ArrowGoBack from "../components/arrowGoBack";
8 import { useDispatch } from "react-redux";
9 import { setProduct } from "../slice/productSlice";
10
11 export default function ProductsScreen() {
12   const { params } = useRoute();
13   const dispatch = useDispatch();
14
15   let item = params;
16   console.log("item : ", item);
17
18   useEffect(() => {
19     if (item && item.id) {
20       dispatch(setProduct({...item}))
21     }
22   }, []);
23   return (
24     <>
25     <ArrowGoBack />
26
27     <ShopCart />
28
29     <ScrollView>
30       <View style={styles.viewimg}>
31         <Image style={styles.img} source={item.image} />
32       </View>
33
34       /* presentation */
35       <View style={styles.viewcontainerpresentation}></View>
36       /* menu */
37       <View style={styles.viewmenu}>
38         /* listado de productos */
39         {item.itemproducts.map((iproducts, listCategories) => (
40           <ProductsRow item={{ ...iproducts }} key={listCategories} />
41         ))}
42       </View>
43     </ScrollView>
44
45     <Text style={{ textAlign: "center" }}>ProductsScreen</Text>
46   </>
47 );
48 }
49
50 const styles = StyleSheet.create({
51   viewimg: {
52     position: "relative",
53   },
54   img: {
55     width: "100%",
56     height: 250,
57   },
58   arrowleft: {
59     zIndex: 50,
60   },
61 });
```

Fuente: elaboración propia.

Uno de los aspectos esenciales en el diseño del código fuente de ProductScreen es la modularidad, esta implica dividir el código en componentes independientes que se pueden desarrollar, probar y mantener de manera autónoma, lo cual es fundamental para gestionar la complejidad de un sistema que maneja grandes volúmenes de datos relacionados con productos.

Figura 7. Welcome UUZAJI

```
WelcomeScreen.js X
screens > WelcomeScreen.js > ...
1 import { SafeAreaView, StyleSheet, Text, View } from "react-native";
2 import React from "react";
3 import { Button, Image } from "@rneui/themed";
4 import { useNavigation } from "@react-navigation/native";
5
6 export default function WelcomeScreen() {
7   const navigation = useNavigation();
8   return (
9     <SafeAreaView style={styles.safeareaview}>
10
11       <View style={styles.viewwelcome}>
12         <Text style={styles.textwelcome}>Bienvenidos a Winkel</Text>
13
14         <View style={styles.viewimg}>
15           <Image
16             source={require("../assets/UuzajiconUzi.png")}
17             style={styles.img}
18           />
19         </View>
20         <Button
21           onPress={() => navigation.navigate("login")}
22           buttonStyle={styles.button}
23           title={"Empezemos"}
24           titleStyle={styles.btntitle}
25         />
26       </View>
27     </SafeAreaView>
28   );
29 }
30
31 const styles = StyleSheet.create({
32   safeareaview: {
33     flex: 1,
34     justifyContent: "center",
35   },
36   viewwelcome: {
37     justifyContent: "space-around",
38     flex: 1,
39   },
40   textwelcome: {
41     textAlign: "center",
42     fontWeight: "bold",
43     fontSize: 25,
44   },
45   img: {
46     width: 370,
47     height: 200,
48   },
49 }
```

Fuente: elaboración propia.

Dado que WelcomeScreen es la primera interacción que un usuario tiene con la aplicación, es crucial que la pantalla de bienvenida sea atractiva, intuitiva y fácil de usar. Esto implica diseñar una interfaz de usuario (UI) que sea visualmente atractiva y que invite al usuario a explorar más la aplicación, mientras que el código subyacente debe soportar estas interacciones de manera rápida y eficiente.

Figura 8. Profile UUZAJI

```
Profile.js x
screens > accounts > Profile.js > ...
1  import React from "react";
2  import { StyleSheet, Text, View } from "react-native";
3  import { Button } from "@rneui/base";
4  import { closeSession } from "../../utils/actions";
5
6  export default function Profile() {
7    return (
8      <View
9        style={{
10         flex: 1,
11         justifyContent: "center",
12         alignItems: "center",
13         gap: 10,
14       }}
15     >
16       <Text>Perfil</Text>
17
18       <Text> Completar Perfil </Text>
19
20       <View style={{ justifyContent: "center", alignItems: "center", gap: 12 }}>
21         <Button title={"Completar perfil"} />
22
23         <Button
24           title={"cerrar sesion"}
25           onPress={() => {
26             closeSession();
27           }}
28         />
29       </View>
30     </View>
31   );
32 }
33
34 const styles = StyleSheet.create({});
35
```

Fuente: elaboración propia.

El código subyacente debe estar optimizado para soportar estas interacciones de manera rápida y eficiente, garantizando que las modificaciones realizadas en el perfil se reflejen inmediatamente. De esta manera, el diseño del perfil debe considerar aspectos como la accesibilidad para las personas, la coherencia visual en diferentes dispositivos y la capacidad de personalización de la interfaz para adaptarse a las preferencias individuales.

Figura 9. Login UUAJL

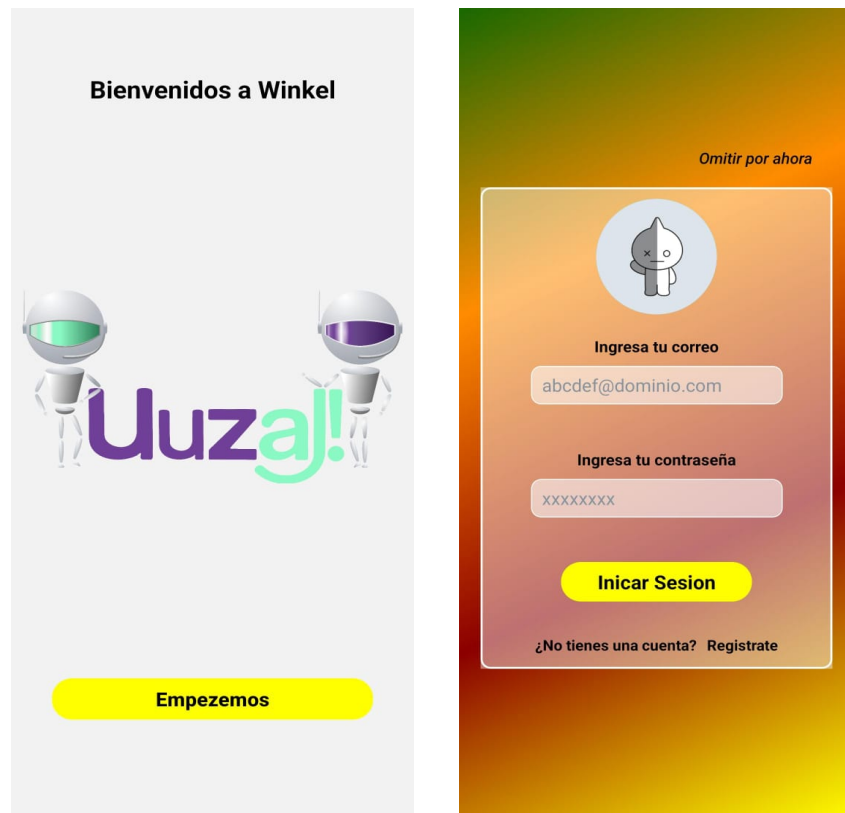
```
Login.js
screens > accounts > Login.js > styles > image
1 import React from "react";
2 import { ScrollView, StyleSheet, Text, View, Image } from "react-native";
3 import { BlurView } from "expo-blur";
4 import { Avatar } from "@react-native-material/core";
5 import { LinearGradient } from "expo-linear-gradient";
6
7 import LoginForm from "../../components/forms/LoginForm";
8 import { TouchableOpacity } from "react-native-gesture-handler";
9 import { useNavigation } from "@react-navigation/native";
10
11 export default function Login() {
12   const navigation = useNavigation();
13
14   return (
15     <View style={styles.container}>
16       <LinearGradient
17         colors={["darkgreen", "darkorange", "darkred", "yellow"]}
18         style={styles.linearGradient}
19         start={{ x: 0, y: 0 }}
20         end={{ x: 1, y: 1 }}
21       >
22         <ScrollView contentContainerStyle={styles.scrollview}>
23           <View style={styles.viewtouch}>
24             <TouchableOpacity onPress={() => navigation.navigate("home")}>
25               <Text style={styles.text}>Omitir por ahora</Text>
26             </TouchableOpacity>
27           </View>
28           <BlurView intensity={100} style={styles.blur}>
29             <View style={styles.viewlogin}>
30               <Avatar image={require("../../assets/VAN.png")} size={120} />
31               <LoginForm />
32             </View>
33           </BlurView>
34         </ScrollView>
35       </LinearGradient>
36     </View>
37   );
38 }
39
40 const styles = StyleSheet.create({
41   scrollview: {
42     flex: 1,
43     width: "100%",
44     height: "100%",
45     alignItems: "center",
46     justifyContent: "center",
47   },
48   viewlogin: {
49     width: 350,
50     height: 500,
51     borderColor: "white",
52     borderRadius: 10,
53     borderWidth: 2,
54     padding: 10,
55     alignItems: "center",
56   },
57   container: {
58     flex: 1,
59     backgroundColor: "#fff",
60   },
61 });
```

Fuente: elaboración propia.

El diseño del código fuente de LOGIN, es un componente esencial que permite a los usuarios acceder al aplicativo móvil, ya que es un proceso que exige una atención rigurosa a la seguridad, la eficiencia y la experiencia del usuario. Este actúa como la puerta de entrada al sistema, donde los usuarios deben autenticarse para acceder a las funcionalidades y datos del aplicativo. Por lo tanto, la estructura de del código fuente no solo debe garantizar que el proceso de autenticación sea seguro y confiable, sino que también debe ser intuitivo y rápido, proporcionando una experiencia de usuario sin fricciones.

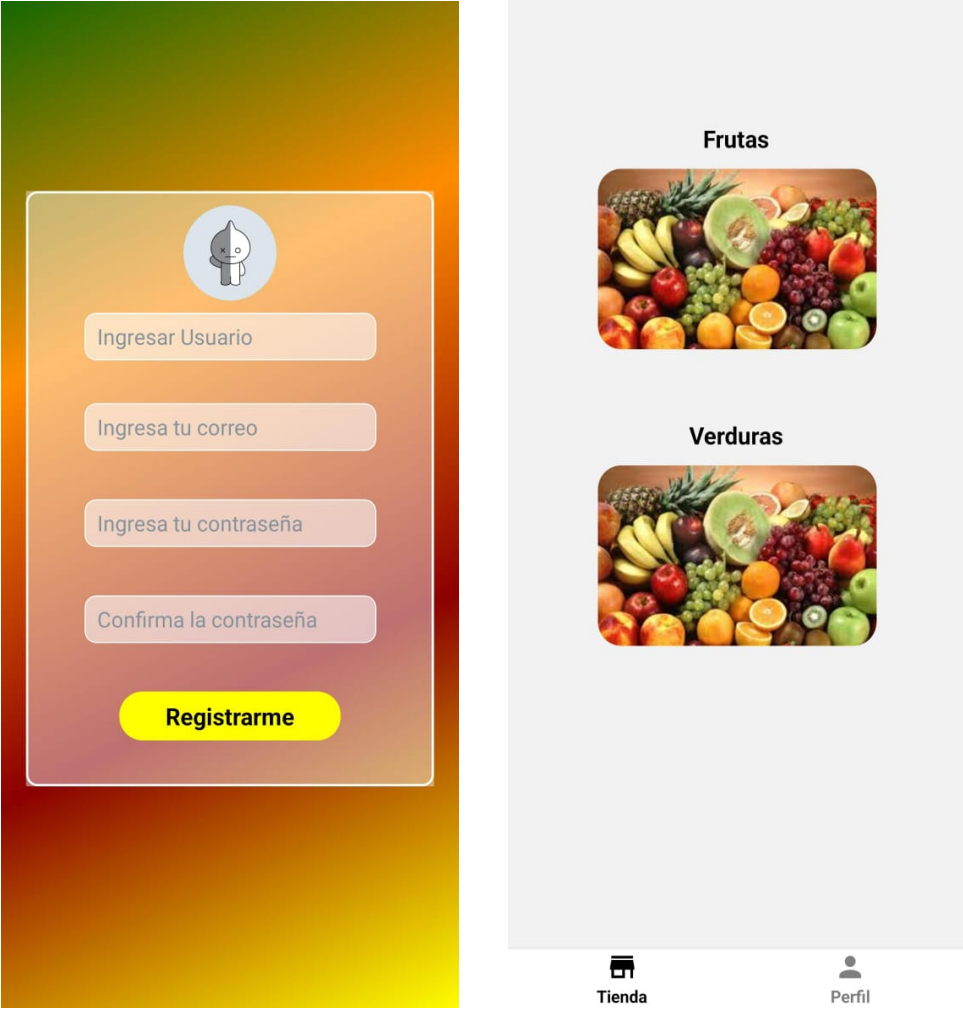
7.3 Cumplimiento objetivo 3: Establecer la ejecución del Frontend mediante una herramienta de simulación.

Figura 10. Bienvenida UUZAJI



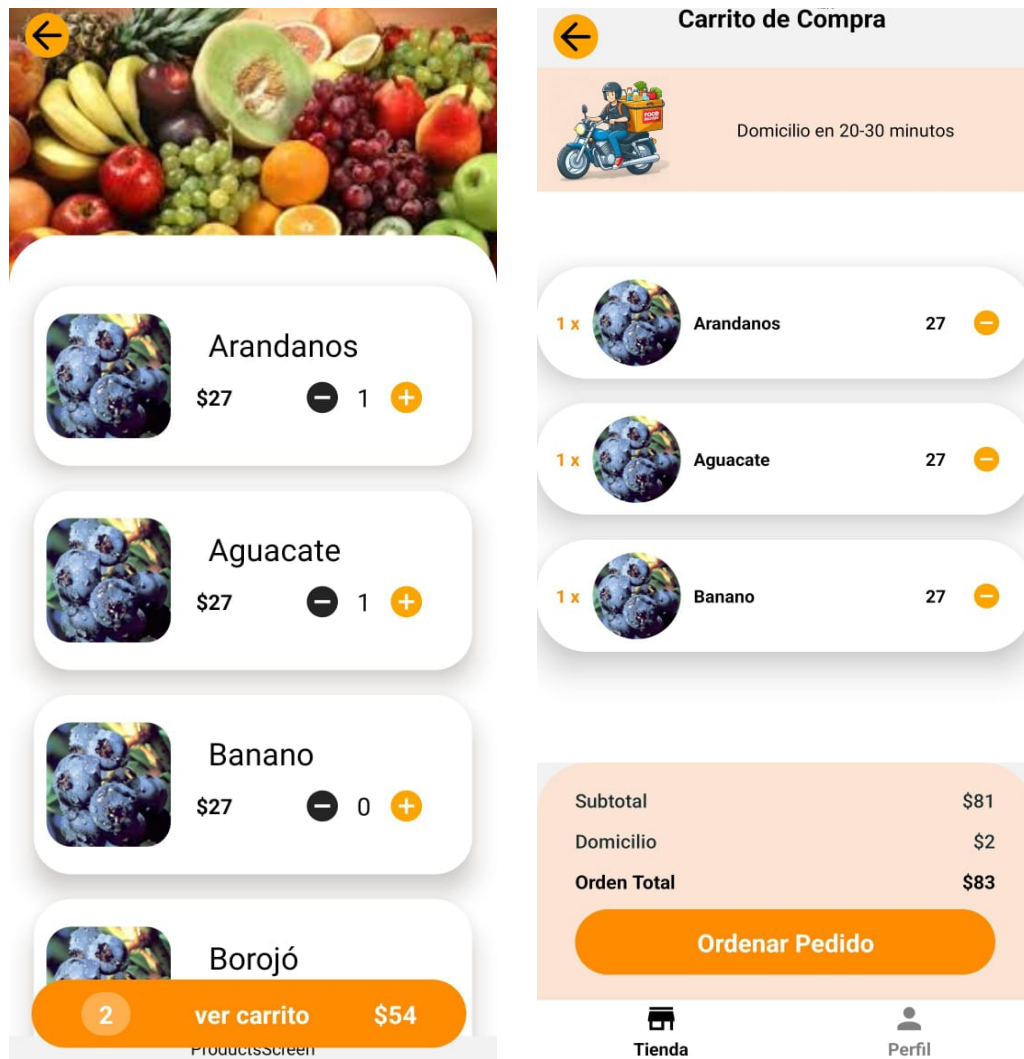
Fuente: elaboración propia.

Figura 12. Registrarse UUZAJI



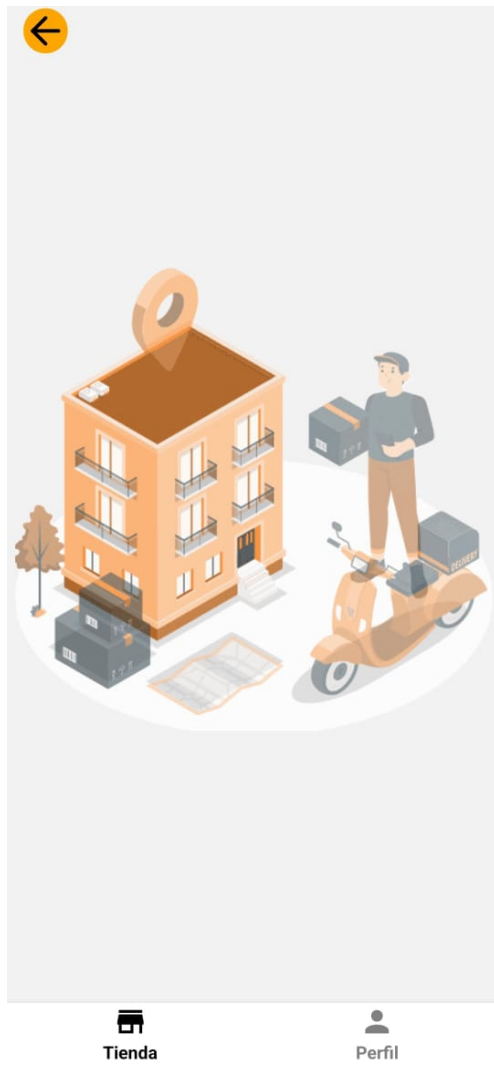
Fuente: elaboración propia.

Figura 15. Añadir al carrito - UUAZI



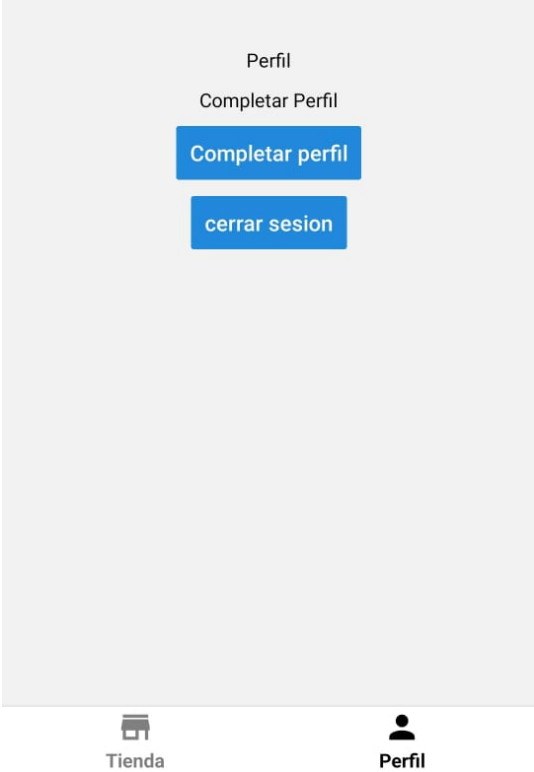
Fuente: elaboración propia.

Figura 17. Ubicación del domicilio - UUZAJI



Fuente: elaboración propia.

Figura 19. Cerrar sesión - UUZAJI



Fuente: elaboración propia.

Conclusiones

Lograr definir los requerimientos funcionales necesarios para el desarrollo del Frontend del aplicativo móvil es fundamental para establecer una base sólida que guíe la construcción de una interfaz de usuario que no solo es intuitiva y fácil de usar, sino que también está alineada con las necesidades operativas de la empresa. La correcta identificación y documentación de estos requerimientos ha asegurado que todas las funcionalidades clave del aplicativo estén orientadas a optimizar las ventas y mejorar significativamente el servicio al cliente.

El diseño de los módulos del Frontend utilizando el entorno de ejecución de React Native se completó con éxito, asegurando que cada componente de la aplicación esté optimizado para su funcionalidad específica. Estos módulos fueron diseñados para ser altamente escalables, permitiendo futuras expansiones y mejoras sin comprometer el rendimiento de la aplicación. El uso de React Native ha permitido crear una experiencia de usuario fluida y coherente en múltiples plataformas, garantizando que el aplicativo móvil brinde un soporte efectivo a las estrategias de ventas y ofrezca un servicio al cliente de alta calidad.

La implementación del aplicativo móvil se realizó conforme a los requerimientos funcionales y el diseño modular establecidos, logrando un producto final que cumple con los objetivos de mejorar las ventas y optimizar el servicio al cliente. El aplicativo móvil ha demostrado ser una herramienta eficaz para facilitar las transacciones y mejorar la interacción con los clientes, contribuyendo a la eficiencia operativa y a la satisfacción del usuario. Esta implementación ha permitido que la empresa se posicione de manera competitiva en el mercado, ofreciendo una solución tecnológica que responde a las necesidades actuales de los consumidores.

Recomendaciones

1. Para garantizar la escalabilidad y el desempeño óptimo del aplicativo, se recomienda implementar herramientas de monitoreo y análisis continuo que permitan evaluar el comportamiento del usuario, la carga del sistema y el rendimiento general de la aplicación.
2. Se recomienda que la arquitectura del sistema esté diseñada desde el principio con la escalabilidad en mente. Esto incluye la utilización de tecnologías y servicios en la nube que permitan ajustar los recursos de manera flexible según la demanda.
3. Se recomienda diseñar el aplicativo con una estructura modular que facilite la incorporación de nuevas funcionalidades y mejoras sin afectar el rendimiento o la estabilidad del sistema.

Referencias

- Aristizábal, C., Rojas, J., & Ramírez, T. (2020). Aplicación móvil para la ubicación de buses y usuarios de transporte público colectivo. *IX Jornada de Ciencia y Tecnología y I Virtual de la Universidad Popular del Cesar Sede Sabanas, Valledupar*. Obtenido de https://www.researchgate.net/publication/345765419_Aplicacion_movil_para_la_ubicacion_de_buses_y_usuarios_de_transporte_publico_colectivo
- Bangare, S., Gupta, S., Dalal, M., & Inamdar, A. (2016). Using Node.JS to Build High Speed and Scalable BAcKend Database Server. *International Journal of Research in Advent Technology*. Obtenido de https://www.researchgate.net/profile/Sunil-Bangare-2/publication/301788361_Using_NodeJs_to_Build_High_Speed_and_Scalable_Backend_Database_Server/links/57285d6c08aee491cb416ad6/Using-NodeJs-to-Build-High-Speed-and-Scalable-Backend-Database-Server.pdf
- Blancarte, O. (2017). *Introducción a NodeJS (JavaScript del lado del Servidor)*.

Obtenido de [https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/#:~:text=Que%20es%20NodeJS%20\(Seg%C3%BAAn%20su%20creador\)%3A&text=js%20usa%20un%20modelo%20de,c%C3%B3digo%20abierto%20en%20el%20mundo.](https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/#:~:text=Que%20es%20NodeJS%20(Seg%C3%BAAn%20su%20creador)%3A&text=js%20usa%20un%20modelo%20de,c%C3%B3digo%20abierto%20en%20el%20mundo.)

Cobo Yara, G., & Barrios Vega, J. (2024). Desarrollo de una aplicación móvil de domicilios para modernizar la experiencia de compra en pueblos tradicionales de Colombia. *Universitaria Uniagustiniana*, 1-89. Obtenido de <https://repositorio.uniagustiniana.edu.co/bitstream/handle/123456789/2581/CoboYara-Geovanny-1-2024.pdf?sequence=4&isAllowed=y>

Desarrollo Web. (2022). *NodeJS*. Obtenido de <https://desarrolloweb.com/home/nodejs>

Donayre Malpartida, V. (2024). Desarrollo e aplciativo Ecommerce para mejorar las ventas de materiales de construcción en los clientes distribuidores de CEMEX Perú S.A LIMA 2021-2022. *Universidad San Ignacio de Loyola*, 1-106. Obtenido de <https://repositorio.usil.edu.pe/server/api/core/bitstreams/0267ccf3-cdd9-4b74-8daf-176148ffb37a/content>

EGA Futura. (18 de Marzo de 2022). *Las estadísticas de servicio al cliente más importantes para 2022*. Obtenido de EGA Futura: <https://discover.egafutura.com/las-estadisticas-de-servicio-al-cliente-mas-importantes-para-2022/>

Florido, L. (2016). La implementación del mobile marketing como herramienta multidisciplinar en el sector turístico y aeroportuario. *EUMED, Biblioteca Virtual de Deecho, Economía, Ciencias Sociales y Tesis Doctorales*. Obtenido de <https://www.eumed.net/libros-gratis/2016/1539/aplicacion.htm>

Inuk, D. (30 de Agosto de 2023). *9 Fallos del servicio de atención al cliente y cómo*

evitarlos. Obtenido de CHATY: <https://chaty.app/es/blog/9-customer-service-fails-and-how-to-avoid-them/>

Íscar, J. (2015). *NODE.JS Do's and Don'ts*. Trabajo de pregrado, Universitat Politècnica de València. Obtenido de <https://riunet.upv.es/bitstream/handle/10251/55344/%C3%8DSCAR%20-%20NODE.JS%20Do%27s%20and%20Don%27ts.pdf?sequence=1>

Izquierdo, C. (2018). *Cómo se hace: prueba de API con Postman*. Obtenido de <https://medium.com/@cesiztel/c%C3%B3mo-se-hace-api-testing-con-postman-978a521552f4>

Lagos, J. (2017). *Desarrollo del Backend de una aplicación para vinculación de clientes en una entidad bancaria*. Tesis de Ingeniería de Sistemas y Computación, Universidad Católica de Colombia. Obtenido de <https://repository.ucatolica.edu.co/bitstream/10983/14605/1/Julian%20Estic%20Lagos%20Galindo%20625180.pdf>

Martínez Tarira, N. (2024). Atención al cliente en el comercial "solano" de la ciudad de babahoyo, periodo 2023. *Universidad Técnica de Babahoyo*, 1-39. Obtenido de <http://dspace.utb.edu.ec/bitstream/handle/49000/15841/E-UTB-FAFI-COM-000404.pdf?sequence=1&isAllowed=y>

Mines Paz, A., & Mines Paz, V. (2024). Desarrollo de un Chatbot basado en Redes Neuronales con IBM Watson para la Atención al Cliente en Axiska Constructores en el año 2023. *Universidad Privada Antenor Orrego*, 1-59. Obtenido de https://repositorio.upao.edu.pe/bitstream/handle/20.500.12759/28511/REP_ALBERTO.MINES_VICTOR.MINES_DESARROLLO.DE.UN.CHATBOT.pdf?sequence=1&isAllowed=y

Mora, M. (2021). *Construir una aplicación móvil para Caarry Soluciones S.A. que se encargue de juntar los servicios de personas independientes y que tengan conocimientos en una determina área*. trabajo de pregrado, Universidad

Católica de Santiago de Guayaquil. Obtenido de <http://repositorio.ucsg.edu.ec/bitstream/3317/17566/1/T-UCSG-PRE-ING-CIC-11.pdf>

Nogales Shagñay, S., & Salazar Villarroel, A. (2024). Desarrollo de una aplicación Web y Móvil para la gestión de ventas por catálogo usando los Frameworks Laravel e Ionic para la empresa Sumicomp S.A. *Escuela Superior Politécnica de Chimborazo*, 1-349. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/21992>

Obando Zambrano, B. (2024). Desarrollo de primera defensa de trabajo final de investigación aplicada a la maestría realizada en el metaverso en Costa Rica. *Revista Educación*, 48(2), 1-16. doi:<http://doi.org/10.15517/revedu.v48i2.58523>

Orientación Universia. (2020). *¿Qué es XP y cómo usarlo en el desarrollo de un proyecto?* Obtenido de <https://orientacion.universia.edu.pe/infodetail/orientacion/consejos/que-es-xp-y-como-usarlo-en-el-desarrollo-de-un-proyecto-6157.html>

Peña, A., & Figueroa Restrepo, W. (2024). Desarrollo de una aplicación para dispositivos móviles que oferte los servicios de reconstrucción, remodelación y rediseño de espacios interiores. *Universidad de la Costa*, 1-83. Obtenido de <https://repositorio.cuc.edu.co/bitstream/handle/11323/13019/Desarrollo%20de%20una%20aplicaci%3b%20para%20dispositivos%20m%3b%20vil%20es%20que%20oferte%20los%20servicios%20de%20reconstrucci%3b%202c%20remodelaci%3b%20y%20redise%3b%20de%20espacios>

Pérez, A., Arias, L., & Romero, M. (2021). Aplicación para promover la cultura y el turismo de la comunidad de Atanquez. *Revista INNDECOMM*. Obtenido de <https://revistas.sena.edu.co/index.php/RINDECOMM/article/view/4342/45>
45

- Quiroga Llanos, J., & Jaimes Foronda, D. (2023). Aplicación Móvil de venta de productos de consumo masivo para automatizar la toma de pedidos de los comerciantes minoristas usando Android. *Universidad Peruana de Ciencias Aplicadas*, 1-63. Obtenido de https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/671091/Quiroga_LJ.pdf?sequence=1&isAllowed=y
- Raebum, A. (2022). *La programación extrema (XP) produce resultados, pero ¿es la metodología adecuada para ti?* Obtenido de <https://asana.com/es/resources/extreme-programming-xp>
- Red Hat. (2022). *¿Qué es la metodología ágil?* Obtenido de <https://www.redhat.com/es/devops/what-is-agile-methodology>
- Sánchez, Y., & Molinares, A. (2020). *Desarrollo de aplicación móvil para el mejoramiento de los procesos de recolección de datos para inventarios forestales de la Fundación BIMA*. Trabajo de pregrado, Universidad Nacional Abierta y a Distancia. Obtenido de <https://repository.unad.edu.co/bitstream/handle/10596/31478/yjsanchezpi.pdf?sequence=3&isAllowed=y>
- Solis Lopez, A., & Valenzuela Peceros, W. (2024). Aplicación Móvil para la gestión de ventas de la empresa Mahpsa S.A.C, Lima, 2023. *Universidad César Vallejo*, 1-94. Obtenido de https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/137821/Solis_LAR-Valenzuela_PWR-SD.pdf?sequence=1&isAllowed=y
- Sommerville, I. (2011). *Ingeniería de Software*. México: Pearson Educación.
- Subramani, K., Hemapriya, V., Anto, A., & Yazhini, V. (2016). Node JS: Building an High Performance Event Manager in Android Platform. *Australian Journal of Basic and Applied Sciences*, 10(1), 174-177. Obtenido de <https://deliverypdf.ssrn.com/delivery.php?ID=581084069120106098022095104025030076054014001051010007006068069074005111117021106088>

024033002115006099007022110087119113017110048006086079029005
094127086119080006110086041022087064085069068064101071106074
067001

UOC. (2021). *Tecnologías y herramientas para el desarrollo web*. Obtenido de http://cv.uoc.edu/annotation/a9c35c372dcee6e6b92afad6993cd048/620334/PID_00250214/PID_00250214.html

Vallejo, K. (2022). *Estudio comparativo de las Tecnologías para el desarrollo del Back-end "NodeJS y PHP"*. Tesis de pregrado, Universidad Técnica de Babahoyo. Obtenido de <http://dspace.utb.edu.ec/bitstream/handle/49000/11684/E-UTB-FAFI-SIST-000338.pdf?sequence=1&isAllowed=y>

Vivar Nuñez , A., & Mendro Vasquez, J. (2024). Diseño de una aplicación móvil para mejorar la gestión de inventario en la Botica Inkasalud, 2023. *Universidad Nacional del Callao*, 1-131. Obtenido de <https://repositorio.unac.edu.pe/bitstream/handle/20.500.12952/8836/TESES%20-%20VIVAR-MENDRO.pdf?sequence=1&isAllowed=y>

Zendesk. (20 de Diciembre de 2023). *8 problemas comunes en servicio al cliente cómo resolverlos*. Obtenido de Zendesk: <https://www.zendesk.com.mx/blog/problemas-comunes-con-clientes/>

