



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE SANTO DOMINGO

Coordinación de Tecnologías de la Información y Diseño

APLICACIÓN WEB Y MÓVIL PARA LA OPTIMIZACIÓN DE RUTAS DE DISTRIBUCIÓN
DE AGUA EN LA EMPRESA STILL WATER DEL CANTÓN SANTO DOMINGO

TRABAJO DE TITULACIÓN

Previo a la obtención del título de Ingeniero en Tecnologías de la Información

Línea de investigación: Tecnologías de la información y la comunicación

Autoría:

Andrade Aguilar Robert Steeven

Sacheri Solórzano Anthony

Dirección:

Ocampo Pazos Willian Javier, Mg.

Santo Domingo – Ecuador
Febrero, 2025



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE SANTO DOMINGO

Coordinación de Tecnologías de la Información y Diseño

HOJA DE APROBACIÓN

**APLICACIÓN WEB Y MÓVIL PARA LA OPTIMIZACIÓN DE RUTAS DE DISTRIBUCIÓN
DE AGUA EN LA EMPRESA STILL WATER DEL CANTÓN SANTO DOMINGO**

Línea de investigación: Tecnologías de la información y la comunicación

Autoría:

Andrade Aguilar Robert Steeven

Sacheri Solórzano Anthony

Revisado por:

Ocampo Pazos Willian Javier, Mg.
DIRECTOR DEL TRABAJO DE TITULACIÓN

Rodolfo Sirilo Córdova Gálvez, Mg.
CALIFICADOR

Luis Javier Ulloa Meneses, Mg.
CALIFICADOR

Franklin Andrés Carrasco Ramírez, Mg.
COORDINADOR DE LA CARRERA DE GRADO

Santo Domingo – Ecuador
Febrero, 2025

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Nosotros, Andrade Aguilar Robert Steeven, portador de la cédula de ciudadanía 1723776280, Sacheri Solórzano Anthony, portador de la cédula de ciudadanía 1719079632, declaramos que los resultados obtenidos en la investigación que presentamos como informe final, previo a la obtención del Grado de Ingeniero en Tecnologías de la Información son absolutamente originales, auténticos y personales.

En tal virtud, declaramos que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

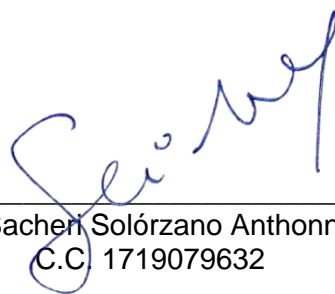
Igualmente, declaramos que todo resultado académico que se desprenda de esta investigación y que se difunda tendrá como filiación la Pontificia Universidad Católica del Ecuador Sede Santo Domingo, reconociendo en las autorías al director del Trabajo de Titulación y demás profesores que amerita.

Además, declaramos que el presente trabajo, producto de las actividades académicas y de investigación, forma parte del capital intelectual de la Pontificia Universidad Católica del Ecuador, Sede Santo Domingo, de acuerdo con lo establecido en el artículo 16, literal j), de la Ley Orgánica de Educación Superior.

En tal razón, autorizo a la Pontificia Universidad Católica del Ecuador, Sede Santo Domingo, para que pueda hacer uso, con fines netamente académicos, del Trabajo de Titulación, ya sea de forma impresa, digital y/o electrónica o por cualquier medio conocido o por conocerse, siendo el presente documento la constancia del consentimiento autorizado; y, para que sea ingresado al Sistema Nacional de Información de la Educación Superior del Ecuador para su conocimiento público, en cumplimiento del artículo 103 de la Ley Orgánica de Educación Superior.



Andrade Aguilar Robert Steeven
C.C. 1723776280



Sacher Solórzano Anthony
C.C. 1719079632

INFORME DE TRABAJO DE TITULACIÓN ESCRITO DE GRADO

Cano de la Cruz, Yullio, PhD

Dirección de Investigación y Postgrados

Pontificia Universidad Católica del Ecuador Sede Santo Domingo


De mi consideración,

Por medio del presente informe en calidad de director del Trabajo de Titulación del Grado de Ingeniería en Tecnologías de la Información titulado: APLICACIÓN WEB Y MÓVIL PARA LA OPTIMIZACIÓN DE RUTAS DE DISTRIBUCIÓN DE AGUA EN LA EMPRESA STILL WATER DEL CANTÓN SANTO DOMINGO, realizado por los estudiantes: Andrade Aguilar Robert Steeven con cédula de ciudadanía 1723776280 y Sacheri Solórzano Anthonny con cédula de ciudadanía 1719079632, previo a la obtención del título de Ingeniero en Tecnologías de la Información, informo que el presente Trabajo de Titulación escrito se encuentra finalizado conforme a la guía y al formato de la Sede vigente.

Además, certifico haber verificado la originalidad y autenticidad del trabajo de titulación por medio del programa anti plagio Turnitin, en respuesta a la normativa institucional vigente.

Santo Domingo, 28/02/2025.

Atentamente,


Mg. Willian Javier Ocampo Pazos
Profesor Titular Auxiliar II

RESUMEN

La distribución de agua es un proceso esencial en el cantón Santo Domingo, sin embargo, en la empresa *Still Water*, se han identificado problemas en la planificación de rutas, debido a la falta de tecnología en la gestión logística, esto genera ineficiencias y pérdida de competitividad en el mercado. Con el fin de solucionar este problema, se planteó la implementación de una aplicación móvil y una aplicación web para optimizar las rutas de distribución, mejorando la asignación de entregas y el seguimiento de pedidos. En el desarrollo de la aplicación *web* se emplearon distintas tecnologías como *PHP*, *HTML*, *CSS* y *JavaScript*, las cuales permitieron crear una plataforma funcional y dinámica, mientras que para el desarrollo de la aplicación móvil se aplicó el lenguaje *Dart* con el *framework Flutter*, que facilitó la creación de una interfaz amigable y eficiente, optimizando la experiencia del usuario. En la investigación emplea un enfoque cuantitativo, con un diseño preexperimental y métodos de recolección de datos de campo, aplicando encuestas a una muestra representativa de 84 clientes, 8 empleados y la entrevista a la gerencia. La hipótesis fue validada, se demostró que el uso de esta tecnología puede optimizar los procesos de distribución, se puede potenciar la escalabilidad de la empresa *Still Water* del cantón Santo Domingo.

Palabras clave: Aplicación móvil, aplicación web, optimización de rutas.

ABSTRACT

Water distribution is an essential process in the Santo Domingo canton, however, in the Still Water company, problems have been identified in route planning, due to the lack of technology in logistics management, this generates inefficiencies and loss of competitiveness in the market. In order to solve this problem, the implementation of a mobile application and a web application was proposed to optimize distribution routes, improving delivery assignment and order tracking. In the development of the web application, different technologies were used such as PHP, HTML, CSS and JavaScript, which allowed the creation of a functional and dynamic platform, while for the development of the mobile application the Dart language was applied with the Flutter framework, which facilitated the creation of a friendly and efficient interface, optimizing the user experience. The research uses a quantitative approach, with a pre-experimental design and field data collection methods, applying surveys to a representative sample of 84 clients, 8 employees and an interview with management. The hypothesis was validated, it was demonstrated that the use of this technology can optimize distribution processes and enhance the scalability of the Still Water company in the Santo Domingo canton.

Keywords: Mobile application, web application, route optimization.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN	10
1.1. Antecedentes	10
1.2. Planteamiento y delimitación del problema.....	12
1.3. Preguntas de investigación.....	13
1.4. Justificación.....	14
1.5. Objetivos de investigación	16
1.5.1. Objetivo general.....	16
1.5.2. Objetivos específicos	16
2. REVISIÓN DE LA LITERATURA.....	17
2.1. Fundamento teórico.....	17
2.1.1. Aplicación Móvil	18
2.1.2. Aplicación web.....	21
2.1.3. Optimización de rutas de distribución.....	32
2.2. Predicción científica.....	36
3. METODOLOGÍA	37
3.1. Enfoque y tipo de investigación	37
3.2. Unidades de análisis	37
3.3. Técnicas e instrumentos de investigación	38
3.4. Técnicas de análisis de datos.....	38
3.5. Operacionalización de las variables	40
4. RESULTADOS	43
4.1. Resultado del primer objetivo específico	43
4.1.1. Resultados de la entrevista al gerente de la empresa.....	43
4.1.2. Análisis e Interpretación de la entrevista al gerente	45
4.1.3. Diagrama de flujo de actividades del proceso de generación de rutas.....	46
4.1.4. Resultados de encuesta a los clientes	47

4.1.5.	Resultados de encuesta a los empleados de la empresa.....	48
4.1.6.	Análisis e Interpretación de los resultados de las encuestas a los clientes	50
4.1.7.	Análisis e Interpretación de los resultados de las encuestas a los empleados	50
4.2.	Resultado del segundo objetivo específico	50
4.2.1.	Tecnologías y herramientas.....	50
4.3.	Resultado del tercer objetivo específico	55
4.3.1.	Nomenclatura y Logotipo	55
4.3.2.	Marco de Trabajo Scrum.....	56
4.3.3.	Sprint 1	56
4.3.4.	Sprint 2	85
4.4.	Validación de la propuesta	100
4.4.1.	Validación de los clientes.....	100
4.4.2.	Validación de los empleados	103
4.5.	Validación de hipótesis.....	106
5.	DISCUSIÓN	109
6.	CONCLUSIONES Y RECOMENDACIONES	111
6.1.	Conclusiones.....	111
6.2.	Recomendaciones.....	112
7.	REFERENCIAS	114
8.	ANEXOS.....	121

1. INTRODUCCIÓN

En la actualidad la globalización ha brindado acceso a mercados internacionales y ha generado la libre competencia, logrando que todas las empresas enfaticen cada vez más en la mejora de procesos o en la innovación, con la finalidad de lograr diferenciarse del resto. En consonancia con lo dicho por Gamboa et al. (2019), la implementación y la adopción de una logística eficaz y accesible representa un pilar de vital importancia para que las empresas, y en concreto las *Pymes* dispongan de mayor número de oportunidades sobre la competencia en este emergente contexto global (p. 253).

Por otra parte, en Ecuador es ampliamente conocido que la mayor parte de las empresas donde la mayoría de sus actividades son de carácter logístico, todavía confían en gran medida en la planificación manual para administrar las rutas de distribución, una estrategia eficaz, sin embargo, esta se torna cada vez más ineficiente y complicada a medida que se incrementa el volumen de las operaciones. En la actualidad, se pueden implementar herramientas tecnológicas en las empresas, que se enfoquen de manera primordial en identificar los trayectos más cortos y mejorar rutas de transporte en general. Como resultado de esto, la empresa puede evidenciar: aumento de productividad, mejor aprovechamiento de recursos logísticos, mayor satisfacción y fidelización del cliente mayorista o minorista.

1.1. Antecedentes

El aumento de la complejidad que se da sobre las cadenas de suministro, y la necesidad de incrementar la eficiencia en la distribución, han contribuido a impulsar el desarrollo de aplicaciones dedicadas a la planificación y optimización de rutas, con el fin de alcanzar este objetivo se debe integrar algoritmos de análisis de datos y tecnología *GPS*. Por esta razón, se lleva a cabo una indagación bibliográfica, en donde se recopila información relevante de diversas bases de datos, que incluyen: *Google académicos*, *Scielo* y *Scopus*.

La investigación de Ramos et al. (2024) en Brasil, tuvo como objetivo, determinar las rutas más adecuadas y óptimas para el transporte de madera. Para lograrlo, implementaron una metodología cualitativa donde utilizaron herramientas de análisis espacial, técnicas de geoprocusamiento, fragmentación de la red de carreteras en el programa *ArcGIS*, y, por último, generaron la ruta óptima con *Network Analyst*. Los resultados obtenidos muestran que las nuevas carreteras creadas han reducido las distancias, disminuyendo el tiempo de viaje, aumentado la seguridad y han permitido su clasificación de carreteras según su frecuencia de uso, generando mayor control y eficiencia logística (pp. 2, 15).

De manera similar, se propone otro problema comparable tratado por Bustamante et al. (2024) en Colombia, quienes utilizaron una metodología cuantitativa para definir su objetivo, que fue, optimizar las rutas de acceso a los residuos en sus distintos puntos estratégicos. Asimismo, realizaron el levantamiento de datos directos de campo, y trazaron los recorridos con el programa *ArcGIS*, donde se crearon las rutas más eficientes empleando *Network Analyst*. Finalmente, aplicaron el algoritmo *Dijkstra* con la finalidad de identificar y evaluar los resultados. Como conclusión, se logró una ruta más corta en los 62.5% de las veces, además, se minimizó las distancias desplazadas en porcentajes desde el 3.67% hasta 28.61% kilómetros transitados diariamente (pp. 1, 5-6, 11-12).

Finalmente, en la investigación realizada por Herrera et al. (2024) en Loja-Ecuador, desarrollaron un procedimiento heurístico que mejoró las rutas de recolección de residuos municipales, empleando como primer paso los datos del *software ArcGIS* y su complemento *Network Analyst*. Este procedimiento, buscó reducir inconsistencias operativas, como giros en "U" innecesarios y accesos a zonas difíciles, optimizando así las distancias recorridas y el tiempo de recolección. Se empleó una metodología mixta, combinando los enfoques cualitativos y cuantitativos para abordar tanto la optimización de las distancias recorridas como la distribución de la carga de trabajo. Como resultado, se logró una reducción significativa en la distancia total recorrida, disminuyendo 295.84 km entre el modelo propuesto, lo que demuestra una mejora en la eficiencia logística (pp. 1, 20).

1.2. Planteamiento y delimitación del problema

Con base a los antecedentes analizados, se pudo evidenciar el uso de herramientas y algoritmos de optimización para mejorar las rutas de transporte. Los avances realizados en estos estudios previos llevados a cabo en Brasil, Colombia y Ecuador, llegaron a satisfacer y obtener resultados positivos, mejorando significativamente la eficiencia en términos de distancia y tiempo. Sin embargo, aún persisten desafíos específicos en la integración de datos y la optimización de rutas en los diferentes contextos logísticos. Considerando este enfoque, se analizaron datos estadísticos sobre el aumento de la complejidad logística, así como sus altos costes.

Según NTT DATA y el Centro Español de Logística (2023), las campañas comerciales que generan un gran volumen de actividad logística pueden causar picos de demanda de entre el 41% y el 60%. Sin embargo, la gestión inadecuada de estos incrementos puede provocar problemas logísticos, de los cuales el 27 y 17% están relacionados con retrasos y entregas fallidas, respectivamente. Como resultado, el 41.5% de las empresas se ve afectada su reputación, debido a estos problemas durante los picos de demanda (pp. 73, 76, 77).

El crecimiento exponencial del comercio electrónico en Colombia, ha creado un crecimiento notable en la demanda de servicios logísticos. Según el Departamento Nacional de Planeación (2022), las empresas han enfrentado desafíos crecientes en sus operaciones logísticas debido a no contar con condiciones adecuadas de infraestructura, la falta de tecnologías avanzadas y la fragmentación de las rutas de distribución. Estas dificultades han provocado un aumento significativo en el costo logístico nacional que alcanzó el 17.9%, donde el transporte es el de mayor costo participación con un 35.9%. Además, en el costo logístico por actividades económicas sobresale el sector comercial por tener el mayor costo del 22.7% (pp. 16-17). Este escenario se agrava con la importancia de satisfacer con las proyecciones de los consumidores, quienes demandan entregas rápidas y precisas. La falta de integración tecnológica, han complicado la capacidad de las empresas para optimizar

sus rutas, resultando en ineficiencias y altos costos operativos que limitan la competitividad en el mercado.

En el mismo contexto, según el Clúster Logístico del Ecuador y ASOLOG (2023), el costo logístico representa un 17.9% del total de ventas de las empresas, evidenciando la elevada carga financiera que enfrentan las organizaciones. Además, el transporte constituye el 40% del costo logístico, mientras que el almacenamiento representa el 23%. Estas cifras, resaltan las ineficiencias en la cadena de suministro que se ven exacerbadas por la necesidad de cumplir con los tiempos de entrega esperados por los consumidores. La disparidad en los costos entre empresas grandes y pequeñas también es significativa, con las pequeñas empresas incurriendo en costos logísticos hasta tres veces mayor que las grandes, lo que limita aún más su competitividad en un mercado en rápida expansión (p. 31, 93).

Por consiguiente, se visitó la empresa *Still Water*, con la finalidad de evaluar y diagnosticar los problemas logísticos que estaban afectando su eficiencia operativa, en particular aquellos relacionados con la administración y optimización de rutas de distribución. Como resultado, se evidenció una planificación deficiente que ha llevado a un uso ineficiente de los recursos de transporte, lo que a su vez ha resultado en demoras en las entregas. Además, la inexistente utilización de tecnología en los procesos logísticos ha contribuido al incremento de los costos operativos. Todos estos factores combinados han ocasionado una significativa pérdida de competitividad para la empresa.

1.3. Preguntas de investigación

Los aspectos mencionados anteriormente llevaron al planteamiento del siguiente desafío de investigación: ¿Cómo optimizar el proceso de generación de rutas de distribución en la empresa *Still Water* del cantón Santo Domingo?

- ¿Qué elementos se considera en el proceso de generación de rutas de distribución?

- ¿Cuáles son las principales herramientas para la optimización de rutas de distribución de agua?
- ¿Qué solución tecnológica puede ofrecer mejoras en la optimización de las rutas de distribución en la empresa *Still Water* del cantón Santo Domingo?

1.4. Justificación

Las empresas o *Pymes* cuyas actividades son mayoritariamente de carácter logístico, afrontan retos cada vez más complicados al momento de querer escalar su red de distribución, lo que genera que la curva de crecimiento de estas empresas cada vez más se asemeje a una línea plana. Una de las principales causas es por la planificación manual para administrar las rutas de distribución, una estrategia eficaz con un volumen de operaciones bajo, sin embargo, al confiar en esta estrategia mucho tiempo se elimina toda posibilidad de un crecimiento exponencial de la empresa. La falta de conciencia sobre la automatización en estas tareas frena el crecimiento de las empresas, logrando como resultado un aumento de sus costos operativos, en comparación con aquellas que ya implementan tecnologías avanzadas de seguimiento y gestión logística.

Este trabajo de titulación está alineado con la Constitución del Ecuador (2008, actualizado ciertos artículos 2021), en el título VII, sección octava, específicamente el artículo 385, denota la relevancia de la innovación y desarrollo de tecnologías, que debe potenciar la producción nacional, debe a su vez incrementar la eficiencia y productividad. Además, este enfoque vanguardista tiene como finalidad el aumento continuo de la calidad de vida de la población y aportar en gran medida a la consecución del buen vivir (p. 185). Esta declaración se encarrila con una visión integral de progreso y desarrollo sustentable que incentiva el bienestar social y económico del país.

De acuerdo con la Asamblea Nacional del Ecuador en base a lo expuesto en la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial (2018), en el artículo 1, narra sobre el incentivo y control de los distintos ámbitos del transporte en Ecuador, incluyendo

los distintos términos para conseguir una movilidad más eficiente como la: planificación de rutas, las operaciones de transporte de mercancías y pasajeros (p. 2). La optimización de rutas de distribución se haya dentro de esta ley, puesto que mejora diversos procesos logísticos aumentando su eficiencia. En definitiva, esta ley beneficia significativamente el transporte en el país al promover una planificación de rutas más eficiente, logrando fomentar y potenciar una logística más competitiva y sostenible.

De igual forma, el Código Orgánico de la Producción, Comercio e Inversiones (2019), en el título I, capítulo I, concretamente el artículo 5, se narra que el Estado estimulará la inversión productiva mediante el incentivo y potenciación de la logística e infraestructura. El principal objetivo es lograr un fortalecimiento de la productividad, generando condiciones óptimas necesarias para desarrollar e incrementar la eficiencia en el transporte terrestre, aéreo y marítimo con un enfoque holístico y una operación integradora (pp. 5-6).

Además, en los Objetivo de Desarrollo Sostenible, en Agenda 2030 presentado por las Naciones Unidas (2018), el objetivo 9, promueve la industrialización sostenible y fomenta la innovación. La meta de este objetivo para 2030 es la adopción de tecnologías innovadoras que brinden mayor eficiencia en la utilización de recursos, disminuyendo en gran medida las emisiones y fomentando la sostenibilidad industrial, logrando que todos los países incorporen la máxima cantidad de medidas de acuerdo a sus capacidades respectivas (pp. 43-44). Además, Naciones Unidas (2018) en su objetivo de desarrollo 11, donde se enmarca la importancia de mejorar la infraestructura urbana y los servicios básicos, promoviendo prácticas que reduzcan el impacto ambiental y fomenten una mayor calidad de vida en las áreas urbanas. En particular, la meta para el 2030 es lograr un crecimiento poblacional sostenible en conjunto con una reducción del impacto ambiental negativo (pp. 52-53).

1.5. Objetivos de investigación

1.5.1. Objetivo general

Implementar una aplicación *web* y móvil para la optimización de rutas de distribución en la empresa *Still Water* del cantón Santo Domingo.

1.5.2. Objetivos específicos

- Identificar los elementos principales a considerar en el proceso de generación de rutas de distribución para definir los requerimientos necesarios.
- Identificar herramientas y tecnologías adecuadas para que se considere en el desarrollo de una aplicación *web* y móvil.
- Desarrollar una aplicación *web* y móvil de optimización de rutas para la distribución en la empresa *Still Water* del cantón Santo Domingo.

2. REVISIÓN DE LA LITERATURA

2.1. Fundamento teórico

Según Hernández y Mendoza (2018), se sugieren elaborar un índice preliminar que abarque de manera global el contenido, el cual deberá ser refinado progresivamente hasta alcanzar un alto grado de especificidad. Posteriormente, se procede a insertar la información referenciada en las secciones correspondientes del esquema (p. 89). La figura 1 define la variable independiente Aplicación Móvil, la figura 2 establece la variable independiente Aplicación Web, la figura 3 delimita la variable dependiente Optimización de Rutas de Distribución.

Figura 1. Panorama bibliográfico de la variable independiente Aplicación Móvil

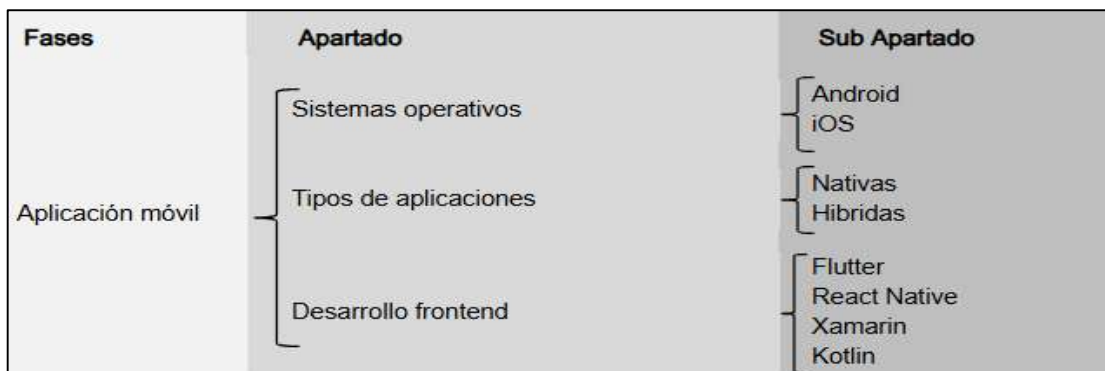


Figura 2. Panorama bibliográfico de la variable independiente Aplicación Web

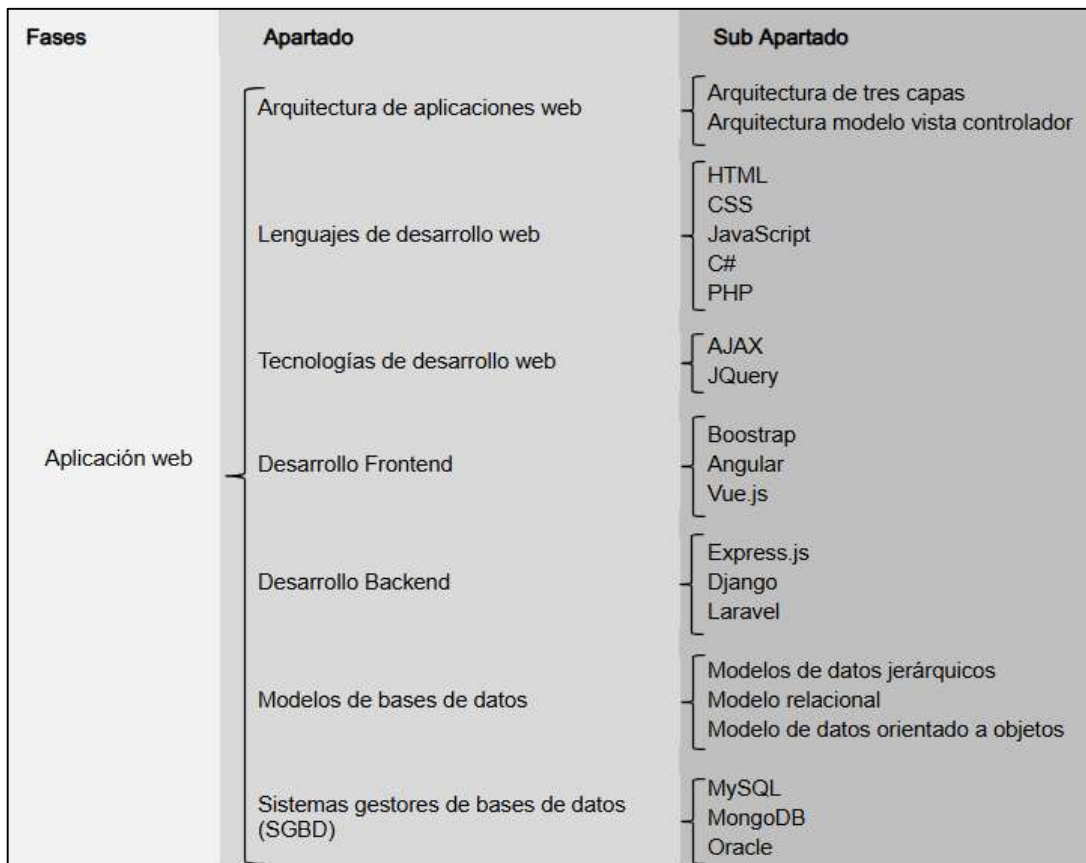
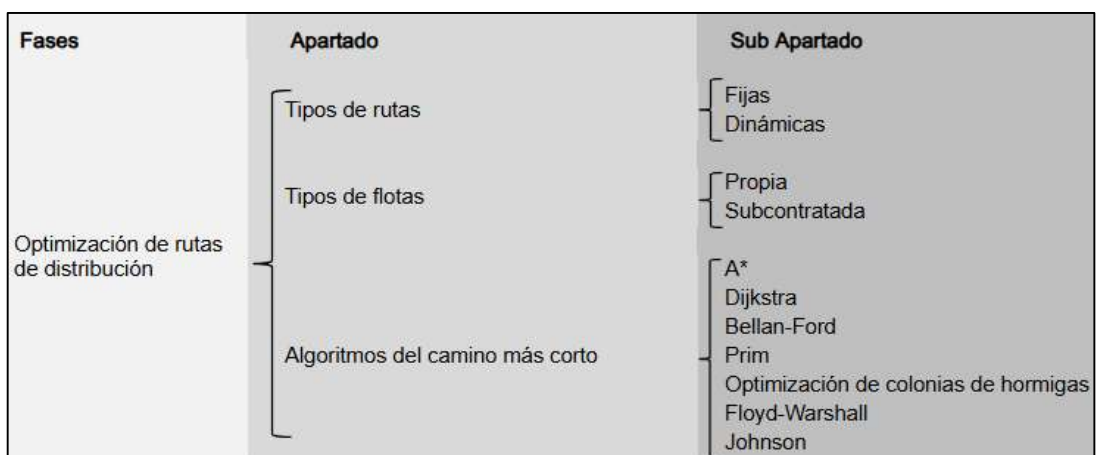


Figura 3. Panorama bibliográfico de la variable dependiente Optimización de Rutas de Distribución



2.1.1. Aplicación Móvil

De acuerdo con Corilla (2022), que citó a Geddes (2004), el concepto de aplicación móvil se refiere a un tipo de software de aplicación creado exclusivamente para dispositivos móviles, como smartphones y tabletas, y está diseñado para sacar provecho de las

capacidades y características propias de estos equipos, tales como la interfaz táctil y la conectividad a redes inalámbricas (p. 19).

Por su parte, Mascarell-Palau (2021) afirma que estas aplicaciones pueden satisfacer prácticamente cualquier exigencia de los usuarios en el ámbito tecnológico de forma accesible en todo el mundo, ya que son accesibles a través de descargas en línea, y pueden ser gratuitas o de pago (p. 82).

2.1.1.1. Sistemas operativos

2.1.1.1.1. Android

De acuerdo con Quisaguano et al. (2022) Android, es un sistema operativo desarrollado por *Google*, sustentado en *Linux* y presentado en 2007. Pensado para distintos equipos con pantalla táctil como celulares, televisores, tabletas y *smartwatches*, *Android* ha sido ampliamente adoptado debido a su carácter de *open-source*, en continuo desarrollo por parte de *Google* y la disponibilidad gratuita de partes de la plataforma para fabricantes de *hardware* y operadores de telefonía (p. 4605).

2.1.1.1.2. iOS

Quisaguano et al. (2022) señala que *iOS* desarrollado por *Apple*, se trata de un sistema operativo móvil disponible públicamente desde el 29 de junio de 2007, basado en *macOS*, el cual, tiene como base el sistema *Darwin BSD*, clasificándose como un sistema operativo tipo *Unix*. Además, Quisaguano et al. (2022), que citó a Antonio (2020), afirma que *iOS* a contraste de *Android*, su sistema operativo se limita a dispositivos de *Apple* como el *iPhone*, *iPod*, *iPad* y *Apple Watch*, y su kit de desarrollo solo está disponible en dispositivos del fabricante, lo que puede incrementar los coste de desarrollo debido al hardware determinado requerido (p. 4606).

2.1.1.2. Tipos de aplicaciones móviles

Según Afreen (2021) existen tres tipos de aplicaciones: *web*, nativas e híbridas, cada una con ventajas y desventajas. La elección óptima depende de componentes como la experiencia de usuario deseada, los recursos necesarios, las características de la aplicación, el tiempo, el costo, el presupuesto y los dispositivos de destino. Por lo que, un

enfoque estratégico que considere todos los factores mencionados permitirá crear una solución tecnológica más efectiva y alineada con los objetivos del proyecto (p. 8).

2.1.1.2.1. Aplicaciones nativas

Para Quisaguano et al. (2022) las aplicaciones nativas son creadas exclusivamente para un determinado dispositivo o sistema, permitiéndoles operar sin la necesidad de programas externos, ya que están desarrolladas en el lenguaje de programación definido de cada sistema. Esta especificidad asegura un rendimiento óptimo y una integración profunda con el *hardware* del dispositivo (p. 4603).

2.1.1.2.2. Aplicaciones híbridas

Por otro lado, Puetate y Ibarra (2020) mencionan a las aplicaciones híbridas como el resultado de combinar aplicaciones nativas y aplicaciones *web*, desarrolladas con herramientas *web* como *HTML*, *CSS* y *JavaScript*, que se compilan y estructuran de manera que el producto final es una aplicación destinada a dispositivos móviles (p. 20).

2.1.1.3. Desarrollo Frontend

2.1.1.3.1. Flutter

Según Napoli (2020), *Flutter* es un *framework* desarrollado por *Google* que maneja *Dart* como lenguaje de programación basado en el enfoque orientado a objetos. Este *framework* de código abierto cuya responsabilidad de desarrollo y mantenimiento son responsabilidad tanto de *Google* como de la comunidad de desarrolladores, se ha solidificado como una poderosa herramienta para la creación de aplicaciones nativas avanzadas, compatibles tanto con las plataformas *Android* como *iOS* (p. 4).

2.1.1.3.2. React Native

Para Lazcano et al. (2019) definen a *React Native* como una tecnología que admite la creación de aplicaciones móviles basándose solo en un lenguaje, *JavaScript*. Esto resulta ventajoso para los programadores web ya que no requieren adquirir conocimiento en un nuevo lenguaje de programación. Además, recalcan que, a diferencia de las aplicaciones *web* móviles, basadas en *HTML5* o aplicaciones híbridas, *React Native* permite desarrollar aplicaciones móviles nativas que son prácticamente idéntico de una aplicación nativa desarrollada con lenguajes como *Java* en *Android* o *Objective C* en *iOS* (p. 1).

2.1.1.3.3. Xamarin

Conforme a lo mencionado por Microsoft (2023), *Xamarin* es una tecnología *open-source* que agiliza la creación de aplicaciones de gran rendimiento y modernas para *Android*, *iOS* y *Windows* con *.NET*, resaltando que *Xamarin* se ejecuta en un entorno administrado, para con ello, obtener distintas ventajas como la recolección de elementos no manejados y asignación de memoria permitiendo a los desarrolladores compartir cerca del 90% del código de una aplicación entre diversas plataformas. A su vez, las aplicaciones creadas con *Xamarin* pueden ser escritas en *PC* o *Mac* y se compilan en formatos de aplicaciones nativas, tales como los archivos *.apk* en *Android* o *.ipa* en *iOS* (párr 2-4).

2.1.1.3.4. Kotlin

De acuerdo con lo que menciona Aceñero (2023), *Kotlin* es un lenguaje de código abierto que combina programación funcional con orientación a objetos. Se destaca por una sintaxis más sencilla a *Java* y similar a *Scala*, lo cual proporciona su interoperabilidad con *Java*, permitiendo así a los desarrolladores aumentar su productividad al integrar funcionalidades de *Java* en programas *Kotlin* sin generar conflictos al ser llamadas, por lo tanto, se convierte en una opción ideal para programadores, inclusive para principiantes, interesados en desarrollar aplicaciones nativas para *Android* de manera eficaz (p. 6).

2.1.2. Aplicación web

En palabras de Cardador (2024), menciona a una aplicación *web* como un programa de computador codificado en un lenguaje de programación compatible con navegadores *web*, permitiendo la comunicación del usuario con el servidor *web*. El desarrollo de aplicaciones web se realiza utilizando lenguajes de programación o a su vez en entornos de desarrollo integrado, donde se diseña la aplicación *web* mediante código y se prueba con herramientas que dispone para garantizar que el código sea correcto y cumpla con sus objetivos (p. 139).

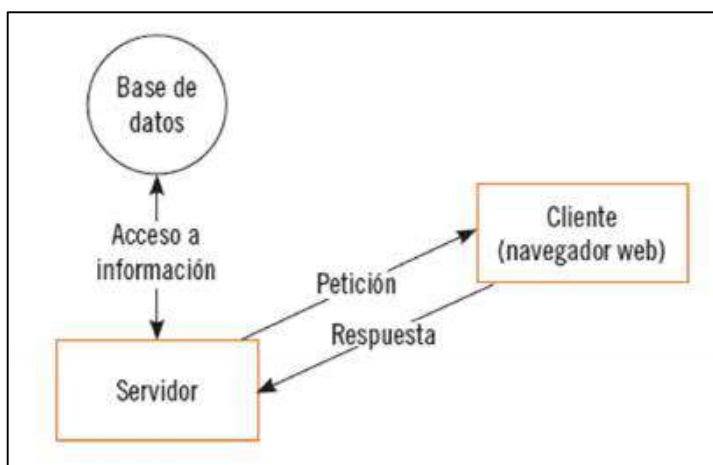
Según Sánchez (2015), el diseño y programación de aplicaciones web es basado en un conjunto de tecnologías estándar, siendo *HTML*, *CSS* y *JavaScript* las más fundamentales. *HTML* proporciona la estructura de la página, *CSS* se encarga del estilo

visual y *JavaScript* agrega interactividad. Aunque los navegadores no pueden compilar código de lenguajes de programación tradicionales como C o C++, son capaces de interpretar eficientemente *JavaScript*, gracias a motores de ejecución como V8 de *Chrome* (p. 14).

2.1.2.1. Arquitectura de aplicaciones web

La estructura fundamental de una aplicación web, como se muestra en la figura 4 sigue un ciclo de funcionamiento que puede desglosarse en varios pasos.

Figura 4. Estructura fundamental de una aplicación web



Nota. Granados La Paz (2023, p. 250) Estructura básica de una aplicación web.

En términos generales, según Granados La Paz (2023), el sistema funciona de la siguiente manera:

1. El proceso comienza cuando el usuario envía una solicitud mediante el cliente, que suele ser un navegador *web* como *Google Chrome*, *Firefox* o cualquier otro. Esta petición puede ser tan simple como solicitar el acceso a una página o interactuar con alguna funcionalidad de la aplicación *web* (p. 250).
2. Habitualmente, el servidor cuenta con disponibilidad a una base de datos, que es donde se guarda la información necesaria para ejecutar las consultas solicitadas por el usuario (p. 251).

3. Luego de obtener o actualizar la información relevante de la base de datos, el servidor procesa y forma una respuesta que envía de vuelta al cliente (p. 251).
4. Posterior, el cliente se encarga de interpretar y renderizar la respuesta proporcionada por el servidor, mostrando los datos de forma comprensible e interactiva para el usuario (p. 251).

Integrando diferentes entidades como el cliente, servidor y base de datos en este ciclo de funcionamiento, es lo que permite que las aplicaciones *web* sean eficientes y dinámicas.

2.1.2.1.1. Arquitectura de tres capas

Como señalan López et al. (2016), la arquitectura de tres capas se establece como un paradigma esencial en la ingeniería de *software*, principalmente en el ámbito de las aplicaciones *web*. Esta arquitectura separa el desarrollo de una aplicación en tres niveles bien precisos: presentación, lógica de negocio y persistencia (almacenamiento de datos). Cada capa agrupa funciones concretas, lo que beneficia una clara separación de responsabilidades y facilita la modularidad del *software*. La estructura se basa en la idea de que cada capa está compuesta por un conjunto de módulos que interactúan entre sí o con otras capas, por tanto, puede suplir varios servicios que permiten la funcionalidad completa de la aplicación (p. 74).

2.1.2.1.2. Capas

López et al. (2016) señalan que la arquitectura *web* de tres capas establece un paradigma de prototipo de software que promueve la estructuración y la separación de responsabilidades en la construcción de aplicaciones web (p. 74), este modelo estructural descompone una aplicación en tres niveles bien específicos, los cuales se presentan a continuación:

Capa de presentación

Para Urbano (2015), la capa de presentación actúa como la interfaz para el usuario, usualmente a través del navegador, cuyas funciones incluyen recuperar la información del

usuario, enviarla a la capa correspondiente, recibir los resultados una vez procesada la información y presentarlos al usuario, lo que asegura una comunicación efectiva y continua entre el usuario y la aplicación (p. 182).

Capa de negocio

Según Sánchez (2015), explica que es la capa intermedia en una aplicación que comunica la capa de presentación con la de datos, proporcionando la lógica del negocio mediante servidores, en otras palabras, la que contiene la información empresarial. Su función es procesar las peticiones del cliente, recibir solicitudes de la capa de presentación, comunicarse con la capa de datos y enviar la información nuevamente a la capa de presentación, certificando una comunicación efectiva entre todas las capas de la aplicación (p. 27).

Capa de datos

En base a lo mencionado por Urbano (2015), la capa de datos se constituye de la base de datos, y sus principales funciones incluyen almacenar, mantener y recuperar datos, además de garantizar su integridad (p. 182).

2.1.2.1.3. Arquitectura modelo vista controlador (MVC)

En palabras de Pinzón et al. (2019), el modelo de arquitectura Modelo-Vista-Controlador es uno de los más usados y aplicados en el desarrollo de sistemas de información con interfaces de usuario. Esta arquitectura facilita el desarrollo de *software* con un ciclo de vida más eficaz, haciendo una mayor funcionalidad, mantenibilidad, reutilización del código y escalabilidad. Además, el MVC separa la estructura de codificación en capas que definen claramente las responsabilidades de cada componente, lo que permite una mejor organización del sistema. Este esquema asegura que los distintos elementos, como el modelo, la vista y el controlador, se desarrollen de manera independiente y puedan relacionarse de manera eficiente (p. 19).

Para Pinzón et al. (2019) la principal característica del modelo MVC radica en que el modelo, la vista y el controlador se manejan como entidades independientes. Esto permite

que cualquier modificación en el modelo se manifieste instantáneamente en las vistas (p. 19).

2.1.2.1.4. Modelo: Programación de acceso a datos

Según Coronado (2024), el modelo constituye la parte de la aplicación encargado de la administración de los datos, y es en esta capa donde se establecen las estructuras y los procesos necesarios para almacenar, recuperar y manipular la información. Asimismo, el modelo se comunica de manera inmediata con la base de datos y otras fuentes de datos (p. 152).

2.1.2.1.5. Vista: Desarrollo de aplicaciones en cliente. Eventos e interfaz de usuario

En palabras de Pinzón et al. (2019), la vista es el componente de la aplicación encargado de facilitar la comunicación directa con el usuario, además, su función principal es presentar los datos y gestionar las entradas proporcionadas por el usuario (p. 20). Además, según Coronado (2024) en el ámbito de una aplicación *web*, la vista suele estar compuesta por tecnologías como *HTML*, *CSS* y *JavaScript*. Los eventos generados en la vista, tales como el clic en un botón, son capturados y transmitidos al controlador para su procesamiento (p. 152).

2.1.2.1.6. Programación del controlador

Como afirma Coronado (2024), el controlador funciona como un enlace entre el modelo y la vista. Su tarea principal es recibir las solicitudes del usuario que son obtenidas por medio de la vista, procesarlas y determinar los datos necesarios del modelo. Posteriormente, transmite dicha información a la vista para que sea presentada al usuario (p. 152).

2.1.2.2. Desarrollo web

Según lo planteado por Sánchez (2015) afirma que el *frontend* es la parte visible de una aplicación, con la cual interactuamos directamente, a través de botones, imágenes y menús, los diseñadores de *frontend* crean bocetos y maquetas para visualizar cómo se verá la aplicación final, usando *software* especializado para elaborar *wireframes* y *mockups*,

gracias a esta parte del desarrollo, las aplicaciones no solo son funcionales, sino también atractivas y fáciles de usar (p. 27).

2.1.2.2.1. HTML

De acuerdo con Fernández (2023) describe *HTML (HyperText Markup Language)* como un lenguaje de marcado especializado en la creación de páginas web, compuesto por entidades que estructuran el contenido y definen sus distintas partes. Estos elementos, conocidos como etiquetas, contienen atributos que desempeñan funciones particulares y pueden estar vinculados a comportamientos específicos, proporcionando flexibilidad y control sobre la presentación y funcionalidad del contenido web (p. 18).

2.1.2.2.2. CSS

Como señala Fernández (2023), las hojas de estilo en cascada (*Cascading Style Sheets*), es un lenguaje de diseño que se utiliza para la personalización de documentos estructurados, facilita el desarrollo de las páginas web en conjunto con el *HTML*. Aunque, su uso debe centrarse en contenidos no esenciales y no debe ser un factor diferenciador en términos de accesibilidad, *CSS* puede utilizarse para potenciar y mejorar la accesibilidad (p. 55).

2.1.2.2.3. JavaScript

De acuerdo con Fernández (2023), *JavaScript* es un lenguaje interpretado de programación que se fundamenta en el estándar *ECMAScript (European Computer Manufacturer's Association Script)*, caracterizándose por su enfoque orientado a eventos y su modelo basado en prototipos, además de ser dinámico y no rigurosamente tipado. Su uso es en lado del cliente y son los navegadores quienes lo implementan como parte de su potencial, por tal motivo, algunas funciones, sentencias, eventos y métodos no funcionen igual, ya que influye el navegador con el que se encuentra trabajando, llegando incluso a que algunas funcionalidades ni siquiera sean reconocidas (p. 146).

2.1.2.2.4. C#

Según Layedra et al. (2022), el lenguaje *C#* fue desarrollado por *Microsoft*, producto del lenguaje *C*, poseyendo una sintaxis con muchas similitudes sin embargo con la diferencia que *C#* es orientado a objetos, además que fue diseñado para elaborar

aplicaciones con un mayor nivel de abstracción. Este lenguaje de programación es multiplataforma, diseñado para la infraestructura común de los lenguajes, los usos más destacados son en el desarrollo de aplicaciones de consola o escritorio, para desarrollo de aplicaciones en móviles con *Xamarin*, páginas *web* con *ASP.NET*, también para el desarrollo de servicios *API*, entre otros (pp. 1617-1618).

2.1.2.2.5. PHP

De acuerdo con lo dicho por Escarcena (2020) donde explica que *PHP*, siglas de *Hypertext Preprocessor*, es un lenguaje de programación del lado del servidor destacando por ser autónomo, gratuito, de código abierto y eficiente. Agregando que es compatible con diferentes dispositivos y facilita una integración robusta con diversas bases de datos, como *MySQL*. Además, *PHP* es un lenguaje interpretado, es decir que su código permite la ejecución sin necesidad que sea compilado previamente, lo que le concede independencia del *hardware* en uso. Su simplicidad radica en su capacidad para incorporar pequeños scripts en página *HTML*, permitiendo así implementar funcionalidades dinámicas en el entorno de *backend* (p. 53).

2.1.2.3. Tecnologías de desarrollo web

2.1.2.3.1. AJAX

En palabras de Flórez y Hernández (2021), *AJAX* es un acrónimo de *Asynchronous JavaScript and XML*, no constituye una tecnología por si sola, sino de una serie de principios. Estos principios son derivados de investigaciones de *Google*, enfocados en la gestión de *JavaScript* en aplicaciones web de alta demanda, desarrollando así interfaces dinámicas, de alto rendimiento y eficientes. Se destaca por el uso eficiente de *JavaScript*, permitiendo la conexión asíncrona con un servidor para descargar nuevos datos sin recargar la página completa, generalmente, se desarrolla mediante *jQuery*, compatible con los navegadores más utilizados como *Google Chrome*, *Microsoft Edge*, *Opera*, *Firefox* y otros. Esto asegura y facilita su integración en muchas aplicaciones *web* diferentes (pp. 5, 217).

2.1.2.3.2. JQuery

Por otro lado, según Flórez y Hernández (2021), *JQuery* es una librería que proporciona la manipulación del *Document Object Model*, ofreciendo diferentes funciones para gestionar eventos, animaciones, agregar otros elementos del *DOM*, etc. Además, posterior a la carga de la página *web*, *JQuery* permite hacer consultas *HTTP* a través *AJAX*, haciendo posible la comunicación dinámica entre el cliente y el servidor (pp. 4, 89).

2.1.2.4. Desarrollo frontend

2.1.2.4.1. Bootstrap

Como afirma Flórez y Hernández (2021), *Bootstrap* fue creado por *Twitter* a mediados de 2010, se ha convertido en un *framework* que destaca por su capacidad para permitir que los sitios *web* sean responsivos, es decir, permite adaptar el contenido de manera óptima según el dispositivo. Desde su primera versión, conocida como *Twitter Blueprint*, ha avanzado a *Bootstrap 4*, siendo ampliamente utilizado y compatible con los distintos navegadores como *Google Chrome*, *Microsoft Edge*, *Firefox*, *Opera* y *Safari* (con la excepción para *Windows*), e incluso ofrece soporte para versiones *IEQ* y versiones antiguas de *Firefox*, garantizando su funcionalidad en una amplia gama de navegadores y entornos (p. 75).

2.1.2.4.2. Angular

De acuerdo con Bhaskar (2020), *Angular* es un *framework* de desarrollo *web* construida sobre *TypeScript*, diseñado especialmente para programar aplicaciones *web* de una sola página, sin embargo, esto no lo limita a desarrollar otras aplicaciones dinámicas. Además, como característica de *Angular* es que su arquitectura se basa en componentes, donde cada componente tiene su propio archivo de plantilla *HTML*, estilos *CSS* y lógica en *TypeScript* (pp. 7613-7614). Importante el añadir que permite manipular el *DOM* y la incorporación de la consola de comandos agiliza la creación, construcción, pruebas y despliegue de aplicaciones mediante comandos simples. Para finalizar destacar que *Angular* es multiplataforma y ofrece considerables ventajas como: desarrollo rápido, reutilización del código, un óptimo rendimiento, una interfaz de usuario atractiva, escalabilidad, entre otros (pp. 7616-7617).

2.1.2.4.3. Vue.js

Según So (2018), considerado un *framework* progresivo, *Vue.js* facilita desarrollar interfaces de usuario, es marco está diseñado de forma que se pueda adoptar gradualmente. So hace énfasis en tres características de *Vue.js*: su accesibilidad para que desarrolladores de la triada *HTML*, *CSS* y *JavaScript* lo puedan entender con relativa facilidad, su versatilidad ya que es muy flexible y su alto rendimiento del *DOM* virtual (p. 381).

2.1.2.5. Desarrollo backend

2.1.2.5.1. Express.js

Tal como señala OpenJS Foundation (2024), Express es un entorno de aplicaciones web de *Node.js*, es minimalista y ágil, que proporciona un grupo de funciones y cualidades para mejorar el desarrollo de aplicaciones *web* y móviles. Además, permite una fácil y rápida creación de *APIs* robustas. Express posee una estructura de enrutamiento flexible y ligera con atributos básicos mínimos, está diseñado con el objetivo de ser ampliado con el uso de módulos *middleware* (párr. 1-2, 4).

2.1.2.5.2. Django

Según lo planteado por Coronado (2024), *Django* es un *framework* para la programación *web* en *Python* que permite desarrollar aplicaciones *web* de manera ágil y eficaz, utilizando menos código. Además, se enfoca en la seguridad y ofrece distintas herramientas necesarias como autenticación de usuarios, organización de formularios y autorización a bases de datos para estar en un entorno seguro, haciendo aún más sencillo el desarrollo completo de aplicaciones desde cero (p. 110).

2.1.2.5.3. Laravel

Para Aguirre (2021), *Laravel* es un *framework open-source* que agiliza el desarrollo de aplicaciones escalables y elegantes, quitando redundancias. Se basa en una arquitectura Modelo-Vista-Controlador que organiza y conserva la separación entre lógica, datos y diseño. Además, su repositorio en *GitHub* facilita a los programadores la colaboración y el acceso libre a su código fuente (p. 11).

2.1.2.6. Modelos de bases de datos

Según Campos et al. (2022), una base de datos como una colección estructurada de datos organizados que representan aspectos específicos del mundo real y su administración se lleva a cabo en un Sistema Gestor de Bases de Datos en el cual se almacenan, consultan e integran los datos. Un SGBD es un *software* que proporciona administrar, organizar y facilitar el acceso y la manipulación eficiente de la información registrada en la base de datos, proporcionando así una gestión optimizada y estructurada de los datos (p. 19).

2.1.2.6.1. Modelos de datos jerárquicos

Siguiendo la perspectiva de Postigo (2021), la información se organiza mediante un esquema en forma de un árbol, que es esencialmente el modelo que utiliza *XML* para organizar los datos de manera jerárquica (p. 5). Además, según Ladrón de Guevara (2024) este modelo es muy útil cuando existe gran cantidad de datos y sumado a esto los datos son compartidos, sin embargo, su mayor defecto es la imposibilidad de lograr una correcta redundancia de datos (p. 54).

2.1.2.6.2. Modelo relacional

Conforme a lo mencionado por Ladrón de Guevara (2024), señala que el modelo relacional es el modelo de bases de datos más popular en el presente ya que facilita la gestión dinámica de datos y de representar de manera adecuada las situaciones reales. Su principal característica es el almacenamiento de información en tablas, que se dividen en campos (columnas) y registros (filas). Las tablas se interrelacionan, proporcionando una estructura lógica para vincular los datos, y la metodología más común para obtener información en estas bases de datos es a través de consultas, facilitando el acceso y la manipulación eficiente de la información almacenada (p. 54).

2.1.2.6.3. Modelo de datos orientado a objetos

Según lo planteado por Ladrón de Guevara (2024), este tipo de bases de datos utiliza el paradigma de programación orientada a objetos, permitiendo que los datos tengan tanto un estado como un comportamiento, además de integrar características inherentes a

este paradigma como la herencia, el polimorfismo, el encapsulamiento y la abstracción (p. 55).

2.1.2.7. Sistemas gestores de bases de datos

En palabras de Postigo (2021), los sistemas gestores de bases de datos (SGBD) son *softwares* esenciales que organizan y manipulan grandes volúmenes de datos, facilitando el acceso directo a archivos y registros para extraer información, además, proporcionan una interfaz de abstracción que facilita a los usuarios gestionar eficientemente los datos, asegurando su integridad y disponibilidad (p. 4). De este modo, los SGBD se establecen como elementos fundamentales para mejorar la eficiencia de procesos en una variedad de aplicaciones y entornos.

2.1.2.7.1. MySQL

Según Ohyver et al. (2019), *MySQL* es un servidor de bases de datos relacional que soporta el conocido lenguaje *SQL (Structured Query Language)*, según *Dataconomy*, la última versión de *MySQL* se posiciona como uno de los gestores de bases de datos más utilizados a nivel internacional debido a que es una solución de código abierto confiable y compatible con todos los principales proveedores de *hosting*, además de ser rentable y fácil de gestionar, lo que facilita su adopción en diversos entornos (p. 397).

Además, Pavón y Llarena (2015), mencionan que *MySQL* se posiciona como la base de datos favorita por la gran parte de desarrolladores, debido a que este gestor de bases de datos soporta *SQL* y permite la conexión simultánea de múltiples usuarios, sin embargo, su uso suele estar enfocado principalmente en aplicaciones de pequeño y mediano tamaño. Es una herramienta gratuita, lo que contribuye significativamente en diversos proyectos de desarrollo *web* (p. 18).

2.1.2.7.2. MongoDB

En palabras de López et al. (2023), *MongoDB* carece de un lenguaje de definición de datos específico, sin embargo, emplea el formato *BSON* para el almacenamiento interno de datos y el formato *JSON* para la representación de la información ante los usuarios, siendo *BSON* la representación binaria de *JSON*. Los servidores de *MongoDB* pueden almacenar

múltiples bases de datos, cada una de ellas compuesta por un conjunto de colecciones, lo que las clasifica como bases de datos distribuidas por naturaleza, esta arquitectura permite una gestión eficiente y escalable de los datos en diversos entornos (p. 264).

2.1.2.7.3. Oracle

Según Postigo (2022), Oracle es un motor de base de datos bajo el enfoque relacional, diseñado para ser multihilo, multiusuario y multiplataforma, cuya arquitectura principal es cliente-servidor y está expandiéndose hacia otros modelos. Además, incorpora el lenguaje *JSON* y dispone de una versión gratuita, *Express Edition* de código abierto, permitiendo un acceso más amplio y flexible, posicionándolo como un SGBD altamente competitivo (p. 18).

2.1.3. Optimización de rutas de distribución

Según Sarder (2021), cada empresa posee necesidades particulares por ende las redes de transporte deben de tener un análisis y modelado cuidadoso para cubrir adecuadamente todas estas necesidades. Este proceso debe tomar en cuenta diversas variables, como el tipo de producto a transportar y el destino al cual se dirige. Además, afirma que no existe un estándar que se pueda implementar de manera óptima en cada una de las posibles situaciones existentes, ya que las circunstancias y variables involucradas, como la geografía, los costos y la demanda, fluctúan continuamente (p. 57).

2.1.3.1. Tipos de rutas

2.1.3.1.1. Fijas

Tal como afirma Ragàs (2024), las rutas fijas están elaboradas de forma en que se conectan puntos predefinidos de forma regular y sin alteraciones, conservando un volumen de carga constante para cada trayecto. Estas rutas son caracterizadas por la afinidad del conductor con cada uno de los distintos aspectos del recorrido, desde cada zona de embarque y desembarque hasta las posibles restricciones horarias que puedan existir en determinados puntos (p. 68).

Del mismo modo Castell (2021) detalla los principales beneficios al aplicar este tipo de ruta, afirma que hay normalidad y regularidad en los recorridos, facilitando a los clientes

organizar sus actividades con mayor precisión y certeza. Además, al realizar rutas estándar, la carga administrativa es relativamente menor, puesto que no requieren ajustes de manera frecuente (p. 44).

2.1.3.1.2. Dinámicas

Según Ragàs (2024) las rutas dinámicas se elaboran y diseñan para amoldarse a los requerimientos particulares de cada cliente, logrando una distribución más personalizada en función del origen, destino y ofreciendo variaciones del volumen de la carga. Entre sus puntos diferenciadores de las rutas fijas se evidencian la variabilidad de los distintos puntos de embarque y desembarque, del mismo modo como en el volumen de carga transportada (p. 68).

De igual manera, Castell (2021) denota las distintas ventajas de las rutas dinámicas, como la posibilidad de ofrecer un servicio más personalizado, la capacidad de adaptación para coordinar la recolección y distribución, la optimización de los transportes según la demanda, la posibilidad de ajustarse a ventanas de tiempo y controlar situaciones no previstas, además de mejorar la flexibilidad en la asignación de vehículos y conductores para potenciar la eficiencia del servicio (p. 45).

2.1.3.2. Tipos de flotas

2.1.3.2.1. Propia

Según Castell (2021) existen varias opciones para que una empresa pueda adquirir una flota, como: la compra directa, *leasing* o *renting*, la selección se ajustara acorde a las distintas necesidades y presupuesto de las empresas. El *renting* es un contrato que ofrece de beneficio ajustar la flota en función de la demanda durante un par de años que es la duración promedio que se maneja en esta clase de contratos. Por otro lado, el *leasing* es un contrato de compra de largo plazo con un bajo valor residual, donde se puede adquirir el vehículo al terminar el contrato (pp. 54-55).

2.1.3.2.2. Subcontratada

Por otro lado, Ragàs (2024) afirma que la mayoría de los vendedores no poseen un medio de transporte propio y deben de contratar empresas de transporte externas (p. 62), bajo esta perspectiva, Castell (2021) explica que la subcontratación de flotas permite a las

empresas hacer frente a picos de demanda sin incurrir en los altos costos fijos asociados con el mantenimiento de una flota propia. Una de las modalidades más comunes es la flota contratada dedicada, que ofrece un servicio exclusivo y flexible, lo que ayuda a las empresas a reducir gastos relacionados con personal, mantenimiento de los vehículos y seguros. Asimismo, las empresas que gestionan grandes volúmenes de carga suelen optar por una estrategia mixta, combinando el uso de flotas propias con la subcontratación de vehículos adicionales (p. 56).

2.1.3.3. Algoritmos del camino más corto

2.1.3.3.1. A*

De acuerdo con Urquhart (2022), el algoritmo A^* es más eficiente que *Dijkstra*, si se busca encontrar rutas en grafos de mayor tamaño. Este algoritmo hace uso de una lista donde guarda únicamente los nodos que se están evaluando en cada momento y emplea una heurística para instruir la búsqueda al nodo final. A^* evalúa la distancia desde el nodo origen y estima la distancia al nodo destino empleando la fórmula de *Haversine*, que ofrece una estimación en línea recta. Lo que lo diferencia de *Dijkstra*, es que A^* optimiza el proceso de búsqueda al priorizar los nodos que se encuentran en la ruta más directa, seleccionando cada siguiente nodo a partir de la suma de la distancia calculada desde el punto inicial y la distancia estimada al punto de llegada (pp. 165-166).

2.1.3.3.2. Dijkstra

De acuerdo con Urquhart (2022) el algoritmo de *Dijkstra* evalúa y busca el camino más corto desde un punto de origen hacia todos los demás puntos de un grafo. Asigna etiquetas de distancia y nodo anterior a cada vértice, y las actualiza conforme evalúa los vecinos del nodo actual. Este proceso sigue su curso hasta que cada uno de los nodos han sido visitados, construyendo así un árbol de caminos más cortos. Además, el algoritmo se detiene cuando el nodo actual es el mismo al nodo final, evitando el cálculo de distancias innecesarias (pp. 162-164).

2.1.3.3.3. Bellman-Ford

Como sostiene Kumawat (2021), el algoritmo de *Dijkstra* no es adecuado para grafos con aristas de peso negativo, por lo que se recurre al algoritmo de *Bellman-Ford*, que sí

puede manejar estos casos. *Bellman-Ford* calcula los caminos más cortos desde un punto de inicio hacia los demás puntos restantes, actualizando progresivamente las estimaciones de distancia en cada iteración. Ajusta cada una de las aristas y repite el proceso n cantidad de veces (siendo n el número total de vértices). Inicializa el valor de la fuente en 0 y asigna infinito a los demás nodos (p. 179).

2.1.3.3.4. Prim

Según lo señalado por Kumawat (2021), el algoritmo de *Prim* describe un procedimiento que selecciona las aristas de menor peso, comenzando con un nodo y añadiendo gradualmente nuevos nodos a un grafo no conectado. En cada iteración, se selecciona la conexión de menor peso desde los nodos disponibles. El algoritmo opera manteniendo dos conjuntos de vértices y, en cada ciclo, identifica la arista mínima entre ambos conjuntos, añadiendo nuevos vértices hasta completar el grafo (p. 179).

2.1.3.3.5. Optimización de colonias de hormigas

Según Hemmati-Sarapardeh (2020) describe este algoritmo como una metaheurística basada en el comportamiento de las hormigas al buscar alimento. Las hormigas exploran su entorno aleatoriamente, dejando rastros de feromonas cuando encuentran una fuente de alimento, lo que guía a otras hormigas hacia los caminos más eficientes. Con el tiempo, los trayectos más cortos se refuerzan, optimizando el recorrido. Originalmente, este algoritmo se aplicaba a problemas en dominios discretos, pero luego se extendió a dominios continuos utilizando un archivo de soluciones y modelos probabilísticos basados en mezclas gaussianas, lo que se conoce como algoritmo de estimación de distribución (pp. 59-60).

2.1.3.3.6. Floyd-Warshall

De acuerdo con Toroslu (2021), afirma que el algoritmo *Floyd-Warshall* aborda el problema de encontrar los caminos más eficientes en términos de distancia, entre todos y cada uno de los pares de vértices existentes en un grafo. Lo hace mediante una relación de recurrencia que, en iteraciones sucesivas, actualiza las distancias mínimas entre vértices al agregar intermediarios en los caminos. Comienza con una matriz de adyacencia que

representa los pesos entre los vértices y aplica una operación de relajación en cada iteración para mejorar las distancias. El algoritmo es eficiente para grafos pequeños o densos, pero no es adecuado para grafos grandes y dispersos sin modificaciones (p. 1).

2.1.3.3.7. Johnson

Tal como señala Kumawat (2021), el algoritmo de *Johnson* se emplea para calcular los caminos más cortos en grafos, especialmente cuando estos contienen ciclos de peso negativo. En una primera etapa, utiliza el algoritmo de *Bellman-Ford* para detectar y eliminar tanto los ciclos como las aristas con pesos negativos. Luego, vuelve a ponderar las aristas para garantizar que todas posean pesos positivos. Finalmente, hace uso del algoritmo *Dijkstra* calculando los caminos más cortos. Este enfoque resulta eficiente, sobre todo cuando se trata de grafos dispersos (p. 179).

2.2. Predicción científica

H0: La aplicación *web* y móvil no incide significativamente en la optimización de rutas de distribución en la empresa *Still Water* del cantón Santo Domingo.

H1: La aplicación *web* y móvil incide significativamente en la optimización de rutas de distribución en la empresa *Still Water* del cantón Santo Domingo.

3. METODOLOGÍA

3.1. Enfoque y tipo de investigación

Para el desarrollo de este trabajo de titulación se inclinó por emplear un enfoque cuantitativo, el cual según Hernández y Mendoza (2018), sigue un proceso secuencial que pretende probar la hipótesis mediante la recolección y análisis de datos numéricos, desde la elaboración de preguntas hasta la ponderación estandarizada de resultados que valida o rechaza las hipótesis planteadas, proporcionando evidencia estadística (p. 6).

Se estimó el diseño preexperimental debido a que posee un nivel de control básico, de acuerdo con lo presentado por Hernández y Mendoza (2018), el diseño preprueba/posprueba de un solo grupo, implica aplicar una medición anticipada al tratamiento experimental y otra medición posterior al estímulo surgido por el tratamiento experimental (p. 163).

Asimismo, este trabajo de titulación hizo uso de la investigación aplicada, que según palabras de Baena (2017), se orienta a resolver problemas concretos mediante soluciones prácticas y acciones inmediatas. Aunque puede contribuir al conocimiento teórico, su enfoque está limitado a casos específicos, sin ser generalizable a otras situaciones (p. 18).

Finalmente, se consideró utilizar la investigación de campo, donde se emplean técnicas de investigación con la intención recolectar datos, donde según Baena (2017), el propósito es recopilar y registrar de manera ordenada los datos correspondientes al tema de estudio seleccionado, empleando técnicas como la observación y la interrogación (p. 70).

3.2. Unidades de análisis

En el presente proyecto de titulación, se aprovecharon los datos brindados por el gerente de la distribuidora de agua Still Water, que cuenta con una población finita de 84 clientes, como se evidencia en el Anexo 7, y una población de 8 conductores. La población, según la define Vizcaíno (2023), como el conjunto total de individuos, elementos o

fenómenos que comparten una característica en común y que son considerados dentro de una investigación (p. 9746). En este trabajo se empleó un muestro censal.

3.3. Técnicas e instrumentos de investigación

Las técnicas que se implementaron en esta investigación, son la entrevista y la encuesta de opinión. En palabras de Baena (2017), se describe la entrevista como un proceso de investigación dirigido a una o más personas, con el fin de obtener testimonios sobre experiencias personales, opiniones o comentarios (p. 80). Además, Hernández y Mendoza (2018), afirman que la entrevista, es una técnica cualitativa que permite obtener información más detallada mediante la interacción entre entrevistador y entrevistado (p. 449). En consecuencia, se llevó a cabo la entrevista al director general, el cual proporcionó con información adicional para el desarrollo del producto.

En relación con esto, Baena (2017) afirma que, la encuesta es una técnica que utiliza un cuestionario dirigido a una muestra representativa, con el objetivo de obtener información que permita identificar características o comportamientos de la población completa (p. 82). En relación con esto, Hernández y Mendoza (2018) afirma que, un cuestionario consiste en una serie de preguntas vinculadas con una o más variables que se pretende analizar (p. 250). De este modo, las encuestas enfocadas a clientes y empleados se estructuraron con preguntas cerradas.

3.4. Técnicas de análisis de datos

A partir de la hipótesis planteada, se empleó un enfoque estadístico descriptivo para examinar los datos obtenidos y ofrecer una síntesis de su comportamiento. Se empleó *Google Forms* como herramienta para la recolección de datos a través de encuestas, mientras que se usó la herramienta de *Microsoft Excel* para organizar, tabular los datos y la creación de gráficas. Los resultados obtenidos a partir de las encuestas se presentan en forma de porcentajes, lo que facilita su análisis e interpretación.

En relación con esto, Hernández y Mendoza (2018) afirma que, cuando los datos han sido codificados, corregidos y organizados en una matriz, es posible realizar un análisis empleando un programa de computadora (p. 312). Finalmente, para el análisis de las encuestas se hizo uso del *software IBM SPSS*, empleando el modelo de regresión logístico binario.

3.5. Operacionalización de las variables

Tabla 1. Aplicación Móvil – Variable Independiente

Conceptualizaciones	Dimensión	Indicadores	Preguntas	Herramienta
De acuerdo con Corilla (2022), que citó a Geddes (2004), el concepto de "App móvil" se refiere a un tipo de software de aplicación creado exclusivamente para dispositivos móviles, como smartphones y tabletas, y está diseñado para sacar provecho de las capacidades y características propias de estos equipos, tales como la interfaz táctil y la conectividad a redes inalámbricas (p. 19).	Tipos de aplicaciones	Nativas	¿Con qué frecuencia solicita productos por medio de una aplicación?	Encuesta a los clientes
		Híbridas	¿Qué tan importante considera que las instrucciones para las rutas de distribución diarias en la aplicación móvil sean claras? ¿Qué tan de acuerdo está con la implementación de una aplicación móvil y web que asigne a los clientes en sectores o zonas específicas para optimizar las rutas de distribución?	Encuesta a los empleados
	Desarrollo Frontend	Flutter	¿Qué tan difícil le resulta acceder a internet?	Encuesta a los clientes y empleados
		React Native	¿Qué tan importante considera que una aplicación sea intuitiva y requiera poco entrenamiento para su uso?	Encuesta a los empleados
		Xamarin	¿Cuál considera que sería el principal beneficio de tener una aplicación móvil para facilitar tareas cotidianas, como la gestión y optimización de las rutas de distribución de agua para la empresa?	Entrevista al gerente
		Kotlin	¿Qué tipo de experiencia espera que los clientes tengan cuando utilizan la aplicación móvil para solicitar sus productos?	
	Sistemas operativos	Android	¿Tiene a su disposición un dispositivo inteligente?	Encuesta a los clientes y empleados
		iOS	¿Cuál es el sistema operativo de su dispositivo móvil?	

Tabla 2. Aplicación Web – Variable Independiente

Conceptualizaciones	Dimensión	Indicadores	Preguntas	Herramienta
En palabras de Cardador (2024), menciona a una aplicación web como un programa de computador codificado en un lenguaje de programación compatible con navegadores web, permitiendo la interacción del usuario con el servidor web. El desarrollo de aplicaciones web se realiza utilizando lenguajes de programación o a su vez Entornos de Desarrollo Integrado (IDE), donde se diseña la aplicación web mediante código y se prueba con herramientas que dispone para garantizar que el código sea correcto y cumpla con sus objetivos (p. 139).	Arquitecturas de aplicaciones web	Arquitectura de tres niveles Arquitectura modelo vista controlador	¿Qué tan importante consideraría que la aplicación tenga una versión web complementaria para gestionar la información de los clientes y conductores?	Entrevista al gerente
	Lenguajes de desarrollo web	HTML CSS JavaScript C# PHP	¿Está de acuerdo en que la aplicación web debe ser ligera y optimizada para cargar rápidamente?	Encuesta a los empleados
	Tecnologías de desarrollo web	AJAX jQuery	¿Qué tan importante consideraría que los datos en la aplicación web se actualicen automáticamente en tiempo real?	Encuesta a los empleados
	Desarrollo Frontend	Bootstrap Angular Vue.js	¿Está de acuerdo en que la aplicación web debe ser capaz de adaptarse a distintos formatos de dispositivos?	Encuesta a los empleados
	Desarrollo Backend	Express.js Django Laravel	¿Qué tan importante es para usted la privacidad y seguridad de los datos cuando usa aplicaciones móviles y web?	Entrevista al gerente
	Modelos de bases de datos	Modelos de datos jerárquicos Modelo relacional Modelo de datos orientado a objetos	¿Qué tan importante considera que es tener datos fiables y actualizados de los clientes?	Encuesta a los empleados
	Sistema gestor de la base de datos	MySQL MongoDB Oracle	¿Qué beneficios cree que le puede proporcionar a la empresa contar con una base de datos actualizada de sus clientes?	Entrevista al gerente

Tabla 3. Optimización de Rutas de Distribución – Variable Dependiente

Conceptualizaciones	Dimensión	Indicadores	Preguntas	Herramienta	
Según Sarder (2021) en la actualidad las redes de transporte requieren un análisis y modelado cuidadoso para ajustarse a las necesidades particulares de cada empresa. Este proceso debe tomar en cuenta diversos factores, tales como el tipo de producto que se transporta y el destino al cual se dirige (p. 57).	Tipos de rutas	Fijas	¿Cuáles son los procesos principales para la planificación y coordinación de las entregas dentro de la empresa? ¿Cómo se maneja la planificación de rutas de distribución cuando la demanda es alta?	Entrevista al gerente	
		Dinámicas	¿Cuáles son los principales desafíos que enfrentan los empleados en las rutas de distribución de los productos?		
			¿Cuál es la cantidad de bidones de agua que consume semanalmente en la actualidad?		Encuesta a los clientes
	¿Cómo evalúa actualmente la eficiencia en la distribución de agua de la empresa Still Water?				
	¿Qué tan difícil le resulta modificar la ruta al incorporar un cliente nuevo?		Encuesta empleados		
	Tipos de flotas	Propia	¿Cuántas son las rutas de distribución que maneja la empresa y con qué frecuencia se cubren?	Entrevista al gerente	
		Subcontratada	¿Cómo se evalúa el rendimiento de los conductores y cuáles son los criterios más importantes para medir su desempeño?	Entrevista al gerente	
		Algoritmos de camino más corto	A*	¿Con qué frecuencia ha recibido quejas de clientes que no se entregó el servicio? Actualmente, ¿Qué tan de acuerdo está con que la gestión y planificación de rutas de distribución se controla de manera eficiente?	Encuesta a los empleados
			Dijkstra	¿Qué tan satisfecho está con la distancia y los recorridos que realiza utilizando los métodos actuales de planificación de rutas?	
			Bellman-Ford	¿Qué tan importante es para usted contar con una aplicación que le informe sobre los detalles de su pedido en todo momento?	
			Prim	¿Con qué frecuencia ha experimentado retrasos significativos en la entrega de agua?	
			Optim. colonias de hormigas	¿Qué tan satisfecho está con el servicio general de entrega de agua de la empresa Still Water?	
		Floyd-Warshall	¿Considera que un sistema de gestión y optimización de rutas de distribución podría mejorar la eficiencia del servicio y aumentar la satisfacción del cliente?	Entrevista al gerente	
Johnson					

4. RESULTADOS

4.1. Resultado del primer objetivo específico

4.1.1. Resultados de la entrevista al gerente de la empresa

Pregunta 1: ¿Cuál considera que sería el principal beneficio de tener una aplicación móvil para facilitar tareas cotidianas, como la gestión y optimización de las rutas de distribución de agua para la empresa?

Respuesta: Considero que tiene muchos beneficios relevantes, mencionaré algunos de los que considero más importantes: que el servicio sea más rápido y se proveche mejor al personal, mejorar los gastos logísticos, mayor cantidad de entregas y a su vez mayor producción.

Pregunta 2: ¿Qué tipo de experiencia espera que los clientes tengan cuando utilizan la aplicación móvil para solicitar sus productos?

Respuesta: Esperaría que la experiencia del cliente al usar la aplicación sea intuitiva y fluida. Queremos que perciba la aplicación como confiable y que los pedidos los pueda realizar rápidamente.

Pregunta 3: ¿Qué tan importante consideraría que la aplicación tenga una versión web complementaria para gestionar la información de los clientes y conductores?

Respuesta: Considero que es muy importante, poder acceder a la aplicación desde cualquier lugar y así tener mayor control sobre los clientes, sus pedidos, los conductores, entre otros.

Pregunta 4: ¿Qué tan importante es para usted la privacidad y seguridad de los datos cuando usa aplicaciones web o móviles?

Respuesta: Considero que la privacidad y seguridad son fundamentales en una aplicación y más aún cuando en la aplicación se maneja información personal o delicada.

Pregunta 5: ¿Qué beneficios cree que le puede proporcionar a la empresa contar con una base de datos actualizada de sus clientes?

Respuesta: El mayor beneficio que considero al tener una base de datos actualizada de los clientes es el poder darles seguimiento a sus pedidos, enviar recordatorios del servicio y aumentar la fidelización del cliente.

Pregunta 6: ¿Cuántas son las rutas de distribución que maneja la empresa y con qué frecuencia se cubren?

Respuesta: No contamos con rutas de distribución fijas, en la empresa consideramos un enfoque de rutas dinámicas donde se basa en la demanda del cliente, sin embargo, si contamos con ciertas empresas como clientes, las cuales si visitamos con cierta frecuencia aun así no nos hayan solicitado el servicio.

Pregunta 7: ¿Cuáles son los procesos principales para la planificación y coordinación de las entregas dentro de la empresa?

Respuesta: El proceso es todo manual, no tenemos un plan de rutas específico, recibimos los pedidos de los clientes por *WhatsApp*, se carga el vehículo con bastantes productos, mientras se carga el vehículo revisamos y analizamos manualmente cual es el posible pedido más cercano, realizamos una ruta mentalmente, y luego nos dirigimos a distribuir los pedidos en la forma en la que consideramos que es más rápida.

Pregunta 8: ¿Cómo se maneja la planificación de rutas de distribución cuando la demanda es alta?

Respuesta: El proceso es el mismo, recibimos los pedidos por *WhatsApp*, cargamos el vehículo y luego empezamos la distribución de los pedidos en la forma en la que consideramos que es más rápida. Un factor por considerar es que cuando la demanda es alta no nos alcanzamos a cubrir con todos los pedidos dentro de nuestro horario laboral.

Pregunta 9: ¿Cuáles son los principales desafíos que enfrentan los empleados en las rutas de distribución de los productos?

Respuesta: Los empleados enfrentan varios desafíos, dos desafíos son muy frecuentes y considero importantes a mejorar, el primero es la mala comunicación entre la persona de ventas que administra el *WhatsApp* de los pedidos y el conductor. Hay casos en los que surge un nuevo pedido, lo podemos considerar un pedido intermedio, cerca de donde se encuentra el conductor. Sin embargo, el conductor se demora en atender al nuevo pedido que se le envía por mensajes de *WhatsApp*, y se aleja de la ubicación hasta que se percata del nuevo pedido, y debe de regresar a un lugar cercano de donde estaba anteriormente. Este desafío es muy frecuente y nos dificulta ser más eficientes.

El segundo desafío es cuando considero importante es cuando el conductor, con el ajetreo constante de nuevos pedidos a veces se pasa por alto una entrega y no la realiza, lo que genera quejas y malestar en los clientes.

Pregunta 10: ¿Cómo se evalúa el rendimiento de los conductores y cuáles son los criterios más importantes para medir su desempeño?

Respuesta: Evaluamos el rendimiento de los conductores en términos de cantidad de entregas que realizan por horas. Siendo los criterios más importantes para medir su desempeño: el tiempo, la distancia y la cantidad de entregas que realizan.

Pregunta 11: ¿Considera que un sistema de gestión y optimización de rutas de distribución podría mejorar la eficiencia del servicio y aumentar la satisfacción del cliente?

Respuesta: Si, por supuesto, considero que un sistema de gestión y optimización de rutas de distribución mejoraría en gran medida actividades que son de gran impacto para la empresa. Actualmente dichas actividades al llevarlas de forma manual dan paso a fallos humanos y disminuye nuestro nivel de eficiencia. Además, también considero que aumentaría la satisfacción del cliente puesto que le ofreceríamos un servicio más ágil.

4.1.2. Análisis e Interpretación de la entrevista al gerente

A partir de la entrevista realizada al gerente de la empresa distribuidora de agua *Still Water*, de identificó una deficiencia en los procesos de gran impacto para el tipo de negocio,

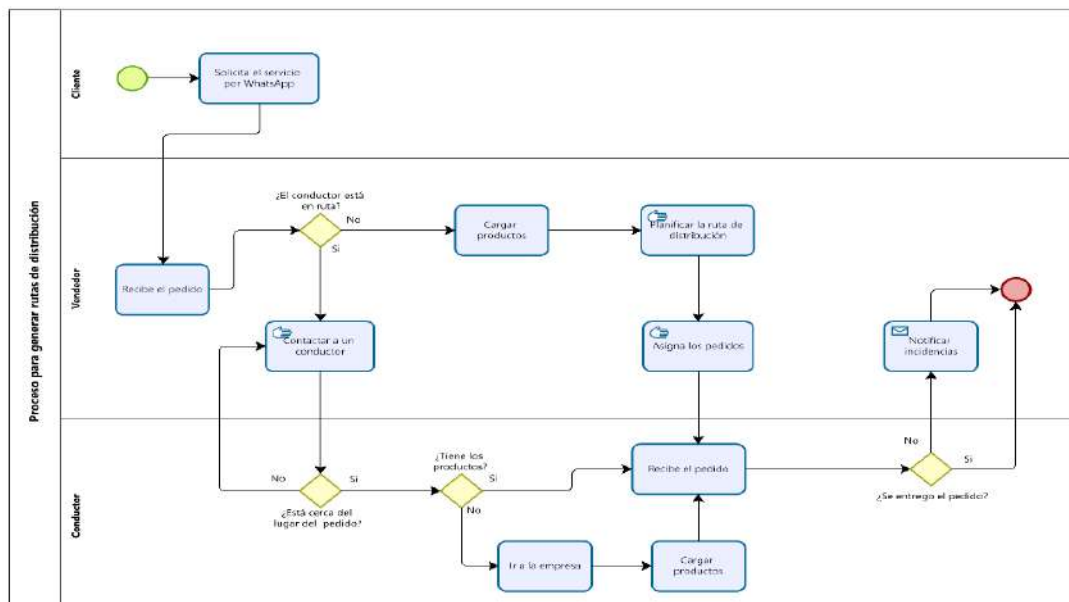
como la distribución de sus productos, actualmente se gestiona los pedidos por medio de la aplicación de mensajería *WhatsApp*. Estos pedidos pasan del cliente al vendedor y del vendedor al conductor, lo que suele resultar en una comunicación ineficiente y retrasos en las entregas. Además, enfrentan desafíos para modificar su trayecto constantemente, cuando hay nuevos pedidos mientras está en ruta, como resultado, se producen errores humanos y disminución de la eficiencia en el servicio.

El gerente demostró estar abierto hacia la posibilidad de implementar una aplicación para optimizar las rutas de distribución de agua, con la finalidad de aumentar la eficiencia del servicio y la satisfacción de sus clientes.

4.1.3. Diagrama de flujo de actividades del proceso de generación de rutas

Se determinó el flujo de actividades que realiza la distribuidora, desde el momento de recibir los pedidos, carga de vehículo, proceso de distribución y entrega. Utilizando la herramienta *BPMN* de *Bizagi*, se representó en la figura 5, la intervención de cada parte involucrada en el proceso de distribución.

Figura 5. Diagrama del flujo de actividades BPMN



4.1.4. Resultados de encuesta a los clientes







Tabla 4. Resultados pre test de los clientes








N°	Preguntas	Escala y %				Figuras	
		Sí	No				
1	¿Tiene a su disposición un dispositivo inteligente?	97,60%	2,40%				
2	¿Cuál es el sistema operativo de su dispositivo móvil?	Android	iOS	Windows Phone	Otro		
3	¿Qué tan difícil le resulta acceder a internet?	Muy difícil	Difícil	Moderado	Fácil	Muy fácil	
4	¿Cuál es la cantidad de bidones de agua que consume semanalmente en la actualidad?	1	2 a 4	4 a 7	7 a 10	Más de 10	
5	¿Con qué frecuencia ha experimentado retrasos significativos en la entrega de agua?	Siempre	Frecuentemente	A veces	Rara vez	Nunca	
6	¿Cómo evalúa actualmente la eficiencia en la distribución de agua de la empresa Still Water?	Muy eficiente	Eficiente	Moderadamente eficiente	Poco eficiente	Nada eficiente	
7	¿Qué tan satisfecho está con el servicio general de entrega de agua de la empresa Still Water?	Muy satisfecho	Satisfecho	Neutral	Insatisfecho	Muy insatisfecho	
8	¿En qué rango de edad se encuentra?	18-25	26-30	31-40	40 o más		

Nota. Fuente: Datos registrados en el pre test enfocada a los clientes

4.1.5. Resultados de encuesta a los empleados de la empresa

Tabla 5. Resultados pre test de los empleados

N°	Preguntas	Escala y %					Figuras
		Sí	No				
1	¿Tiene a su disposición un dispositivo inteligente?	100,00%	0,00%				
2	¿Cuál es el sistema operativo de su dispositivo móvil?	Android	iOS	Windows Phone	Otro		
2	¿Qué tan de acuerdo está con la implementación de una aplicación móvil y web que asigne a los clientes en sectores o zonas específicas para optimizar las rutas de distribución?	Totalmente de acuerdo	De acuerdo	Neutral	En desacuerdo	Totalmente en desacuerdo	
3	¿Qué tan importante considera que las instrucciones para las rutas de distribución diarias en la aplicación móvil sean claras?	Muy importante	Importante	Neutral	Poco importante	Nada importante	
4	¿Qué tan importante considera que es tener datos fiables y actualizados de los clientes?	Muy importante	Importante	Neutral	Poco importante	Nada importante	
5	¿Qué tan importante consideraría que los datos en la aplicación web se actualicen automáticamente en tiempo real?	Muy importante	Importante	Neutral	Poco importante	Nada importante	
6	¿Está de acuerdo en que la aplicación web debe ser	Totalmente de acuerdo	De acuerdo	Moderadamente de acuerdo	Poco de acuerdo	Nada de acuerdo	

	ligera y optimizada para cargar rápidamente?	100,00%	0,00%	0,00%	0,00%	0,00%	
7	¿Está de acuerdo en que la aplicación web debe ser capaz de adaptarse a distintos formatos de dispositivos?	Totalmente de acuerdo	De acuerdo	Moderadamente de acuerdo	Poco de acuerdo	Nada de acuerdo	
		87,50%	12,50%	0,00%	0,00%	0,00%	
8	¿Qué tan difícil le resulta modificar la ruta al incorporar un cliente nuevo?	Muy difícil	Difícil	Moderado	Fácil	Muy fácil	
		0,00%	62,50%	25,00%	12,50%	0,00%	
9	¿Con qué frecuencia ha recibido quejas de clientes que no se entregó el servicio?	Siempre	Frecuentemente	A veces	Rara vez	Nunca	
		0,00%	37,50%	37,50%	25,00%	0,00%	
10	¿Qué tan importante es para usted contar con una aplicación que le informe sobre los detalles de su pedido en todo momento?	Muy importante	Importante	Neutral	Poco importante	Nada importante	
		87,50%	12,50%	0,00%	0,00%	0,00%	
11	¿Qué tan satisfecho está con la distancia y los recorridos que realiza utilizando los métodos actuales de planificación de rutas?	Muy satisfecho	Satisfecho	Neutral	Insatisfecho	Muy insatisfecho	
		25,00%	0,00%	12,50%	62,50%	0,00%	
12	Actualmente, ¿Qué tan de acuerdo está con que la gestión y planificación de rutas de distribución se controla de manera eficiente?	Totalmente de acuerdo	De acuerdo	Moderadamente de acuerdo	Poco de acuerdo	Nada de acuerdo	
		25,00%	0,00%	25,00%	50,00%	0,00%	

Nota. Fuente: Datos registrados en el pre test enfocada a los empleados

4.1.6. Análisis e Interpretación de los resultados de las encuestas a los clientes

Por medio de las encuestas que se llevaron a cabo a los 84 clientes de la empresa, se consiguieron los resultados correspondientes al *pre test*, donde se evidenció un gran porcentaje de los encuestados constantemente sufren retrasos significativos en la entrega del agua. Además, consideran que la empresa es poco o moderadamente eficiente en la distribución de agua. Asimismo, los clientes indicaron que no se sienten satisfechos con el servicio en general que ofrece la empresa *Still Water*.

4.1.7. Análisis e Interpretación de los resultados de las encuestas a los empleados

Después de aplicar la encuesta a los 8 empleados de la empresa *Still Water*, en la tabla 5 hizo se puede ver que los conductores están de acuerdo con una aplicación que asigne a los clientes en sectores para optimizar las rutas de distribución. Además, consideran muy importante que las indicaciones sobre las rutas en la aplicación sean claras, asimismo, consideran importante que la aplicación les indique en todo momento los detalles de los pedidos que transporta. Por otro lado, los empleados afirman en tener complicaciones para modificar su ruta al momento de incorporar un cliente nuevo, afirmando que no están de acuerdo en los métodos actuales para la planificación de rutas de distribución, evidenciando la necesidad de implementar una aplicación móvil y una aplicación web que solvante los problemas anteriormente ya mencionados.

4.2. Resultado del segundo objetivo específico

4.2.1. Tecnologías y herramientas

4.2.1.1. Sistema gestor de bases de datos para la aplicación web y la aplicación móvil

Para poder almacenar los datos y manipularlos, se debe implementar un sistema gestor de base de datos (SGBD), de tal manera que se hizo la comparativa de los gestores de datos más empleados y aceptados como se refleja en la tabla 6.

Tabla 6. Comparativa de SGBD

Características	Mongo ^a	MySQL ^b	Oracle ^c
Lenguaje de consulta	<i>MongoDB Query Language</i>	SQL	SQL y PL/SQL
Modelo de datos	NoSQL	Relacional	Relacional
Escalabilidad	Horizontal	Vertical y horizontal	Horizontal y vertical principalmente
Tipo de datos	Documentos JSON	Basado en tablas	Tablas relacionales con soporte de JSON y XML
Uso principal	Big data y análisis en tiempo real	Aplicaciones transaccionales y de almacenamiento estructurado	Aplicaciones empresariales y de misión crítica
Transacciones	BASE	ACID	ACID
Licencia	Open source y comercial	Open source y comercial	Comercial
Gestión de datos	Semiestructurado o sin esquema	Estructurado con esquema rígido	Estructurado con esquemas flexibles

Nota: Obtenido de: ^aMongoDB (2024), ^bOracle (2024b), ^cOracle (2024a)

Gracias al análisis presentado de la tabla 6, se optó por el sistema gestor de bases de datos *MySQL*, ya que permite un almacenamiento estructurado, necesario para el tipo de proyecto, además ofrecer un rendimiento eficiente y facilidad de uso.

4.2.1.2. Desarrollo backend para aplicación web y móvil

Con respecto a la organización de propiedades, funciones y lógica de la aplicación móvil y la aplicación web, se analizó diversos *frameworks*, los cuales permiten un desarrollo ágil para crear *APIs*, aplicaciones moderna y servicios en tiempo real.

Tabla 7. Comparativa del desarrollo backend

Características	Express.js ^a	Django ^b	Laravel ^c
Lenguaje	<i>JavaScript (Node.js)</i>	<i>Python</i>	<i>PHP</i>
Arquitectura	Flexible, Modelo-Vista-Controlador (MVC)	Modelo-Vista-Plantilla (MVT)	Modelo-Vista-Controlador (MVC)
Curva de aprendizaje	Moderada; requiere conocimiento de <i>JavaScript</i>	Moderada; requiere conocimiento de <i>Python</i>	Moderada; requiere conocimiento de <i>PHP</i>
Rendimiento	Alto; eficiente en manejo de múltiples solicitudes	Alto; optimizado para aplicaciones complejas	Moderado a alto; depende de la optimización
Escalabilidad	Alta	Alta	Alta
Seguridad	Depende de la implementación del desarrollador	Alta; incluye medidas de seguridad integradas	Alta; incluye medidas de seguridad integradas

Comunidad y soporte	Amplia y activa	Amplia y activa	Amplia y activa
Facilidad de integración	Alta; compatible con múltiples herramientas y servicios	Alta; incluye muchas funcionalidades integradas	Alta; incluye muchas funcionalidades integradas
Uso común	APIs y aplicaciones en tiempo real	Aplicaciones web complejas y de gran escala	Aplicaciones web con funcionalidades complejas

Nota: Obtenido de: ^aOpenJS (2024), ^bDjango (2024), ^cLaravel Holdings Inc. (2024)

Se decidió usar el *framework Express.js* en el *backend* para la creación de APIs debido a su alto rendimiento en el manejo de múltiples solicitudes y por su arquitectura muy flexible.

4.2.1.3. Arquitectura de desarrollo para aplicaciones web y móviles

Un factor de vital importancia para asegurar que cualquier aplicación web o móvil se desarrolle y se florezca, es la elección correcta de una arquitectura de desarrollo.

Desarrollar el programa con buenas prácticas y una arquitectura limpia y congruente garantiza, escalabilidad, además de un mejor mantenimiento y potencia el rendimiento y la seguridad.

Tabla 8. Comparativa de arquitecturas de desarrollo

Características	Arquitectura de tres capas ^a	Arquitectura Modelo-Vista-Controlador (MVC) ^b
Capas/Componentes	Capa de presentación, Capa de negocio, capa de datos	Modelo, vista, controlador
Objetivo	Cada capa agrupa funciones concretas, lo que favorece una clara división de responsabilidades y facilita la modularidad del <i>software</i> .	Distribuir la lógica de negocio de la interfaz de usuario para facilitar el desarrollo modular y el mantenimiento
Funcionalidad por nivel/componente	Capa de presentación: En este nivel es donde el usuario interactúa con la aplicación	Modelo: maneja los datos de la aplicación
	Capa de negocio: Es el nivel lógico, el núcleo de la aplicación.	Vista: muestra los datos del modelo y refleja los cambios en la interfaz de usuario
	Capa de datos: Es donde se guarda y gestiona los datos procesados por la aplicación	Controlador: procesa las entradas del usuario y actualiza el modelo o la vista según se requiera

Nota: Obtenido de: ^aLópez (2016), ^aUrbano (2015), ^bSarrion (2022)

Por medio de la comparación de estas dos arquitecturas, se decidió por implementar la arquitectura *MVC*, gracias a su enfoque de separar las responsabilidades, la reutilización de código, la facilidad de mantenimiento y realización de pruebas unitarias.

4.2.1.4. Desarrollo frontend para la aplicación móvil

La variedad de herramientas disponibles para el desarrollo *frontend* permite a los desarrolladores crear interfaces de usuario que son tanto dinámicas como fáciles de usar. Con el objetivo de identificar el *framework* más adecuado para el proyecto, se realizó una comparativa de los tres *frameworks* móviles más utilizados.

Tabla 9. Comparativa de frameworks frontend para desarrollo móvil

Características	Flutter ^a	React Native ^b	Xamarin ^c
Lenguaje de programación	<i>Dart</i>	<i>JavaScript</i>	<i>C#</i>
Desarrollador	<i>Google</i>	<i>Meta</i>	<i>Microsoft</i>
Arquitectura	Compilación directa a código nativo	Puente <i>JavaScript</i> para comunicación con componentes nativos	Compilación a código intermedio y luego a código nativo
Rendimiento	Alto; compilación nativa y motor gráfico propio	Bueno; depende del puente <i>JavaScript</i> y puede requerir optimizaciones	Alto; cercano al rendimiento nativo
Componentes de UI	Amplia gama de <i>widgets</i> personalizables propios	Utiliza componentes nativos	Acceso a controles nativos y personalizados
Soporte multiplataforma	<i>iOS, Android, Web, Windows, macOS, Linux</i>	<i>iOS, Android, Web</i>	<i>iOS, macOS, Android, Windows</i>
Casos de uso comunes	Aplicaciones con interfaces personalizadas y alto rendimiento	Aplicaciones que requieren desarrollo rápido y reutilización de código <i>web</i>	Aplicaciones empresariales con integración en el ecosistema <i>Microsoft</i>

Nota: Obtenido de: ^aGoogle (2024b), ^bMeta Platforms (2024), ^cMicrosoft (2024)

Con un enfoque en el desarrollo ágil, se optó por elegir *Flutter* gracias a su capacidad para desarrollar aplicaciones multiplataforma desde una única base de código. Además, cuenta con herramientas como "*hot reload*" que facilita la visualización en tiempo real sin necesidad de reiniciar la aplicación agilizando el proceso de desarrollo y prueba.

4.2.1.5. Desarrollo CSS para la aplicación web

Las herramientas de desarrollo CSS no solamente ayudan en apariencia y funcionalidad de la aplicación *web*, sino que, además, brindan eficiencia y escalabilidad al proyecto, a continuación, se presenta una tabla con la comparación de características de tres herramientas para agilizar y mejorar significativamente el desarrollo en una aplicación *web*.

Tabla 10. Comparativa de frameworks CSS para desarrollo web

Características	Bootstrap ^a	TailwindCSS ^b
Arquitectura	Basado en una estructura de grilla responsiva y componentes	Utiliza clases utilitarias para diseñar sin necesidad de componentes predefinidos
Lenguaje Principal	CSS, HTML, JavaScript	CSS, HTML, JavaScript
Descripción	Framework CSS para diseño responsivo y componentes reutilizables	Framework CSS basado en utilidades para una mayor personalización y flexibilidad
Soporte para Aplicaciones Móviles	Ofrece diseño responsivo adaptable	Soporte completo mediante clases utilitarias para adaptabilidad móvil
Rendimiento	Ligero, depende de la implementación	Mayor rendimiento debido a la generación de CSS optimizado y eliminación de estilos no usados

Nota: Obtenido de: ^aBootstrap (2024), ^bTailwind Labs Inc. (2024)

Con relación a la tabla comparativa de los *frameworks* CSS para desarrollo web, se optó por *Bootstrap* debido a su rápida implementación y facilidad de uso. Además, posee amplia variedad de componentes y su diseño basado en grillas, facilita que las interfaces se adapten de forma automática a diversos dispositivos.

4.2.1.6. Selección del algoritmo

Para lograr la correcta optimización de rutas de distribución de manera eficiente y automatizada, es necesario optar por la implementación con un algoritmo del camino más corto, en la tabla 11 se compara los algoritmos de generación del camino más corto y más usados.

Tabla 11. Comparativa de los algoritmos de ruta más corta

Características	A*	Dijkstra	Bellman-Ford	Floyd-Warshall
Tipo de grafo	Grafos sin aristas negativas	Grafos sin aristas negativas	Grafos con aristas negativas permitidas	Grafos con o sin aristas negativas

Ciclos negativos	No permite	No permite	Detecta ciclos negativos, reporta si existe	Detecta y permite ciclos negativos
Heurística usada	Depende del problema	No requiere	No aplica	No aplica
Alcance de la solución	Camino más corto desde un nodo al objetivo	Camino más corto desde un nodo	Camino más corto desde un nodo	Camino más corto entre todos los pares
Escenario de uso	Objetivo conocido (con heurística adecuada)	Grafos grandes sin aristas negativas	Grafos con algunas aristas negativas	Redes densas o con análisis exhaustivo
Precisión de resultados	Aproximada, depende de la heurística	Exacta	Exacta	Exacta

Nota: Obtenido de: Kumawat (2021)

En función a la tabla comparativa entre los distintos algoritmos de camino más corto, se optó por la utilización del algoritmo *Dijkstra*, esta elección se basa en que la precisión de sus resultados es exacta, puede manipular grandes grafos y no requiere de una heurística para su mejoramiento. Además, añade valor que es un algoritmo ampliamente usado por diversos motores de rutas debido a su exactitud en resultados. Como menciona Urquhart (2022), este algoritmo es usado y empleado por el motor de ruta *Google Maps*, el cual ha invertido mucho esfuerzo en lograr ofrecer rutas optimizadas empleando una variedad de técnicas como almacenamiento en caché, jerarquías de contracción y jerarquía de autopistas, logrando alimentar e implementar de manera más eficaz el algoritmo (p. 188).

4.3. Resultado del tercer objetivo específico

4.3.1. Nomenclatura y Logotipo

Se decidió utilizar la nomenclatura de “ANRO” por motivos de simplicidad y facilidad de memorizar. Este nombre corto fue seleccionado porque mejora la experiencia del usuario, facilitando la identificación de la plataforma y su integración en interfaces. Esto optimiza su funcionalidad como solución de *software* en la gestión logística. El logotipo correspondiente puede observarse en la figura 6.

Figura 6. Logotipo del sistema diseñado en la aplicación Adobe Illustrator



4.3.2. Marco de Trabajo Scrum

Según Schwaber y Sutherland (2020), este marco de trabajo incentiva la transparencia, la inspección y la adaptación de forma continua, generando elementos esenciales para responder a los cambios y necesidades que puedan surgir en cada fase de desarrollo. Por lo que, la estructura de *Scrum* está basada en ciclos cortos (*sprints*), y garantizan una revisión constante de la incorporación de mejoras y los avances, lo cual es ideal para proyectos que requieren ajustes continuos para lograr que los objetivos estén alineados a los objetivos de la empresa (pp. 3-4).

4.3.3. Sprint 1

4.3.3.1. Sprint 1 – Planificación

En base a lo mencionado por Schwaber y Sutherland (2020), la etapa inicial en el proceso de desarrollo incluye la planificación, considerando el *sprint* como un evento cuya duración es de un mes o menos, y cada *sprint* comienza inmediatamente después de que finaliza el anterior (pp. 8-9). Por lo tanto, se considera que esta fase es importante para asegurar una implementación exitosa del sistema. En el contexto del marco de trabajo *Scrum*, se ejecutaron dos *sprints* que se estructuraron en función de requerimientos específicos y prioridades del cliente.

4.3.3.1.1. Roles

En *Scrum*, como se muestra en la tabla 12, se asignaron roles específicos para el funcionamiento del marco de trabajo. Según Rubin (2012), el propietario del producto, conocido como *product owner*, es responsable de garantizar que se alcancen los objetivos. Por otro lado, el *scrum master* facilita que el equipo siga el marco de trabajo de manera óptima, siendo un defensor de su aplicación. Finalmente, el equipo de desarrollo se encarga de diseñar y completar el producto final (pp. 15-16).

Tabla 12. Distribución de Roles

Persona	Roles	Área
Patricio Ulloa	<i>Product Owner</i>	Gerente general de <i>Still Water</i>
Mg. Willian Ocampo	<i>Scrum Master</i>	Docente de la PUCESD
Anthony Sacheri	Equipo de Desarrollo y Testeo	Desarrollador <i>Frontend</i> y <i>Tester</i>
Robert Andrade	Equipo de Desarrollo y Testeo	Desarrollador <i>Backend</i> y <i>Tester</i>

4.3.3.1.2. Modelo Vista Controlador

Para evitar la complejidad del código en el desarrollo de la aplicación *web* y la aplicación móvil, se estableció el modelo *MVC* por motivo de que permite dividir la lógica, facilitando la comprensión, modificación y prueba de la aplicación. Esta arquitectura está compuesta por:

Controlador: Procesa las entradas del usuario y actualiza el modelo o la vista según se requiera, siendo el intermediario entre el modelo y la interfaz.

Modelo: Se recurre a la base de datos *SQL MySQL* para manejar los datos de la aplicación.

Vista: En esta se muestra los datos del modelo y refleja los cambios en la interfaz de usuario.

4.3.3.1.3. Parametrización

Con la finalidad de aumentar la adaptación a posibles cambios y mejoras futuras se empleó una parametrización de métodos, clases, *widgets*, constantes, variables, etc., tanto

para la aplicación móvil y la aplicación *web*, esta parametrización se evidencia en la próxima tabla.

Tabla 13. Parametrización

Categoría	Elemento	Nombre (en idioma español)
Controlador	Métodos	crearUsuario()
		guardarDatos()
		actualizarDatos(id)
		inactivarDatos(id)
		activarDatos(id)
Vista (<i>Flutter</i>)	Clase	nombreControlador
	Métodos	nombreMetodo
	Widgets	widNombreWidget
	Pantalla	nombrePantallaScreen
Vista (<i>PHP + HTML</i>)	Elementos <i>HTML</i>	formNombre
	Tabla	tabNombre
Modelo API (<i>Express.js</i>)	<i>Endpoints</i>	<i>GET</i> /listar
		<i>POST</i> /guardar
		<i>PUT</i> /editar/:id
		<i>PUT</i> /inactivar/:id
		<i>PUT</i> /activar/:id
Variables (<i>Flutter</i>)	Constantes	<i>const</i> Nombre
	Variables	<i>final</i> Nombre
Variables (<i>PHP</i>)	Constantes	<i>const</i> Nombre
	Variables	<i>let</i> Nombre
Variables (<i>Express.js</i>)	Constantes	<i>const</i> Nombre
	Variables	<i>let</i> Nombre
		<i>final</i> Nombre

4.3.3.1.4. Control de versiones

Con el objetivo de gestionar y controlar las versiones del proyecto, se utilizó *GitHub*, una plataforma que facilitó la colaboración en equipo de forma simultánea. Además, permitió manipular el repositorio y sus distintas versiones de manera remota, ofreciendo también la oportunidad de compartir y evaluar el trabajo de otros, promoviendo una colaboración segura y eficiente, tal como se ilustran en las figuras 7, 8, 9 y 10.

Figura 7. Plataforma GitHub en la administración de versiones backend

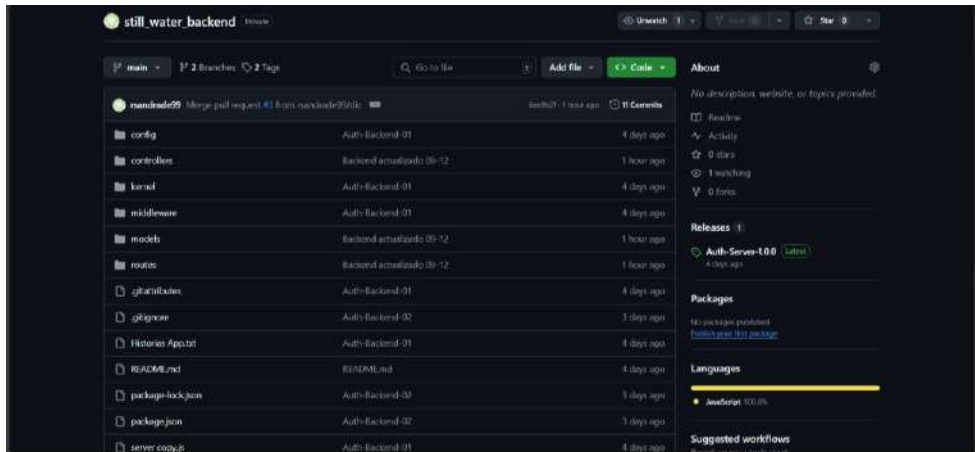


Figura 8. Plataforma GitHub en la administración de versiones frontend web

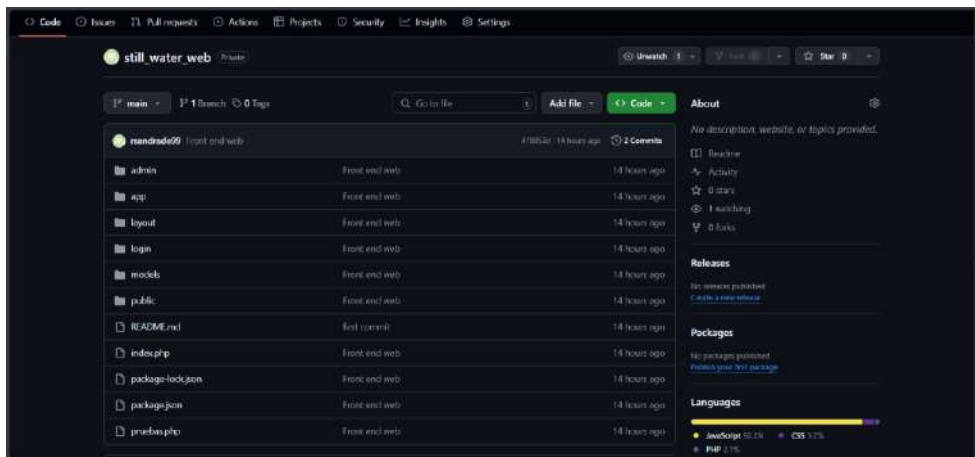


Figura 9. Plataforma GitHub en la administración de versiones frontend móvil cliente

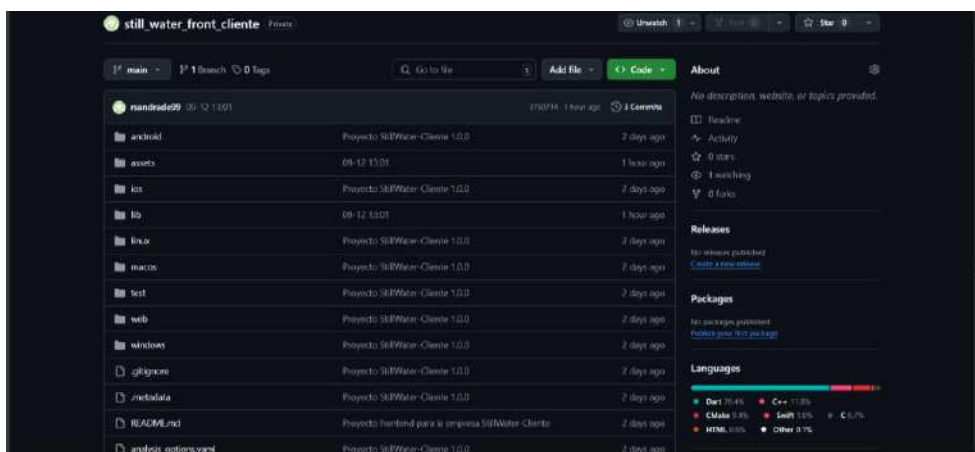
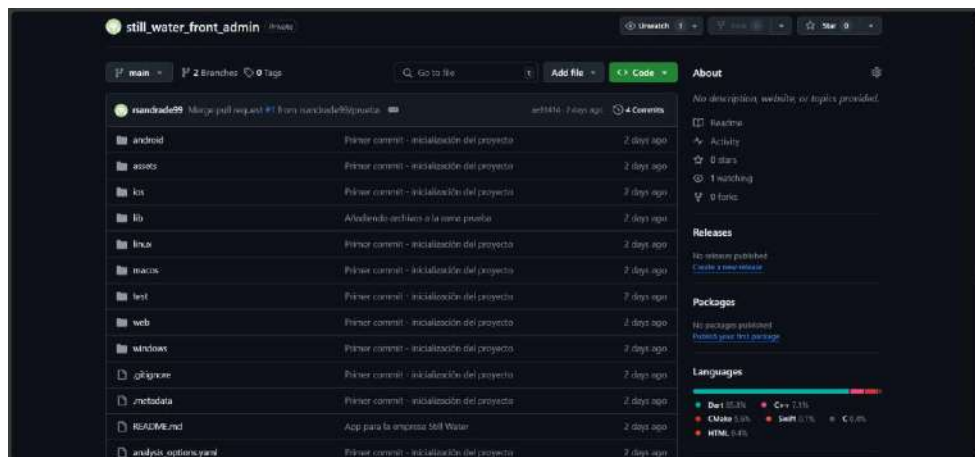


Figura 10. Plataforma GitHub en la administración de versiones frontend móvil gerente



4.3.3.1.5. Product Backlog

En palabras de Schwaber y Sutherland (2020), el *product backlog* es un listado organizado de todo el trabajo pendiente a desarrollar, necesario para mejorar y refinar el producto (p. 11). La tabla 14, indica un listado ordenado de las historias de usuario, respecto a la prioridad de negocio, el riesgo en desarrollo y su estimación de desarrollo.

Tabla 14. Backlog priorizado para el desarrollo del Producto (Product Backlog)

Nº	Historia de Usuario	Prioridad de Negocio	Riesgo en Desarrollo	Estimación
1	Registro de pedido (Aplicación móvil)	100	Bajo	3
2	Registro de conductor (Aplicación web)	95	Bajo	3
3	Registro de vehículo (Aplicación web)	90	Bajo	3
4	Registro de cliente (Aplicación web y móvil)	85	Bajo	3
5	Registro de producto (Aplicación web y móvil)	80	Bajo	3
6	Login (Aplicación web y móvil)	75	Medio	5
7	Visualizar pedidos (Aplicación web)	70	Bajo	3
8	Asignación de pedido a conductor (Aplicación móvil)	65	Medio	3
9	Ruta óptima (Aplicación móvil)	60	Alto	8
10	Visualizar pedidos en el mapa (Aplicación móvil)	55	Bajo	3
11	Historial de pedidos (Aplicación web y móvil)	50	Bajo	2

12	Registro de entrega del pedido (Aplicación móvil)	45	Bajo	2
13	Visualizar perfil (Aplicación web)	40	Bajo	2
14	Editar perfil (Aplicación web)	35	Bajo	2

4.3.3.1.6. Estimación

Para lograr una estimación coherente se la realizó con la métrica de puntos de historias, también, se empleó la técnica de *planning poker* basada en la sucesión *Fibonacci*, posibilitando evaluar la incertidumbre y complejidad de cada historia.

Tal como afirma Rubin (2012), la técnica del *planning poker* está basada en un consenso grupal para estimar el esfuerzo de desarrollo. Durante la sesión el *product owner* presenta y describe cada *item* del *product backlog*, con ayuda del *Scrum master* para lograr mayor comprensión y participación de todo el equipo. Luego cada miembro de desarrollo escoge una tarjeta con un número (escala de estimación *Fibonacci*) que según cada uno representa la complejidad del *item*. Estas tarjetas se revelan entre todos al mismo tiempo, si hay mucha disparidad de complejidad entre los miembros se repite el proceso presentando argumentos de ciertos miembros del grupo de desarrollo, buscando llegar a un número de complejidad en el que todos estén de acuerdo (pp. 129-133).

Tabla 15. Calendario de trabajo

Calendario de Trabajo			
Mes	Semana	Día	Horas
1	4	6	6

4.3.3.1.7. Velocidad de desarrollo

Durante el primer *sprint*, las primeras cinco historias poseen una estimación de 3 puntos cada una, mientras que la sexta historia se estima con 5 puntos y la séptima historia tiene una valoración de 3.

4.3.3.1.8. Escenarios de prueba

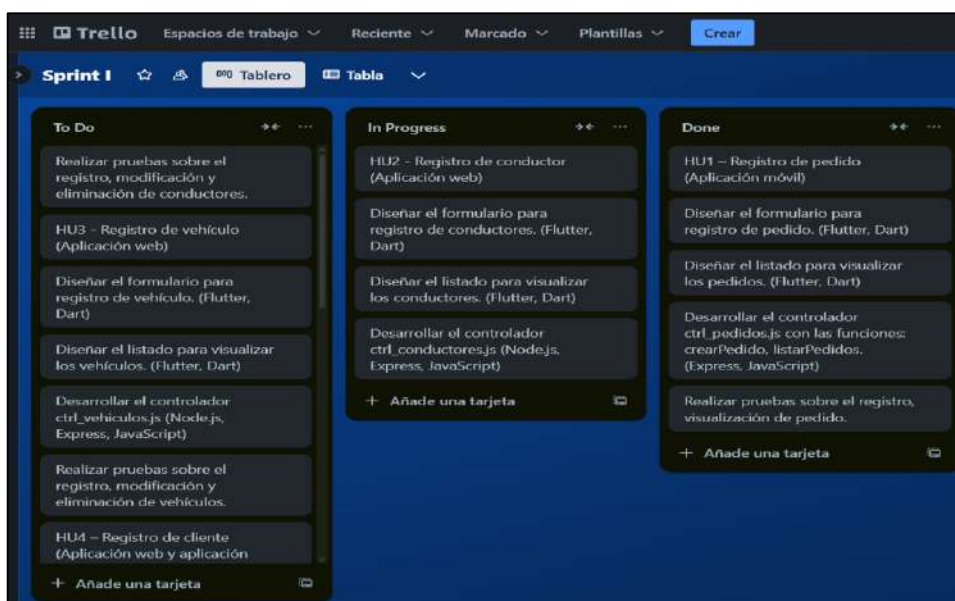
Al terminar las historias de usuario, se debe evaluar el cumplimiento total de los requerimientos que solicita la historia, esto se evidencia en un documento llamado pruebas de aceptación, se encuentra en el Anexo 6.

Tal como afirma Rubin (2012), cada elemento del *product backlog* que se incorpora al *sprint* debe tener un conjunto de criterios de satisfacción, especificados por el *product owner*. Estos criterios de aceptación se verificarán, en última instancia, en pruebas de aceptación que el *product owner* confirma para determinar si el elemento del *backlog* funciona como se ambiciona (p. 77).

4.3.3.1.9. Gestión de incidencias

Para el monitoreo y control de las tareas, de cada historia de usuario, se empleó la plataforma *Trello* para la gestión de proyectos y tareas, esta herramienta está basada en la metodología *Kanban*, se evidencia en la figura 11.

Figura 11. Tablero de Trello



4.3.3.1.10. Sprint Backlog

Para el primer *sprint*, se seleccionó las primeras 7 HU. de mayor prioridad, acumulando un total de 23 puntos, como se puede apreciar en la tabla 16. De acuerdo con lo dicho por Schwaber y Sutherland (2020), el *sprint backlog* es una proyección del tiempo

real del trabajo que los desarrolladores planean invertir con la finalidad de alcanzar el objetivo del *sprint* (p. 11).

Tabla 16. Sprint Backlog 1

Sprint Backlog						
Objetivo: Desarrollar las funciones necesarias para los registros (clientes, conductores, vehículos y productos), autenticar el acceso de usuarios (<i>login</i>) y gestionar pedidos (registro y visualización) tanto en la aplicación web como en la móvil según corresponda.						
Historia usuario	Est	Categ	Tareas de ingeniería	Est	Respon s	Esta
HU1 – Registro de pedido (Aplicación móvil)	3	Diseño	Diseñar el formulario para registro de pedido. (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Diseño	Diseñar el listado para visualizar los pedidos. (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Desarrollo	Desarrollar el controlador <i>ctrl_pedidos.js</i> con las funciones: <i>crearPedido, listarPedidos</i> . (<i>Express, JavaScript</i>)	1	Robert	C
		Prueba	Realizar pruebas sobre el registro, visualización de pedido.	1	Sacheri	C
HU2 - Registro de conductor (Aplicación web)	3	Diseño	Diseñar el formulario para registro de conductores. (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Diseño	Diseñar el listado para visualizar los conductores. (<i>Flutter, Dart</i>)	0.5	Robert	C
		Desarrollo	Desarrollar el controlador <i>ctrl_conductores.js</i> con las funciones: <i>crearConductor, editarConductor, listarConductores, eliminarConductor, activarConductor, getRegistroConductor</i> . (<i>Node.js, Express, JavaScript</i>)	1	Robert	C
		Prueba	Realizar pruebas sobre el registro, modificación y eliminación de conductores.	1	Sacheri	C
HU3 - Registro de vehículo (Aplicación web)	3	Diseño	Diseñar el formulario para registro de vehículo. (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Diseño	Diseñar el listado para visualizar los vehículos. (<i>Flutter, Dart</i>)	0.5	Robert	C
		Desarrollo	Desarrollar el controlador <i>ctrl_vehiculos.js</i> con las funciones: <i>crearVehiculo, listarVehiculos, obtenerVehiculo, editarVehiculo, activarVehiculo, eliminarVehiculo</i> . (<i>Node.js, Express, JavaScript</i>)	1	Robert	C
		Prueba	Realizar pruebas sobre el registro, modificación y eliminación de vehículos.	1	Sacheri	C
HU4 – Registro	3	Diseño	Diseñar el formulario para registro de cliente. (<i>Flutter, Dart</i>)	0.5	Robert	C

de cliente (Aplicación web y aplicación móvil)		Diseño	Diseñar el listado para visualizar los clientes. (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Desarrollo	Desarrollar el controlador <i>ctrl_clientes.js</i> con las funciones: <i>crearCliente, listarClientes, actualizar, eliminarCliente, activarCliente, actualizarContra</i> . (<i>Node.js, Express, JavaScript</i>)	1	Robert	C
		Prueba	Realizar pruebas sobre el registro, modificación y eliminación de clientes.	1	Robert	C
HU5 - Registro de producto (Aplicación web y aplicación móvil)	3	Diseño	Diseñar el formulario para registro de producto. (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Diseño	Diseñar el listado para visualizar los productos. (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Desarrollo	Desarrollar el controlador <i>ctrl_productos.js</i> con las funciones: <i>crearProducto, editarProducto, listarProducto, comboProductos, activarProducto, eliminarProducto, obtenerProducto</i> . (<i>Node.js, Express, JavaScript</i>)	1	Robert	C
		Prueba	Realizar pruebas sobre el registro, modificación y eliminación de productos.	1	Sacheri	C
HU6 - Login (Aplicación web y aplicación móvil)	5	Diseño	Diseñar el formulario para <i>Login</i> (<i>Flutter, Dart</i>)	1	Robert	C
		Desarrollo	Desarrollar el controlador <i>ctrl_autenticacion.js</i> con la función <i>login</i> . (<i>Node.js, Express, JavaScript</i>)	3	Robert	C
		Prueba	Realizar pruebas de <i>Login</i> .	1	Sacheri	C
HU7 - Visualizar pedidos (Aplicación web)	3	Diseño	Diseñar el formulario para Visualizar pedidos (<i>Flutter, Dart</i>)	1	Sacheri	C
		Desarrollo	Desarrollar en el controlador <i>ctrl_pedidos.js</i> la función: <i>listarPedidos</i> . (<i>Node.js, Express, JavaScript</i>)	1	Robert	C
		Prueba	Realizar pruebas de Visualizar pedidos.	1	Robert	C

4.3.3.2. Sprint 1 – Reuniones diarias

Con la finalidad de comprender y evidenciar el trabajo pendiente, se optó por incorporar las reuniones diarias, de una duración máxima de 15min, con el equipo de desarrollo. Según afirma Schwaber y Sutherland (2020), las reuniones diarias promueven la comunicación entre miembros del equipo logrando identificar inconvenientes de manera rápida y favoreciendo a la toma de decisiones, adaptando el *sprint backlog* según sea necesario (p. 9).

4.3.3.2.1. Historia de Usuario 1: Registro de pedido (Aplicación móvil)

En el anexo 5, se encuentra la información completa sobre la historia. En cuanto al desarrollo de la historia 1, se creó el archivo *pedidosFormScreen.dart*, el cual define la interfaz de usuario para gestionar la creación de pedidos dentro de la aplicación. La interfaz presentada en la figura 12, corresponde a un formulario de pedido diseñado en *Flutter*, orientado a la gestión eficiente de solicitudes de productos. Este formulario integra diversos componentes visuales y funcionales que facilitan la experiencia del usuario. En la parte superior, se encuentra un campo destinado a "Indicaciones", el cual permite agregar notas específicas relacionadas con el pedido. Además, incluye una lista desplegable para seleccionar productos, donde se puede visualizar la selección actual, junto con la cantidad correspondiente. También, destacando un campo que muestra el "Total a pagar", y por último un menú desplegable que permite optar por la forma de pago. (Figura 13 y 14).

Figura 12. Pantalla "Formulario de pedido"

The screenshot displays the 'Formulario de pedido' (Order Form) interface. At the top, a blue header bar contains a back arrow and the title 'Formulario de pedido'. Below the header, the form consists of several sections: a text input field for 'Indicaciones'; a 'Producto' dropdown menu currently showing 'Paca de 24 botellas'; two quantity input fields, one for 'Bidón de agua (x2)' and another for 'Paca de 24 botellas (x1)'; a bold text label indicating the total amount: 'Total a pagar: \$7.20'; an 'Información de pago' dropdown menu set to 'Pago en efectivo'; and finally, a prominent blue button at the bottom labeled 'Registrar pedido'.

Figura 13. Código fuente del botón “Registrar pedido” del formulario de pedido

```

1 ElevatedButton(
2   onPressed: () async {
3     if (_formKey.currentState!.validate() &&
4       productosSeleccionados.isNotEmpty) {
5       var pedido = PedidoModel(
6         detallePedido: _indicacionesController.text,
7         infoPago: _informacionPago!,
8         productos: productosSeleccionados.map((item) {
9           return {
10            'id_producto': item['id_producto'],
11            'cantidad': item['cantidad'],
12          };
13        }).toList(),
14        idUsuario: -1,
15        valor_total: _calcularTotalPedido());
16
17       try {
18         setState(() {
19           isLoading = true;
20         });
21
22         bool exito =
23           await _pedidosController.guardarPedido(pedido);
24
25         if (exito) {
26           mostrarCorrecto(context, '¡Se ha realizado el pedido!');
27           setState(() {
28             _formKey.currentState!.reset();
29             productosSeleccionados.clear();
30             _informacionPago = 'Pago en efectivo';
31           });
32         } else {
33           mostrarInfo(context, 'Ya tiene un pedido en proceso');
34         }
35       }
36     }
37   )
38 )

```

Figura 14. Código de guardarPedido

```

1 Future<bool> guardarPedido(PedidoModel pedido) async {
2   var apiUrl = Uri.parse('http://$ROOT:5000/pedidos/crear');
3   try {
4     String? token = await obtenerToken();
5     if (token != null) {
6       var response = await http.post(
7         apiUrl,
8         headers: {"Content-Type": "application/json", "authorization": token},
9         body: json.encode(pedido.toJson()),
10      );
11      if (response.statusCode == 200) {
12        return true;
13      } else {
14        return false;
15      }
16    }
17    return false;
18  } catch (e) {
19    return false;
20  }
21 }

```

Para el desarrollo de la funcionalidad de registro de pedidos, se desarrolló la función asíncrona *guardarPedido* en *Dart*, diseñada para enviar los datos de un pedido a la *API* que corresponde del *backend*. La función utiliza la librería *http* para ejecutar solicitudes *POST* al *endpoint* establecido con la *URL* del servidor (Figura 15).

Figura 15. Código fuente del backend crearPedido

```

38
39 objVal.validarNoRepetir(tablaPedidosRob, params, async (err, noRepetido) => {
40   if (noRepetido) {
41
42     // Campos de la tabla "pedidos"
43     const camposPedido = [
44       'PEDI_DETALLE', 'PEDI_ESTADO', 'PEDI_FECREALIZO', 'PEDI_FORMA_PAGO',
45       'USUA_CODIGO', 'PEDI_CONDUCTOR', 'PEDI_VALOR_TOTAL'
46     ];
47
48     // Insertar datos en la tabla "tablaPedidosRob"
49     const resultPedido = await objSql.insertarDatos(tablaPedidosRob, camposPedido, valoresPedido);
50
51     const pedidoCodigo = resultPedido.insertId; // Obtener el código del pedido recién insertado
52     console.log("El idpedido es " + pedidoCodigo);
53
54     // Insertar los productos en la tabla "pedido_productos"
55     const valoresProductos = productos.map(producto => [
56       pedidoCodigo, // PEDI_CODIGO
57       producto.id_producto, // PROD_CODIGO
58       producto.cantidad // PEDI_CANTIDAD
59     ]);
60
61     const camposProductos = ['PEDI_CODIGO', 'PROD_CODIGO', 'PEDI_CANTIDAD'];
62
63     // Insertar los productos asociados al pedido
64     try {
65       await objSql.insertarDatos(tablaPedidoProductos, camposProductos, valoresProductos);
66       // Si todo salió bien, respondemos con el éxito
67       res.status(200).json({ result: 'Pedido creado correctamente' });
68     } catch (error) {
69       console.error(error);
70       // Si ocurre un error, respondemos con un mensaje de error
71       res.status(500).json({ error: 'Error al crear el pedido', error: error.message });
72     }
73
74   } else {
75     console.log("EL PEDIDO DEL CLIENTE AUN ESTA EN PENDIENTE, NO PUEDE REALIZAR UN PEDIDO");
76     res.status(406).json({ error: 'Aún no puede realizar un pedido' });
77   }
78 });

```

En el *backend* se hizo uso de *Express.js*, configurando un servidor que implementa autorización mediante *JSON Web Tokens (JWT)* para encargarse de la seguridad de las solicitudes. El servidor se basa en una estructura modular que permite manejar diferentes controladores y rutas. Con esta estructura, se creó la función *crearPedido*, como se evidencia en la figura 14, la cual se encarga de procesar los pedidos enviados desde el *frontend*. En el desarrollo, se realiza la validación del usuario a través del *token* de autorización y se obtienen los datos necesarios para proceder a registrar el pedido. Además, se utilizan funciones para insertar los datos del pedido en la tabla correspondiente, así como los productos. Con este diseño modular y flexible se permitió adaptar el *backend* a las necesidades de la aplicación, asegurando así un manejo eficiente y seguro de los datos.

4.3.3.2. Historia de Usuario 2: Registro de conductor (Aplicación web)

En el anexo 5, se presentan los detalles completos de la historia. Para la elaboración de la historia 2, se creó un archivo *.php* configurando un formulario dentro de un diseño responsivo utilizando *Bootstrap*. Este formulario que se aprecia en la figura 16, 17 y 18,

donde contiene los campos necesarios para el registro de conductores, como cédula, nombres, apellidos, teléfono y contraseña. La interfaz también incorpora un botón para buscar datos automáticamente al ingresar el número de cédula.

El diseño y la funcionalidad del formulario fueron realizados dentro de una tarjeta de estilo *Bootstrap*, proporcionando una experiencia amigable al usuario.

Figura 16. Pantalla “Creación de un nuevo conductor”

Figura 17. Código fuente de la vista "Creación de un nuevo conductor"

```

41         <div class="col-md-4">
42             <div class="form-group">
43                 <label for="">Contraseña</label><b> *</b>
44                 <input type="password" name="usua_password" class="form-control">
45             </div>
46         </div>
47         <div class="col-md-4">
48             <div class="form-group">
49                 <label for="">Verificación de contraseña</label><b> *</b>
50                 <input type="password" name="usua_password_ver1" class="form-control">
51             </div>
52         </div>
53     </div>
54     <div class="row">
55         <div class="col-md-12">
56             <div class="form-group">
57                 <a href="{APP_URL}/admin/conductores" class="btn btn-secondary">Cancelar</a>
58                 <button type="submit" class="btn btn-primary">Registrar conductor</button>
59             </div>
60         </div>
61     </div>

```

Figura 18. Código de consumo de la API para crear un conductor

```

1 $url = "http://", $iproot = "15000/conductores/crear"; // Cambia esta URL a la de tu API
2
3 // Los datos que quieres enviar a la API
4 $data = [
5     'cedula' => $usua_cedula,
6     'nombre' => $usua_nombre,
7     'apellido' => $usua_apellido,
8     'direccion' => $na,
9     'telefono' => $usua_telefono,
10    'password1' => $usua_password,
11    'password2' => $usua_password_veri
12 ];
13
14 // Inicializar cURL
15 $ch = curl_init($url);
16
17 // Configurar cURL para enviar los datos por POST
18 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
19 curl_setopt($ch, CURLOPT_POST, true);
20 curl_setopt($ch, CURLOPT_HTTPHEADER, [
21     'Content-Type: application/json', // Definir el tipo de contenido como JSON
22     'Accept: application/json', // Aceptar respuesta en formato JSON
23     "Authorization: $token", // El token de la sesión actual
24 ]);
25
26 curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data)); // Enviar los datos en formato JSON
27
28 // Ejecutar la solicitud
29 $response = curl_exec($ch);
30
31 // Obtener el código de estado HTTP de la respuesta
32 $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
33

```

El código implementa un proceso en *PHP* utilizando *cURL* para enviar datos a una *API* que registra conductores en un sistema, como se observa en la figura 19. Incluye la configuración de la solicitud *POST*, el envío de datos en formato *JSON* y el uso de un *token* de autorización para garantizar seguridad. Además, gestiona respuestas del servidor, mostrando mensajes de éxito o error según el caso, y asegura la liberación de recursos tras completar la comunicación. Este enfoque prioriza la eficiencia, la seguridad y la claridad en la interacción con la *API*.

Figura 19. Código fuente del backend crearConductor

```

40
41 objVal.validarNoRepetir(tabla, params, async(err, noRepetido) => {
42     if (noRepetido) {
43         var pass = objEncriptacion.encriptar_sha512(password1);
44         var valores =
45             [
46                 pass,
47                 cedula,
48                 nombre.toUpperCase(), // Convertir nombre a mayúsculas
49                 apellido.toUpperCase(), // Convertir apellido a mayúsculas
50                 "SN",
51                 telefono,
52                 'ACTIVO',
53                 "00",
54                 "00"
55             ];
56
57
58         const campos = [
59             'USUA_CLAVEX',
60             'PIRE_CODIGO',
61             'USUA_CEDULA',
62             'USUA_NOMBRE',
63             'USUA_APELLI',
64             'USUA_DIRECC',
65             'USUA_TELFNO',
66             'USUA_ESTADO',
67             'USUA_LATITUD',
68             'USUA_LONGITUD'
69         ];
70
71         try {
72             await objUsu.insertarUsuario(campos, valores);
73             res.status(200).json({ result: "Conductor creado correctamente" });
74         } catch (error) {
75             console.error("Error al crear conductor:", error);
76             res.status(500).json({ error: "Error al crear conductor", error: error.message });
77         }
78     } else {
79         console.log("== Existe");
80         res.status(400).json({ error: "El conductor ya existe" });
81     }
82 });

```

El código presentado es una función asíncrona denominada *crearConductor*, diseñada para manejar la creación de conductores, en un sistema mediante distintas validaciones y operaciones sobre la base de datos. Además, verifica la unicidad de la cédula para evitar registros duplicados y encripta la contraseña utilizando *SHA-512* antes de almacenarla.

4.3.3.2.3. Historia de Usuario 3: Registro de vehículo (Aplicación web)

Los detalles de la funcionalidad se encuentran en el anexo 5. Para la implementación de la creación de vehículos, se desarrolló un archivo *.php* que integra un formulario en un diseño responsivo utilizando *Bootstrap*, asegurando una experiencia intuitiva y consistente para el usuario. Este formulario que se evidencia en la figura 20, permite registrar nuevos vehículos mediante campos como placa, marca, modelo, conductor y un campo opcional para observaciones. La interfaz incluye botones para cancelar la operación o registrar el vehículo.

Figura 20. Pantalla “Creación de un nuevo vehículo”

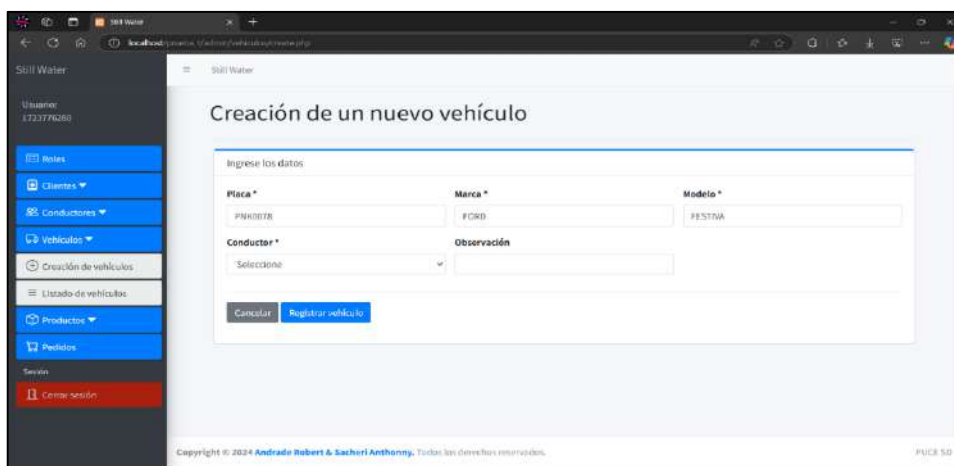


Figura 21. Código fuente de la vista "Creación de un nuevo vehículo"

```

50     <div class="row">
51         <div class="col-md-4">
52             <div class="form-group">
53                 <label for="Conductor">Conductor</label><input type="text" class="form-control" />
54                 <select name="conductor" id="" class="form-control">
55                     <option value="">Seleccione</option>
56                     <?php
57                     foreach ($conductores as $conductor) {
58                         $nombreCompleto = trim($conductor['USUA_NOMBRE'] . ' ' . $conductor['USUA_APELLI']); // Combinar nombre y apellido
59                         $codigo = $conductor['USUA_CODIGO'];
60                         // $selected = (isset($_POST['conductor']) && $_POST['conductor'] == $codigo) ? 'selected' : '';
61                         echo "<option value='$codigo' $selected:$nombreCompleto/>";
62                     }
63                 </select>
64             </div>
65         </div>
66     </div>
67 </div>
68 <div class="col-md-4">
69     <div class="form-group">
70         <label for="Observación">Observación</label>
71         <input type="text" class="form-control" name="vehl_observacion">
72     </div>
73 </div>
74 </div>
75 </div>
76 <div class="row">
77     <div class="col-md-12">
78         <div class="form-group">
79             <a href="#"> APP_URL; ?>/admin/vehiculos/" class="btn btn-secondary">Cancelar</a>
80             <button type="submit" class="btn btn-primary">Registrar vehículo</button>
81         </div>
82     </div>

```

4.3.3.2.4. Historia de Usuario 4: Registro de cliente

Los detalles de la historia están disponibles en el anexo 5. Para la elaboración de la historia 4, se creó un archivo *.php*, configurando un formulario dentro de un diseño responsivo utilizando *Bootstrap*. Este formulario que se exhibe en la figura 22 incluye

campos para el registro de clientes, como cédula, nombres, apellidos, teléfono, y referencia del domicilio, además de un mapa interactivo para seleccionar la ubicación del cliente. La interfaz también incorpora un botón para usar la ubicación actual mediante la geolocalización del navegador y otro para buscar datos automáticamente al ingresar el número de cédula.

El diseño y la funcionalidad del formulario fueron implementados dentro de una tarjeta de estilo *Bootstrap*, proporcionando una experiencia amigable al usuario. Adicionalmente, se integró un *script* de *Google Maps API*, que permite elegir y actualizar las coordenadas (latitud y longitud) mediante *clíc* en el mapa o arrastrando un marcador. El proceso completo de desarrollo y su implementación se detalla en las figuras 22 y 23.

Figura 22. Pantalla "Creación de un nuevo cliente"

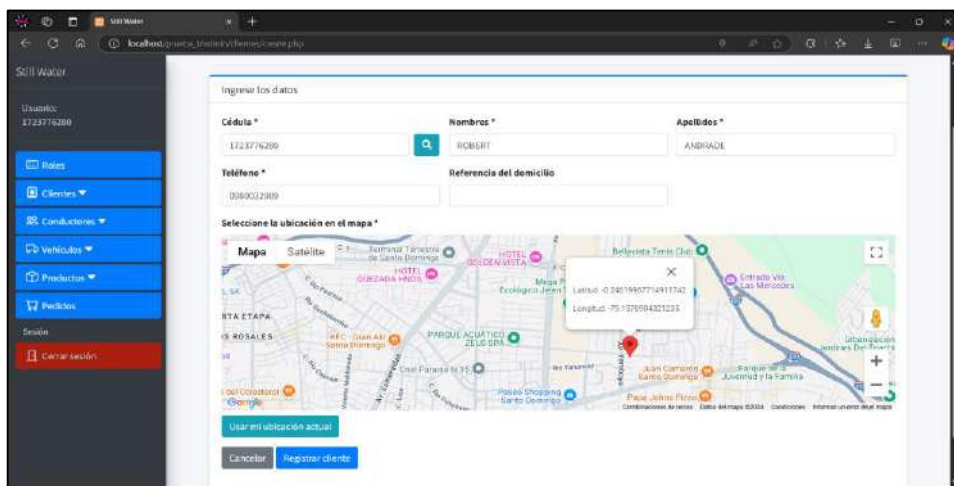


Figura 23. Código fuente de la vista "Creación de un nuevo cliente"

```

1 <form action="<? APP_URL; ?>/app/controllers/clientes/create.php" method="post">
2   <div class="row">
3     <div class="col-md-4">
4       <div class="form-group">
5         <label for="">Cédula</label><b> *</b>
6         <div class="form-group d-flex">
7           <input type="text" class="form-control" name="clie_cedula">
8           <button type="button" class="btn btn-info ml-2" id="btnBuscarCedula">
9             <i class="fas fa-search"></i>
10          </button>
11        </div>
12      </div>
13    </div>
14    <div class="col-md-4">
15      <div class="form-group">
16        <label for="">Nombres</label><b> *</b>
17        <input type="text" class="form-control" name="clie_nombre">
18      </div>
19    </div>
20    <div class="col-md-4">
21      <div class="form-group">
22        <label for="">Apellidos</label><b> *</b>
23        <input type="text" class="form-control" name="clie_apellido">
24      </div>

```

Figura 24. Código de consumo de la API para crear un cliente

```

16
17 $url = "http://" . $iproot . ":5000/clientes/crear"; // Cambia esta URL a la de tu API
18
19 // Los datos que quieres enviar a la API
20 $data = [
21     'tipo_doc' => "Cédula",
22     'cedula' => $clie_cedula,
23     'nombre' => $clie_nombre,
24     'apellido' => $clie_apellido,
25     'direccion' => $clie_direccion,
26     'telefono' => $clie_telefono,
27     'latitud' => $clie_latitud,
28     'longitud' => $clie_longitud
29 ];
30
31 // Inicializar cURL
32 $ch = curl_init($url);
33
34 // Configurar cURL para enviar los datos por POST
35 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
36 curl_setopt($ch, CURLOPT_POST, true);
37 curl_setopt($ch, CURLOPT_HTTPHEADER, [
38     'Content-Type: application/json', // Definir el tipo de contenido como JSON
39     'Accept: application/json', // Aceptar respuesta en formato JSON
40     'Authorization: $token', // El token de la sesión actual
41 ]);
42
43 curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data)); // Enviar los datos en formato JSON
44
45 // Ejecutar la solicitud
46 $response = curl_exec($ch);
47
48 // Obtener el código de estado HTTP de la respuesta
49 $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
50

```

Para la gestión del registro de clientes, se implementa un *script* en *PHP* reflejado en la figura 24, que utiliza *cURL* para enviar datos a la *API* de *Express.JS*, y se observa en la figura 25. El archivo inicia cargando la configuración del sistema y recuperando los datos enviados mediante el método *POST*, como cédula, nombre, apellido, teléfono, dirección, latitud y longitud. Posteriormente, se verifica el estado de la sesión para asegurar que exista un *token* de autenticación válido antes de realizar la solicitud. El *script* configura una solicitud *HTTP POST* hacia el *endpoint* de la *API*, incluyendo encabezados que definen el contenido como *JSON* y el *token* de autorización para validar la comunicación. (Figura 25).

Figura 25. Código fuente del backend crearCliente

```

46
47 objVal.validarNoRepetir(tabla, params, async (err, noRepetido) => {
48     if (noRepetido) {
49
50         var pass = objEncriptacion.encriptar_sha512(cedula);
51         var valores =
52             [
53                 pass,
54                 12,
55                 cedula,
56                 nombre.toUpperCase(), // Convertir nombre a mayúsculas
57                 apellido.toUpperCase(), // Convertir apellido a mayúsculas
58                 direccion,
59                 telefono,
60                 'ACTIVO',
61                 latitud,
62                 longitud
63             ];
64
65         const campos = [
66             'USUA_CLAVE',
67             'PERF_CODIGO',
68             'USUA_CEDULA',
69             'USUA_NOMBRE',
70             'USUA_APELLI',
71             'USUA_DIRECC',
72             'USUA_TELEFO',
73             'USUA_ESTADO',
74             'USUA_LATITUD',
75             'USUA_LONGITUD'
76         ];
77
78         try {
79             await objUsu.insertarUsuario(campos, valores);
80             res.status(200).json({ result: "Cliente creado correctamente" });
81         } catch (error) {
82             console.error("Error al crear usuario:", error);
83             res.status(500).json({ error: "Error al crear cliente", error: error.message });
84         }
85     } else {
86         res.status(406).json({ error: "El cliente ya existe" });
87     }
88 }

```

El método *crearCliente* es una función asíncrona programada con *Node.js* y *Express* para el registro de clientes en la aplicación móvil y aplicación *web*. La programación sigue las mejores prácticas de validación, encriptación y manejo de errores para mantener la integridad de los datos.

Figura 26. Pantalla "Formulario de cliente"

El formulario de cliente presenta un diseño claro y funcional para registrar información básica del cliente. Incluye los campos necesarios para el registro como el seleccionar el tipo de identificación (cédula o RUC), cédula, nombres, apellidos, teléfono y una referencia del domicilio. Además, permite seleccionar una ubicación específica en el mapa, con un mensaje que confirma cuando esta ha sido seleccionada. El formulario está complementado con un botón principal llamado "Registrar cliente", que realiza el envío de los datos ingresados (Figura 26).

Figura 27. Código fuente del botón "Registrar cliente" del formulario de cliente

```

1 ElevatedButton(
2
3   onPressed: () {
4     if (_formKey.currentState!.validate()) {
5       var data = json.encode({
6         'cedula': _cedulaController.text,
7         'nombre': _nombreController.text,
8         'apellido': _apellidoController.text,
9         'telefono': _telefonoController.text,
10        'direccion': _direccionController.text,
11        'latitud': _latitudController.text,
12        'longitud': _longitudController.text,
13      });
14
15      if (_hasChanges()) {
16        if (widget.condCodigo > 0) {
17          actualizarUsuario(
18            data,
19            widget.condCodigo,
20            context,
21          );
22        } else {
23          guardarUsuario(data, context);
24        }
25      } else {
26        mostrarInfo(
27          context, 'No hubo cambios para actualizar');
28      }
29    }
30  },

```

Figura 28. Código de guardarUsuario

```

1 Future<void> guardarUsuario(data, context) async {
2   var apiUrl = Uri.parse('http://$ROOT:5000/clientes/crear');
3   var userData = await obtenerTokenSesion();
4   if (userData != null) {
5     var response = await http.post(apiUrl, body: data, headers: {
6       "Content-Type": "application/json",
7       "authorization": userData.toString()
8     });
9
10    if (response.statusCode == 200) {
11      var responseBody = json.decode(response.body);
12      var mensaje = responseBody['result'] ?? 'Usuario guardado';
13      mostrarCorrecto(context, mensaje);
14      Navigator.pop(context, true);
15    } else {
16      var responseBody = json.decode(response.body);
17      var errorMensaje = responseBody['error'] ?? 'Error al guardar usuario';
18      mostrarError(context, errorMensaje);
19    }
20  }
21 }

```

La función asíncrona *guardarUsuario* implementa un proceso para registrar la información de un cliente haciendo uso de la API utilizando *HTTP POST*. Primero se obtiene un *token* de sesión mediante la función *obtenerTokenSesion*. Si el *token* es válido, permite enviar una solicitud con encabezados que incluyen *Content-Type* como *JSON* y un campo de autorización con el *token*. Al recibir una respuesta de la API se verifica mediante su código de estado: si es 200, se interpreta como éxito, se decodifica la respuesta, y se muestra un mensaje de confirmación mediante *mostrarCorrecto()*. Si ocurre un error, se decodifica para obtener el mensaje de error y se utiliza *mostrarError()* para notificar al usuario. Además, la función emplea *Navigator.pop()* para retornar a la pantalla anterior tras un registro exitoso, garantizando una experiencia de usuario fluida y manejando errores de forma clara y eficiente.

4.3.3.2.5. Historia de Usuario 5: Registro de producto (Aplicación web y móvil)

Los detalles de la historia están disponibles en el anexo 5. Para la elaboración de la Historia de Usuario 5, se desarrollaron formularios independientes para la versión *web* y la versión *móvil*. En la aplicación *web* que se visualiza en la figura 29, se creó un archivo *.php* como se observa en la figura 30, configurando un formulario dentro de un diseño responsivo utilizando *Bootstrap*. Este formulario incluye campos para ingresar el nombre, detalle y precio del producto, junto con botones para cancelar o registrar el producto.

En la versión móvil, como se aprecia en la figura 33, el formulario mantiene los mismos campos principales en un diseño optimizado para dispositivos móviles, con una interfaz minimalista y un botón principal que facilita el registro del producto. Las dos aplicaciones aseguran un flujo intuitivo y eficiente para los usuarios, basándose en los principios de diseño responsivo y una experiencia similar en distintas plataformas.

Figura 29. Pantalla “Creación de un nuevo producto”

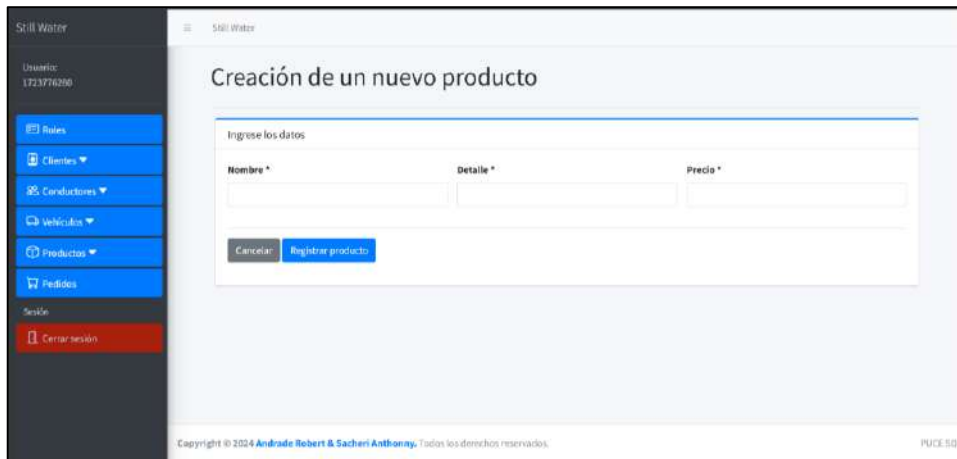


Figura 30. Código fuente de la vista "Creación de un nuevo producto"

```

1 <div class="row">
2     <div class="col-md-4">
3         <div class="form-group">
4             <label for="">Nombre</label><b> *</b>
5             <input type="text" class="form-control" name="prod_nombre">
6         </div>
7     </div>
8     <div class="col-md-4">
9         <div class="form-group">
10            <label for="">Detalle</label><b> *</b>
11            <input type="text" class="form-control" name="prod_detalle">
12        </div>
13    </div>
14    <div class="col-md-4">
15        <div class="form-group">
16            <label for="">Precio</label><b> *</b>
17            <input type="text" class="form-control" name="prod_precio">
18        </div>

```

Figura 31. Código de consumo de la API para crear un producto

```

11 $url = "http:// . iproot . ":5000/productos/crear"; // Cambia esta URL a la de tu API
12
13 // Los datos que quieres enviar a la API
14 $data = [
15     'nombre' => $prod_nombre,
16     'detalle' => $prod_detalle,
17     'precio' => $prod_precio
18 ];
19
20 // Inicializar cURL
21 $ch = curl_init($url);
22
23 // Configurar cURL para enviar los datos por POST
24 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
25 curl_setopt($ch, CURLOPT_POST, true);
26 curl_setopt($ch, CURLOPT_HTTPHEADER, [
27     'Content-Type: application/json', // Definir el tipo de contenido como JSON
28     'Accept: application/json',     // Aceptar respuesta en formato JSON
29     "Authorization: $token",        // El token de la sesión actual
30 ]);
31
32 curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data)); // Enviar los datos en formato JSON
33
34 // Ejecutar la solicitud
35 $response = curl_exec($ch);
36
37 // Obtener el código de estado HTTP de la respuesta
38 $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
39

```

El código presentado en *PHP* permite registrar un producto en un sistema utilizando *cURL* para interactuar con una *API REST*, reflejado en la figura 32. Inicialmente, se capturan los datos del producto enviados mediante el método *POST*, como el nombre, detalle y precio. A continuación, se verifica el estado de la sesión para obtener un *token* de autenticación que asegura la validez de la operación.

Figura 32. Código fuente del backend crearProductos

```

27 if (id_usuario) {
28     const campos = ['PROD_NOMBRE', 'PROD_DETALLE', 'PROD_PRECIO', 'PROD_ESTADO'];
29     const valores = [nombre, detalle, precio, 'ACTIVO'];
30     const params = [
31         { field: 'PROD_NOMBRE', value: nombre },
32         // { field: 'PROD_ESTADO', value: 'ACTIVO' },
33     ];
34     objVal.validarNoRepetir(tablaProductos, params, (err, noRepetido) => {
35         if (err) {
36             return res.status(400).json({ result: err.result });
37         }
38         if (noRepetido) {
39             try {
40                 objSql.insertarDatos(tablaProductos, campos, valores);
41                 res.status(200).json({ result: 'Producto creado correctamente' });
42             } catch (error) {
43                 return res.status(500).json({ error: 'Error al insertar productos del pedido', error: error.message });
44             }
45         } else {
46             return res.status(400).json({ error: "El producto ya existe" });
47         }
48     });
49 }

```

El código implementa la función asíncrona *crearProductos* establecida en la figura 32, diseñada para gestionar el registro de productos mediante una *API*. Se extraen del

cuerpo de la solicitud (*req.body*) los datos esenciales: nombre, detalle y precio, y se valida cada uno utilizando funciones específicas que verifican su formato y contenido. Si las validaciones no son exitosas, se genera un mensaje de error con un código *HTTP 400*.

Figura 33. Pantalla "Formulario de producto"



Figura 34. Código fuente del botón "Registrar producto" del formulario de producto

```

1 ElevatedButton(
2   onPressed: () {
3     if (_formKey.currentState!.validate()) {
4       var data = {
5         'nombre': _nombreController.text,
6         'detalle': _detalleController.text,
7         'precio': _precioController.text,
8       };
9       _guardarProducto(widget.prodCodigo, data);
10    }
11  },

```

Figura 35. Código de consumo de la API para crear o editar un producto.

```

1 Future<void> _guardarProducto(
2   int prodCodigo, Map<String, dynamic> data) async {
3   var apiUrl = buttonText == 'Registrar producto'
4     ? Uri.parse('http://$ROOT:5000/productos/crear')
5     : Uri.parse('http://$ROOT:5000/productos/editar/$prodCodigo');
6
7   try {
8     setState(() {
9       isLoading = true;
10    });
11    String? token = await _getToken();
12    if (token != null) {
13      var response = await http.post(apiUrl,
14        body: json.encode(data),
15        headers: {
16          "Content-Type": "application/json",
17          "authorization": token
18        });
19
20    if (response.statusCode == 200) {
21      mostrarConnecto(context, jsonDecode(response.body)['result']);
22      _limpiarCampos();
23      Navigator.pop(context, true);
24    } else {
25      mostrarError(context, jsonDecode(response.body)['error']);
26    }
27  }

```

El método asíncrono *guardarProducto* permite registrar o actualizar un producto en el sistema a través de una solicitud *HTTP*. Dependiendo del valor de *buttonText*, la función determina si la operación es de creación o edición y ajusta la *URL* del *endpoint* de la *API* visualizado en la figura 32, los datos del producto, codificados en formato *JSON*, se envían

mediante una solicitud *POST* junto con un *token* de autenticación para garantizar la seguridad.

4.3.3.2.6. Historia de Usuario 6: Login

Para la autenticación, se implementa un *script* en *PHP* que utiliza *cURL* para enviar datos a la *API* de *Express.JS*, que se puede visualizar en la figura 39. Se envían las credenciales (cédula y clave) en una solicitud *HTTP POST* en donde incluye los datos del usuario en formato *JSON* y espera una respuesta de la *API*. Si ocurre un error en la conexión, se informa al usuario que no hay acceso al servidor. Si la *API* responde con un código *HTTP 200*, los datos del usuario y el *token* se almacenan en la sesión para autenticaciones futuras en el sistema.

Figura 36. Pantalla "Login" en la aplicación web

Figura 37. Código fuente de la vista web "Login"

```

1 <div class="card-body">
2   <p class="login-box-msg">Inicio de sesión</p>
3   <form action="controller_ingreso.php" method="post">
4     <div class="input-group mb-3">
5       <input type="text" class="form-control" name="email" placeholder="Cédula">
6       <div class="input-group-append">
7         <div class="input-group-text">
8           <span class="fas fa-id-card"></span>
9         </div>
10      </div>
11    </div>
12    <div class="input-group mb-3">
13      <input type="password" class="form-control" name="password" placeholder="Contraseña">
14      <div class="input-group-append">
15        <div class="input-group-text">
16          <span class="fas fa-lock"></span>
17        </div>
18      </div>
19    </div>
20    <button type="submit" class="btn btn-primary btn-block">Ingresar</button>
21  </form>
22 </div>

```

Figura 38. Código de consumo de la API para el login

```

2  $url = "http://". iproot . ":5000/login/"; // Cambia esta URL a la de tu API
3
4  // Los datos que quieres enviar a la API
5  $data = [
6    'cedula' => $email,
7    'clave' => $password
8  ];
9
10 // Inicializar cURL
11 $ch = curl_init($url);
12
13 // Configurar cURL para enviar los datos por POST
14 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
15 curl_setopt($ch, CURLOPT_POST, true);
16 curl_setopt($ch, CURLOPT_HTTPHEADER, [
17   'Content-Type: application/json', // Definir el tipo de contenido como JSON
18   'Accept: application/json', // Aceptar respuesta en formato JSON
19 ]);
20 curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data)); // Enviar los datos en formato JSON
21
22 // Ejecutar la solicitud
23 $response = curl_exec($ch);
24
25 // Obtener el código de estado HTTP de la respuesta
26 $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
27
28 // Verificar si hubo un error
29 if (curl_errno($ch)) {
30   // Mostrar un mensaje si no hay conexión al servidor
31   $_SESSION['mensaje'] = "No hay conexión con el servidor. Por favor, inténtalo más tarde.";
32   $_SESSION['icono'] = "error";
33   ?<script>
34     window.history.back();
35   </script>?php
36     exit;
37   } else {
38     // Verificar si el código de estado es 200 (OK)
39     if ($http_code == 200) {

```

Además, dependiendo del rol del usuario, ya sea gerente o conductor se redirige al panel de administración estableciendo el menú según corresponda, en caso de ser cliente, se informa que el acceso solo está permitido desde la aplicación móvil. En caso de credenciales incorrectas o una respuesta diferente a 200, aparece un mensaje de error notificando que las credenciales no son válidas. Posteriormente, se libera la sesión *cURL* para garantizar un uso eficiente de los recursos.

Figura 39. Código fuente del backend login

```

1  const login = asyncHandler(async (req, res) => {
2    const { cedula, clave } = req.body;
3    console.log(req.body);
4    try {
5      const usuarios = await objUser.obtenerUsuarioPorCedulaClaveEstado(cedula, objEncryptacion.encryptar_sha512(clave), "ACTIVO");
6
7      if (usuarios.length === 0) {
8        return res.status(401).json({ result: "USUARIO O CONTRASEÑA INCORRECTO" });
9      }
10
11     const usuario = usuarios[0];
12     const token = objToken.generateToken_JWT(usuario.USUA_CODIGO);
13
14     if (usuario.USUA_ESTADO == "INACTIVO") {
15       return res.status(401).json({ result: "Usuario inactivo, favor hablar con el administrador para que le permita ingresar" });
16     }
17
18     return res.status(200).json({
19       codigo: usuario.USUA_CODIGO,
20       cedula: usuario.USUA_CEDULA,
21       nombre: usuario.USUA_NOMBRE,
22       apellido: usuario.USUA_APELLI,
23       telefono: usuario.USUA_TELEFO,
24       direccion: usuario.USUA_DIRECC,
25       latitud: usuario.USUA_LATITUD,
26       longitud: usuario.USUA_LONGITUD,
27       id_perfil: usuario.PERF_CODIGO,
28       token
29     });
30

```

La función *login* autentica usuarios mediante su cédula y clave encriptada, además de validar que su estado sea "ACTIVO". Si las credenciales son incorrectas o el usuario está inactivo, retorna mensajes de error con códigos *HTTP 401*. Si la autenticación es exitosa, se genera un *token JWT* y responde con los datos del usuario, y en caso de errores internos, maneja las excepciones con un código *HTTP 500*, asegurando un proceso seguro y claro para la gestión del *login*.

Figura 40. Pantalla "login" en la aplicación móvil



Los detalles de la historia están disponibles en el anexo 5. Se diseñó una interfaz intuitiva y minimalista utilizando componentes modernos. La pantalla presenta el logotipo de la aplicación seguido del título "*Still Water*", representando la marca del sistema.

El formulario contiene dos campos principales: uno destinado al nombre de usuario y otro a la contraseña, con un ícono que permite alternar la visibilidad del texto en el campo de la contraseña, mejorando la usabilidad. En la zona inferior, se incluye un botón destacado con el texto "INGRESAR", para hacer uso del *API Rest* que se visualiza en la figura 39, con ello, facilita la autenticación del usuario.

El diseño responsivo asegura una compatibilidad con los distintos dispositivos móviles, garantizando una experiencia visual agradable y funcional. Esta pantalla fue diseñada para ser simple y clara, optimizando el acceso rápido al sistema.

Figura 41. Código fuente de la vista móvil "Login"

```

38     widCampoTexto(
39         hintText: 'Usuario',
40         icon: Icons.person,
41         controller: _usuarioController,
42         backgroundColor: ColorFondo.PRIMARY.withOpacity(0.1),
43         borderColor: ColorFondo.PRIMARY,
44     ),
45     const SizedBox(height: 20.0),
46     widCampoTexto(
47         hintText: 'Contraseña',
48         icon: Icons.lock,
49         isPassword:
50             !_isPasswordVisible, // Controlar visibilidad de la contraseña
51         controller: _claveController,
52         backgroundColor: ColorFondo.PRIMARY.withOpacity(0.1),
53         borderColor: ColorFondo.PRIMARY,
54         suffixIcon: IconButton(
55             icon: Icon(
56                 !_isPasswordVisible
57                 ? Icons.visibility
58                 : Icons.visibility_off,
59             ),
60             onPressed: () {
61                 setState(() {
62                     _isPasswordVisible = !_isPasswordVisible;
63                 });
64             },
65         ),
66     ),
67     const SizedBox(height: 40.0),
68     _buildSignInButton(context),

```

Figura 42. Código de consumo de la API para el login

```

1  static Future<void> login(String usuario, String clave) async {
2    try {
3      var url = Uri.parse('http://$ROOT:5000/login/');
4      var body = json.encode({'cedula': usuario, 'clave': clave});
5
6      var response = await http.post(
7        url,
8        body: body,
9        headers: {"Content-Type": "application/json"},
10     ).timeout(const Duration(seconds: 10), onTimeout: () {
11       throw ('Tiempo de espera agotado. Verifica tu conexión a internet.');
```

La función *login* permite autenticar a los usuarios mediante su cédula y clave, enviando la información a la API ubicada en la figura 39. Si la autenticación es exitosa y el usuario tiene los permisos correspondientes, se guarda su información para el inicio de sesión. En caso de cédula o clave incorrecta, falta de datos o problemas con el servidor, se muestran mensajes al usuario indicando el error.

4.3.3.2.7. Historia de usuario 7: Visualizar pedidos (Aplicación móvil)

El anexo 5 contiene los detalles de la historia. En el desarrollo de la Historia de Usuario 7, se diseñó una pantalla para la gestión y visualización de pedidos en la aplicación móvil. En la Figura 43, se muestra la interfaz que organiza los pedidos en tarjetas individuales con la información necesaria: nombre del cliente, estado del pedido ("en proceso", "entregado", "no entregado" o "pendiente"), detalle, y la fecha y hora del pedido. Además, se agregaron pestañas para filtrar los pedidos por estado: "todos", "en proceso" y "no entregado", "pendientes". Para visualizar información del pedido, se presiona en la

tarjeta y a continuación abre una nueva pantalla con el detalle del pedido como se presenta en la figura 41.

La pantalla también incluye accesos rápidos, para buscar los pedidos con diferentes filtros y un botón para un nuevo pedido. Este diseño, es claro y funcional, para garantizar que los usuarios puedan gestionar y encontrar los pedidos de manera sencilla y eficiente.

Figura 43. Pantalla "Pedidos"



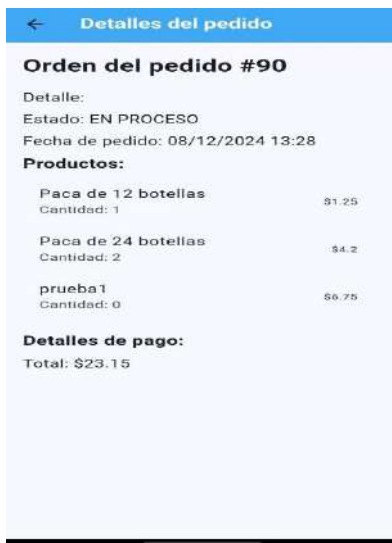
Figura 44. Código fuente de la vista "Pedidos"

```

2      onRefresh: obtenerPedidos,
3      child: ListView.builder(
4        itemCount: filteredPedidos.length,
5        itemBuilder: (context, index) {
6          final pedido = filteredPedidos[index];
7          final estado = pedido['estado'];
8          final cliente =
9            "${pedido['cliente_nombres'] ?? "Desconocido"}";
10         final estadoColor = _getEstadoColor(estado);
11
12         return Card(
13           elevation: 3,
14           margin: const EdgeInsets.symmetric(
15             vertical: 8, horizontal: 16),
16           child: ListTile(
17             leading: CircleAvatar(
18               backgroundColor: estadoColor,
19               child:
20                 const Icon(Icons.receipt, color: Colors.white),
21             ),
22             title: Text(
23               "Cliente: $cliente",
24               style: const TextStyle(fontWeight: FontWeight.bold),
25             ),
26             subtitle: Text(
27               "Estado: $estado\nDetalle: ${pedido['detalle']}\nFecha del pedido: ${pedido['fecha_pedido']}",
28               style: const TextStyle(fontSize: 12),
29             ),

```

Figura 45. Pantalla "Detalles del pedido"



Para visualizar el detalle de un pedido en específico, se diseñó una pantalla la cual presenta toda la información del pedido seleccionado, incluyendo el número del pedido real en el sistema, estado, fecha y hora, y la lista de productos elegidos, con su cantidad y precio unitario. En la zona inferior, se visualiza "Detalles de pago" que muestra el total del pedido calculado automáticamente.

Este diseño busca facilitar al usuario una vista clara y detallada de los pedidos, facilitando el seguimiento de los productos seleccionado y el total del pedido.

Figura 46. Código fuente de la vista "Detalle del pedido"

```

76     if (pedido['estado'].toUpperCase() == "ENTREGADO") {}
77     pedido['estado'].toUpperCase() == "PENDIENTE") ...{
78     Builder(builder, context) {
79         String mensaje;
80         switch (pedido['estado'].toUpperCase()) {
81             case "ENTREGADO":
82                 mensaje = 'Conductor que entregó el pedido: ';
83                 break;
84             case "PENDIENTE":
85                 mensaje = 'Pedido pendiente, conductor asignado: ';
86                 break;
87             default:
88                 mensaje = 'Estado del pedido no identificado: ';
89         }
90         return Column(
91             crossAxisAlignment: CrossAxisAlignment.start,
92             children: [
93                 Text(
94                     mensaje,
95                     style: const TextStyle(
96                         fontSize: 18, fontWeight: FontWeight.bold),
97                 ),
98                 const SizedBox(height: 8),
99                 Text(
100                    'Nombre: ${pedido['conductor.nombre']}',
101                    style: const TextStyle(fontSize: 16),
102                ),
103                const SizedBox(height: 16),
104            ],
105        );
106    },
107 ],
108 // Pago y total
109 const Text(
110     'Detalles de pago:',
111     style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
112 ),
113 const SizedBox(height: 8),
114 Text(
115     'Total: \${pedido['total']}',
116     style: const TextStyle(fontSize: 16),
117 ),

```

4.3.3.3. Sprint 1 – Revisión

Una vez finalizado el desarrollo del primer sprint, se realizó una reunión entre el gerente de empresa y el equipo de desarrollo, donde se presentaron todos los avances y funcionalidades de la aplicación. Se revisó cada historia de usuario y sus respectivos escenarios de prueba junto con sus pruebas de aceptación correspondientes, las cuales se evidencian en el anexo 6.

4.3.3.3.1. Pruebas unitarias

El objetivo principal de las pruebas unitarias, consiste en verificar la funcionalidad de las distintas partes que conforman un sistema, las partes o unidades pueden ser un método, un módulo o una función que desempeña una tarea concreta dentro del sistema y posee cierta lógica en su funcionamiento.

Mientras se desarrollaba cada historia de usuario, se realizaron pruebas unitarias, empleando la técnica de caja blanca, la cual se enfoca en la optimización y calidad del código permitiendo la detección y corrección de errores en las primeras etapas de desarrollo. Para realizarlas, se empleó el *framework* de *Mocha* que funciona en el entorno de *Node.js*.

4.3.3.4. Sprint 1 – Retrospectiva

Una vez finalizado el primer *sprint*, con la finalidad de fomentar la mejora continua se reunió al equipo de desarrollo, para obtener una perspectiva clara de lo que salió bien o mal y lo que se planea mejorar para el siguiente *sprint*.

Tabla 17. Sprint 1 - Retrospectiva

¿Qué funcionó bien en el <i>sprint</i> ?	¿Qué aspectos no funcionaron en el <i>sprint</i> ?	¿En qué áreas podemos trabajar para mejorar?
La comunicación entre los miembros del equipo y el <i>product owner</i> fue cordial y muy provechosa.	Algunos requisitos cambiaron en mitad del <i>sprint</i> lo que generó pequeños retrasos en algunas tareas.	Se identificó que se debe mejorar la estimación del esfuerzo requerido para las tareas y mejorar la gestión de los cambios en los requerimientos.

4.3.4. Sprint 2

4.3.4.1. Sprint 2 – Planificación

Para el segundo *sprint*, se seleccionaron las últimas 7 historias de usuario pendientes del *product backlog*. Hacen un total de 22 puntos de estimación estas historias de usuario juntas, de manera similar, se elaboró un *sprint backlog* que se evidencia en la tabla 18.

Tabla 18. Sprint Backlog 2

Sprint Backlog						
Objetivo: Desarrollar las funciones necesarias para los registros (clientes, conductores, vehículos y productos), autenticar el acceso de usuarios (login) y gestionar pedidos (registro y visualización) tanto en la aplicación web como en la móvil según corresponda.						
Historia usuario	Est	Categ	Tareas de ingeniería	Est	Respon	Est
HU8 – Asignación de pedido a conductor (Aplicación móvil)	3	Diseño	Diseñar el botón asignar pedidos (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Desarrollo	Desarrollar en el controlador <i>ctrl_pedidos.js</i> la función <i>asignarPedidosAConductores</i> . (<i>Node.js, Express, JavaScript</i>)	0.5	Robert	C
		Prueba	Realizar pruebas de asignar pedidos a los conductores	1	Sacheri	C
HU9 – Ruta óptima (Aplicación web)	8	Desarrollo	Desarrollar en el controlador <i>ctrl_pedidos.js</i> la función <i>clasificarPedidosPorZona</i> . (<i>Node.js, Express, JavaScript</i>)	6	Sacheri	C
		Prueba	Realizar pruebas sobre la ruta óptima.	2	Robert	C
HU10 - Visualizar pedidos en el mapa (Aplicación web)	3	Diseño	Diseñar el formulario para visualizar pedidos en el mapa (<i>Flutter, Dart</i>)	1	Sacheri	C
		Desarrollo	Desarrollar el generar los marcadores de los pedidos en el mapa (<i>Flutter, Dart</i>)	1	Robert	C
		Prueba	Realizar pruebas de visualizar los pedidos en el mapa	1	Sacheri	C
HU11 – Historial de pedidos (Aplicación web y aplicación móvil)	2	Diseño	Diseñar el formulario para Visualizar el historial de pedidos (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Desarrollo	Desarrollar en el controlador <i>ctrl_pedidos.js</i> la función <i>historialPedidos</i> . (<i>Node.js, Express, JavaScript</i>)	0.5	Robert	C
		Prueba	Realizar pruebas de visualizar el historial de pedidos.	1	Robert	C
HU12 - Registro de entrega del pedido	3	Diseño	Diseñar el formulario para Registrar el historial de pedidos (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Desarrollo	Desarrollar en el controlador <i>ctrl_pedidos.js</i> la función	1.5	Robert	C

(Aplicación móvil)			<i>cambiarEstadoPedido. (Node.js, Express, JavaScript)</i>			
		Pruebas	Realizar pruebas de registro de entrega del pedido	1	Sacheri	C
HU13 - Visualizar perfil (Aplicación web y aplicación móvil)	2	Diseño	Diseñar el formulario para Visualizar perfil (<i>Flutter, Dart</i>)	0.5	Robert	C
		Desarrollo	Desarrollar en el controlador <i>ctrl_clientes.js</i> la función <i>getDatosCliente</i> y en el controlador <i>ctrl_conductores.js</i> la función <i>getRegistroConductor. (Node.js, Express, JavaScript)</i>	1	Robert	C
		Prueba	Realizar pruebas de visualizar de visualizar perfil	0.5	Sacheri	C
HU14 - Editar perfil (Aplicación web y aplicación móvil)	2	Diseño	Diseñar el formulario para editar perfil (<i>Flutter, Dart</i>)	0.5	Sacheri	C
		Desarrollo	Desarrollar en el controlador <i>ctrl_clientes.js</i> las funciones: <i>editarCliente, actualizarContra;</i> y en el controlador <i>ctrl_conductores.js</i> las funciones <i>editarConductor, actualizarContra (Node.js, Express, JavaScript)</i>	1	Robert	C
		Prueba	Realizar pruebas de editar perfil	0.5	Robert	C

4.3.4.2. Sprint 2 – Reuniones diarias

Gracias a la retrospectiva del primer *sprint*, se identificó que se llevó a cabo de forma favorable la comunicación entre el equipo de trabajo, el *product owner* y el *scrum master*.

Con la intención de seguir manteniendo una buena comunicación se volvió a adoptar las reuniones diarias, de una duración máxima de 15min, entre los miembros involucrados de la aplicación. Además, se continuó empleando la herramienta de gestión de proyectos y tareas *Trello*, de esta manera se pudo llevar a cabo un correcto control de las tareas.

4.3.4.2.1. Historia de Usuario 8: Asignación de pedidos a conductor (Aplicación móvil)

Los detalles de la historia están disponibles en el anexo 5. Para la elaboración de la historia de usuario 8, se desarrolló una pantalla destinada a la gestión de pedidos en la aplicación móvil. En la figura 47 y 48, se muestra una lista de pedidos pendientes, detallando información como el nombre del cliente, distancia hasta la ubicación del pedido, productos solicitados, cantidad, y el total a cobrar.

Figura 49. Código de consumo de la API para asignar los pedidos a conductores

```

8     if (pedidos.isEmpty || conductores.isEmpty) {
9         throw Exception('Debe proporcionar los pedidos y los conductores.');
```

Figura 50. Código fuente del backend para asignar los pedidos a conductores

```

1 function asignarPedidosAConductores(pedidos, conductores) {
2     // Clasificamos los pedidos en zonas (norte, sur, este, oeste)
3     const zonas = clasificarPedidosPorZona(pedidos);
4
5     let asignaciones = [];
6     let conductorIndex = 0;
7
8     // Asignamos los pedidos de cada zona a los conductores
9     Object.keys(zonas).forEach(zona => {
10        const pedidosZona = zonas[zona];
11        pedidosZona.forEach(pedido => {
12            asignaciones.push({
13                conductorId: conductores[conductorIndex],
14                pedidoId: pedido.PEDI_CODIGO,
15                zona: zona
16            });
17            conductorIndex = (conductorIndex + 1) % conductores.length; // Vuelve a empezar cuando se acaban los conductores
18        });
19    });
20
21    return asignaciones;
22 }
```

La función *asignarPedidosAConductores* se encarga de distribuir pedidos entre conductores de una manera equitativa. Primero, clasifica los pedidos en zonas (norte, sur, este, oeste) haciendo uso de la función *clasificarPedidosPorZona*. Después, recorre cada zona y distribuye los pedidos entre los conductores de forma cíclica, asegurándose de que todos los conductores reciban pedidos de manera uniforme.

Para cada pedido, se guarda una asignación que incluye el *index* del conductor, el *index* del pedido y la zona correspondiente. Si se agotan los conductores disponibles, la asignación comienza de nuevo desde el primer conductor. Al finalizar, la función retorna una lista con todas las asignaciones realizadas. Este diseño garantiza una distribución equilibrada y organizada de los pedidos.

Figura 51. Código fuente del backend para clasificar los pedidos por zona

```

1  function clasificarPedidosPorZona(pedidos) {
2      const zonas = {
3          norte: [],
4          sur: [],
5          este: [],
6          oeste: []
7      };
8
9      pedidos.forEach(pedido => {
10         // Clasificación basada en latitud y longitud
11         if (pedido.latitud > puntoCentral.lat && pedido.longitud > puntoCentral.lon) {
12             zonas.norte.push(pedido); // Norte-Este
13         } else if (pedido.latitud > puntoCentral.lat && pedido.longitud < puntoCentral.lon) {
14             zonas.oeste.push(pedido); // Norte-Oeste
15         } else if (pedido.latitud < puntoCentral.lat && pedido.longitud > puntoCentral.lon) {
16             zonas.sur.push(pedido); // Sur-Este
17         } else if (pedido.latitud < puntoCentral.lat && pedido.longitud < puntoCentral.lon) {
18             zonas.oeste.push(pedido); // Sur-Oeste
19         }
20     });
21
22     return zonas;
23 }

```

La función *clasificarPedidosPorZona* organiza pedidos en diferentes zonas geográficas (norte, sur, este y oeste) según su latitud y longitud en relación con un punto central. Se crea un objeto *zonas* con listas vacías para cada dirección, y luego se recorren los pedidos.

4.3.4.2.2. Historia de Usuario 9: Ruta óptima (Aplicación móvil)

Para la elaboración de la historia de usuario 9, se desarrolló una pantalla destinada a la gestión de pedidos en la aplicación móvil del lado del perfil conductor. En la figura 52 se visualiza una lista de pedidos con información relevante como el nombre del cliente, productos solicitados, cantidad de productos, distancia desde la ubicación del usuario y el total a cobrar por cada pedido. Además, en la zona inferior de la pantalla, se disponen botones clave para visualizar la ruta y en caso de ser necesario finalizar todos los pedidos, permitiendo optimizar la gestión y planificación de la ruta de distribución. Los detalles de la historia están a disposición en el anexo 5.

Figura 52. Pantalla "Gestión de pedidos"



Figura 53. Código fuente de la vista "Gestión de pedidos"

```

46         Expanded(
47             child: _pedidosOrdenados.isNotEmpty
48             ? ListView.builder(
49                 itemCount: _pedidosOrdenados.length,
50                 itemBuilder: (context, index) {
51                     return _buildPedidoItem(
52                         _pedidosOrdenados[index],
53                         index,
54                     );
55                 },
56             )
57             : const Center(
58                 child: Text('No hay pedidos para mostrar'),
59             ),
60         ),
61         if (_pedidosOrdenados.isNotEmpty)
62         ElevatedButton.icon(
63             onPressed: _visualizarEnMapa,
64             icon: const Icon(Icons.map),
65             label: const Text('Visualizar Ruta'),
66         ),

```

Figura 54. Código de consumo de la API para obtener pedidos del conductor

```

1 Future<List<Map<String, dynamic>>> obtenerPedidosConductor({
2     required String codigoConductor,
3     required double latCond,
4     required double lonCond,
5 }) async {
6     List<Map<String, dynamic>> pedidos = [];
7     var apiUrl = Uri.parse(
8         'http://$ROOT:5000/pedidos/pedidos_conductor/$codigoConductor');
9     final userData = await obtenerTokenSesion();
10
11     if (userData != null) {
12         final response = await http.post(
13             apiUrl,
14             headers: {
15                 "Content-Type": "application/json",
16                 "authorization": userData.toString(),
17             },
18             body: json.encode({
19                 "latCond": latCond,
20                 "lonCond": lonCond,
21             })),
22         );
23
24     if (response.statusCode == 200) {
25         final data = json.decode(response.body);
26         pedidos = List<Map<String, dynamic>>.from(data['result']);
27     } else if (response.statusCode == 404) {
28         final data = json.decode(response.body);
29         throw Exception(
30             'Error al cargar los pedidos del conductor: ${data['error']}');
31     } else {
32         throw Exception(
33             'Error al cargar los pedidos del conductor: ${response.statusCode}');
34     }
35 }
36 return pedidos;
37 }

```

La función *listarPedidosPendientesCond* permite obtener una lista de pedidos pendientes asignados a un conductor, ordenados por proximidad a su ubicación actual. Primero, verifica la validez del *token* de autorización y obtiene el identificador del usuario. A continuación, se extraen las coordenadas del conductor y se filtran los pedidos en estado "en proceso" asociados al conductor en cuestión.

Se calcula la distancia entre la ubicación del conductor y cada pedido, utilizando las coordenadas del cliente. Los pedidos con coordenadas válidas se incluyen en la lista final, ordenados de menor a mayor distancia. Si no se encuentran pedidos pendientes o hay problemas con las coordenadas, la función responde con mensajes de error claros. Este enfoque asegura una organización eficiente de los pedidos, priorizando los más cercanos para optimizar la logística.

Figura 55. Código fuente del backend de listar los pedidos pendientes del conductor

```

18     const pedidos = await objSql.seleccionarTodosDatosTabla(
19       'vista_historial_pedidos', filtro
20     );
21
22     //console.log('Contenido de pedidos:', pedidos);
23
24     if (!pedidos || pedidos.length === 0) {
25       return res.status(404).json({ error: "No hay pedidos pendientes" });
26     }
27
28     // 2. Obtener las coordenadas del usuario actual
29     const usuarioLatitud = latCond;
30     const usuarioLongitud = lonCond;
31
32     // 3. Calcular la distancia de cada pedido desde la ubicación del usuario
33     const pedidosConDistancia = pedidos.map(pedido => {
34       // Validar si las coordenadas del pedido son nulas
35       if (pedido.cliente_latitud == null || pedido.cliente_longitud == null) {
36         console.log('Pedido con código ${pedido.pedido_codigo} tiene coordenadas nulas.~');
37         return { ...pedido, distancia: null }; // Asignar null a la distancia
38       }
39
40       // Calcular la distancia solo si las coordenadas son válidas
41       const distancia = geolib.getDistance(
42         { latitude: usuarioLatitud, longitud: usuarioLongitud },
43         { latitude: pedido.cliente_latitud, longitud: pedido.cliente_longitud }
44       );
45       return { ...pedido, distancia };
46     });
47
48     // 4. Filtrar pedidos válidos y ordenar por distancia
49     const pedidosValidos = pedidosConDistancia.filter(pedido => pedido.distancia != null);
50     pedidosValidos.sort((a, b) => a.distancia - b.distancia);
51
52     // 5. Responder con los pedidos ordenados por distancia
53     res.status(200).json({ result: pedidosValidos });

```

4.3.4.2.3. Historia de Usuario 10: Visualizar pedidos en el mapa (Aplicación móvil)

Para la elaboración de la historia de usuario 10, se diseñó una pantalla en la aplicación móvil que permite a los usuarios de perfil conductor visualizar la ruta de entrega de pedidos. Los detalles de la historia están a disposición en el anexo 5. En la figura 56, se muestra un mapa interactivo que traza el recorrido planificado para los pedidos asignados a

dicho conductor, destacando los puntos de entrega con marcadores específicos que cambian su color según el estado en el que se encuentre el pedido:

Marcador Turquesa: Ubicación de partida

Marcador Azul: Pedido en proceso

Marcador Verde: Pedido entregado

Marcador Rojo: Pedido no entregado

La interfaz incluye en la parte superior un tiempo estimado de la ruta para que el conductor tenga una referencia para completar el recorrido. Además, en la zona inferior, se agregó un botón etiquetado como Guardar estado pedido(s), que permite actualizar el estado de los pedidos conforme se cambia el estado de uno o varios.

Figura 56. Pantalla "Visualización de ruta"



Figura 57. Código fuente de la vista "Visualización de ruta"

```

15     Expanded(
16       child: GoogleMap(
17         initialCameraPosition: CameraPosition(
18           target: LatLng(widget.ubicacionActual.latitude,
19             widget.ubicacionActual.longitude),
20           zoom: 14.0,
21         ),
22         markers: _marcadores,
23         polylines: _polilineas,
24         onMapCreated: (GoogleMapController controller) {
25           mapaController = controller;
26         },
27       ),
28     ),
29   ],
30 ),
31 floatingActionButton: FloatingActionButton.extended(
32   backgroundColor: ColorFondo.PRIMARY,
33   onPressed: () {
34     Navigator.pop(
35       context, _estadoCambiado); // Retornar el estado de cambio
36   },
37   icon: const Icon(
38     Icons.check,
39   ),
40   label: const Text(
41     'Guardar estado pedido(s)',

```

4.3.4.2.4. Historia de Usuario 11: Historial de pedidos (Aplicación web y móvil)

Para el desarrollo de la historia de usuario 11, se diseñó una pantalla en la aplicación *web* que permite al usuario de perfil gerente visualizar el listado de pedidos. Los detalles de la historia están a disposición en el anexo 5. En la pantalla se muestra una tabla que presenta información necesaria de cada pedido, como número, estado, fecha de creación, cliente, productos solicitados, forma de pago, total y acciones disponibles.

La interfaz incluye un buscador para buscar rápidamente pedidos específicos. Además, existen un botón en la columna "Acciones" que permiten visualizar detalles cada pedido de forma individual.

Figura 58. Pantalla "Listado de pedidos"

Nro.	Detalle	Estado	Fecha pedido	Cliente	Productos	Forma de pago	Total	Acciones
1		PENDIENTE	12/12/2024	ANTHONY SACHERI SOLORZANO	Paquete de botellas x24	Pago en efectivo	\$8.40	
2		PENDIENTE	12/12/2024	ALBERTO PATRICIO ULLOA SEGURA	Botellón	Transferencia bancaria	\$1.50	

Mostrando 1 a 2 de 2 pedidos

Anterior Siguiente

Figura 59. Código fuente de la vista "Listado de pedidos"

```

2      <h1><span class="fa fa-box"></span> Listado de pedidos</h1>
3    </div>
4    <hr>
5    <div class="row">
6      <div class="col-md-12">
7        <div class="card card-outline card-primary">
8          <div class="card-body">
9            <table id="example1" class="table table-striped table-bordered table-hover table-sm">
10             <thead>
11               <tr>
12                 <th>Nro</th>
13                 <th>Detalle</th>
14                 <th>Estado</th>
15                 <th>Fecha pedido</th>
16                 <th>Cliente</th>
17                 <th>Productos</th>
18                 <th>Forma de pago</th>
19                 <th>Total</th>
20                 <th>Acciones</th>
21               </tr>
22             </thead>
23             <tbody>
24               <!-- Datos dinámicos con AJAX -->
25             </tbody>
26           </table>

```

4.3.4.2.5. Historia de Usuario 12: Registro de entrega del pedido (Aplicación móvil)

Para la historia de usuario 12, los detalles de la historia están disponibles en el anexo 5, se diseñó una funcionalidad en la aplicación móvil que permite confirmar el estado de entrega de los pedidos. En la figura 60, se muestra una ventana emergente asociada a un punto de entrega en el mapa. El usuario de perfil conductor cuenta con opciones para ubicar en el mapa y botones para confirmar si el Pedido fue entregado o si Pedido no fue entregado, permitiendo registrar el estado en tiempo real. Este diseño asegura una gestión eficiente de entregas, logrando actualizar el estado de los pedidos directamente desde la aplicación móvil.

Figura 60. Ventana de "Registro de entrega"



Figura 61. Código fuente de la ventana “Registro de entrega”

```

27     onPressed: () {
28       _confirmarAccion(
29         mensaje: "¿Confirmar que el pedido fue entregado?",
30         onConfirmar: () async {
31           Navigator.of(context).pop(); // Cierra este cuadro
32           bool exito =
33             await PedidosController().cambiarEstadoPedido(
34               cliente['pedido_codigo'], // ID del pedido
35               'ENTREGADO',
36             );
37           if (exito) {
38             mostrarCorrecto(
39               context, 'Pedido marcado como entregado');
40             actualizarEstadoMarcador(
41               cliente['cliente_nombres'], 'ENTREGADO');
42           } else {
43             mostrarError(context, 'Error al marcar como entregado');
44           }
45         },
46       );
47     },
48   ),
49   const SizedBox(height: 8),
50   _crearBotonAccion(
51     texto: "Pedido no entregado",
52     color: Colors.red,
53     icono: Icons.cancel,
54     onPressed: () {
55       _confirmarAccion(
56         mensaje: "¿Confirmar que el pedido no fue entregado?",
57         onConfirmar: () async {
58           Navigator.of(context).pop(); // Cierra este cuadro
59           bool exito =
60             await PedidosController().cambiarEstadoPedido(
61               cliente['pedido_codigo'], // ID del pedido
62               'NO ENTREGADO',
63             );

```

Figura 62. Código de consumo de la API para cambiar estado del pedido

```

1  Future<bool> cambiarEstadoPedido(int idPedido, String nuevoEstado) async {
2    try {
3      var apiUrl =
4        Uri.parse('http://$ROOT:5000/pedidos/cambiar_estado/$idPedido');
5      String? token = await obtenerToken();
6      if (token != null) {
7        var response = await http.put(
8          apiUrl,
9          headers: {"Content-Type": "application/json", "authorization": token},
10         body: json.encode({
11           'nuevo_estado': nuevoEstado,
12           'observacion': '',
13         })),
14       );
15       if (response.statusCode == 200) {
16         return true; // Indicar éxito
17       }
18     }
19   } catch (e) {
20     throw Exception('Error al cambiar el estado del pedido: $e');
21   }
22   return false; // Indicar fallo
23 }

```

La función *cambiarEstadoPedido* actualiza el estado de un pedido enviando una solicitud al servidor con el nuevo estado. Si la operación es exitosa, devuelve *true*; de lo contrario, maneja los errores y devuelve *false*. Este método asegura que los estados de los pedidos puedan modificarse de manera sencilla y confiable.

Figura 63. Código fuente del backend cambiarEstadoPedido

```

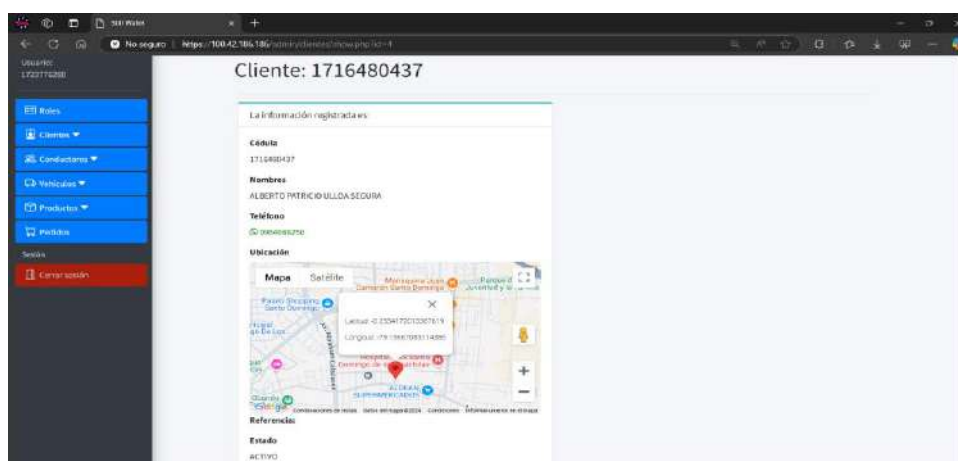
3     const { nuevo_estado, observacion = '' } = req.body; // Recibimos el código del pedido y el nuevo estado
4
5     const condicion = `PEDI_CODIGO = ${req.params.codigo}`;
6
7     if (!nuevo_estado) {
8         return res.status(400).json({ error: "Faltan datos: Nuevo estado" });
9     }
10
11    let token = req.headers.authorization || null;
12    let id_usuario = objToken.obtener_id_usuario(token) || null;
13    if (id_usuario) {
14
15        // Obtenemos la fecha actual y ajustamos a la zona horaria de Ecuador (UTC-5)
16        const fechaActual = new Date();
17        fechaActual.setHours(fechaActual.getHours() - 5); // Restamos 5 horas para ajustar a UTC-5
18
19        // Formateamos la fecha como 'YYYY-MM-DD HH:MM:SS'
20        const fechaFormateada = fechaActual.toISOString().slice(0, 19).replace('T', ' ');
21
22        // Construimos los valores a actualizar
23        const valoresActualizados = {
24            'PEDI_ESTADO': nuevo_estado.toUpperCase(),
25            'PEDI_OBSERVACION': observacion
26        };
27
28        // Si el nuevo estado es "entregado", actualizamos también la fecha de entrega
29        if (nuevo_estado.toUpperCase() === 'ENTREGADO' || nuevo_estado.toUpperCase() === 'NO ENTREGADO') {
30            valoresActualizados['PEDI_FECENTREGA'] = fechaFormateada;
31        }
32
33        // Realizamos la actualización del estado en la tabla 'al_pedidos_rob'
34        objSql.actualizarDatos(
35            tablaPedidos, // Tabla
36            valoresActualizados, // Nuevos valores para las columnas
37            condicion, // Condición para seleccionar el pedido
38            res
39        );

```

4.3.4.2.6. Historia de Usuario 13: Visualizar perfil (Aplicación web)

La pantalla muestra la información completa de un cliente específico en la aplicación web como se exhibe en la figura 64. Incluye toda la información registrada del cliente como la cédula, nombres, teléfono y el estado del cliente (por ejemplo, "activo"). Además, presenta el mapa con el marcador y coordenadas de la ubicación registrada.

Figura 64. Pantalla de "Información usuario"



El diseño organiza la información de manera clara, permitiendo al usuario acceder rápidamente a los datos del cliente, y el mapa facilita la identificación del domicilio, mejorando la gestión y visualización en el sistema.

Figura 65. Código fuente de la vista de “Información usuario”

```

12 <div class="card-body">
13 <div class="row">
14 <div class="col-md-12">
15 <div class="form-group">
16 <label for="">Cédula</label>
17 <p><? $info_cliente[ 'USUA_CEDULA' ] ?</p>
18 </div>
19 </div>
20 <div class="col-md-12">
21 <div class="form-group">
22 <label for="">Nombres</label>
23 <p><? $info_cliente[ 'USUA_NOMBRE' ] . " " . $info_cliente[ 'USUA_APELLI' ] ?</p>
24 </div>
25 </div>
26 <div class="col-md-12">
27 <div class="form-group">
28 <label for="">Teléfono</label>
29 <p>
30 <a href="https://api.whatsapp.com/send?phone=593<? $info_cliente[ 'USUA_TELEFO' ] ?" style="color: green;" target="_blank" rel="noopener noreferrer">
31 <i class="fab fa-whatsapp"></i>
32 <? $info_cliente[ 'USUA_TELEFO' ] ?>
33 </a>
34 </p>
35 </div>
36 </div>
37 <div class="col-md-12">
38 <div class="form-group">
39 <label for="">Ubicación</label>
40 <div id="map" style="height: 300px; width: 100%;</div>
41 <p><b>Referencia:</b> <? $info_cliente[ 'USUA_DIRECC' ] ?</p>
42 </div>
43 </div>
44 <div class="col-md-12">
45 <div class="form-group">
46 <label for="">Estado</label>
47 <p><? $info_cliente[ 'USUA_ESTADO' ] ?</p>
48 </div>

```

Figura 66. Código de consumo de la API para obtener datos del cliente

```

4 $url = "http://" . $iproot . ":5000/clientes/" . $codi_clie;
5
6 // Inicializar cURL
7 $ch = curl_init($url);
8
9 // Configurar opciones de cURL
10 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); // Para recibir la respuesta como string
11
12 // Verificar el estado de la sesión
13 if (session_status() === PHP_SESSION_NONE) {
14     session_start(); // Iniciar sesión si no está activa
15 }
16
17 if (isset($_SESSION['sesion_token'])) {
18     $token = $_SESSION['sesion_token']; // Obtener el token de la sesión
19     curl_setopt($ch, CURLOPT_HTTPHEADER, [
20         'Content-Type: application/json', // Definir el tipo de contenido como JSON
21         'Accept: application/json', // Aceptar respuesta en formato JSON
22         "Authorization: $token",
23     ]);
24 } else {
25     // Si no hay token, redirigir al login
26     header('Location: ' . APP_URL . '/login');
27     exit();
28 }
29
30 // Ejecutar la solicitud
31 $response = curl_exec($ch);
32
33 // Verificar si hubo un error
34 if (curl_errno($ch)) {
35     echo 'Error: ' . curl_error($ch);
36     return null;
37 } else {
38     // Decodificar la respuesta JSON
39     $datos_p = json_decode($response, true);
40 }

```

Figura 67. Código fuente del backend getRegistroCliente

```

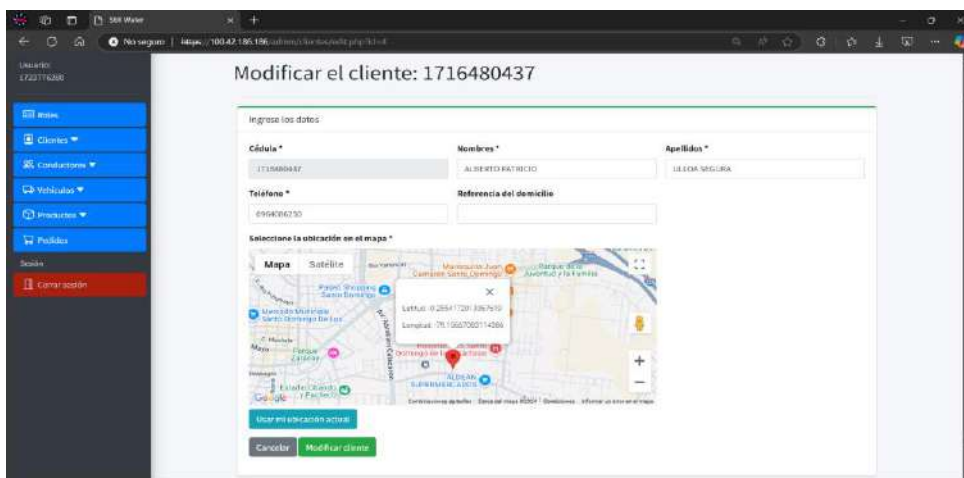
1 const getRegistroCliente = asyncHandler(async (req, res) => {
2     try {
3         let token = req.headers.authorization || null;
4         let id_usuario = objToken.obtener_id_usuario(token) || null;
5         if (id_usuario) {
6             const campos = [
7                 'USUA_CODIGO',
8                 'USUA_CEDULA',
9                 'USUA_NOMBRE',
10                'USUA_APELLI',
11                'USUA_DIRECC',
12                'USUA_LATITUD',
13                'USUA_LONGITUD',
14                'USUA_TELEFO',
15                'USUA_ESTADO'
16            ]
17            objSql.seleccionarUnRegistro(tabla, campos, `USUA_CODIGO=${req.params.codigo}`, [req.params.codigo], res);
18        } else {
19            console.log('No se encontro el ID del usuario')
20            res.status(500).json({ error: "No se encontro el ID del usuario" })
21        }

```

La función *getRegistroCliente*, es una función asíncrona programada para recuperar información de un cliente específico desde la base de datos. Utiliza un *token* de autorización para determinar al usuario que realiza la solicitud. Si se obtiene un identificador válido del usuario, se recuperan los campos que se están especificando en campos = {}, como código, cédula, nombres, apellidos, dirección, latitud y longitud, teléfono y estado.

4.3.4.2.7. Historia de Usuario 14: Editar perfil (Aplicación web)

Figura 68. Pantalla "Modificar cliente"



La pantalla "Modificar el cliente" permite editar la información de un cliente específico. Incluye campos para actualizar los nombres, apellidos, teléfono, referencia del domicilio y la ubicación en el mapa, con la opción de usar la ubicación actual del dispositivo. En la zona inferior, se disponen los botones "Cancelar cliente" y "Modificar cliente", que permiten confirmar o descartar los cambios realizados.

Figura 69. Código fuente de la vista de "Modificar cliente"

```

48     <div class="row">
49         <div class="col-md-8">
50             <div class="form-group">
51                 <label for="">Seleccione la ubicación en el mapa</label><b * />
52                 <div id="map" style="height: 300px; width: 100%;></div>
53                 <button type="button" onclick="getUserLocation()" class="btn btn-info mt-2">
54                     Usar mi ubicación actual
55                 </button>
56                 <!-- Campos ocultos para latitud y longitud -->
57                 <input type="hidden" value="{?=$cliente['USUA_LATITUD']; ?}" name="clie_latitud" id="latitud">
58                 <input type="hidden" value="{?=$cliente['USUA_LONGITUD']; ?}" name="clie_longitud" id="longitud">
59             </div>
60         </div>
61     </div>
62     <div class="row">
63         <div class="col-md-12">
64             <div class="form-group">
65                 <a href="{?=$APP_URL; ?}/admin/clientes" class="btn btn-secondary">Cancelar</a>
66                 <button type="submit" class="btn btn-success">Modificar cliente</button>
67             </div>

```

Figura 70. Código de consumo de la API para editar datos del cliente

```

1  $url = "http://" . $iproot . ":5000/clientes/editar/" . $cliente_codigo; // Cambia esta URL a la de tu API
2
3  // Los datos que quieres enviar a la API
4  $data = [
5      'cedula' => $clie_cedula,
6      'nombre' => $clie_nombre,
7      'apellido' => $clie_apellido,
8      'direccion' => $clie_direccion,
9      'telefono' => $clie_telefono,
10     'latitud' => $clie_latitud,
11     'longitud' => $clie_longitud
12 ];
13
14 // Inicializar cURL
15 $ch = curl_init($url);
16
17 // Configurar cURL para enviar los datos por PUT
18 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
19 curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT');
20 curl_setopt($ch, CURLOPT_HTTPHEADER, [
21     'Content-Type: application/json', // Definir el tipo de contenido como JSON
22     'Accept: application/json', // Aceptar respuesta en formato JSON
23     "Authorization: $token",
24 ]);
25
26 curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data)); // Enviar los datos en formato JSON
27
28 // Ejecutar la solicitud
29 $response = curl_exec($ch);
30
31 // Obtener el código de estado HTTP de la respuesta
32 $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
33

```

Figura 71. Código fuente del backend actualizar

```

41     const camposValores = {
42         // 'USUA_CLAVEX': pass,
43         "USUA_CEDULA": cedula,
44         "USUA_NOMBRE": nombre.toUpperCase(),
45         "USUA_APELLI": apellido.toUpperCase(),
46         "USUA_DIRECC": direccion,
47         "USUA_TELEFO": telefono,
48         "USUA_LATITUD": latitud,
49         "USUA_LONGITUD": longitud
50     };
51
52     const condicion = `USUA_CODIGO = ${req.params.codigo}`;
53     objSql.actualizarDatos(tabla, camposValores, condicion, res);
54 } else {
55     console.log('No se encontro el ID del usuario')
56     res.status(500).json({ error: "No se encontro el ID del usuario" })
57 }

```

4.3.4.3. Sprint 2 – Revisión

Una vez finalizado el desarrollo del segundo *sprint*, se realizó una reunión junto al equipo de desarrollo y gerente de empresa, donde se expusieron todos los avances, funcionalidades y la totalidad de la aplicación. Se revisó cada historia de usuario y sus respectivos escenarios de prueba junto con sus pruebas de aceptación correspondientes, las cuales se evidencian en el anexo 6.

4.3.4.4. Sprint 2 – Retrospectiva

Una vez concluida la revisión del segundo *sprint*, se especificaron tres preguntas para generar la retrospectiva del *sprint*, en la tabla 19 se presentan los resultados.

Tabla 19. Sprint 2 - Retrospectiva

¿Qué funcionó bien en el <i>sprint</i> ?	¿Qué aspectos no funcionaron en el <i>sprint</i> ?	¿En qué áreas podemos trabajar para mejorar?
Se cumplió con la meta del <i>sprint</i> según lo planificado.	Algunos requisitos no fueron claros y generaron dudas en su implementación.	Establecer un plan de coordinación para manejar dependencias entre equipos.

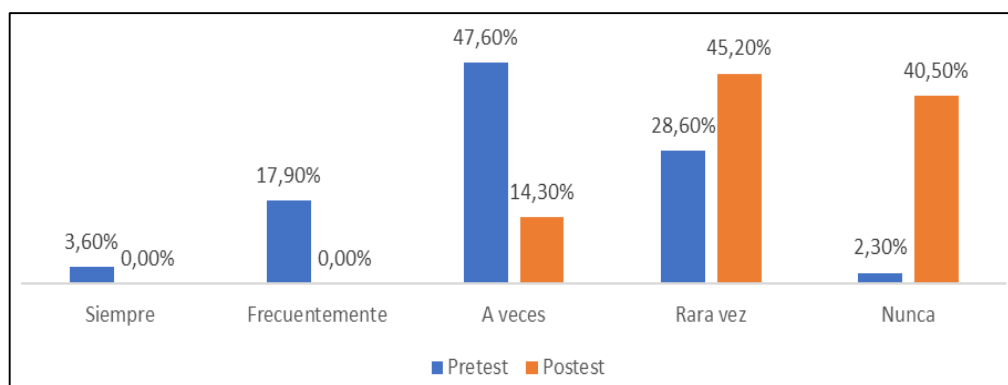
4.4. Validación de la propuesta

4.4.1. Validación de los clientes

Por medio de la realización de encuestas a los clientes de la empresa *Still Water* del cantón Santo Domingo, se logró evidenciar el nivel de satisfacción de los clientes, sobre la eficiencia de distribución del servicio, esta encuesta fue realizada en base a las preguntas planteadas en el instrumento de recolección de información presente en el anexo 3. Luego de realizar el *pre test* y un *post test*, sobre la muestra de 84 clientes, se pudo evidenciar que hubo cambios significativos al momento de comparar los resultados, se pueden observar en las siguientes figuras, donde se presentan tres de las diez preguntas.

Pregunta 1: ¿Con qué frecuencia ha experimentado retrasos significativos en la entrega de agua?

Figura 72. Frecuencia de retrasos en la entrega de agua según los clientes



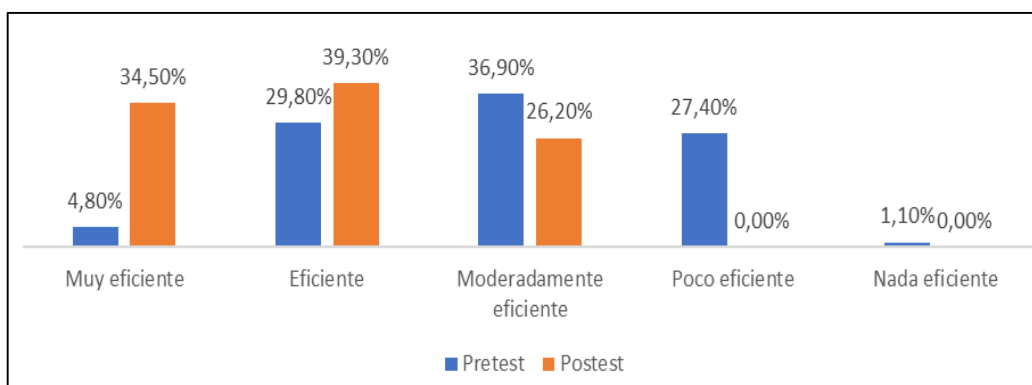
Análisis e interpretación: Los resultados del *pre test* revelan que una gran parte de los clientes afrontan retrasos considerables en la entrega de agua de forma recurrente. Un 3.60% menciona que los retrasos ocurrían siempre, mientras que un 17.90% señala que eran frecuentes. La mayoría de los encuestados que formaban el 47.60% indica que estos

inconvenientes suceden a veces, y un 28.60% afirma que se presentaban rara vez. En contraste, sólo un 2.30% asegura que nunca experimentaba problemas de retraso del servicio, lo que evidenciaba un servicio deficiente antes de la implementación de los aplicativos.

Tras el estímulo generado por la aplicación móvil y *web*, los datos obtenidos en el *post test* reflejaron un avance significativo. Ningún participante reporta retrasos en las categorías de siempre o frecuentemente. Además, la proporción de quienes indican que los retrasos ocurren a veces se redujo considerablemente a un 14.30%, mientras que la cantidad de personas que respondieron rara vez aumenta notablemente hasta un 45.20%. Lo más relevante a destacar es que el 40.50% de los encuestados afirma que nunca experimenta demoras en la entrega de agua, lo que demuestra una mejora sustancial en la regularidad de las entregas, ocasionado por una mejora en la gestión de pedidos y la optimización de rutas de distribución ofrecidas por la aplicación móvil y la aplicación *web*.

Pregunta 2: ¿Cómo evalúa actualmente la eficiencia en la distribución de agua de la empresa *Still Water*?

Figura 73. Grado de eficiencia en la distribución de agua



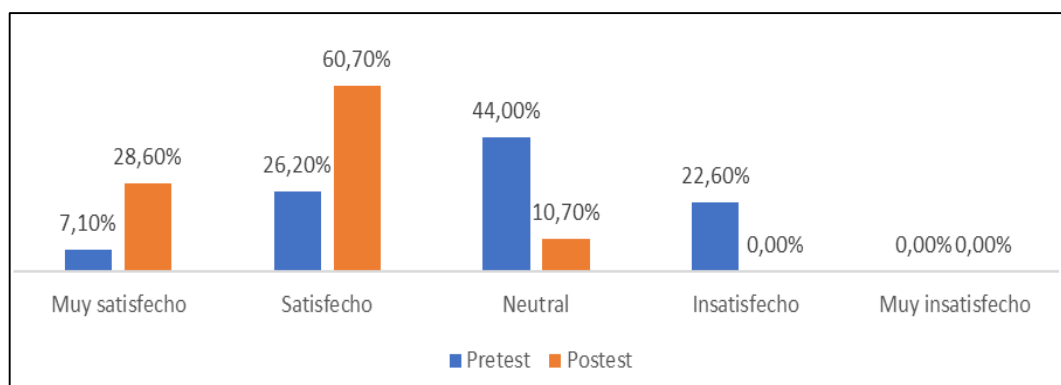
Análisis e interpretación: Los resultados pertenecientes al *pre test* demuestran que la percepción sobre la eficiencia en la distribución de agua por parte de la empresa *Still Water* era en su mayoría negativa. Apenas un 4.80% de los encuestados califica el servicio como muy eficiente, mientras que un 29.80% lo considera eficiente. La mayoría lo evalúa como moderadamente eficiente (36.90%) o poco eficiente (27.40%), e incluso un 1.10% lo

percibe como nada eficiente. Estos resultados reflejaban una opinión generalizada de deficiencia en el servicio ofrecido por la empresa *Still Water*.

Después de la implementación de los aplicativos, los datos del *post test* evidencian una transformación positiva en la percepción del servicio. La proporción de usuarios que lo califican como muy eficiente se eleva significativamente hasta un 34.50%, y el 39.30% lo valora como eficiente, lo que sugiere una mejora considerable. A su vez, la categoría moderadamente eficiente disminuye a 26.20%, mientras que las categorías restantes de poco eficiente y nada eficiente desaparecen por completo. Lo que indica un aumento notable en la percepción de los usuarios en cuanto a la eficiencia del servicio.

Pregunta 3: ¿Qué tan satisfecho está con el servicio general de entrega de agua de la empresa *Still Water*?

Figura 74. Grado de satisfacción de los clientes sobre el servicio



Análisis e interpretación: Antes de la implementación de las aplicaciones, la satisfacción general con el servicio de entrega de agua de la empresa *Still Water* es baja. La mayoría conformado con el 44.00% expresa una opinión neutral, mientras que un 22.60% señala estar insatisfecho a nivel general con el servicio. Apenas un 7.10% de los encuestados declara sentirse muy satisfecho y un 26.20% indica estar satisfecho. Estos resultados reflejaban una percepción general de deficiencia en el servicio.

Tras la implementación, los resultados del *post test* revelaron que el porcentaje de quienes se consideran muy satisfechos aumenta a 28.60%, mientras que la categoría

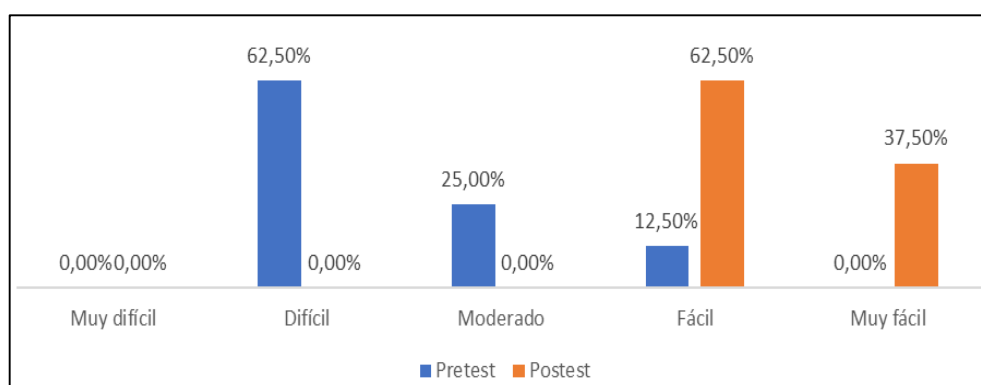
satisfecho crece considerablemente hasta el 60.70%, lo que evidencia un cambio positivo en la percepción del servicio. La categoría neutral disminuye hasta el 10.70%. Estos datos confirman una mejora sustancial en la calidad del servicio, reflejando el impacto positivo de las aplicaciones implementadas.

4.4.2. Validación de los empleados

Del mismo modo, luego de realizar el *pre test* y un *post test*, a los 8 empleados, se observó un aumento en el grado de satisfacción respecto a las distancias y las nuevas rutas de distribución, se pueden observar en las siguientes figuras, donde se presentan cuatro de las dieciséis preguntas. Esta encuesta fue realizada en base a las preguntas planteadas en el instrumento de recolección de información presente en el anexo 3.

Pregunta 1: ¿Qué tan difícil le resulta modificar la ruta al incorporar un cliente nuevo?

Figura 75. Grado de dificultad modificar la ruta



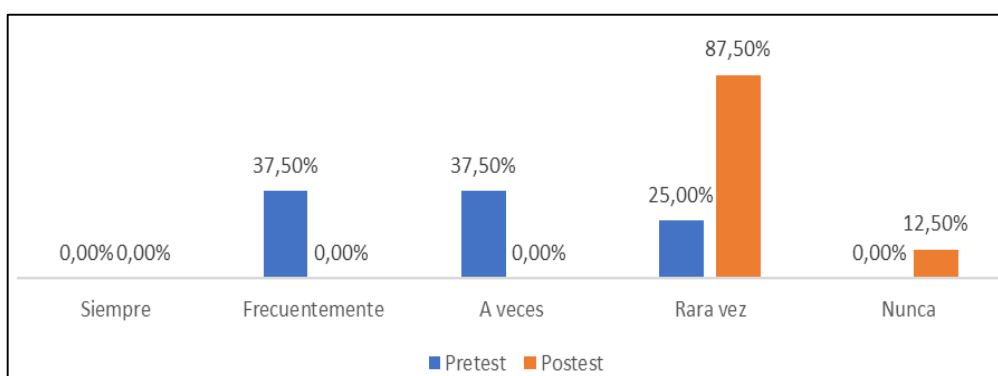
Análisis e interpretación: Los resultados del *pre test* indicaron que la mayoría de los encuestados que conformaban el 62.50% percibe el hecho de modificar la ruta al incorporar un nuevo cliente como una tarea difícil. Un 25.00% lo califica como un proceso de dificultad moderada, mientras que sólo un 12.50% lo considera fácil, lo que sugiere que la flexibilidad en la modificación de rutas era limitada y representa un desafío significativo.

Por otro lado, los resultados del *post test* reflejan un cambio significativo en la facilidad para modificar rutas al aumentar los clientes. La categoría difícil desaparece por

completo, y ahora la mayoría de los encuestados (62.50%) considera que ajustar la ruta es una tarea fácil, mientras que el 37.50% lo califica como muy fácil. Estos datos indican el impacto generado al utilizar las aplicaciones, logrando optimizar de manera significativa la gestión de rutas, permitiendo una adaptación más ágil y eficiente de nuevos clientes.

Pregunta 2: ¿Con qué frecuencia ha recibido quejas de clientes que no se entregó el servicio?

Figura 76. Grado de frecuencia ha recibido quejas

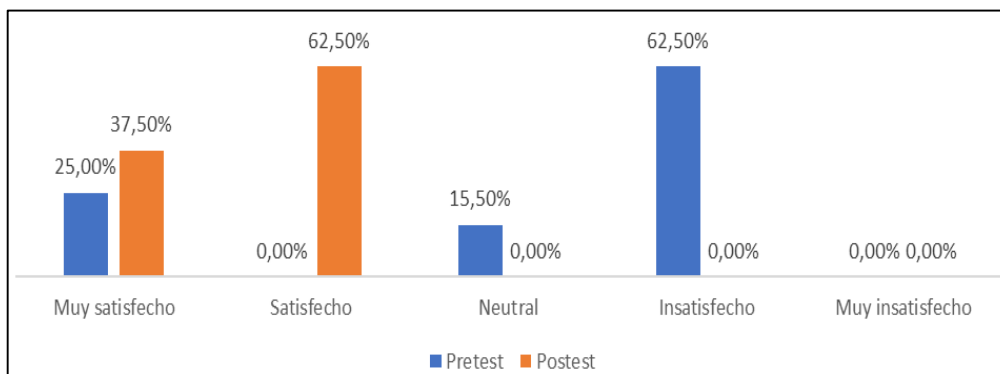


Análisis e interpretación: Los datos obtenidos del *pre test* indican que las quejas de los clientes por falta de servicio eran un problema recurrente. Un 37.50% de los empleados encuestados indica que recibe reclamaciones de manera frecuentemente, mientras que otro 37.50% señala que esto ocurría a veces. Un 25.00% menciona que las quejas sucedían rara vez. Estos resultados reflejan una deficiencia notable del servicio y una gestión inadecuada de pedidos.

Por otro lado, los resultados del *post test* muestran una reducción significativa en la cantidad de quejas de clientes que no se entrega el servicio. Las categorías frecuentemente y a veces desaparecieron por completo, ahora el 87.50% de los encuestados reporta que las quejas ocurren solo rara vez. Además, un 12.50% afirma que nunca recibe reclamos, lo que indica que gracias al uso de la aplicación móvil y aplicación *web* se ha reducido drásticamente la insatisfacción de los clientes y se ha logrado entregar el servicio de manera eficaz.

Pregunta 3: ¿Qué tan satisfecho está con la distancia y los recorridos que realiza utilizando los métodos actuales de planificación de rutas?

Figura 77. Grado de satisfecho está con la distancia y los recorridos

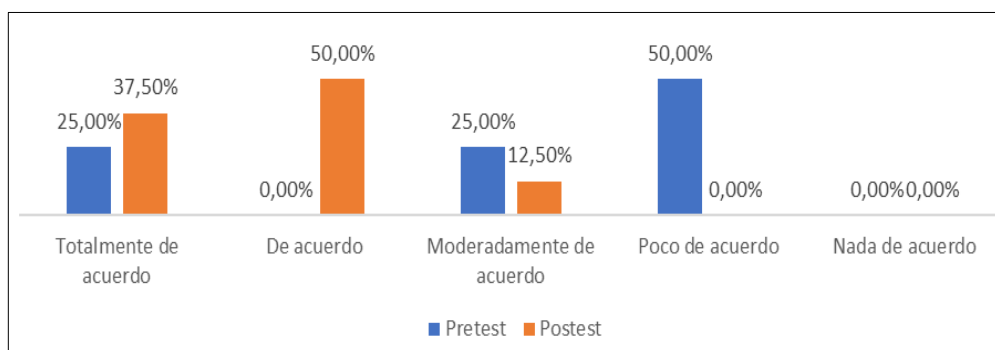


Análisis e interpretación: Los resultados del *pre test* indicaron la insatisfacción con los métodos de planificación de rutas. Un 62.50% de los empleados encuestados declara estar insatisfecho con la distancia y los recorridos realizados, mientras que un 15.50% expresó una opinión neutral, por otro lado, solo un 25.00% afirma estar muy satisfecho. Estos resultados sugieren que, la optimización de rutas representaba un desafío significativo para la empresa.

Luego de la implementación de los aplicativos, los resultados del *post test* evidencian una mejora sustancial en la de planificación de rutas. El 62.50% de los encuestados consideran sentirse satisfechos. Asimismo, el porcentaje de quienes se sienten muy satisfechos con las distancias y recorridos aumenta a 37.50%. Con estos datos, se confirman que al utilizar un algoritmo de camino más corto como Dijkstra, se mejoran y optimizan de manera efectiva las rutas de distribución, mejorando tanto la eficiencia operativa como la satisfacción de los usuarios.

Pregunta 4: Actualmente, ¿Qué tan de acuerdo está con que la gestión y planificación de rutas de distribución se controla de manera eficiente?

Figura 78. Gestión y planificación de rutas



Análisis e interpretación: Los datos obtenidos del *pre test* evidencian que la percepción de los empleados sobre la eficiencia en la gestión y planificación de rutas es mayormente negativa. Sólo un 25.00% de los encuestados se muestra totalmente de acuerdo con que la gestión era eficiente. Un 25.00% expresa una opinión moderadamente de acuerdo, mientras que la mitad de los participantes (50.00%) indica estar poco de acuerdo en que la gestión y planificación de rutas de distribución es llevada de manera eficiente.

Por otro lado, los resultados del *post test* muestran un cambio significativo. La categoría totalmente de acuerdo aumenta a 37.50% y la opinión de 50.00% de los encuestados ahora están de acuerdo en que la gestión y planificación se maneja de forma eficiente. Por su parte, la opción moderadamente de acuerdo se redujo a 12.50%. Estos datos indican que las estrategias implementadas lograron una mejora sustancial en la gestión de rutas, además deja abierta la posibilidad de implementar mejorar y buscar nuevas funcionalidades para aportar valor a la empresa o mejorar los procesos que realiza.

4.5. Validación de hipótesis

Es fundamental llevar a cabo la recodificación de los distintos escenarios tanto en el *pre test* como en el *post test*. Este proceso permitió realizar un análisis más preciso mediante la regresión logística binaria, proporcionando así un respaldo estadístico sólido para la validación de la hipótesis planteada, tal como se muestra en la tabla 20.

Tabla 20. Recodificación de escenarios

Recodificación	Escenario
0	Sin aplicación web y móvil para la optimización de rutas de distribución
1	Con aplicación web y móvil para la optimización de rutas de distribución

La recodificación se realizó y se observa en el anexo 14 en base a los resultados del *pre test* y *post test* de las preguntas realizadas a los clientes y empleados de la empresa, empleando el *software* de IBM SPSS, este proceso se evidencia en la siguiente figura 79.

Figura 79. Proceso de análisis en el software SPSS

The figure shows two screenshots of the IBM SPSS Statistics Editor de datos interface. The top screenshot displays a data table with 14 rows and 7 columns. The columns are: Escenario, ¿Qué tan difícil es usar el sistema?, ¿Cuáles son las características de la oferta actual?, ¿Con qué frecuencia ha experimentado retrasos significativos en la entrega?, ¿Cómo valora actualmente la eficiencia de la distribución?, and ¿Qué tan satisfecho está con el servicio general de entrega? The data in this table is qualitative, using terms like 'Muy fácil', 'Fácil', 'A veces', 'Eficiente', and 'Satisfecho'. The bottom screenshot displays a data table with 14 rows and 7 columns, which is a numerical recoding of the data from the top screenshot. The columns are: Escenario, ¿Qué tan difícil es usar el sistema?, ¿Cuáles son las características de la oferta actual?, ¿Con qué frecuencia ha experimentado retrasos significativos en la entrega?, ¿Cómo valora actualmente la eficiencia de la distribución?, and ¿Qué tan satisfecho está con el servicio general de entrega?. The data in this table is numerical, using values from 0 to 4.

Escenario	¿Qué tan difícil es usar el sistema?	¿Cuáles son las características de la oferta actual?	¿Con qué frecuencia ha experimentado retrasos significativos en la entrega?	¿Cómo valora actualmente la eficiencia de la distribución?	¿Qué tan satisfecho está con el servicio general de entrega?
1	Muy fácil	5 a 7	A veces	Eficiente	Satisfecho
2	Fácil	1	Rara vez	Eficiente	Satisfecho
3	Fácil	1	Rara vez	Muy eficiente	Muy satisfecho
4	Muy fácil	2 a 4	A veces	Eficiente	Satisfecho
5	Muy fácil	2 a 4	Rara vez	Moderadamente eficiente	Satisfecho
6	Fácil	2 a 4	A veces	Moderadamente eficiente	Neutral
7	Muy fácil	1	Rara vez	Eficiente	Satisfecho
8	Muy fácil	1	Rara vez	Eficiente	Satisfecho
9	Muy fácil	2 a 4	Siempre	Moderadamente eficiente	Neutral
10	Muy fácil	2 a 4	A veces	Eficiente	Satisfecho
11	Muy fácil	2 a 4	A veces	Moderadamente eficiente	Neutral
12	Fácil	1	Frecuentemente	Poco eficiente	Muy satisfecho
13	Muy fácil	1	Nunca	Muy eficiente	Muy satisfecho
14	Muy fácil	1	Rara vez	Muy eficiente	Muy satisfecho

Escenario	¿Qué tan difícil es usar el sistema?	¿Cuáles son las características de la oferta actual?	¿Con qué frecuencia ha experimentado retrasos significativos en la entrega?	¿Cómo valora actualmente la eficiencia de la distribución?	¿Qué tan satisfecho está con el servicio general de entrega?
1	0	2	2	3	3
2	0	1	0	3	3
3	0	1	0	4	4
4	0	1	2	3	3
5	0	1	1	2	3
6	0	1	2	2	2
7	0	0	1	3	3
8	0	0	1	3	3
9	0	1	4	2	2
10	0	1	2	3	3
11	0	1	2	2	2
12	0	0	3	1	4
13	0	0	0	4	4
14	0	0	1	4	4

Los resultados obtenidos de clientes se muestran en la tabla 21, mientras que el de los empleados en la tabla 22, con un nivel de significancia menor 0.05, en las dos tablas.

Tabla 21. Resultados de preguntas a clientes

Preguntas	gl	Sig.
¿Con qué frecuencia ha experimentado retrasos significativos en la entrega de agua?	1	,000

¿Cómo evalúa actualmente la eficiencia en la distribución de agua de la empresa Still Water?	1	,000
¿Cómo evalúa actualmente la eficiencia en la distribución de agua de la empresa Still Water?		
¿Qué tan satisfecho está con el servicio general de entrega de agua de la empresa Still Water?	1	,006

Tabla 22. Resultados a preguntas a empleados

Preguntas	gl	Sig.
¿Qué tan difícil le resulta modificar la ruta al incorporar un cliente nuevo?	1	,001
¿Con qué frecuencia ha recibido quejas de clientes que no se entregó el servicio?	1	,004
¿Qué tan satisfecho está con la distancia y los recorridos que realiza utilizando los métodos actuales de planificación de rutas?	1	,014
Actualmente, ¿Qué tan de acuerdo está con que la gestión y planificación de rutas de distribución se controla de manera eficiente?	1	,032

Por lo tanto, los resultados obtenidos validan la hipótesis, dando evidencia que la implementación de la aplicación *web* y *móvil* influye considerablemente en la optimización de rutas de distribución de la empresa *Still Water* del cantón Santo Domingo.

5. DISCUSIÓN

Por medio del empleo de herramientas de recolección de datos, se desarrollaron encuestas a empleados y clientes, que aportaron a facilitar información respecto a las rutas de distribución y un modelo logístico, logrando evidenciar de este modo, las pautas donde se aumentó la eficiencia del servicio general de distribución de agua. Sumado a esto, lo mencionado por los empleados, los cuales del mismo modo se encuentran satisfechos con las distancias y recorridos realizados para ofrecer un servicio más rápido y ágil, reduciendo los tiempos de espera, aumentando la cantidad de entregas y se disminuyeron los costos logísticos. Este criterio se alinea con Mahmoud et al. (2022) que, aparece citado dentro de una obra de Bustamante et al. (2024), donde los modelos logísticos optimizan distancias y/o tiempos, para garantizar que las operaciones asociadas con el usuario final puedan hacer efectivo el despacho del producto o servicio solicitado (p. 4).

En relación con el segundo objetivo específico, se eligieron e integraron de diversas tecnologías, las cuales permitieron construir una aplicación práctica y eficiente. Esta propuesta combina algoritmos de búsqueda y optimización, como el de *Dijkstra*, con bases de datos relacionales como *MySQL*, el desarrollo del *backend* y *APIs* con *Express.js* y *JavaScript*. Asimismo, el desarrollo del *frontend* empleando diversos lenguajes de programación como *HTML*, *CSS*, *Bootstrap* y *JavaScript* para desarrollo *web*, y *Flutter*, para el desarrollo móvil. Este enfoque tiene sustento en investigaciones previas, como las de Bustamante et al. (2024), donde evidenciaron que los algoritmos avanzados pueden reducir significativamente los tiempos y las distancias en la logística (p. 12). De manera similar, la implementación de aplicaciones *web* y móviles favorecen la accesibilidad y flexibilidad para los usuarios, tal como señalan Cardador (2024), quien afirma que con las aplicaciones *web* se puede llegar a miles de usuarios de cualquier parte del mundo (p. 174).

El tercer objetivo corresponde al desarrollo de la aplicación móvil y la aplicación *web*. El desarrollo de las aplicaciones incorporó la implementación de funcionalidades

indispensables para mejorar la optimización de rutas de distribución. Entre las principales funcionalidades incluyen el registro de cliente, inicio de sesión, registro de pedidos, asignación de pedidos a conductores, generación de ruta óptima y visualizar pedido en el mapa. Estas funcionalidades se implementaron de forma exitosa empleando el marco de desarrollo ágil *Scrum*, que permitió una priorización de tareas reduciendo el desperdicio de tiempo y esfuerzos de desarrollo. De igual manera, con este marco de trabajo se entregó el producto de manera incremental, mejorando la planificación con el equipo de trabajo en conjunto con el *product owner*. Este criterio es respaldado por Schwaber y Sutherland (2020), en donde la estructura de *Scrum* está basada en ciclos cortos llamados *sprints*, que garantizan una revisión constante de la incorporación de mejoras y los avances, lo cual es ideal para proyectos que requieren ajustes continuos (pp. 3-4).

El análisis de los datos evidenció un aumento en la satisfacción de los clientes relacionado a la reducción en los tiempos de entrega, además, los empleados también indicaron que están satisfechos con las distancias y los recorridos que realizan. Esto también, se respalda a lo anunciado por Mascarell-Palau (2021), sobre el impacto beneficioso de las aplicaciones móviles para responder a las necesidades específicas, en entornos digitales (p. 82).

De tal manera, se logró validar la hipótesis del estudio, donde, la aplicación web y la aplicación *web* incide significativamente en la optimización de rutas de distribución en la empresa *Still Water* del cantón Santo Domingo.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

En definitiva, se concluye que, gracias al proceso de recolección de datos y a la identificación de los elementos críticos para la generación de rutas, se evidencia las limitaciones de los métodos utilizados. Las encuestas y entrevistas realizadas indican que la falta de herramientas tecnológicas afectaba la eficiencia operativa y a su vez genera insatisfacción en los clientes. Este diagnóstico es de vital importancia al permitir definir los requisitos funcionales necesarios en la aplicación móvil y la aplicación *web*, logrando que respondiera a determinadas necesidades de la empresa y a los retos logísticos de su entorno.

El segundo objetivo, centrado en la selección de herramientas tecnológicas adecuadas para el desarrollo de las aplicaciones, permite integrar soluciones adaptadas a las capacidades de la empresa. Se realizaron comparativas de las diversas herramientas tecnológicas que permiten la correcta selección de estas herramientas en base a sus virtudes y a los requerimientos de la empresa.

Por lo tanto, para las arquitecturas de desarrollo se elige la arquitectura Modelo-Vista-Controlador, con el propósito de distribuir la lógica de negocio de la interfaz de usuario facilitando el desarrollo modular y el mantenimiento, respecto a los lenguajes de desarrollo *frontend web*, se emplea *Bootstrap* en conjunto con la triada *HTML*, *CSS* y *JavaScript*, ofreciendo un diseño e interfaz intuitiva y agradable. Del lado del desarrollo móvil en el *frontend* se emplea *Flutter*, el cual ofrece un solo entorno de desarrollo para diversos sistemas operativos, en este caso los más relevantes en este trabajo son *Android* e *iOS*. En cuando al *backend* y desarrollo de *APIs*, se emplea *Express.js* junto con *JavaScript*, además, como sistema gestor de bases de datos se usa *MySQL*, por la estructuración de datos y el manejo de datos relacionales. Por otro lado, de los diversos algoritmos de caminos más cortos, se emplea *Dijkstra* el cual destaca porque la precisión de sus

resultados es exacta, además añade valor que es un algoritmo ampliamente usado y aceptado por grandes motores de rutas como lo es *Google Maps*.

Finalmente, gracias al marco de trabajo *Scrum*, el cual favorece el desarrollo incremental de las aplicaciones y al manejo de ajustes continuos lo que permite cumplir con los tiempos de entrega. Los resultados de las pruebas realizadas con datos reales indican una reducción en las distancias recorridas, además acompañado de una disminución en los tiempos de entrega de agua, generando de un incremento en la satisfacción del cliente. La herramienta permite mejorar la asignación de rutas y la gestión de pedidos, demostrando que la optimización tecnológica es clave para la competitividad en el sector logístico.

6.2. Recomendaciones

Se recomienda llevar a cabo evaluaciones periódicas a los procesos logísticos de la empresa, en busca de desarrollar nuevas funcionalidades que aporten valor a la empresa y aumenten el grado de satisfacción del cliente y el servicio que se le ofrece. Sin embargo, en cada nueva funcionalidad implementada se debe obtener retroalimentación y aprobación por parte de los clientes, empleando herramientas de recolección de datos como las encuestas, con la finalidad de garantizar que cumpla con las expectativas y necesidades de los clientes.

Por otra parte, se recomienda ampliamente adoptar herramientas complementarias de monitoreo en tiempo real, que permitan supervisar las rutas y el progreso de las entregas. Esto facilita la respuesta inmediata ante imprevistos y una mayor eficiencia en los procesos de distribución. Además, es esencial capacitar de manera continua al personal en el uso de la herramienta. Esto garantiza un manejo adecuado de la aplicación y su correcto funcionamiento.

Finalmente, se recomienda realizar análisis periódicos de los datos generados por las aplicaciones para evaluar el desempeño de las rutas y los tiempos de entrega. Estas evaluaciones permiten identificar áreas de mejora y realizar ajustes dinámicos en las

estrategias de distribución según se requiera, asegurando que las operaciones se mantengan optimizadas con el tiempo.

7. REFERENCIAS

- Aceñero, R. (2023). *Kotlin y Jetpack Compose: desarrollo de aplicaciones Android (1 ed.)*. RA-MA Editorial.
- Afreen, C. F. (2021). *Mobile Applications Development*. Book Rivers. ISBN: 978-93-90548-23-1.
https://islamiahwomensartsandsciencecollege.com/CRITERIA/C3%20DVV/3.3.2/3.3.2_2020-2021/3.3.2_2020-2021_Mobile%20Applications%20Development-Full%20Book.pdf
- Aguirre, S. (2021). *Framework total Vol.1: crea apps desde cero con Laravel + Bootstrap + MySQL*. Plandos. Obtenido de <https://books.google.es/books?hl=es&lr=&id=bvs-EAAAQBAJ&oi=fnd&pg=PP1&dq=integrar+framework&ots=Jp0bBFntWv&sig=OKavOzhVXwyMjHpQwBVBzW5uB5E#v=onepage&q=integrar%20framework&f=false>
- Asamblea Nacional de la República del Ecuador. (2018). *Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial*. Registro Oficial Suplemento 398 de 07-ago.-2008, última modificación 21-ago.-2018. <https://www.lexis.com.ec/biblioteca/ley-organica-transporte-terrestre-transito-seguridad-vial>
- Asamblea Nacional de la República del Ecuador. (2019). *Código Orgánico de la Producción, Comercio e Inversiones*. Registro Oficial Suplemento 351 de 29-dic.-2010, última modificación 31-dic.-2019. <https://www.lexis.com.ec/biblioteca/copci>
- Avalos, O. (2020). *Programación con GO - Vol.2: Parámetros - Estructuras de Control - Arreglos - Sectores*. RedUsers. Obtenido de https://www.google.com.ec/books/edition/Programaci%C3%B3n_con_GO_Vol_2/p6EJEAAAQBAJ
- Baena, P. G. (2017). *Metodología de la investigación (3a. ed.)*. Obtenido de http://www.biblioteca.cij.gob.mx/archivos/materiales_de_consulta/drogas_de_abuso/articulos/metodologia%20de%20la%20investigacion.pdf

- Bhaskar, A. & Manjunath, A. (2020). *An Interpretation and Anatomization of Angular: A Google Web Framework*. International Research Journal of Engineering and Technology (IRJET). <https://www.irjet.net/archives/V7/i5/IRJET-V7I51471.pdf>
- Bootstrap. (2024). *Build fast, responsive sites with Bootstrap*. Obtenido de <https://getbootstrap.com/>
- Bustamante, L. C., Alzate, A. M., & Alzate, M. (2024). *Un modelo de rutas para la recolección de residuos sólidos domiciliarios: un caso de estudio*. Revista De Métodos Cuantitativos Para La Economía Y La Empresa, 37, 1–14. <https://doi.org/10.46661/rev.metodoscuant.econ.empresa.7820>
- Campos, M., Campos, E. M., & López, J. (2022). *Bases de datos avanzadas e ingeniería del software*. RA-MA Editorial. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/230564>
- Cardador, A. L. (2024). *Implantación de aplicaciones web en entornos internet, intranet y extranet*. IFCD0210. IC Editorial. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/269379>
- Castell, R. (2021). *Manual de gestión de tráfico de mercancías*. https://biblioteca.isuc.edu.ec/cgi-bin/koha/opac-detail.pl?biblionumber=1197&query_desc=pl%3A%22%5BBarcelona%2C%20Espa%C3%B1a%20%3A%20%22
- Clúster Logístico del Ecuador, Ministerio de Producción, Comercio Exterior, Inversiones y Pesca, & ASOLOG. (2023). *Encuesta Nacional Logística Ecuador 2023*. Obtenido de <https://clusterlogistico.ec/wp-content/uploads/2023/12/2023-12-11-Resultados-ENL-EC-2023.pdf>
- Constitución del Ecuador, (2008). Trabajo y producción. *Ciencia, tecnología, innovación y saberes ancestrales*. Obtenido de <https://educacion.gob.ec/wp-content/uploads/downloads/2012/08/Constitucion.pdf>
- Corilla, K. V. (2022). *Desarrollo de aplicaciones móviles usando el lenguaje Kotlin*. 1(1), 22-33. doi:<https://doi.org/10.32654/DialogosAbiertos.1-1.3>

- Coronado, B. (2024). *Desarrollo de aplicaciones web en el entorno servidor*. UF1844. Madrid: Editorial Tutor Formación. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/273999>
- Departamento Nacional de Planeación. (2022). *Encuesta Nacional Logística 2022*. DNP. <https://rutadelasostenibilidad.org/wp-content/uploads/2023/12/Encuesta-nacional-logistica-1.pdf>
- Django. (2024). *Django: The web framework for perfectionists with deadlines*. Obtenido de <https://www.djangoproject.com/>
- Escarcena, M. (2020). *Programación páginas Web: JavaScript y PHP*. Madrid: RA-MA Editorial. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/222656>
- Evan You. (2024). *Vue.js - The Progressive JavaScript Framework | Vue.js*. Obtenido de <https://vuejs.org/>
- Fernández, P. (2023). *Construcción y diseño de páginas web con HTML, CSS y JavaScript*. Madrid: RA-MA Editorial. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/235052>
- Flórez, H., & Hernández, J. (2021). *Aplicaciones web con PHP*. Madrid: RA-MA Editorial. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/230578>
- Gamboa, J., Armijo, G., Pluas, R. & Tovar, G. (2019). *La logística como instrumentos de gestión empresarial*. <https://doi.org/10.23857/fipcaec.v4i1%20ESPECIAL.112>
- Google. (2024a). *Angular*. Obtenido de <https://angular.dev/overview>
- Google. (2024b). *Flutter*. Obtenido de <https://flutter.dev/>
- Granados La Paz, E. L. (2023). *Desarrollo de aplicaciones web en el entorno servidor*. IFCD0210. Antequera, Málaga: IC Editorial. <https://www.iceditorial.com/desarrollo-de-aplicaciones-con-tecnologia-web-ifcd0210/10830-desarrollo-de-aplicaciones-web-en-el-entorno-servidor-uf1844-9788411841764.html>
- Hemmati-Sarapardeh, A., Larestani, A., Amar, M & Hajirezaie, S. (2020). *Aplicaciones de técnicas de inteligencia artificial en la industria petrolera*. Obtenido de <https://doi.org/10.1016/C2018-0-04421-7>

- Hernández, R., & Mendoza, C., P. (2018). *Metodología de la investigación: las rutas: cuantitativa y cualitativa y mixta*. México: Mc Graw Hill- educación.
<http://repositorio.uasb.edu.bo:8080/handle/54000/1292>
- Herrera, I.D., Cadena J., León, J.C., Herrera, E.P., Chávez, D., & Rosales, A. (2024). A *Heuristic Procedure for Improving the Routing of Urban Waste Collection Vehicles Using ArcGIS*. *Sustainability*, 16(13), 5660. <https://doi.org/10.3390/su16135660>
- Kumawat, S., Dudeja, C. & Kumar, P. (2021). *An Extensive Review of Shortest Path Problem Solving Algorithms*. <https://ieeexplore.ieee.org/document/9432275>
- Ladrón de Guevara, M. Á. (2024). *Utilización de las bases de datos relaciones en el sistema de gestión y almacenamiento de datos. UF0348 (1 ed.)*. Editorial Tutor Formación.
Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/256855>
- Laravel Holdings Inc. (2024). *Laravel - The PHP Framework For Web Artisans*. Obtenido de <https://laravel.com/>
- Layedra, N. P., Ramos, M. V., Salazar, S. A., & Baldeón, B. A. (2022). *Análisis de los lenguajes de programación más utilizados en el desarrollo de aplicaciones web y móviles*. *Dominio de las Ciencias*, 8(3), 1601-1625. <https://doi.org/10.23857/dc.v8i3>
- Lazcano, R., Valencia, L., Baena, D., & Venegas, R. (2019). *React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma*. *Revista Digital Universitaria*, 20. doi:<http://doi.org/10.22201/codeic.16076079e.2019.v20n5.a5>
- López, J., Campos, E. M., & Campos, M. (2023). *Algoritmia y bases de datos (1 ed.)*. Madrid: RA-MA Editorial. Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/230563>
- López, M., Vara, J., Soltero, F., & Moreno, Á. (2016). *MF0492_3 Programación Web en el Entorno Servidor*. España: RA-MA Editorial. Obtenido de https://www.google.com.ec/books/edition/MF0492_3_Programaci%C3%B3n_Web_en_el_Entorno/L6e6EAAAQBAJ?hl=es&gbpv=1
- Mascarell-Palau, D. (2021). *Aportaciones científicas sobre apps para dispositivos móviles. Vinculaciones educativas y a las artes visuales. Revisa de Imagen, Artes y*

Educación Crítica y Social, 6, 80-91.

doi:<https://dx.doi.org/10.12795/Communiars.2021.i06.06>

Meta Platforms (2024). *React Native*. Obtenido de <https://reactnative.dev/>

Microsoft (2023). *¿Qué es Xamarin?* Obtenido de <https://learn.microsoft.com/es-es/previous-versions/xamarin/get-started/what-is-xamarin>

Microsoft (2024). *Xamarin*. Obtenido de <https://dotnet.microsoft.com/es-es/apps/xamarin>

MongoDB (2024). *MongoDB Atlas: Sitio Oficial*. Obtenido de <https://www.mongodb.com/es>

Naciones Unidas (2018). *La Agenda 2030 y los Objetivos de Desarrollo Sostenible: una oportunidad para América Latina y el Caribe* (LC/G.2681-P/Rev.3), Santiago.

<https://repositorio.cepal.org/server/api/core/bitstreams/cb30a4de-7d87-4e79-8e7a-ad5279038718/content>

Napoli, M. (2020). *Beginning Flutter®: A Hands On Guide To App Development*. John Wiley & Sons, Inc.

NTT DATA & Centro Español de Logística (2023). *III Estudio de la Logística del E-commerce: Desafíos y soluciones en la logística del e-commerce*.

<https://es.nttdata.com/documents/informe-ecommerce-2023.pdf>

Ohyver, M., Moniaga, J., Sungkawa, I., Subagyo, B., & Chandra, I. (2019). *The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test*. doi:10.1016/j.procs.2019.08.231

OpenJS. (2024). *Express - Node.js web application framework*. Obtenido de <https://expressjs.com/es/>

Oracle. (2024a). *Database | Oracle*. Obtenido de <https://www.oracle.com/database/>

Oracle. (2024b). *MySQL*. Obtenido de <https://www.mysql.com/products/workbench/>

Pavón, J. & Llarena, E. (2015). *Creación de un sitio web con PHP y MySQL: (5 ed.)*. RA-MA Editorial. <https://elibro.puce.elogim.com/es/ereader/puce/106491?page=17>

Pinzón, S., Rodríguez, R., & Vanegas, C. (2019). *Java y el patrón Modelo-Vista-Controlador (MVC)*. Colombia: UD.

- Postigo, A. (2021). *Bases de datos*. Ediciones Paraninfo, S.A. Obtenido de https://www.google.com.ec/books/edition/Bases_de_datos/DHE-EAAAQBAJ?hl=es419&gbpv=0
- Postigo, A. (2022). *Gestión de bases de datos*. Ediciones Paraninfo. Obtenido de <https://books.google.es/books?id=4Yh3EAAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false>
- Puetate, G., & Ibarra, J. (2020). *Aplicaciones Móviles Híbridas*. Centro de Publicaciones Pontificia Universidad Católica.
- Quisaguano, L., Pallasco, M., Andaluz, A., Martínez, M., & Corrales, S. (2022). *Desarrollo híbrido con flutter*. *Ciencia Latina Revista Científica Multidisciplinar*, 6(4), 4594-4609. doi:https://doi.org/10.37811/cl_rcm.v6i4.2959
- Ragàs, I. (2024). *Logística Urbana. Manual para operadores logísticos y administraciones públicas*. <https://www.studocu.com/ec/document/universidad-tecnica-particular-de-loja/logitica-urbana/logistica-urbana-libro-de-logistica/82730638>
- Ramos, M. de P., Gonçalves, I. A., Nero, M. A., Bortone, T. P., Martini, R. J., & Temba, P. C. (2024). *Otimização de rotas de transporte em estradas florestais: um estudo de caso em Paraopeba/MG*. *Revista De Gestão E Secretariado*, 15(4), e3633. <https://doi.org/10.7769/gesec.v15i4.3633>
- Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley.
- Sánchez, J. (2015). *Implantación de Aplicaciones Web*. Madrid: Ibergarceta Publicaciones. <https://www.garceta.es/catalogo/libro.php?ISBN=978-84-1622-830-0&idd=12>
- Sarder, M. (2021). *Logistics Transportation Systems*. Elsevier. <https://doi.org/10.1016/C2017-0-04218-0>
- Sarrion, E. (2022). *JavaScript from Frontend to Backend: Learn full stack JavaScript development using the MEVN stack with quick and easy steps*. Packt Publishing Ltd. https://books.google.com.ec/books/about/JavaScript_from_Frontend_to_Backend.html?id=qOV5EAAAQBAJ

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum:*

The Rules of the Game. Obtenido de

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

So, P. (2018). *Architect and Implement Decoupled Drupal Architectures Across the Stack*.

<https://doi.org/10.1007/978-1-4842-4072-4>

Tailwind Labs Inc. (2024). *Construya rápidamente sitios web modernos sin tener que salir*

de su HTML. Obtenido de <https://tailwindcss.com/>

Toroslu, I. (2021). *Improving The Floyd-Warshall All Pairs Shortest Paths Algorithm*.

<https://doi.org/10.48550/arXiv.2109.01872>

Urbano, M. (2015). *Administración y auditoría de los servicios Web*. IFCT0509. IC Editorial.

Obtenido de <https://elibro.puce.elogim.com/es/ereader/puce/45242>

Urquhart, N. (2022). *Nature Inspired Optimisation for Delivery Problems*.

<https://doi.org/10.1007/978-3-030-98108-2>

Vizcaíno, P., Cedeño, R. & Maldonado, I. (2023). *Metodología de la investigación científica:*

guía práctica. Ciencia Latina Internacional. https://doi.org/10.37811/cl_rcm.v7i4.7658

8. ANEXOS

Anexo 1: Carta de asignación, registro de recursos y cronograma



RECURSOS	CANTIDAD	VALOR UNITARIO	VALOR TOTAL USD
GASTOS			
Humano			
Estudiantes	2	-	-
Tecnología			
Computadora	2	\$ 750,00	\$ 1.500,00
Punto de Acceso Wifi	1	\$ 50,00	\$ 50,00
Regulador de Voltaje	2	\$ 20,00	\$ 40,00
Smartphones (Testing)	2	\$ 300,00	\$ 600,00
Materiales			
Generador EG1000 - 120V - 1000W	1	\$ 500,00	\$ 500,00
Aceite 10w30			
Havoline Synthetic Technology 1/4g	4	\$ 10,00	\$ 40,00
Gasolina Extra (litros)	357,5	\$ 0,71	\$ 253,83
Servicios			
Servidor	1	\$ 240,00	\$ 240,00
Google API maps	1	\$ 50,00	\$ 50,00
API WebServices Ecuador	1	\$ 10,00	\$ 10,00
TOTAL:			\$ 3.283,83
INGRESOS			
Recursos propios	-	-	\$ 400,00
TOTAL:			\$ 400,00

N°	Actividades	Estado	2024												2025			
			Producto Mínimo Viable I = 7° nivel						Producto Mínimo Viable II = 8° nivel									
			Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Enero	Febrero	Marzo			
ÉPICA 1	Desarrollo de la propuesta	HECHO				☹️												
ÉPICA 2	Ejecución del Proyecto	HECHO																☹️
ÉPICA 3	Revisión literaria	HECHO					☹️											
ÉPICA 4	Metodología de la investigación	HECHO					☹️											
ÉPICA 5	Resultados	HECHO																☹️
ÉPICA 5.1	Del primer objetivo específico	HECHO										☹️						
ÉPICA 5.2	Del segundo objetivo específico	HECHO										☹️						
ÉPICA 5.3	Del tercer objetivo específico	HECHO											☹️					
ÉPICA 6	Desarrollo de la propuesta de intervención (Gestión adaptativa con incremento iterativo "Scrum")	HECHO																☹️
ÉPICA 6.1	Prototipo de diseño operativo, llamado solución en punta	HECHO																☹️
ÉPICA 6.2	Sprint 1	HECHO																☹️
ÉPICA 6.3	Sprint 2	HECHO																☹️
ÉPICA 7	Conclusiones y Recomendaciones	HECHO																☹️
ÉPICA 8	Informe Final del Trabajo de Titulación de Grado	HECHO																☹️
ÉPICA 8.1	Correcciones de Lectores	HECHO																☹️
ÉPICA 8.2	Anexos	HECHO																☹️
ÉPICA 9	Disertación de Grado	HECHO																☹️

Nota: PMV=Producto mínimo viable; ☹️= Peligoso la semana con "Done"

Anexo 2: Carta de impacto y consentimiento informado



Santo Domingo, 28 de diciembre de 2024

PhD. Yulio Cano de la Cruz
Director de Investigación y Postgrados
Pontificia Universidad Católica del Ecuador Sede Santo Domingo
Presente.-

De mi consideración:

Reciba un cordial saludo y deseos de éxitos en sus delicadas funciones.


Por medio del presente, pongo en su conocimiento que el proyecto de disertación de grado titulado **APLICACIÓN MÓVIL Y WEB PARA LA OPTIMIZACIÓN DE RUTAS DE DISTRIBUCIÓN DE AGUA EN LA EMPRESA STILL WATER DEL CANTÓN SANTO DOMINGO**, elaborado por el Sr. ANDRADE AGUILAR ROBERT STEEVEN y el Sr. SACHERI SOLÓRZANO ANTHONNY, ha generado un impacto positivo en nuestras operaciones diarias, incrementando la eficiencia y maximizando el rendimiento de la empresa.

Por la atención dada a la presente, me suscribo de usted.

Atentamente,



Sr. Alberto Patricio Ulloa Segura
Gerente de Still Water

 **SANTO DOMINGO**
Será en mis fealdades

Consentimiento informado

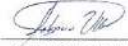
La empresa Still Water, libre y voluntariamente participa en el proyecto de Trabajo de Titulación de Grado de la Pontificia Universidad Católica del Ecuador Sede Santo Domingo, con el título "**Aplicación móvil y web para la optimización de rutas de distribución de agua en la empresa Still Water del cantón Santo Domingo**", elaborado por el Sr. Andrade Aguilar Robert Steeven y el Sr. Sacheri Solórzano Anthony, estudiantes de la carrera de Ingeniería de Tecnologías de la Información.

Luego de firmar este documento certifico lo siguiente:


- Recibimos una copia de este documento de consentimiento informado.
- Estamos de acuerdo en que los datos recopilados, fotografías y resultados de este proyecto de Trabajo de Titulación de Grado se publiquen en artículos académicos, conferencias, en páginas web institucionales y en otros medios de comunicación.
- No esperamos recibir beneficios o pago por la participación.


Y a los efectos que procedan, firmamos el presente consentimiento informado.
Santo Domingo, 22 de noviembre del 2024.

Firma del Gerente de la empresa Still Water


Sr. Ulloa Segura Alberto Patricio

Firma de los autores del Trabajo de Titulación de Grado:


Sr. Andrade Aguilar Robert Steeven


Sr. Sacheri Solórzano Anthony

Dirección: Vía a Chone Km. 2
Código postal: 230203 / Teléfono: 593 - 0993285425
Santo Domingo - Ecuador / www.pucsd.edu.ec
f t in d

Anexo 3: Validación de instrumentos de recolección de datos



Santo Domingo, 29 de noviembre del 2024

Estimado Mg. Javier Ulloa,
De mi consideración:

El motivo del presente es que le hemos elegido a usted para redactar la solicitud de revisión y validación de los instrumentos de recolección de datos.

A continuación, encontrará la entrevista y encuesta que contienen las preguntas que permitirán la recolección de información de acuerdo con el trabajo de titulación "APLICACIÓN MÓVIL Y WEB PARA LA OPTIMIZACIÓN DE RUTAS DE DISTRIBUCIÓN DE AGUA EN LA EMPRESA STILL WATER DEL CANTÓN SANTO DOMINGO"; dirigida a los clientes, empleados y al gerente.

Para la validación de los instrumentos se ajusta la operacionalización de variables, con la finalidad de que se visualice la relación de las preguntas con las dimensiones e indicadores. Además, se encuentran divididos los instrumentos en dos partes, la primera corresponde a la entrevista (preguntas de fondo verbal) para las dos variables; y la segunda a la encuesta (preguntas de la variable independiente de fondo no verbal) y las preguntas de la variable dependiente fondo verbal.

Gracias por su valiosa colaboración en este trabajo de titulación de grado.
Atentamente,

Anthony Sacheri Solórzano
asacheri@pucead.edu.ec

Robert Steeven Andrade Aguilera
randrade@pucead.edu.ec



Operacionalización de las variables

Operacionalización de la variable independiente

Operacionalización variable independiente aplicación móvil

Table with 4 columns: Dimension, Instrumento, Preguntas, Herramientas. Rows include 'Tipo de aplicación' with questions about user experience and tools like Google Forms, and 'Desarrollo Frontend' with questions about development process and tools like Visual Studio.



Operacionalización de la variable dependiente

Operacionalización variable dependiente aplicación web

Table with 4 columns: Dimension, Instrumento, Preguntas, Herramientas. Rows include 'Tipo de ruta' with questions about route optimization and tools like ArcGIS, and 'Desarrollo Backend' with questions about development process and tools like Java, Spring Boot.

1

2



Operacionalización variable dependiente aplicación web

Table with 4 columns: Dimension, Instrumento, Preguntas, Herramientas. Rows include 'Tipo de ruta' with questions about route optimization and tools like ArcGIS, and 'Desarrollo Backend' with questions about development process and tools like Java, Spring Boot.

4

Preguntas

Objetivo:

Identificar los procedimientos sobre la generación y manejo de rutas de distribución en la empresa Still Water para presentar una propuesta que aumente la eficiencia en estas actividades.

Baremo:

Claridad: Se refiere si la pregunta está comprendida por los destinatarios.

Pertinencia: Se refiere si la pregunta corresponde con lo que se quiere indagar

Las preguntas en cuanto a su claridad y pertinencia se encuentran bajo la escala valorativa Likert del 1 al 5 (donde 1 es el menor valor y 5 el mayor). Podrá añadir una formulación alternativa y observación, en caso de que considere necesario

Marque con una cruz (X) el tramo del baremo que exprese mejor su juicio "Claridad" y "Pertinencia" sobre los ítems propuestos:

Entrevista dirigida gerente de la empresa

Variable independiente: Aplicación móvil (Dimensión: Desarrollo Frontend)

1. ¿Cuál considera que sería el principal beneficio de tener una aplicación móvil para facilitar tareas cotidianas, como la gestión y optimización de las rutas de distribución de agua para la empresa?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Aplicación móvil (Dimensión: Desarrollo Frontend)

2. ¿Qué tipo de experiencia espera que los clientes tengan cuando utilicen la aplicación móvil para solicitar sus productos?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Optimización de rutas de distribución (Dimensión: Tipos de rutas)

8. ¿Cómo se maneja la planificación de rutas de distribución cuando la demanda es alta?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Optimización de rutas de distribución (Dimensión: Tipos de rutas)

9. ¿Cuáles son los principales beneficios que obtienen los empleados en las rutas de distribución de los productos?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Optimización de rutas de distribución (Dimensión: Tipos de flotas)

10. ¿Cómo se evalúa el rendimiento de los conductores y cuáles son los criterios más importantes para medir su desempeño?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Optimización de rutas de distribución (Dimensión: Algoritmo de camino más corto)

11. ¿Considera que un sistema de gestión y optimización de rutas de distribución podría mejorar la eficiencia del servicio y aumentar la satisfacción del cliente?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Aplicación web (Dimensión: Arquitectura de aplicaciones web)

3. ¿Qué tan importante consideraría que la aplicación tenga una versión web complementaria para gestionar la información de los clientes y conductores?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Aplicación web (Dimensión: Desarrollo Backend)

4. ¿Qué tan importante es para usted la privacidad y seguridad de los datos cuando usa aplicaciones web o móviles?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable independiente: Aplicación web (Dimensión: Sistema gestor de base de datos)

5. ¿Qué beneficios cree que le puede proporcionar a la empresa contar con una base de datos actualizada de sus clientes?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Optimización de rutas de distribución (Dimensión: Tipos de rutas)

6. ¿Cuáles son las rutas de distribución que maneja la empresa y con qué frecuencia se cubren?

CLARIDAD					PERTINENCIA					FORMULACIÓN ALTERNATIVA	OBSERVACIÓN
1	2	3	4	5	1	2	3	4	5		
				x					x		

Variable dependiente: Optimización de rutas de distribución (Dimensión: Tipos de rutas)

7. ¿Cuáles son los procesos principales para la planificación y coordinación de las entregas dentro de la empresa?

Encuesta dirigida a los clientes

Tema del Trabajo de Titulación de grado: Aplicación móvil y web para la optimización de rutas de distribución en la empresa Still Water del Cantón Santo Domingo

Objetivo: Recolectar información para validar la propuesta de intervención enfocada al mejoramiento del proceso logístico de gestión y optimización de rutas de distribución.

Instrucciones al público objetivo: La encuesta está dirigida a los clientes de la empresa de distribución Still Water, en base a la información obtenida, permitirá conocer el manejo de nuevas tendencias tecnológicas en el manejo de procesos de logística.

PREGUNTAS

Variable independiente: Aplicación móvil (Dimensión: Sistemas operativos)

1. ¿Tiene a su disposición su dispositivo inteligente?

Relevancia		Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2		3	4	5	
				x					x							

Variable independiente: Aplicación móvil (Dimensión: Sistemas operativos)

2. ¿Cuál es el sistema operativo de su dispositivo móvil?

Relevancia		Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2		3	4	5	
				x					x							

Variable independiente: Aplicación móvil (Dimensión: Tecnologías Frontend)

3. ¿Que tan difícil le resulta acceder a internet?

Relevancia		Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2		3	4	5	
				x					x							

Variable independiente: Aplicación móvil (Dimensión: Tipos de aplicaciones)

4. ¿Con qué frecuencia solicita productos por medio de una aplicación?

- a) Siempre
- b) Frecuentemente
- c) A veces
- d) Rara vez
- e) Nunca

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable dependiente: Optimización de rutas de distribución (Dimensión: Tipos de rutas)

5. ¿Cuál es la cantidad de bidones de agua que consume semanalmente en la actualidad?

- a) 1
- b) 2 a 4
- c) 4 a 7
- d) 7 a 10
- e) Más de 10

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable dependiente: Optimización de rutas de distribución (Dimensión: Algoritmos de camino más corto)

6. ¿Con qué frecuencia ha experimentado retrasos significativos en la entrega de agua?

- a) Siempre
- b) Frecuentemente
- c) A veces
- d) Rara vez
- e) Nunca

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable dependiente: Optimización de rutas de distribución (Dimensión: Tipos de rutas)

Encuesta dirigida a los empleados

Tema del Trabajo de Titulación de grado: Aplicación móvil y web para la optimización de rutas de distribución en la empresa Still Water del Cautín Santo Domingo

Objetivo: Recolectar información para validar la propuesta de intervención enfocada al mejoramiento del proceso logístico, de gestión y optimización de rutas de distribución.

Instrucciones al público objetivo: La encuesta está dirigida a los empleados de la empresa de distribución Still Water, en base a la información obtenida, permitirá conocer el manejo de nuevas tendencias tecnológicas en el manejo de procesos de logística.

PREGUNTAS

Variable independiente: Aplicación móvil (Dimensión: Sistemas operativos)

1. ¿Tiene a su disposición un dispositivo inteligente?

- a) Sí
- b) No

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable independiente: Aplicación móvil (Dimensión: Sistemas operativos)

2. ¿Cuál es el sistema operativo de su dispositivo móvil?

- a) Android
- b) IOS
- c) Windows Phone
- d) Otro

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable independiente: Aplicación móvil (Dimensión: Desarrollo Frontend)

3. ¿Qué tan difícil le resulta acceder a sistemas?

- a) Muy difícil
- b) Difícil
- c) Moderado
- d) Fácil
- e) Muy fácil

7. ¿Cómo evalúa actualmente la eficiencia en la distribución de agua de la empresa Still Water?

- a) Muy eficiente
- b) Eficiente
- c) Moderadamente eficiente
- d) Poco eficiente
- e) Nada eficiente

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable dependiente: Optimización de rutas de distribución (Dimensión: Algoritmos de camino más corto)

8. ¿Que tan satisfecho está con el servicio general de entrega de agua de la empresa Still Water?

- a) Muy satisfecho
- b) Satisfecho
- c) Neutral
- d) Insatisfecho
- e) Muy insatisfecho

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable independiente: Aplicación móvil (Dimensión: Sistemas operativos)

4. ¿Qué tan importante consideran que una aplicación sea intuitiva y requiera poco entrenamiento para su uso?

- a) Muy importante
- b) Importante
- c) Neutral
- d) Poco importante
- e) Nada importante

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable independiente: Aplicación móvil (Dimensión: Tipos de aplicaciones)

5. ¿Qué tan de acuerdo está con la implementación de una aplicación móvil y web que asigne a los clientes en sectores o zonas específicas para optimizar las rutas de distribución?

- a) Totalmente de acuerdo
- b) De acuerdo
- c) Moderadamente de acuerdo
- d) Poco de acuerdo
- e) Nada de acuerdo

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Variable independiente: Aplicación móvil (Dimensión: Tipos de aplicaciones)

6. ¿Que tan importante considera que las instrucciones para las rutas de distribución diarias en la aplicación móvil sean claras?

- a) Muy importante
- b) Importante
- c) Neutral
- d) Poco importante
- e) Nada importante

Relevancia					Claridad					Formulación alternativa:					Observación:				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
			x											x					

Anexo 4: Expertos en evaluación de instrumentos

Expertos en evaluación de instrumentos para recopilación de datos		
Nombres	Título Académico	Área
Luis Javier Ulloa Meneses	Mgtr. En Informática Empresarial	Sistemas
Rodolfo Sirjlo Córdova Gálvez	Mgtr. Redes de Comunicaciones	Sistemas

Anexo 5: Historias de usuario

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre historia: Registro de pedido (Aplicación móvil)	
Prioridad en negocio: 100	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Robert Andrade	
Descripción: Como cliente Quiero ingresar los detalles completos de mi pedido Para que me hagan llegar mi pedido.	
Escenario de prueba: Dado el ingreso de los detalles completos del pedido Cuando presione el botón "Registrar pedido" Entonces se visualiza un mensaje de éxito. Dado el ingreso de los detalles completos del pedido Cuando presione el botón "Registrar pedido" Entonces se visualiza un mensaje de error pedido ya está en proceso. Dado el ingreso de los detalles incompletos del pedido Cuando presione el botón "Registrar pedido" Entonces se visualiza un mensaje de error.	
Historia de Usuario	
Número: 2	Usuario: Gerente
Nombre historia: Registro de conductor (Aplicación web)	
Prioridad en negocio: 95	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Robert Andrade	
Descripción: Como gerente Quiero ingresar datos completos y correctos del conductor Para facilitar la comunicación y coordinación durante las entregas.	
Escenario de prueba: Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar conductor" Entonces se visualiza un mensaje de éxito. Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar conductor" Entonces se visualiza un mensaje de error conductor ya está registrado. Dado el ingreso de datos incompletos y/o incorrectos Cuando se presione el botón "Registrar conductor" Entonces se visualiza un mensaje de error.	

Historia de Usuario	
Número: 3	Usuario: Gerente
Nombre historia: Registro de vehículo (Aplicación web)	
Prioridad en negocio: 90	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Anthonny Sacheri	
Descripción: Como gerente Quiero ingresar datos completos y correctos del vehículo Para asegurar una asignación eficiente de pedidos.	
Escenario de prueba: Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar vehículo" Entonces se visualiza un mensaje de éxito. Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar vehículo" Entonces se visualiza un mensaje de error vehículo ya está registrado. Dado el ingreso de datos incompletos y/o incorrectos Cuando se presione el botón "Registrar vehículo" Entonces se visualiza un mensaje de error.	

Historia de Usuario	
Número: 4	Usuario: Gerente
Nombre historia: Registro de cliente (Aplicación web y aplicación móvil)	
Prioridad en negocio: 85	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Robert Andrade	
Descripción: Como gerente Quiero ingresar datos completos y correctos de clientes Para tener los registros actualizados.	
Escenario de prueba: Dado el ingreso de datos completos y correctos del cliente Cuando presione el botón "Registrar cliente" Entonces se visualiza un mensaje de éxito. Dado el ingreso de datos completos y correctos del cliente Cuando presione el botón "Registrar cliente" Entonces se visualiza un mensaje de error cliente ya está registrado. Dado el ingreso de datos incompletos y/o incorrectos Cuando presione el botón "Registrar cliente" Entonces se visualiza un mensaje de error.	

Historia de Usuario	
Número: 5	Usuario: Gerente
Nombre historia: Registro de producto (Aplicación web y aplicación móvil)	
Prioridad en negocio: 80	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Robert Andrade	
Descripción: Como gerente Quiero ingresar datos completos y correctos de productos Para ofrecer mi catálogo de productos.	
Escenario de prueba: Dado el ingreso de datos completos y correctos del producto Cuando se presione el botón "Registrar producto" Entonces se visualiza un mensaje de éxito. Dado el ingreso de datos completos y correctos del producto Cuando se presione el botón "Registrar producto" Entonces se visualiza un mensaje de error producto ya está registrado. Dado el ingreso de datos incompletos y/o incorrectos del producto Cuando se presione el botón "Registrar producto" Entonces se visualiza un mensaje de error.	

Historia de Usuario	
Número: 6	Usuario: Gerente
Nombre historia: Login (Aplicación web y aplicación móvil)	
Prioridad en negocio: 75	Riesgo en desarrollo: MEDIO
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Robert Andrade	
Descripción: Como gerente Quiero que se ingrese credenciales Para garantizar la seguridad y acceso solo a personal autorizado.	
Escenario de prueba: Dado el ingreso de credenciales correctas Cuando presione el botón "Ingresar" Entonces se redirige a la página principal de la aplicación. Dado el ingreso de credenciales incorrectas Cuando presione el botón de "Ingresar" Entonces se visualiza un mensaje de error. Dado el no ingreso de credenciales Cuando presione el botón de "Ingresar" Entonces se visualiza un mensaje de solicitud de ingreso de credenciales.	
Historia de Usuario	
Número: 7	Usuario: Gerente
Nombre historia: Visualizar pedidos (Aplicación móvil)	
Prioridad en negocio: 70	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Anthonny Sacheri	
Descripción: Como gerente Quiero visualizar el listado de pedidos Para conocer la carga de trabajo y los detalles de cada pedido.	
Escenario de prueba: Dado la visualización del listado de pedidos Cuando presione el icono "Mostrar" Entonces se verifica la información detallada del pedido seleccionado.	
Historia de Usuario	
Número: 8	Usuario: Gerente
Nombre historia: Asignación de pedido a conductor (Aplicación móvil)	
Prioridad en negocio: 65	Riesgo en desarrollo: MEDIO
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Anthonny Sacheri	
Descripción: Como gerente Quiero visualizar los pedidos pendientes Para asignar a los conductores.	
Escenario de prueba: Dado la visualización de pedidos pendientes Cuando presione el botón "Asignar pedidos a conductores" Entonces se muestra un mensaje de éxito.	
Historia de Usuario	
Número: 9	Usuario: Conductor
Nombre historia: Ruta óptima (Aplicación móvil)	
Prioridad en negocio: 60	Riesgo en desarrollo: ALTO
Puntos estimados: 8	Iteración asignada: 2
Programador responsable: Anthonny Sacheri	
Descripción: Como conductor Quiero visualizar los pedidos Para realizar la entrega.	
Escenario de prueba: Dada la visualización de pedidos Cuando presione el botón "Gestión pedidos" Entonces se muestra la información ordenada.	

Historia de Usuario	
Número: 10	Usuario: Conductor
Nombre historia: Visualizar pedidos en el mapa (Aplicación móvil)	
Prioridad en negocio: 55	Riesgo en desarrollo: BAJO
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Anthonny Sacheri	
Descripción: Como conductor Quiero visualizar los pedidos en el mapa Para visualizar la información del cliente y el pedido a entregar.	
Escenario de prueba: Dado la visualización de los pedidos en el mapa Cuando presione un marcador Entonces se muestra la información del cliente y su pedido.	
Historia de Usuario	
Número: 11	Usuario: Gerente, Cliente
Nombre historia: Historial de pedidos (Aplicación web y aplicación móvil)	
Prioridad en negocio: 50	Riesgo en desarrollo: BAJO
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Anthonny Sacheri	
Descripción: Como cliente Quiero visualizar mis pedidos realizados Para tener un registro y control interno.	
Escenario de prueba: Dado la visualización de mis pedidos realizados Cuando presione un pedido de la lista Entonces se abre una nueva ventana mostrando los detalles del pedido.	
Historia de Usuario	
Número: 12	Usuario: Conductor
Nombre historia: Registro de entrega del pedido (Aplicación móvil)	
Prioridad en negocio: 45	Riesgo en desarrollo: BAJO
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Robert Andrade	
Descripción: Como conductor Quiero un registro de entrega del pedido Para evidenciar mi cumplimiento laboral y facilitar el seguimiento de los pedidos entregados.	
Escenario de prueba: Dado el registro de entrega del pedido Cuando se presione el botón de "Pedido entregado" Entonces se muestra un mensaje de pedido entregado. Dado el registro de entrega del pedido Cuando se presione el botón de "Pedido no entregado" Entonces se muestra un mensaje de pedido no entregado.	
Historia de Usuario	
Número: 13	Usuario: Gerente
Nombre historia: Visualizar perfil (Aplicación web y aplicación móvil)	
Prioridad en negocio: 40	Riesgo en desarrollo: BAJO
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Robert Andrade	
Descripción: Como gerente Quiero visualizar la información de los perfiles de usuarios Para corroborar que la información este actualizada y sea correcta.	
Escenario de prueba: Dado la visualización la información de los perfiles de usuarios Cuando presione el icono de "Mostrar" Entonces se redirige en una nueva página la información de dicho usuario.	

Historia de Usuario

Número: 14	Usuario: Gerente
Nombre historia: Editar perfil (Aplicación web y aplicación móvil)	
Prioridad en negocio: 35	Riesgo en desarrollo: BAJO
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Robert Andrade	
Descripción: Como gerente Quiero modificar la información del perfil de usuario Para reemplazar la información desactualizada.	
Escenario de prueba: Dado la modificación de la información del perfil de usuario Cuando presione el botón de "Actualizar" Entonces se un mensaje de éxito.	

Anexo 6: Pruebas de aceptación




CASO DE PRUEBA 01	Fecha: 02/12/2024				
Nombre caso de prueba: Registro de pedido (Aplicación móvil) Sprint: 1					
Módulo/sección a evaluar: Registro de pedido (Aplicación móvil) Historia de usuario asociada: 1					
Técnica de prueba: Caja Negra <input checked="" type="checkbox"/> Caja Blanca <input type="checkbox"/> Tipo: Prueba de Aceptación					
Descripción: Como cliente Quiero ingresar los detalles completos de mi pedido Para que me hagan llegar mi pedido.					
Escenario de prueba: Dado el ingreso de los detalles completos del pedido Cuando presione el botón "Registrar pedido" Entonces se visualiza un mensaje de éxito. Dado el ingreso de los detalles completos del pedido Cuando presione el botón "Registrar pedido" Entonces se visualiza un mensaje de error pedido ya está en proceso. Dado el ingreso de los detalles incompletos del pedido Cuando presione el botón "Registrar pedido" Entonces se visualiza un mensaje de error.					
Precondiciones					
<ul style="list-style-type: none"> • Tener conexión estable a internet • Estar autenticado en la aplicación 					
Pasos y condiciones de ejecución					
<ul style="list-style-type: none"> • Dirigirse a la pantalla principal "Inicio principal" • Pulsar el botón "Nuevo pedido" • Agregar indicaciones • Seleccionar el método de pago • Seleccionar el producto y la cantidad • Pulsar en el botón "Registrar pedido" 					
Resultado esperado					
<ul style="list-style-type: none"> • Mensaje de éxito al registrar • Se presentan mensajes de error, según corresponda 					
Estado de prueba	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Éxito</th> <th style="width: 50%;">Falló</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Si</td> <td style="text-align: center;">No</td> </tr> </tbody> </table>	Éxito	Falló	Si	No
Éxito	Falló				
Si	No				
Errores asociados:					
 Sr. Ulises Segura Alberto Patricio PRODUCT OWNER					

Código postal: 230103
Santo Domingo - Ecuador / www.puce.edu.ec






CASO DE PRUEBA 02	Fecha: 02/12/2024				
Nombre caso de prueba: Registro de conductor (Aplicación web) Sprint: 1					
Módulo/sección a evaluar: Registro de conductor (Aplicación web) Historia de usuario asociada: 2					
Técnica de prueba: Caja Negra <input checked="" type="checkbox"/> Caja Blanca <input type="checkbox"/> Tipo: Prueba de Aceptación					
Descripción: Como gerente Quiero ingresar datos completos y correctos del conductor Para facilitar la comunicación y coordinación durante las entregas.					
Escenario de prueba: Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar conductor" Entonces se visualiza un mensaje de éxito. Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar conductor" Entonces se visualiza un mensaje de error conductor ya está registrado. Dado el ingreso de datos incompletos y/o incorrectos Cuando se presione el botón "Registrar conductor" Entonces se visualiza un mensaje de error.					
Precondiciones					
<ul style="list-style-type: none"> • Tener conexión estable a internet • Contar con un navegador web • Estar autenticado en la aplicación 					
Pasos y condiciones de ejecución					
<ul style="list-style-type: none"> • Dirigirse al menú lateral en el apartado conductores • Presionar submenú "Creación de conductores" • Ingresar el número de cédula y presionar el icono de la lupa • Ingresar teléfono • Ingresar las contraseñas • Pulsar en el botón "Registrar conductor" 					
Resultado esperado					
<ul style="list-style-type: none"> • Mensaje de éxito al registrar • Se presentan mensajes de error, según corresponda 					
Estado de prueba	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Éxito</th> <th style="width: 50%;">Falló</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Si</td> <td style="text-align: center;">No</td> </tr> </tbody> </table>	Éxito	Falló	Si	No
Éxito	Falló				
Si	No				
Errores asociados:					
 Sr. Ulises Segura Alberto Patricio PRODUCT OWNER					

Código postal: 230103
Santo Domingo - Ecuador / www.puce.edu.ec





Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 03 Fecha: 02/12/2024

Nombre caso de prueba: Registro de vehículo (Aplicación web) **Sprint:** 1

Módulo/sección a evaluar: Registro de vehículo (Aplicación web) **Historia de usuario asociada:** 3

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
Como gerente Quiero ingresar datos completos y correctos del vehículo Para asegurar una asignación eficiente de pedidos.

Escenario de prueba:
Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar vehículo" Entonces se visualiza un mensaje de éxito.
Dado el ingreso de datos completos y correctos Cuando se presione el botón "Registrar vehículo" Entonces se visualiza un mensaje de error vehículo ya está registrado.
Dado el ingreso de datos incompletos y/o incorrectos Cuando se presione el botón "Registrar vehículo" Entonces se visualiza un mensaje de error.

Precondiciones

- Tener conexión estable a internet.
- Contar con un navegador web.
- Estar autenticado en la aplicación.

Pasos y condiciones de ejecución

- Dirigirse al menú lateral en el apartado vehículos
- Presionar submenú "Creación de vehículo"
- Ingresar la placa y presionar el icono de la lupa
- Asignar un conductor
- Ingresar una observación en caso de ser necesario
- Pulsar en el botón "Registrar vehículo"

Resultado esperado

- Mensaje de éxito al registrar
- Se presentan mensajes de error, según corresponda

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:



Sr. Ulloa Segura Alberto Patricio
PRODUCT OWNER

Código postal: 330203
Santo Domingo - Ecuador / www.puce.edu.ec

[f](#) [t](#) [@](#) [in](#) [d](#)



Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 05 Fecha: 02/12/2024

Nombre caso de prueba: Registro de producto (Aplicación web y móvil) **Sprint:** 1

Módulo/sección a evaluar: Registro de producto (Aplicación web y móvil) **Historia de usuario asociada:** 5

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
Como gerente Quiero ingresar datos completos y correctos de productos Para ofrecer mi catálogo de productos.

Escenario de prueba:
Dado el ingreso de datos completos y correctos del producto Cuando se presione el botón "Registrar producto" Entonces se visualiza un mensaje de éxito.
Dado el ingreso de datos completos y correctos del producto Cuando se presione el botón "Registrar producto" Entonces se visualiza un mensaje de error producto ya está registrado.
Dado el ingreso de datos incompletos y/o incorrectos del producto Cuando se presione el botón "Registrar producto" Entonces se visualiza un mensaje de error.

Precondiciones

- Tener conexión estable a internet
- Contar con un navegador web
- Estar autenticado en la aplicación

Pasos y condiciones de ejecución

- Dirigirse al menú lateral en el apartado producto
- Presionar submenú "Creación de producto"
- Ingresar el nombre
- Ingresar la descripción
- Ingresar el precio
- Pulsar en el botón "Registrar producto"

Resultado esperado

- Mensaje de éxito al registrar
- Se presentan mensajes de error, según corresponda

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:



Sr. Ulloa Segura Alberto Patricio
PRODUCT OWNER

Código postal: 330203
Santo Domingo - Ecuador / www.puce.edu.ec

[f](#) [t](#) [@](#) [in](#) [d](#)



Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 04 Fecha: 02/12/2024

Nombre caso de prueba: Registro de cliente (Aplicación web y móvil) **Sprint:** 1

Módulo/sección a evaluar: Registro de cliente (Aplicación web y móvil) **Historia de usuario asociada:** 4

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
Como gerente Quiero ingresar datos completos y correctos de clientes Para tener los registros actualizados.

Escenario de prueba:
Dado el ingreso de datos completos y correctos del cliente Cuando presiona el botón "Registrar cliente" Entonces se visualiza un mensaje de éxito.
Dado el ingreso de datos completos y correctos del cliente Cuando presiona el botón "Registrar cliente" Entonces se visualiza un mensaje de error cliente ya está registrado.
Dado el ingreso de datos incompletos y/o incorrectos Cuando presione el botón "Registrar cliente" Entonces se visualiza un mensaje de error.

Precondiciones

- Tener conexión estable a internet
- Contar con un navegador web
- Estar autenticado en la aplicación

Pasos y condiciones de ejecución

- Dirigirse al menú lateral en el apartado clientes
- Presionar submenú "Creación de clientes"
- Ingresar el número de cédula y presionar el icono de la lupa
- Ingresar teléfono
- Ingresar referencia del domicilio
- Seleccionar la ubicación en el mapa
- Pulsar en el botón "Registrar cliente"

Resultado esperado

- Mensaje de éxito al registrar
- Se presentan mensajes de error, según corresponda

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:



Sr. Ulloa Segura Alberto Patricio
PRODUCT OWNER

Código postal: 330203
Santo Domingo - Ecuador / www.puce.edu.ec

[f](#) [t](#) [@](#) [in](#) [d](#)



Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 06 Fecha: 02/12/2024

Nombre caso de prueba: Login (Aplicación web y móvil) **Sprint:** 1

Módulo/sección a evaluar: Login (Aplicación web y móvil) **Historia de usuario asociada:** 6

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
Como gerente Quiero que se ingrese credenciales Para garantizar la seguridad y acceso solo a personal autorizado.

Escenario de prueba:
Dado el ingreso de credenciales correctas Cuando presione el botón "Ingresar" Entonces se mostrará la página principal de la aplicación.
Dado el ingreso de credenciales incorrectas Cuando presione el botón de "Ingresar" Entonces se visualiza un mensaje de error.
Dado el no ingreso de credenciales Cuando presione el botón de "Ingresar" Entonces se visualiza un mensaje de solicitud de ingreso de credenciales.

Precondiciones

- Tener conexión estable a internet
- Contar con un navegador web

Pasos y condiciones de ejecución

- Ingresar usuario
- Ingresar contraseña
- Presionar el botón "Ingresar"

Resultado esperado

- Mensaje de éxito y se dirige a la pantalla principal
- Se presentan mensajes de error, según corresponda

Estado de prueba	Éxito	Falló
	Si	No

Errores asociados:



Sr. Ulloa Segura Alberto Patricio
PRODUCT OWNER

Código postal: 330203
Santo Domingo - Ecuador / www.puce.edu.ec

[f](#) [t](#) [@](#) [in](#) [d](#)

Pontificia Universidad Católica del Ecuador SANTO DOMINGO
Serbia sus testigos

CASO DE PRUEBA 07 Fecha: 02/12/2024

Nombre caso de prueba: Visualizar pedidos (Aplicación web) **Sprint:** 1

Módulo/sección a evaluar: Visualizar pedidos (Aplicación web) **Historia de usuario asociada:** 7

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
 Como gerente Quiero visualizar el listado de pedidos Para conocer la carga de trabajo y los detalles de cada pedido.

Escenario de prueba:
 Dado la visualización del listado de pedidos Cuando presione el icono "Mostrar" Entonces se mostrará la información detallada del pedido seleccionado.

Precondiciones

- Tener conexión estable a internet
- Contar con un navegador web
- Estar autenticado en la aplicación

Pasos y condiciones de ejecución

- Dirigirse al menú lateral y presionar "Pedidos"
- Pulsar en el icono del ojo

Resultado esperado

- Se redirige a una pantalla con el detalle del pedido

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No


 Sr. Ulises Segura Alberto Patricio
 PRODUCT OWNER

Código postal: 230103
 Santo Domingo - Ecuador / www.puce.edu.ec
 f t i n d

Pontificia Universidad Católica del Ecuador SANTO DOMINGO
Serbia sus testigos

CASO DE PRUEBA 08 Fecha: 02/12/2024

Nombre caso de prueba: Asignación de pedido a conductor (Aplicación móvil) **Sprint:** 2

Módulo/sección a evaluar: Asignación de pedido a conductor (Aplicación móvil) **Historia de usuario asociada:** 8

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
 Como gerente Quiero visualizar los pedidos pendientes Para asignar a los conductores.

Escenario de prueba:
 Dado la visualización de pedidos pendientes Cuando presione el botón "Asignar pedidos a conductores" Entonces se mostrará un mensaje de éxito.

Precondiciones

- Tener conexión estable a internet
- Estar autenticado en la aplicación
- Contar con al menos un pedido pendiente

Pasos y condiciones de ejecución

- Dirigirse al acceso rápido de la pantalla principal y presionar "Gestión pedidos"
- Presionar en el botón "Asignar pedidos a conductores"

Resultado esperado

- Mensaje de éxito en la asignación de pedido a conductores

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No


 Sr. Ulises Segura Alberto Patricio
 PRODUCT OWNER

Código postal: 230103
 Santo Domingo - Ecuador / www.puce.edu.ec
 f t i n d

Pontificia Universidad Católica del Ecuador SANTO DOMINGO
Serbia sus testigos

CASO DE PRUEBA 08 Fecha: 02/12/2024

Nombre caso de prueba: Asignación de pedido a conductor (Aplicación móvil) **Sprint:** 2

Módulo/sección a evaluar: Asignación de pedido a conductor (Aplicación móvil) **Historia de usuario asociada:** 8

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
 Como gerente Quiero visualizar los pedidos pendientes Para asignar a los conductores.

Escenario de prueba:
 Dado la visualización de pedidos pendientes Cuando presione el botón "Asignar pedidos a conductores" Entonces se mostrará un mensaje de éxito.

Precondiciones

- Tener conexión estable a internet
- Estar autenticado en la aplicación
- Contar con al menos un pedido pendiente

Pasos y condiciones de ejecución

- Dirigirse al acceso rápido de la pantalla principal y presionar "Gestión pedidos"
- Presionar en el botón "Asignar pedidos a conductores"

Resultado esperado

- Mensaje de éxito en la asignación de pedido a conductores

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No


 Sr. Ulises Segura Alberto Patricio
 PRODUCT OWNER

Código postal: 230103
 Santo Domingo - Ecuador / www.puce.edu.ec
 f t i n d

Pontificia Universidad Católica del Ecuador SANTO DOMINGO
Serbia sus testigos

CASO DE PRUEBA 09 Fecha: 02/12/2024

Nombre caso de prueba: Ruta óptima (Aplicación móvil) **Sprint:** 2

Módulo/sección a evaluar: Ruta óptima (Aplicación móvil) **Historia de usuario asociada:** 9

Técnica de prueba: Caja Negra Caja Blanca **Tipo:** Prueba de Aceptación

Descripción:
 Como conductor Quiero tener pedidos asignados Para generar la ruta de distribución.

Escenario de prueba:
 Dado la asignación de pedidos al conductor Cuando presione el botón "Gestión pedidos" Entonces se abre una nueva pantalla mostrando los pedidos asignados ya ordenados, generando una ruta de distribución óptima.

Precondiciones

- Tener conexión estable a internet
- Estar autenticado en la aplicación
- Contar con al menos un pedido asignado

Pasos y condiciones de ejecución

- Dirigirse al acceso rápido de la pantalla principal y presionar "Gestión pedidos"
- Españar el cálculo y la generación de la ruta del conductor

Resultado esperado

- Lista de pedidos ordenados con respecto a la ubicación del conductor

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No


 Sr. Ulises Segura Alberto Patricio
 PRODUCT OWNER

Código postal: 230103
 Santo Domingo - Ecuador / www.puce.edu.ec
 f t i n d



Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 10 Fecha: 02/12/2024

Nombre caso de prueba: Visualizar pedidos en el mapa (Aplicación móvil) **Sprint:** 2

Módulo/sección a evaluar: Visualizar pedidos en el mapa (Aplicación móvil) **Historia de usuario asociada:** 10

Técnica de prueba: Caja Negra ☒ Caja Blanca ☐ **Tipo:** Prueba de Aceptación

Descripción:
Como conductor Quiero visualizar los pedidos en el mapa Para conocer la información del cliente y el pedido a entregar.

Escenario de prueba:
Dado la visualización de los pedidos en el mapa Cuando presione un marcador Entonces se mostrará la información del cliente y su pedido.

Precondiciones

- Tener conexión estable a internet
- Estar autenticado en la aplicación
- Contar con al menos un pedido asignado

Pasos y condiciones de ejecución

- Dirigirse al acceso rápido de la pantalla principal y presionar "Gestión pedidos"
- Presionar en el botón "Visualizar ruta"

Resultado esperado

- Se redirige a una pantalla con el mapa mostrando los pedidos como marcadores

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No



Sr. Ulises Segura Alberto Patricio
PRODUCT OWNER

Código postal: 030203
Santo Domingo - Ecuador / www.pucead.edu.ec

f t in d



Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 12 Fecha: 02/12/2024

Nombre caso de prueba: Registro de entrega del pedido (Aplicación móvil) **Sprint:** 2

Módulo/sección a evaluar: Registro de entrega del pedido (Aplicación móvil) **Historia de usuario asociada:** 12

Técnica de prueba: Caja Negra ☒ Caja Blanca ☐ **Tipo:** Prueba de Aceptación

Descripción:
Como conductor Quiero un registro de entrega del pedido Para evidenciar mi cumplimiento laboral y facilitar el seguimiento de los pedidos entregados.

Escenario de prueba:
Dado el registro de entrega del pedido Cuando se presione el botón de "Pedido entregado" Entonces se mostrará un mensaje de pedido entregado.
Dado el registro de entrega del pedido Cuando se presione el botón de "Pedido no entregado" Entonces se mostrará un mensaje de pedido no entregado.

Precondiciones

- Tener conexión estable a internet
- Estar autenticado en la aplicación

Pasos y condiciones de ejecución

- Dirigirse al acceso rápido de la pantalla principal y presionar "Gestión pedidos"
- Presionar en el botón "Visualizar ruta"
- Presionar un marcador (pedido) en el mapa
- Presionar el botón "Pedido entregado" o "Pedido no entregado" según corresponda

Resultado esperado

- Mensaje de éxito "Pedido entregado" o "Pedido no entregado" según corresponda

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No



Sr. Ulises Segura Alberto Patricio
PRODUCT OWNER

Código postal: 030203
Santo Domingo - Ecuador / www.pucead.edu.ec

f t in d



Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 11 Fecha: 02/12/2024

Nombre caso de prueba: Historial de pedidos (Aplicación móvil) **Sprint:** 2

Módulo/sección a evaluar: Historial de pedidos (Aplicación móvil) **Historia de usuario asociada:** 11

Técnica de prueba: Caja Negra ☒ Caja Blanca ☐ **Tipo:** Prueba de Aceptación

Descripción:
Como cliente Quiero visualizar mis pedidos realizados Para tener un registro y control interno.

Escenario de prueba:
Dado la visualización de mis pedidos realizados Cuando presione un pedido de la lista Entonces se abrirá una nueva pantalla mostrando los detalles del pedido.

Precondiciones

- Tener conexión estable a internet
- Estar autenticado en la aplicación
- Contar con al menos un pedido solicitado

Pasos y condiciones de ejecución

- Dirigirse al acceso rápido de la pantalla principal y presionar "Historial de pedidos"
- Presionar en un pedido de la lista

Resultado esperado

- Se redirige a una pantalla mostrando la información y detalles del pedido

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No



Sr. Ulises Segura Alberto Patricio
PRODUCT OWNER

Código postal: 030203
Santo Domingo - Ecuador / www.pucead.edu.ec

f t in d



Pontificia Universidad Católica del Ecuador
Santo Domingo

CASO DE PRUEBA 13 Fecha: 02/12/2024

Nombre caso de prueba: Visualizar perfil (Aplicación web) **Sprint:** 2

Módulo/sección a evaluar: Visualizar perfil (Aplicación web) **Historia de usuario asociada:** 13

Técnica de prueba: Caja Negra ☒ Caja Blanca ☐ **Tipo:** Prueba de Aceptación

Descripción:
Como gerente Quiero visualizar la información de los perfiles de usuarios Para corroborar que la información este actualizada y sea correcta.

Escenario de prueba:
Dado la visualización la información de los perfiles de usuarios Cuando presione el ícono de "Mostrar" Entonces se presentará en una nueva página la información de dicho usuario.

Precondiciones

- Tener conexión estable a internet
- Contar con un navegador web
- Estar autenticado en la aplicación

Pasos y condiciones de ejecución

- Dirigirse al menú lateral en el apartado clientes o conductores según corresponda
- Presionar submenú "Listado de clientes" o "Listado de conductores"
- Presionar ícono del ojo (mostrar)

Resultado esperado

- Se redirige a una pantalla mostrando la información de detalle del usuario

Estado de prueba	Éxito	Fallo
Errores asociados:	Si	No



Sr. Ulises Segura Alberto Patricio
PRODUCT OWNER

Código postal: 030203
Santo Domingo - Ecuador / www.pucead.edu.ec

f t in d

Pontificia Universidad Católica del Ecuador		SANTO DOMINGO	
Sereña mis testigos			
CASO DE PRUEBA 14		Fecha: 02/12/2020	
Nombre caso de prueba: Editar perfil (Aplicación web)	Sprint: 2		
Módulo/sección a evaluar: Editar perfil (Aplicación web)	Historia de usuario asociada: 14		
Técnica de prueba: Caja Negra <input checked="" type="checkbox"/> Caja Blanca <input type="checkbox"/>	Tipo: Prueba de Aceptación		
Descripción:			
Como gerente Quiero modificar la información del perfil de usuario Para reemplazar la información desactualizada			
Escenario de prueba:			
Dado la modificación de la información del perfil de usuario Cuando presione el botón de "Actualiza Entonces se un mensaje de éxito.			
Precondiciones			
<ul style="list-style-type: none"> Tener conexión estable a Internet Contar con un navegador web Estar autenticado en la aplicación 			
Pasos y condiciones de ejecución			
<ul style="list-style-type: none"> Dirigirse al menú lateral en el apartado clientes o conductores según corresponda Presionar submenú "Listado de clientes" o "Listado de conductores" Presionar ícono del lápiz (editar) Realizar los cambios Presionar el botón de "Actualizar" 			
Resultado esperado			
<ul style="list-style-type: none"> Se redirige a una pantalla mostrando la información detallada del usuario listo para actualizar 			
Estado de prueba	Éxito		Falló
	Si		No
Errores asociados:			




Sr. Ulloa Segura Alberto Patricio
PRODUCT OWNER

Código postal: 410105
Santo Domingo - Ecuador / www.puce.edu.ec

f t i n d

Anexo 7: Certificado de base de clientes recurrentes – Still Water

CERTIFICADO	
<p>La empresa Still Water, domiciliada en el cantón Santo Domingo, por medio del presente documento deja constancia de que actualmente cuenta con un total de 84 clientes recurrentes que realizan pedidos frecuentes de agua potable a domicilio, consolidándose como parte esencial de nuestra actividad comercial.</p>	
<p>Este certificado se expide a solicitud de parte interesada para los fines que estime convenientes.</p>	
<p>En Santo Domingo, a los 05 días del mes de enero del año 2025.</p>	
<p>Atentamente,</p>	
<div style="text-align: center;">  </div>	
<p>Sr. Alberto Patricio Ulloa Segura Gerente de Still Water</p>	

Anexo 8: Evidencia por video de implementación del instrumento para la recolección de datos

[Fragmento de la Entrevista al Gerente de Still Water.MOV](#)

Anexo 9: Evidencia por video de la implementación de la aplicación web y la aplicación móvil

[Implementación del sistema](#)

Anexo 10: Datos de registros generados

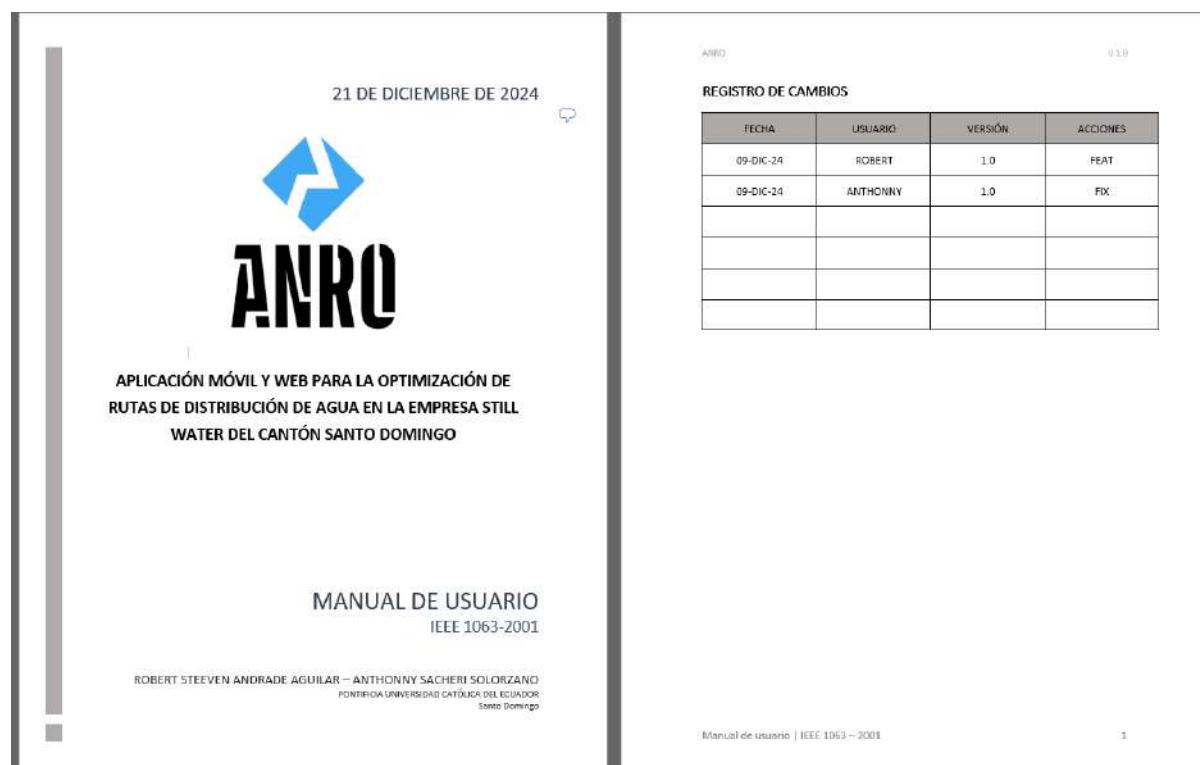
Encuesta a clientes: <https://drive.google.com/drive/folders/1T3D8zol-eltCT1EeZx4fPEZZW50GgYgB?usp=sharing>

Encuesta a empleados:

[https://drive.google.com/drive/folders/1_Uth4DtOIkDOAACwfiWOn-xBkIh2xn2P?usp=drive link](https://drive.google.com/drive/folders/1_Uth4DtOIkDOAACwfiWOn-xBkIh2xn2P?usp=drive_link)

Anexo 11: Manual de usuario

Enlace del manual completo: [Manual de Usuario.pdf](#)



ANRO	V 1.0	ANRO	V 1.0
Tabla de Contenidos			
1	INTRODUCCIÓN 3	3.3	Uso de la aplicación 31
2	CONCEPTO DE LAS OPERACIONES 3	3.4	Acceso como cliente 31
3	PROCEDIMIENTOS 4	3.4.1	Inicio de Sesión 31
3.1	Aplicación web 4	3.4.2	Registro 33
3.1.1	Inicio de sesión 4	3.4.3	Pantalla principal 35
3.1.2	Inicio de sesión con usuario y contraseña 4	3.4.4	Administrar mi cuenta 36
3.1.3	Interfaz principal 5	3.4.5	Historial de pedidos 40
3.1.4	Creación de clientes 6	3.4.6	Cerrar sesión 41
3.1.5	Listado de clientes 7	3.5	Acceso como gerente 42
3.1.6	Editar información de un cliente 8	3.5.1	Inicio de Sesión 42
3.1.7	Activar e inactivar un cliente 9	3.5.2	Pantalla principal 44
3.1.8	Creación de conductores 11	3.5.3	Pedidos 45
3.1.9	Listado de conductores 12	3.5.4	Vehículos 50
3.1.10	Editar información de un conductor 13	3.5.5	Productos 52
3.1.11	Activar e inactivar un conductor 14	3.5.6	Clientes 56
3.1.12	Creación de vehículos 17	3.5.7	Conductores 62
3.1.13	Listado de vehículos 17	3.5.8	Gestionar los pedidos 64
3.1.14	Editar información de un vehículo 19	3.5.9	Cerrar sesión 68
3.1.15	Activar e inactivar un vehículo 20	3.6	Acceso como conductor 69
3.1.16	Creación de productos 22	3.6.1	Inicio de Sesión 69
3.1.17	Listado de productos 22	3.6.2	Pantalla principal 70
3.1.18	Editar información de un producto 24	3.6.3	Pedidos 71
3.1.19	Activar e inactivar un producto 24	3.6.4	Vehículos 73
3.1.20	Listado de pedidos 26	3.6.5	Productos 74
3.2	Aplicación móvil 27	3.6.6	Clientes 75
3.2.1	Fuente de descarga 27	3.6.7	Gestionar los pedidos 80
3.2.2	Instalación 28	3.6.8	Cerrar sesión 84
		4	MENSAJES DE ERROR Y RESOLUCIÓN DE PROBLEMAS 85
Manual de usuario IEEE 1063 – 2001	2	Manual de usuario IEEE 1063 – 2001	3

ANRO	V 1.0	ANRO	V 1.0
5	GLOSARIO 86	1	INTRODUCCIÓN
6	REFERENCIAS 87		En un mundo donde la eficiencia y la rapidez son fundamentales para el éxito empresarial, la gestión optimizada de rutas se convierte en una herramienta clave para las empresas de distribución. ANRO es una plataforma compuesta por una aplicación móvil y una aplicación web, diseñada para facilitar y mejorar el proceso de planificación de rutas de entrega. Su implementación en Still Water permite reducir tiempos, costos y errores en las operaciones logísticas.
7	CARACTERÍSTICAS DE NAVEGACIÓN 87		Este manual proporciona las instrucciones necesarias para el uso eficiente de ANRO, tanto en su versión móvil como en su aplicación web. Su objetivo es garantizar que los usuarios comprendan todas las funcionalidades de la plataforma y puedan aprovechar al máximo sus capacidades para optimizar la gestión de rutas de distribución.
			Se incluyen los requerimientos técnicos, pasos de instalación, instrucciones de configuración y ejemplos prácticos de uso, asegurando una implementación exitosa en las operaciones diarias de la empresa. Con ello, se busca proporcionar una herramienta eficiente y accesible que garantice el uso correcto de la aplicación, maximizando sus beneficios.
		2	CONCEPTO DE LAS OPERACIONES
			Los requerimientos mínimos para que la aplicación móvil ANRO y la aplicación web funcionen correctamente son los siguientes:
			Requerimientos del dispositivo
			<ul style="list-style-type: none"> • Teléfono o computadora inteligente • Poseer un navegador web
			Requerimientos de conectividad
			<ul style="list-style-type: none"> • Conexión a Internet: Se recomienda una conexión estable (Wi-Fi o datos móviles) para funcionalidades que dependen de actualizaciones de rutas en tiempo real. • GPS activado: La aplicación requiere acceso a los servicios de ubicación para generar y optimizar rutas de manera precisa.
			Permisos necesarios
			<ul style="list-style-type: none"> • Ubicación: Para el cálculo y monitoreo de las rutas.
Manual de usuario IEEE 1063 – 2001	4	Manual de usuario IEEE 1063 – 2001	3

ANRO V1.0


- **Almacenamiento:** Para guardar configuraciones locales y registros temporales.

3 PROCEDIMIENTOS

3.1 Aplicación web

3.1.1 Inicio de sesión

Abrimos nuestro navegador de preferencia e ingresamos a la siguiente URL:
<https://100.42.186.185/stillwater/login/>




3.1.2 Inicio de sesión con usuario y contraseña

Procedemos a iniciar sesión con el número de cédula y contraseña anteriormente creada por el administrador.


Manual de usuario | IEEE 1063 – 2001 4

ANRO V1.0



3.1.3 Interfaz principal

Se abrirá la página principal de la aplicación web mostrando información relevante del usuario con el que se ha iniciado sesión, la cantidad de clientes, conductores y vehículos registrados en el sistema.



- Visualizamos las funciones dependiendo el rol (administrador/conductor) en el menú lateral izquierdo y presionamos en cualquiera de ellas según el caso requerido.

Manual de usuario | IEEE 1063 – 2001 5

ANRO V1.0

3.1.4 Creación de clientes

Esta página permite registrar nuevos clientes en el sistema. Asegúrese de completar todos los campos obligatorios marcados con un asterisco (*) y de verificar la información antes de hacer clic en "Registrar cliente".

- **Cédula*:** Campo para ingresar el número de cédula. Incluye un icono de lupa al lado, que al presionarlo busca automáticamente los datos del cliente por el número de cédula.
- **Nombres* y Apellidos*:** Campo para ingresar el nombre completo del cliente.
- **Teléfono*:** Campo para ingresar el número de teléfono del cliente.
- **Referencia del domicilio:** Espacio para describir una referencia del lugar donde vive el cliente.
- **Mapa interactivo:** Permite seleccionar la ubicación del cliente de manera visual. Incluye un botón para usar la ubicación actual.



- En la parte inferior de la página, hay tres botones principales:
 - **Usar mi ubicación actual:** Captura la ubicación del dispositivo.
 - **Cancelar:** Descarta el registro y redirige a la página de listado de clientes.
 - **Registrar cliente:** Guarda la información ingresada.

Manual de usuario | IEEE 1063 – 2001 6

ANRO V1.0

3.1.5 Listado de clientes

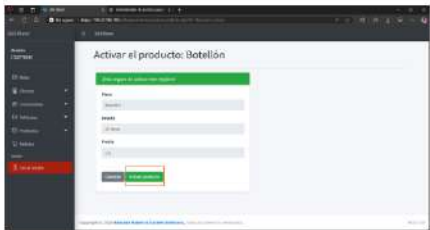
En esta página se visualiza un registro tabular de los clientes previamente ingresados, con las siguientes características:

- **Botón "+ Nuevo cliente":**
 - Ubicado en la parte superior derecha, permite registrar un nuevo cliente.
- **Buscador:**
 - Campo para filtrar clientes ingresando texto (por ejemplo, cédula, nombre o número de teléfono).
- **Tabla de clientes:**
 - Presente los datos organizados en columnas:
 - **No.:** Número secuencial del cliente.
 - **Cédula:** Número de identificación del cliente.
 - **Nombre:** Nombres y apellidos completos del cliente.
 - **Dirección:** Indicada mediante un icono de ubicación azul.
 - **Teléfono:** Número de contacto, con un enlace para abrir WhatsApp (icono verde de WhatsApp).
 - **Acciones:** Contiene tres botones por cada registro:
 - ➔ **Ver (icono de ojo):** Permite visualizar los detalles del cliente.
 - ➔ **Editar (icono de lápiz):** Para modificar los datos del cliente.
 - ➔ **Eliminar (icono de bote de basura):** Inactivar al usuario.
- **Paginación:**
 - Al pie de la tabla, permite navegar entre páginas si hay más registros.

Manual de usuario | IEEE 1063 – 2001 7

ANFO
V.1.0

- Página de confirmación para activar el producto, se muestra la información registrada y al momento de presionar en el botón de "Activar producto" se visualiza un cuadro emergente en el centro de la pantalla con un mensaje "Producto activado exitosamente"



3.1.20 Listado de pedidos

Esta página permite gestionar y monitorear los pedidos realizados en el sistema. Los colores indican el estado del pedido: azul para pendiente, verde para entregado, rojo para no entregado y amarillo para en proceso. Para ver los detalles de un pedido específico, haga clic en el icono de ojo en la columna de acciones.

- **Buscador:**
 - Campo para filtrar pedidos ingresando texto (por ejemplo, indicaciones, estado, cliente, entre las demás columnas.).
- **Tabla de pedidos:**
 - Incluye columnas como:
 - **Nro:** Número secuencial del pedido.
 - **Indicaciones:** Detalles adicionales sobre el pedido.
 - **Estado:** Presente estados como ENTREGADO (verde), NO ENTREGADO (rojo), EN PROCESO (amarillo) y PENDIENTE (azul).
 - **Fecha pedido:** Fecha y hora de creación del pedido.

Manual de usuario | IEEE 1063 – 2001
26

ANFO
V.1.0
Manual de usuario | IEEE 1063 – 2001
27
ANFO
V.1.0
Manual de usuario | IEEE 1063 – 2001
28
ANFO
V.1.0
Manual de usuario | IEEE 1063 – 2001
29

AVRO V.I.O.



- Una vez completada la instalación, podemos hallar la aplicación dentro de nuestro cajón de aplicaciones.



Manual de usuario | IEEE 1063 – 2001 30

AVRO V.I.O.

3.3 Uso de la aplicación

Al momento de abrir por primera vez la aplicación pedirá el acceso a la ubicación del dispositivo para poder cumplir con el objetivo de acceder a la ubicación en tiempo real para usarla en el momento del registro y al editar la información personal.



3.4 Acceso como cliente

3.4.1 Inicio de Sesión

Una vez otorgado los permisos, en la interfaz principal se presentan dos alternativas para acceder a las funcionalidades.

Manual de usuario | IEEE 1063 – 2001 31

AVRO V.I.O.



3.4.1.1 Inicio con Usuario y contraseña

Si cuenta con un usuario y contraseña ya registrados en la aplicación

- llene los campos respectivos y pulse en el botón "Iniciar".



Manual de usuario | IEEE 1063 – 2001 32

AVRO V.I.O.

3.4.2 Registro

- Presione en el botón "Registrar"





- Ingrese el número de cédula o RUC según corresponda y presione en el ícono de la lupa para obtener los datos en tiempo real.





Manual de usuario | IEEE 1063 – 2001 33

Anexo 12: Manual técnico

21 DE DICIEMBRE DE 2024

ANRO

MANUAL TÉCNICO
IEEE 1063-2001

ROBERT STEEVEN ANDRADE AGUILAR – ANTHONNY SACHERI SOLORZANO
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
Santo Domingo

ANRO V 1.0

REGISTRO DE CAMBIOS

FECHA	USUARIO	VERSION	ACCIONES
09-DIC-24	ROBERT	1.0	FEAT
09-DIC-24	ANTHONNY	1.0	FIX

Manual técnico | IEEE 1063 – 2001 1

Tabla de Contenidos

1	INTRODUCCIÓN.....	3
2	CONCEPTO DE LAS OPERACIONES	3
3	PROCEDIMIENTOS.....	4
3.1	Clonar repositorios de GitHub.....	4
3.1.1	Repositorio backend	4
3.1.2	Repositorio frontend web.....	4
3.1.3	Repositorio frontend móvil.....	5
3.2	Archivos del backend	6
3.2.1	Carpeta Controladores.....	6
3.2.2	Carpeta Config.....	7
3.2.3	Carpeta Models.....	8
3.2.4	Carpeta Routes.....	8
3.3	Archivos del frontend móvil.....	9
3.4	Carpeta lib	10
3.5	Archivos del frontend web	11
3.5.1	Carpeta admin.....	11
3.5.2	Carpeta app.....	12
4	ACCESO A LOS DATOS.....	13
5	GLOSARIO.....	15
6	REFERENCIAS	15

Manual técnico | IEEE 1063 – 2001 5

ANRO V 1.0

1 INTRODUCCIÓN

Este manual técnico proporciona una visión detallada de la estructura de la aplicación web y móvil, basándose en los frameworks y tecnologías utilizadas. Para comenzar, se explicará el proceso de clonación del repositorio oficial del proyecto alojado en GitHub. A partir de ahí, se abordará la organización de los archivos del backend y del frontend en la aplicación web, así como la estructura de archivos y componentes en la aplicación móvil.

Al explorar la distribución de carpetas y su funcionalidad, se describirá cómo cada directorio ha sido estratégicamente estructurado para garantizar un rendimiento eficiente y una óptima experiencia de usuario en ambas plataformas. Este manual permitirá obtener una comprensión clara de la arquitectura del software, facilitando así su mantenimiento, escalabilidad y colaboración en futuras actualizaciones y mejoras.

2 CONCEPTO DE LAS OPERACIONES

Los requerimientos mínimos para que la aplicación móvil ANRO y la aplicación web funcionen correctamente son los siguientes:

Requerimientos del dispositivo

- Teléfono o computadora inteligente
- Poseer un navegador web

Requerimientos de conectividad

- **Conexión a Internet:** Se recomienda una conexión estable (Wi-Fi o datos móviles) para funcionalidades que dependen de actualizaciones de rutas en tiempo real.
- **GPS activado:** La aplicación requiere acceso a los servicios de ubicación para generar y optimizar rutas de manera precisa.

Permisos necesarios

- **Ubicación:** Para el cálculo y monitoreo de las rutas.
- **Almacenamiento:** Para guardar configuraciones locales y registros temporales.

Manual técnico | IEEE 1063 – 2001 6

ANRO

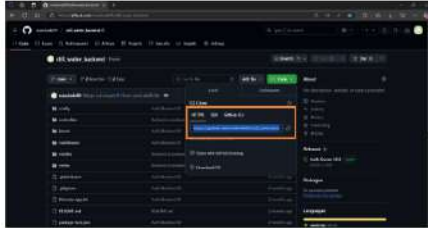
V.1.0

3 PROCEDIMIENTOS

3.1 Clonar repositorios de GitHub

3.1.1 Repositorio backend

Acceda a la página de GitHub donde se encuentra el repositorio del proyecto y copie la URL en formato HTTPS.



Esta URL servirá para descargar una copia local del repositorio utilizando Git.

Para clonar el repositorio, abra la terminal y navegue hasta el directorio donde desea almacenarlo. Una vez ubicado en el directorio correcto, ejecute el siguiente comando:

```
git clone https://github.com/rsandrade99/still_water_backend.git
```

3.1.2 Repositorio frontend web

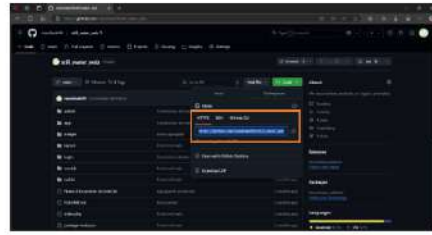
Acceda a la página de GitHub donde se encuentra el repositorio del proyecto y copie la URL en formato HTTPS.

Manual técnico | IEEE 1063 – 2001

7

ANRO

V.1.0



Esta URL servirá para descargar una copia local del repositorio utilizando Git.

Para clonar el repositorio, abra la terminal y navegue hasta el directorio donde desea almacenarlo. Una vez ubicado en el directorio correcto, ejecute el siguiente comando:

```
git clone https://github.com/rsandrade99/still_water_web.git
```

3.1.3 Repositorio frontend móvil

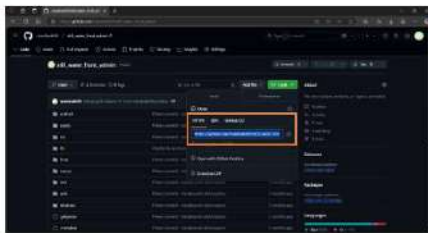
Acceda a la página de GitHub donde se encuentra el repositorio del proyecto y copie la URL en formato HTTPS.

Manual técnico | IEEE 1063 – 2001

8

ANRO

V.1.0



Esta URL servirá para descargar una copia local del repositorio utilizando Git.

Para clonar el repositorio, abra la terminal y navegue hasta el directorio donde desea almacenarlo. Una vez ubicado en el directorio correcto, ejecute el siguiente comando:

```
git clone https://github.com/rsandrade99/still_water_front_admin.git
```

3.2 Archivos del backend

3.2.1 Carpeta Controladores

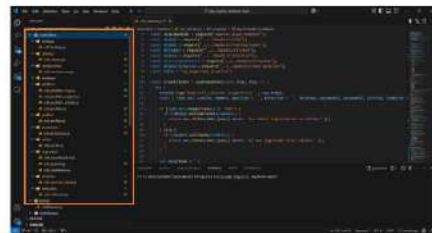
En esta carpeta se muestra una estructura modular de controladores de la aplicación basada en la arquitectura MVC, donde cada archivo (.js) gestiona una funcionalidad específica, como bodega, clientes, conductores, pedidos, perfiles, productos, sectores, seguridad, servicios y vehículos. Estos controladores se encargan de manejar las peticiones HTTP, ejecutar la lógica necesaria y delegar operaciones a otros módulos, manteniendo la organización y la separación de responsabilidades en la aplicación.

Manual técnico | IEEE 1063 – 2001

9

ANRO

V.1.0



3.2.2 Carpeta Config

Contiene el archivo database.js, el cual configura la conexión a una base de datos MySQL mediante el módulo mysql. Define parámetros como host, usuario, contraseña, base de datos y puerto, e incluye un manejo básico de errores para la conexión.



Manual técnico | IEEE 1063 – 2001

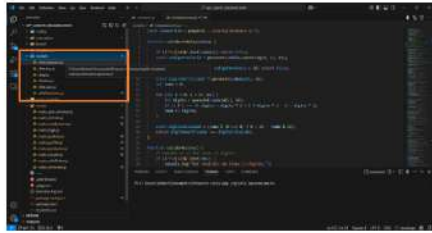
10

ANRO

V.1.0

3.2.3 Carpeta Models

Esta carpeta organiza la lógica de negocio y validaciones en módulos reutilizables, separando las responsabilidades del proyecto. Esto facilita el mantenimiento, pruebas y escalabilidad.

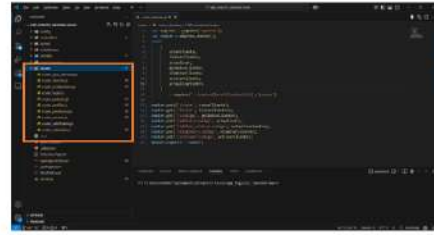


3.2.4 Carpeta Routes

La carpeta routes gestiona el enrutamiento de la aplicación, definiendo los puntos de acceso a las distintas funcionalidades del sistema. Se encarga de mapear las solicitudes HTTP a los controladores correspondientes, garantizando una estructura modular y organizada. Esto facilita la gestión de las rutas, el mantenimiento del código y la implementación de nuevas funcionalidades sin afectar otras partes del proyecto.

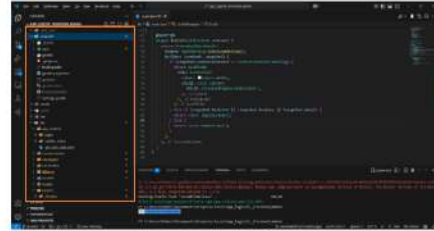
ANRO

V.1.0



3.3 Archivos del frontend móvil

A continuación, se describe la estructura del proyecto y la función de cada carpeta y archivo principal:



- **android/**: Contiene los archivos de configuración y código específico para Android.
- **ios/**: Contiene la configuración y archivos específicos para iOS.
- **assets/**: Directorio donde se almacenan los recursos estáticos como imágenes y fuentes.

ANRO

V.1.0

- **build/**: Carpeta generada automáticamente en el proceso de compilación.
- **test/**: Contiene pruebas unitarias y de integración.

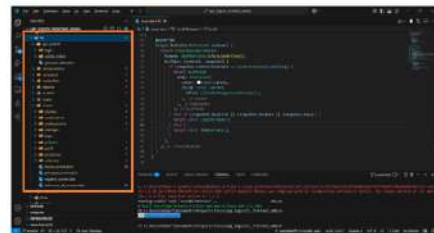
3.4 Carpeta lib

El código fuente principal se encuentra en el directorio lib/, con la siguiente estructura:

- **api_control/**: Contiene la lógica para interactuar con las API del backend.
 - **login/**: Módulos relacionados con la autenticación de usuarios.
 - **valido_token/**: Lógica para la validación de tokens de sesión.
 - **get_user_data.dart**: Método para obtener información del usuario.
- **componentes/**: Contiene los widgets reutilizables dentro de la aplicación.
- **constant/**: Almacena constantes globales utilizadas en toda la aplicación.
- **controllers/**: Controladores que gestionan la lógica de negocio y estado.
- **dijkstra/**: Implementación del algoritmo de Dijkstra para optimización de rutas.
- **models/**: Define las estructuras de datos utilizadas en la aplicación.
- **router/**: Configuración de rutas de navegación entre pantallas.
- **screen/**: Contiene las pantallas principales de la aplicación.
- **main.dart**: Archivo principal de la aplicación, donde se inicia la ejecución.

ANRO

V.1.0



Esta estructura facilita la escalabilidad del proyecto y su mantenimiento. Se recomienda seguir el orden y organización establecidos para futuras mejoras y colaboraciones en el desarrollo.

3.5 Archivos del frontend web

3.5.1 Carpeta admin

Contiene los módulos administrativos del sistema, organizados en subdirectorios:

- **clientes/**: Gestión de clientes.
- **conductores/**: Gestión de conductores.
- **layout/**: Archivos compartidos de diseño.
- **pedidos/**: Gestión de pedidos de clientes.
- **productos/**: Gestión de productos disponibles.
- **roles/**: Administración de roles y permisos.
- **usuarios/**: Gestión de usuarios del sistema.
- **vehiculos/**: Administración de vehículos.
- **index.php**: Archivo principal de la interfaz de administración.

ANRO V1.0



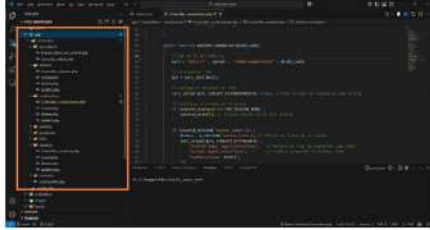
3.5.2 Carpeta app

Aquí se encuentra la lógica del negocio y la configuración del backend:

- **controllers/**: Contiene los controladores que manejan la lógica de cada módulo.
 - **api_externo/**: Módulo para integraciones externas mediante API.
 - **clientes/**, **conductores/**, **pedidos/**, **productos/**, **roles/**, **usuarios/**, **vehiculos/**: Controladores específicos para cada entidad.
 - **solicitudAPI.php**: Módulo para realizar solicitudes a APIs externas.
 - **config.php**: Archivo de configuración general del sistema.

Manual técnico | IEEE 1063 – 2001 15


ANRO V1.0



4 ACCESO A LOS DATOS

La base de datos fue desplegada en un servidor en la nube utilizando Plesk, una plataforma de administración de servidores y hosting. Para acceder a la base de datos, se utilizan las credenciales y direcciones proporcionadas por Plesk.

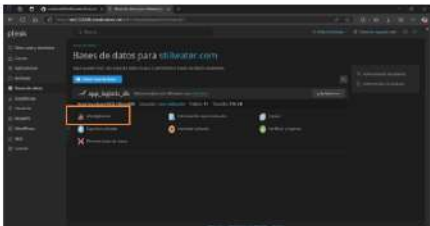
Plesk permite gestionar bases de datos de manera eficiente a través de su interfaz web, facilitando la configuración de usuarios, permisos y conexiones remotas.



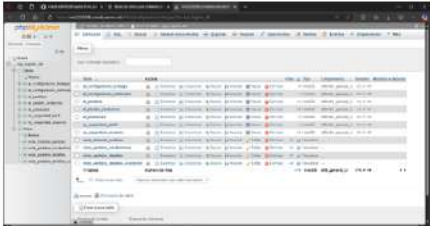
Manual técnico | IEEE 1063 – 2001 16

ANRO V1.0

Para la administración de la base de datos, se utiliza phpMyAdmin, que permite conectarse de forma remota y proporciona una interfaz gráfica para realizar consultas y gestionar los datos de manera más intuitiva.



Esta combinación de herramientas garantiza un acceso seguro y una gestión eficiente de la base de datos, simplificando tareas como la creación de tablas, consultas y copias de seguridad.



Manual técnico | IEEE 1063 – 2001 17

ANRO V1.0

5 GLOSARIO

- **Tecnología**: Conjunto de herramientas, métodos y conocimientos empleados para desarrollar soluciones innovadoras y facilitar diversas tareas.
- **MVC (Modelo-Vista-Controlador)**: Patrón de arquitectura de software que separa la lógica de la aplicación en tres componentes principales: Modelo (datos y lógica), Vista (interfaz de usuario) y Controlador (gestión de eventos y comunicación).
- **Frontend**: Parte de una aplicación con la que interactúa el usuario, generalmente desarrollada con tecnologías como HTML, CSS y JavaScript.
- **Backend**: Lógica y procesamiento de una aplicación que se ejecuta en el servidor, gestionando bases de datos y procesos internos.
- **Repositorio**: Espacio donde se almacena, organiza y gestiona el código fuente de un proyecto, permitiendo el control de versiones y la colaboración entre desarrolladores.
- **APK (Android Application Package)**: Es un archivo especial que se usa para instalar aplicaciones en teléfonos o tabletas con Android.
- **Permisos de aplicación**: Son las cosas que una aplicación necesita permiso para usar en tu teléfono, como la cámara, la ubicación o tus archivos.
- **Framework**: Conjunto de herramientas y librerías que facilitan el desarrollo de software al proporcionar una estructura predefinida y funcionalidad reutilizable, reduciendo el tiempo de implementación y asegurando buenas prácticas.
- **Software**: Conjunto de programas, instrucciones y datos que permiten ejecutar diversas tareas en un sistema informático, incluyendo aplicaciones, sistemas operativos y herramientas de desarrollo.

6 REFERENCIAS

- IEEE (2001). 1063-2001 - IEEE Standard for Software User Documentation [Internet]. Recuperado de: <https://ieeexplore.ieee.org/document/974401>

Manual técnico | IEEE 1063 – 2001 18

Anexo 13: Árbol del problema

EFFECTOS		
Demoras en la entrega	Incrementos de costos	Pérdida de competitividad
Deficiente gestión de rutas de distribución de agua en la empresa Still Water del cantón Santo Domingo		
CAUSAS		
Planificación deficiente	Uso ineficiente de recursos de transporte	Inexistente uso de tecnología

Anexo 14: Recodificación del instrumento de recolección de datos

Recodificación de la escala de Likert

Recodificación	Escala de Likert
Frecuencia	
4	Siempre
3	Frecuentemente
2	A veces
1	Rara vez
0	Nunca
De Acuerdo	
4	Totalmente de acuerdo
3	De acuerdo
2	Moderadamente de acuerdo
1	Poco de acuerdo
0	Nada de acuerdo
Dificultad	
4	Muy difícil
3	Difícil
2	Moderado
1	Fácil
0	Muy fácil
Importancia	
4	Muy importante
3	Importante
2	Neutral
1	Poco importante
0	Nada importante
Eficiencia	
4	Muy eficiente
3	Eficiente
2	Moderadamente eficiente
1	Poco eficiente
0	Nada eficiente
Satisfacción	
4	Muy satisfecho
3	Satisfecho
2	Neutral
1	Insatisfecho
0	Muy insatisfecho

Recodificación de las respuestas que no forman parte de la escala de Likert

Recodificación	Respuestas
Edad	
1	18-25
2	26-30
3	30-40
4	40 o más
Género	
1	Masculino
2	Femenino
Tipo de discapacidad	
0	Ninguna
1	Otro
2	Visual
3	Auditiva
4	Física
Posee un teléfono inteligente	
1	Si
2	No
Sistema operativo del teléfono	
1	Android
2	IOS
3	Windows Phone
4	Otro
Cantidad de consumo de bidones	
0	1
1	2 a 4
2	4 a 7
3	7 a 10
4	Más de 10

Anexo 15: Informe del Turniting

turnitin_FTT_202302O_Sacheri_Andrade_11_02_F.docx			
INFORME DE ORIGINALIDAD			
5%	5%	1%	1%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE
FUENTES PRIMARIAS			
1	repositorio.ucv.edu.pe Fuente de Internet		<1%
2	dspace.esPOCH.edu.ec Fuente de Internet		<1%
3	bibdigital.epn.edu.ec Fuente de Internet		<1%