

**OPTIMIZACIÓN DEL TRANSPORTE PÚBLICO EN VALLEDUPAR MEDIANTE
EL USO DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE RUTAS Y
DESTINOS**

LIBNY DAVID MOLINA RUIZ

ldmolina@unicesar.edu.co

CARLOS DANIEL BOLAÑO RUA

cdbolano@unicesar.edu.co

**UNIVERSIDAD POPULAR DEL CESAR
FACULTAD DE INGENIERÍAS Y TECNOLOGÍAS
PROGRAMA DE INGENIERÍA ELECTRÓNICA**

**Valledupar – Cesar
2023**

**OPTIMIZACIÓN DEL TRANSPORTE PÚBLICO EN VALLEDUPAR MEDIANTE
EL USO DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE RUTAS Y
DESTINOS**

**PROYECTO DE GRADO PRESENTADO COMO REQUISITO PARA OPTAR POR
EL TÍTULO DE INGENIERO ELECTRÓNICO**

PRESENTADO POR:

**LIBNY DAVID MOLINA RUIZ
CARLOS DANIEL BOLAÑO RUA**

DIRECTOR:

MSc Ing. José Ramón Iglesias Gamarra PhD

CODIRECTORA:

Ing. OMAIRA LUZ TAPIAS DIAZ

**UNIVERSIDAD POPULAR DEL CESAR
FACULTAD DE INGENIERÍAS Y TECNOLOGÍAS
PROGRAMA DE INGENIERÍA ELECTRÓNICA**

**Valledupar – Cesar
2023**

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

DEDICATORIA

A toda nuestra familia, especialmente a nuestros padres, quienes con su amor y cariño nos han motivado a terminar este libro y a lo largo de los años nos han apoyado, aconsejado y respaldado en todos los aspectos de nuestra vida, especialmente en nuestra educación y crecimiento personal.

*Carlos Daniel Bolaño Rúa.
Libny David Molina Ruiz.*

AGRADECIMIENTOS

A nuestro director, José Ramón Iglesias Gamarra, por brindarnos su conocimiento, apoyo y herramientas para desarrollar este proyecto de la mejor manera y con estrategias claras. A la Universidad Popular del Cesar por facilitarnos los escenarios, equipos y herramientas que llevaron a la culminación del mismo. Y al SIVA S.A.S, quienes nos facilitaron acceso a información, aconsejaron y apoyaron en el desarrollo de este gran proyecto.

Carlos Daniel Bolaño Rúa.

Libny David Molina Ruiz.

INDICE

DEDICATORIA	4
AGRADECIMIENTOS	5
INDICE DE FIGURAS	8
INDICE DE TABLAS.....	9
RESUMEN	10
ABSTRACT	11
1 INTRODUCCIÓN	12
2 ANTECEDENTES	13
2.1 ANTECEDENTES REGIONALES	13
2.2 ANTECEDENTES INTERNACIONALES	13
3 LÍNEA DE INVESTIGACIÓN	14
4 FORMULACIÓN DEL PROBLEMA	15
5 JUSTIFICACIÓN	16
6 OBJETIVOS	17
6.1 OBJETIVO GENERAL	17
6.2 OBJETIVOS ESPECÍFICOS	17
7 MARCO TEÓRICO.....	18
7.1 APLICACIÓN MÓVIL	18
7.1.1 TIPOS DE APLICACIÓN MÓVIL.....	18
7.2 FLUTTER.....	19
7.3 LENGUAJE DE PROGRAMACIÓN DART	19
7.4 API GOOGLE	20
7.5 API OPEN STREET MAP (OSM)	20
7.6 LENGUAJE DE PROGRAMACIÓN PYTHON.....	21
7.6.1 FAST-API.....	21
7.7 SERVIDORES.....	21
7.8 BASES DE DATOS	22
7.8.1 NOSQL	22
7.9 PROCESAMIENTO DE DATOS.....	22
7.10 ALGORITMOS DE BÚSQUEDA	23
7.10.1 ALGORITMO DE DIJKSTRA	23

7.11 ARCHIVOS JSON	23
8 DISEÑO METODOLÓGICO	25
8.1 Base de datos con las rutas	25
8.2 Diseño de interfaz (front-end).....	26
8.3 Desarrollo del código back-end.....	35
8.4 OPTIMIZACIÓN Y MEJORAS.....	38
8.4.1 Pruebas para detectar errores o puntos a mejorar	38
8.4.2 Prueba de la aplicación por personas aparte de los autores	39
9 RESULTADOS.....	41
10 CONCLUSIONES.....	47
11 RECOMENDACIONES.....	49
12 COSTOS Y RECURSOS GASTADOS	50
13 REFERENCIAS BIBLIOGRÁFICAS	51

INDICE DE FIGURAS

Figura 1. Diagrama de flujo front-end general. Autores.....	27
Figura 2. Diagrama de flujo de botones del menú. Autores.....	27
Figura 3. Pantalla al pulsar abrir app. Autores.....	28
Figura 4. Pantalla principal. Autores.....	28
Figura 5. Menú desplegado (dark). Autores	29
Figura 6. Menú desplegado. Autores.....	29
Figura 7. Pantalla de lista de rutas. Autores.....	30
Figura 8. Pantalla de sectores en mapa. Autores.....	30
Figura 9. Gráfica estadística de relación de sectores sin rutas disponibles. Autores	31
Figura 10. Pantalla de buscador. Autores.....	31
Figura 11. Direcciones digitadas (1). Autores	32
Figura 12. Pantalla de carga (1). Autores.....	32
Figura 13. Recomendaciones (1). Autores	32
Figura 14. Recorrido de origen a destino de ruta seleccionada. Autores	33
Figura 15. Direcciones digitadas (2). Autores.....	33
Figura 16. Pantalla de carga (2). Autores.....	34
Figura 17. Recomendaciones (2). Autores	34
Figura 18. Recorrido de origen a destino con transbordo de ruta. Autores	35
Figura 19. Diagrama back-end general. Autores	36
Figura 20. Diagrama de flujo de almacenamiento interno. Autores	37
Figura 21. Diagrama de flujo para transbordos. Autores	37
Figura 22. Gráfica comparativa de rutas recomendadas. Autores.....	42
Figura 23. Gráfica comparativa de transbordos recomendados. Autores.....	43
Figura 24. Gráfica comparativa de mayor tiempo de rutas recomendadas. Autores	43
Figura 25. Gráfica comparativa de menor tiempo de rutas recomendadas. Autores	44
Figura 26. Gráfica comparativa de tiempo de respuesta del servidor. Autores	44
Figura 27. Transbordo app SIVAlliu. SIVAlliu.....	45
Figura 28. Transbordo app ValleRuta. Autores.....	45

INDICE DE TABLAS

Tabla 1. Resultados de aplicación ValleRuta. Autores.....	41
Tabla 2. Resultados de aplicación SIVAlliu. Autores.....	42
Tabla 3. Resultados de aplicación Moovit. Autores	42
Tabla 4. Análisis de costos y recursos. Autores.....	50

RESUMEN

Este proyecto tuvo como objetivo crear una herramienta tecnológica que brinde a los ciudadanos de Valledupar la facilidad de usar el transporte público urbano de la ciudad, teniendo a la mano información clara y precisa sobre las rutas que lo pueden llevar a su destino, el tiempo de desplazamiento, la distancia del recorrido, entre otra información. Asimismo, contribuir a la empresa operadora del servicio en la ciudad recopilando información estadística para el análisis de posibles modificaciones de rutas y/o implementación de nuevas que puedan abarcar más zonas y brindar un mejor servicio [1].

El primer paso que dimos para desarrollar la idea fue recopilar información, opiniones y sugerencias de una muestra poblacional y empresas, para con ellos establecer las funcionalidades y requisitos con que debe contar la aplicación. Luego procedimos a desarrollar la aplicación, con la herramienta de Google, Flutter [7], se desarrolló el front-end de la app y para el back-end se utilizó el lenguaje de programación Python [9]. Se realizaron pruebas de la versión beta, usando el transporte público urbano siguiendo las instrucciones de la app, detectamos errores y características a mejorar, se corrigió y mejoró el código (se realizó un total de dos veces las pruebas y correcciones).

Como producto final obtuvimos una aplicación móvil funcional, optimizada y fácil de usar que proporciona información precisa, actualizada y de fácil acceso al usuario, esto incentivó al uso del transporte público urbano dentro de la población de prueba (estudiantes de la Universidad Popular del Cesar), también se obtuvo una nueva fuente de información para las empresas que proporciona datos estudiados y analizados con ciencia de datos, el cual da como resultado estadísticas sobre lugares más demandados por los usuarios sin rutas disponibles, para así analizar posibles nuevas rutas con gran demanda. Lo anterior con el fin de dar sugerencias basadas en el estudio de datos y consecuentemente ampliar y/o mejorar el servicio ofrecido.

Palabras clave: Flutter, Python, ciencia de datos, operadora.

ABSTRACT

This project aimed to create a technological tool that provides the citizens of Valledupar with the ease of using the city's urban public transport, having at hand clear and precise information about the routes that can take them to their destination, the travel time, displacement, the distance of the journey, among other information. Likewise, contribute to the service operating company in the city by collecting statistical information for the analysis of possible route modifications and/or implementation of new ones that can cover more areas and provide a better service [1].

The first step we took to develop the idea was to collect information, opinions and suggestions from a sample of the population and companies, in order to establish the functionalities and requirements that the application must have. Then we proceeded to develop the application, with the Google tool, Flutter [7], the front-end of the app was developed and the Python programming language was used for the back-end [9]. Tests of the beta version were carried out, using urban public transport following the instructions of the app, we detected errors and features to improve, the code was corrected and improved (the tests and corrections were carried out a total of two times).

As a final product, we obtained a functional, optimized and easy-to-use mobile application that provides accurate, updated and easily accessible information to the user. This encouraged the use of urban public transportation within the test population (students from the Popular University of Cesar). , a new source of information was also obtained for companies that provides data studied and analyzed with data science, which results in statistics on places most in demand by users without available routes, in order to analyze possible new routes with high demand. The above in order to give suggestions based on the study of data and consequently expand and/or improve the service offered.

Keywords: Flutter, Python, data science, operator.

1 INTRODUCCIÓN

Valledupar es una ciudad pequeña que está en crecimiento, esto incluye el sistema de transporte público urbano, el cual a finales del año 2022 tuvo un crecimiento casi instantáneo, ya que entraron en funcionamiento 130 nuevos autobuses de SIVA S.A.S [2]. Sin embargo, el sistema no tuvo el impacto y el éxito esperado debido a que los ciudadanos carecen de información sobre el sistema. La falta de información precisa, actualizada y de fácil acceso, provoca pérdidas de tiempo e insatisfacción en los usuarios y esto lleva a una consecuencia aún más grave: el sistema no es bien recibido por la ciudadanía, lo que los lleva a optar por el transporte informal, especialmente el mototaxismo [4], aumentando así el riesgo de accidentes, robos y transporte ilegal en la ciudad.

De lo anterior, nace una idea desde el campo de la ingeniería que pretende ofrecer una solución tecnológica mediante el desarrollo y la implementación de una aplicación móvil [19], que les permita a los ciudadanos conocer y utilizar de una manera interactiva y optimizada el sistema integrado de transporte de Valledupar, ayudando a que se desplacen por la ciudad de manera segura y eficiente, brindándoles diferentes opciones e instrucciones precisas sobre como transportarse con el sistema. Y además de ello, aprovechar y recopilar los datos para brindarle a la empresa la información tratada y analizada sobre los destinos sin rutas disponibles.

2 ANTECEDENTES

2.1 ANTECEDENTES REGIONALES

Al realizar una revisión bibliográfica en relación a la temática del proyecto a nivel regional, no se encontró a la fecha ningún antecedente en la ciudad de Valledupar. Sin embargo, en la ciudad de Pereira se encontró que en el trabajo "Diseño de un aplicativo móvil para consultar las rutas de buses y sus recorridos dentro del AMCO" publicado en 2019 por Jaime Valencia y Jesús Molina, presentan un proyecto cuyo objetivo es diseñar e implementar un aplicativo móvil que ayude a los usuarios a tener una mejor experiencia utilizando el servicio público [5]. En esta app se encuentran funciones de consultar las rutas y brinda información sobre puntos de interés.

2.2 ANTECEDENTES INTERNACIONALES

Al realizar la revisión bibliográfica a nivel internacional se encontró que, en un artículo publicado en 2022, Raquel Fulgêncio, Marta Campos Ferreira, Diogo Abrantes y Miguel Coimbra llevaron a cabo un estudio titulado "Restart: un planificador de rutas para incentivar el uso de los servicios de transporte público en un contexto de pandemia", con el objetivo de proponer un planificador de rutas que mejore la seguridad y comodidad en el uso del transporte público [6]. Los hallazgos principales del estudio indican que el prototipo Restart fue altamente valorado por los participantes en términos de usabilidad, obteniendo una puntuación media de 4.7 sobre 5 en el cuestionario post-tarea. Además, los participantes destacaron que la disponibilidad de información en tiempo real sobre diferentes aspectos del viaje fue el factor principal para el uso del sistema en el futuro. Los resultados también mostraron que los participantes proporcionaron importantes conocimientos para la mejora del sistema, sugiriendo la necesidad de incluir información sobre los precios de los viajes y la conexión con una aplicación móvil. En resumen, el estudio propuso un prototipo de aplicación móvil para mejorar la experiencia de viaje en servicios de transporte público, evaluó su usabilidad y utilidad a través de pruebas con usuarios reales, identificó las características y funcionalidades más valoradas por los usuarios en una aplicación móvil de transporte público, y ofreció información relevante para el diseño y desarrollo de futuras aplicaciones móviles de transporte público.

3 LÍNEA DE INVESTIGACIÓN

La temática de este proyecto se enfoca en una de las tres líneas de investigación del programa de Ingeniería Electrónica de la Facultad de Ingeniería y Tecnológicas. Esta línea se denomina "Electrónica, Optoelectrónica y Ciencias Computacionales". Específicamente, el proyecto se centra en la sublínea de "Optoelectrónica y Procesamiento de Datos", que abarca las áreas temáticas de procesamiento digital de datos, análisis de datos, tratamiento de datos y ciencia de datos.

4 FORMULACIÓN DEL PROBLEMA

En Valledupar, el sistema de transporte público urbano ha sido muy limitado para la cantidad de habitantes de la ciudad, hasta finales del mes de diciembre de 2022, cuando entraron en funcionamiento los nuevos autobuses de SIVA S.A.S [2]. Sin embargo, el sistema no ha tenido el impacto y el éxito esperado debido a que los ciudadanos carecen de información sobre el sistema. La falta de información precisa, actualizada y de fácil acceso es una de las causas de esta problemática, lo que dificulta a los ciudadanos conocer el sistema y, por lo tanto, planificar sus desplazamientos, lo que resulta en una falta de eficiencia y optimización, provocando pérdidas de tiempo e insatisfacción en los usuarios. Esto lleva a una consecuencia aún más grave: el sistema no es bien recibido por la ciudadanía, lo que los lleva a optar por el transporte informal, especialmente el mototaxismo, lo que aumenta el riesgo de accidentes, robos y transporte ilegal en la ciudad [4].

5 JUSTIFICACIÓN

Al momento de implementar un nuevo sistema, es importante brindar la información pertinente a los usuarios y posibles usuarios para facilitar e incentivar el uso del nuevo sistema integrado. La información debe ser precisa, actualizada y de fácil acceso para ayudar a los usuarios a utilizar correctamente y de manera eficiente el sistema [3]. En el caso del sistema integrado de transporte de Valledupar, este requisito no se cumple, ya que, aunque cuentan con una página web, esta no proporciona de manera interactiva la información necesaria para el uso eficiente del sistema, ni les da instrucciones precisas sobre cómo desplazarse utilizando el servicio ofrecido [1].

Por consiguiente, se pretende ofrecer una solución tecnológica mediante el desarrollo y la implementación de una aplicación móvil, que le permita a los ciudadanos conocer y utilizar de una manera interactiva y optimizada el sistema integrado de transporte de Valledupar (SIVA S.A.S), además que sea una herramienta práctica y accesible que ayude a los usuarios a desplazarse por la ciudad de manera segura y eficiente, brindándoles diferentes opciones e instrucciones precisas sobre como desplazarse con el sistema.

Además de la justificación anterior, otra motivación significativa es contribuir a SIVA S.A.S. como empresa para ampliar y mejorar su servicio, asegurándonos de que los usuarios estén satisfechos y, al mismo tiempo, aumentar su número. Esto resultará en beneficio tanto para la empresa al aumentar sus ingresos y clientes, como para la ciudadanía al mejorar el servicio. Cabe destacar que SIVA S.A.S ha mostrado un gran interés y apoyo en este proyecto, como se puede ver en los anexos adjuntos.

6 OBJETIVOS

6.1 OBJETIVO GENERAL

Desarrollar una aplicación móvil que facilite a los ciudadanos de Valledupar el uso del sistema de transporte público urbano, con el fin de mejorar la eficiencia y comodidad de los desplazamientos dentro de la ciudad.

6.2 OBJETIVOS ESPECÍFICOS

Desarrollar una interfaz para visualizar información sobre rutas, horarios y paraderos, entre otros elementos.

Implementar un algoritmo para brindar indicaciones precisas sobre dónde, cuál y a qué hora tomar los buses que lleven al usuario a su destino.

Recopilar en una base de datos información sobre las rutas más utilizadas y los lugares más visitados por los usuarios.

Analizar los datos recopilados con ciencia de datos para obtener estadísticas reales y proporcionar sugerencias basadas en el estudio de datos a la empresa encargada sobre posibles nuevas rutas con alta demanda.

7 MARCO TEÓRICO

El objetivo de este capítulo es presentar los fundamentos teóricos del trabajo. Para ello, se abordan los conceptos y temas que contribuyen a una comprensión más profunda del objeto de estudio, así como los conceptos matemáticos, algorítmicos y computacionales que son necesarios para el desarrollo del proyecto.

7.1 APLICACIÓN MÓVIL

Las aplicaciones móviles son programas informáticos desarrollados en diversos lenguajes de programación diseñados específicamente para dispositivos como smartphones y tablets. Su característica principal radica en su utilidad, agilidad y facilidad de instalación, sin dejar de lado la capacidad de ofrecer servicios y experiencias de alta calidad [23].

A diferencia de las aplicaciones diseñadas para computadoras de escritorio, las aplicaciones móviles se caracterizan por su enfoque en brindar funcionalidades independientes y limitadas. Cada aplicación móvil se desarrolla para cumplir con una tarea o servicio particular, en lugar de ser parte de un sistema de software más amplio.

7.1.1 TIPOS DE APLICACIÓN MÓVIL

En el ámbito de las aplicaciones móviles, surge una distinción esencial entre dos categorías predominantes: las aplicaciones web y las aplicaciones nativas. La distinción primordial entre estas radica en su enfoque de desarrollo interno, es decir, en la forma en que han sido concebidas y programadas.

Aplicaciones Nativas: Las aplicaciones nativas son aquellas que se diseñan teniendo en cuenta las particularidades de cada plataforma en la que se desplegarán, ya sea un sistema operativo Android o iOS. Estas aplicaciones se optimizan minuciosamente para ofrecer un rendimiento y una experiencia de usuario excepcionales en su plataforma específica. Además, cualquier anomalía en su funcionamiento o mejoras necesarias se corrigen mediante la descarga de actualizaciones meticulosamente diseñadas con ese fin.

Aplicaciones Híbridas: Las aplicaciones híbridas, también conocidas como aplicaciones web, no se adentran en niveles de especificidad tan profundos como las aplicaciones nativas, lo que significa que pueden adaptarse a diversos entornos en los que se instalan. A pesar de no ofrecer el mismo grado de optimización y rendimiento que las aplicaciones nativas, gozan de la ventaja de su versatilidad,

funcionando en una amplia gama de plataformas sin requerir una programación específica para cada una. Es decir, este tipo de aplicaciones se pueden implementar en sistemas operativos Android y iOS con el mismo código.

7.2 FLUTTER

Flutter es una herramienta gratuita de Google que permite construir aplicaciones que se compilan de manera nativa y son multiplataforma, lo que significa que se puede desarrollar aplicaciones para Android y iOS a partir de una sola base de código. Se trata de un kit de desarrollo de software de código abierto y, dado a que es un software libre, los desarrolladores pueden utilizarlo sin restricciones de licencia ni necesidad de realizar pagos por su uso. Flutter utiliza el lenguaje de programación Dart, el cual también fue desarrollado por Google [7] y, también se puede integrar con códigos en otro lenguaje como Python (del cual se hablará más adelante).

Algunas de sus características más importantes son:

Base de código única: se dirige a Android, iOS, web, Windows, macOS y Linux en una sola base de código.

Compatibilidad con IDE: Crea aplicaciones con editores como VScode para la mejora de trabajo y herramientas de desarrollo integradas.

Siempre de código abierto: Gracias al respaldo de Google y la colaboración de una comunidad global de desarrolladores, Flutter se destaca por su apertura, transparencia y confiabilidad.

7.3 LENGUAJE DE PROGRAMACIÓN DART

Dart es un lenguaje de programación de código abierto desarrollado por Google con la finalidad de optimizar y agilizar el proceso de desarrollo para los programadores. Este lenguaje se caracteriza por ser orientado a objetos y ofrece la opción de tipado estático o dinámico, lo que brinda flexibilidad en el desarrollo. Dart es conocido por su eficiencia y velocidad, gracias a su máquina virtual y la compilación Just-in-Time. Además, cabe recordar y destacar que es el lenguaje principal detrás de Flutter.

Dart también cuenta con herramientas integradas que simplifican el proceso de desarrollo, como lo son diversos compiladores/transpiladores, su propio administrador de paquetes, así como un analizador y formateador. En lo que respecta a la sintaxis, Dart guarda similitudes notables con lenguajes como

JavaScript, Java y C++, lo que facilita el proceso de aprendizaje de Dart para aquellos que ya están familiarizados con alguno de estos lenguajes [8].

Aunque inicialmente se centró en el desarrollo web, Dart se puede transpilar a JavaScript, lo que lo hace compatible con los navegadores modernos. Con el respaldo de Google y una comunidad activa, Dart continúa ganando popularidad en el mundo de la programación.

7.4 API GOOGLE

La API (interfaz de programación de aplicaciones) de Google es un conjunto de interfaces y protocolos que permiten a los desarrolladores interactuar con los servicios y datos proporcionados por Google. Estas API permiten que aplicaciones y sitios web accedan a una amplia gama de servicios de Google, como Google Maps, Google Drive, Google Calendar y muchos otros [16]. Los desarrolladores pueden utilizar estas API para realizar tareas como obtener direcciones y mapas, acceder a archivos en la nube, analizar datos de usuario y mucho más.

Las API de Google son versátiles y están disponibles en múltiples lenguajes de programación, lo que facilita su integración en una amplia variedad de plataformas y aplicaciones sin necesidad de saber cómo están implementados. Además, Google proporciona una plataforma de desarrollo sólida y documentación detallada para ayudar a los desarrolladores a aprovechar al máximo estas API [11].

7.5 API OPEN STREET MAP (OSM)

Open Street Map (OSM) es un proyecto colaborativo de cartografía en línea que permite a los desarrolladores crear y actualizar mapas de todo el mundo de manera gratuita y abierta. A diferencia de otros servicios de mapas, como Google Maps, OSM se basa en datos geoespaciales de código abierto que cualquiera puede usar, modificar y compartir [24].

Algunos de los aspectos clave de la API de OpenStreetMap son:

Datos Abiertos y Colaborativos: La comunidad de OSM mapea carreteras, edificios y muchos otros elementos geoespaciales en todo el mundo y cualquier persona puede contribuir a este mapa y mejorar su precisión.

Personalización de Mapas: Los desarrolladores pueden utilizar la API para crear mapas personalizados y aplicaciones basadas en mapas. Se puede agregar capas

personalizadas, estilos de mapa y funcionalidades específicas para satisfacer las necesidades del proyecto.

Geocodificación y Enrutamiento: OSM ofrece servicios de geocodificación para convertir direcciones en coordenadas geográficas y enrutamiento para calcular rutas entre puntos. Esto es útil para planificación y recomendación de rutas.

Licencia Abierta: Los datos de OSM están disponibles bajo la Licencia Abierta, lo que significa que se puede utilizar en proyectos comerciales y no comerciales, siempre y cuando se reconozcan los derechos de autor y se cumplan las normas de OSM.

7.6 LENGUAJE DE PROGRAMACIÓN PYTHON

Python es un lenguaje de programación de alto nivel, es interpretado, multiplataforma y versátil, permitiendo la programación orientada a objetos, imperativa y funcional. Python brilla por su extensa biblioteca estándar, que abarca manipulación, procesamiento y tratamiento de datos. Su comunidad activa y diversificada impulsa la creación de bibliotecas de terceros y recursos en línea abundantes. Utilizado en una gama ecléctica de aplicaciones, desde desarrollo web hasta análisis de datos y aprendizaje automático, Python es un pilar en la programación moderna, siendo de código abierto y gratuito, fomentando la colaboración y simplificando la escritura de código [9].

7.6.1 FAST-API

FastAPI es un framework de alto rendimiento utilizado para desarrollar APIs en Python 3.6 o versiones posteriores, aprovechando las anotaciones de tipos estándar de Python. FastAPI ha experimentado un aumento significativo en su popularidad debido a su rendimiento excepcional, capacidad para reducir errores humanos, su enfoque intuitivo, velocidad de desarrollo rápida, facilidad de uso y aprendizaje, además de su diseño conciso y robusto [16].

7.7 SERVIDORES

Los servidores son sistemas de hardware y software diseñados para proporcionar servicios y recursos a través de redes. Cumplen un papel esencial en la infraestructura tecnológica, sirviendo aplicaciones, datos y servicios a clientes. Existen varios tipos de servidores, pero en esta ocasión nuestro interés se centra en los servidores de bases de datos adaptado a tareas específicas. La disponibilidad constante, el rendimiento eficiente, la seguridad y la escalabilidad son

características clave de los servidores. En un mundo cada vez más digital, los servidores son fundamentales para empresas, organizaciones, servicios en línea, y por supuesto, para aplicaciones móviles.

7.8 BASES DE DATOS

Las bases de datos son herramientas para recopilar y organizar información sobre una variedad de cosas. A menudo, estas bases de datos comienzan como simples listas en hojas de cálculo o programas de procesamiento de texto. Sin embargo, a medida que estas listas crecen, pueden volverse difíciles de manejar. Los datos se vuelven redundantes e inconsistentes, y encontrar o extraer información específica se vuelve complicado. Cuando surgen estos problemas, suele ser una buena idea migrar los datos a una base de datos creada con un Sistema de Gestión de Bases de Datos (DBMS, por sus siglas en inglés). Los DBMS ayudan a organizar y administrar los datos de manera más eficiente, permitiendo búsquedas más avanzadas y la extracción de subconjuntos de datos de manera más sencilla. Las bases de datos son una herramienta valiosa para manejar grandes cantidades de información de manera estructurada y eficiente.

7.8.1 NOSQL

Las bases de datos NoSQL están diseñadas para modelos de datos específicos y almacenan la información en esquemas flexibles que se pueden escalar fácilmente para las aplicaciones modernas. Estas bases de datos son ampliamente valoradas por su facilidad de desarrollo, funcionalidad y rendimiento a gran escala [25].

Siendo así, algunas de las ventajas de estas bases de datos son:

Flexibilidad: ofrecen esquemas que permiten un desarrollo más rápido e iterativo.

Alto rendimiento: son optimizadas para modelos de datos y patrones de acceso específicos.

Altamente funcional: proporcionan APIs altamente funcionales y tipos de datos diseñados específicamente para cada uno de sus modelos.

7.9 PROCESAMIENTO DE DATOS

El procesamiento de datos es una actividad fundamental en la era de la información. Consiste en la manipulación y transformación de datos de diversas fuentes para obtener información útil y tomar decisiones informadas. Este proceso implica tomar

datos brutos y convertirlos en información útil mediante diversas operaciones como clasificación, cálculos matemáticos, resumen, filtrado y más. Este paso es crucial para convertir datos en conocimiento [26].

7.10 ALGORITMOS DE BÚSQUEDA

Los algoritmos de búsqueda son procedimientos o conjuntos de reglas diseñados para encontrar un elemento específico dentro de un conjunto de datos. Estos algoritmos son fundamentales en ciencias de la computación y se utilizan en una variedad de aplicaciones para buscar información, datos, patrones, o tomar decisiones basadas en la búsqueda de elementos en una colección de datos. Los algoritmos de búsqueda se pueden aplicar en una amplia variedad de contextos. En el caso concreto de búsqueda de rutas, uno de los algoritmos más conocidos es el algoritmo de Dijkstra. Este algoritmo fue desarrollado por el científico de la computación Edsger W. Dijkstra y se utiliza para determinar el camino más corto entre dos puntos determinados.

7.10.1 ALGORITMO DE DIJKSTRA

El algoritmo de Dijkstra fue desarrollado por el científico de la computación Edsger Dijkstra en 1956, es un método fundamental en teoría de grafos y optimización que se utiliza para encontrar el camino más corto entre un nodo de inicio y todos los demás nodos en un grafo ponderado.

La idea detrás del algoritmo de Dijkstra es mantener un conjunto de nodos cuyas distancias más cortas desde el nodo de inicio se han determinado de manera definitiva, y un conjunto de nodos cuyas distancias aún se están explorando. El algoritmo avanza iterativamente, seleccionando el nodo con la distancia más corta en el conjunto de nodos no explorados y actualizando las distancias de sus vecinos si se puede encontrar un camino más corto a través del nodo seleccionado. Este proceso continúa hasta que se haya explorado todo el grafo o se haya alcanzado el nodo de destino, si se está buscando un camino específico. Este algoritmo es una herramienta poderosa para encontrar rutas más cortas en grafos ponderados y ha sido ampliamente utilizado en aplicaciones como enrutamiento de redes, sistemas de navegación, y en general, en cualquier problema donde se requiera encontrar el camino óptimo entre dos puntos en un grafo.

7.11 ARCHIVOS JSON

Los archivos JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero. Su simplicidad lo hace fácilmente legible y escribible para los seres

humanos, al mismo tiempo que es sencillo de interpretar y generar para las máquinas. Son muy utilizados para realizar consultas en API's y/o servidores donde se guarda información utilizada en la app [10].

En un archivo JSON, la organización de los datos se basa en pares de clave-valor, que se asemejan a los diccionarios utilizados en muchos lenguajes de programación (como Python). Estos pares se agrupan en objetos, y es posible anidar objetos para crear estructuras de datos complejas. Los valores que se pueden encontrar en un archivo JSON abarcan desde cadenas de texto, números, booleanos y nulos, hasta objetos y matrices [22].

Ejemplo de archivo JSON:

```
{
  "nombre":"Yupaka",
  "profesion":"Ingeniero",
  "edad":21,
  "lenguajes":["Python","Java","Dart"],
  "disponibilidadParaViajar":true,
  "rangoProfesional": {
    "aniosDeExperiencia": 3,
    "nivel": "Junior"
  }
}
```

El anterior formato y estructura se utilizó para la base de datos de las rutas, tanto en ida como en vuelta.

8 DISEÑO METODOLÓGICO

8.1 Base de datos con las rutas

Lo primero que se creó para el desarrollo de la aplicación fue una base de datos que contiene las rutas disponibles y ofrecidas por SIVA S.A.S, las cuales se obtuvieron en archivos JSON suministrados por la empresa a través de su página web. Este archivo contiene una lista de datos de coordenadas geográficas, que en conjunto forman líneas que hacen parte de la ruta, de tal manera que al unirse y cargar en el mapa se observa todo el recorrido de la ruta en la ciudad.

El archivo está estructurado de la siguiente forma:

```
{
  "type": "FeatureCollection",
  "name": "Layer 100",
  "crs": {
    "type": "name", "properties": {
      "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
    }
  },
  "features": [
    {
      "type": "Feature", "properties": {
        "name": "100_IDA", "Route_ID": 46, "Route_Name": "102_IDA", "HEADWAY": 5,
        "RUTA": 102, "LONGITUD": 13.25, "TIEMPO": 41.01, "ID": 46
      },
      "geometry": {
        "type": "LineString", "coordinates": [
          [ long_1, lat_1, 0.0], [ long_2, lat_2, 0.0 ],..., [ long_n, lat_n, 0.0 ]]
        }
      },
      {
        "type": "Feature", "properties": {
          "name": "100_REGRESO", "Route_ID": 47, "Route_Name": "102_REGRESO",
          "HEADWAY": 5, "RUTA": 102, "LONGITUD": 19.91, "TIEMPO": 59.18, "ID": 47
        },
        "geometry": {
          "type": "LineString", "coordinates": [
            [ long_1, lat_1, 0.0], [ long_2, lat_2, 0.0 ],..., [ long_n, lat_n, 0.0 ]]
          }
        }
      }
    ]
  }
}
```

Estos archivos en formato JSON no requieren mucha capacidad de procesamiento, ni memoria y es el formato más comúnmente usado, ya que es sencillo para el intercambio de datos.

8.2 Diseño de interfaz (front-end)

Bajo un modelo de encuesta presencial y verbal, se consultó con la población estudiantil de la universidad Popular del Cesar cuál era su opinión respecto al transporte público urbano de la ciudad y que si pensaba que había la información suficiente para saber cómo usar el sistema de la forma más óptima, posterior a esto se preguntó sobre si utilizarían una aplicación que proporcione información a detalle para usar el sistema y cuáles funcionalidades debería tener la app.

Las funcionalidades que la mayoría de encuestados sugirieron son las siguientes:

- Tener a disposición el recorrido gráficamente de cada ruta en el mapa de la ciudad.
- Tener la información de la empresa operadora (SIVA S.A.S) a la mano para contactar en caso de petición, queja, reclamos, sugerencias y/o emergencias.
- Tener un indicador a detalle de recorridos y rutas que lleven al usuario de su origen a destino deseado.

Por nuestro lado, se nos ocurrió agregar una funcionalidad de recopilar datos de orígenes y destinos que no cuenten con rutas disponibles, para así tratar los datos y entregar estadísticas a la empresa prestadora del servicio, con el fin de que puedan estudiar posibles modificaciones a rutas existentes o implementaciones de nuevas rutas con gran demanda.

Lo anterior se le comunicó a la empresa SIVA S.A.S, para nutrirnos de opiniones y sugerencias acerca de esta idea. Como resultado, SIVA mostró interés a la idea y además puso a disposición del proyecto la información y apoyo necesario.

Una vez se definieron las funcionalidades y requisitos de la app, comienza la etapa de diseño de la interfaz gráfica, la cual se desarrolló en la herramienta de Google, Flutter, que permite construir aplicaciones en lenguaje de programación Dart que se compilan de manera nativa.

Luego de diseñar la interfaz de la aplicación con los requisitos y funcionalidades planteadas, se muestra el diagrama de flujo de front-end a continuación:

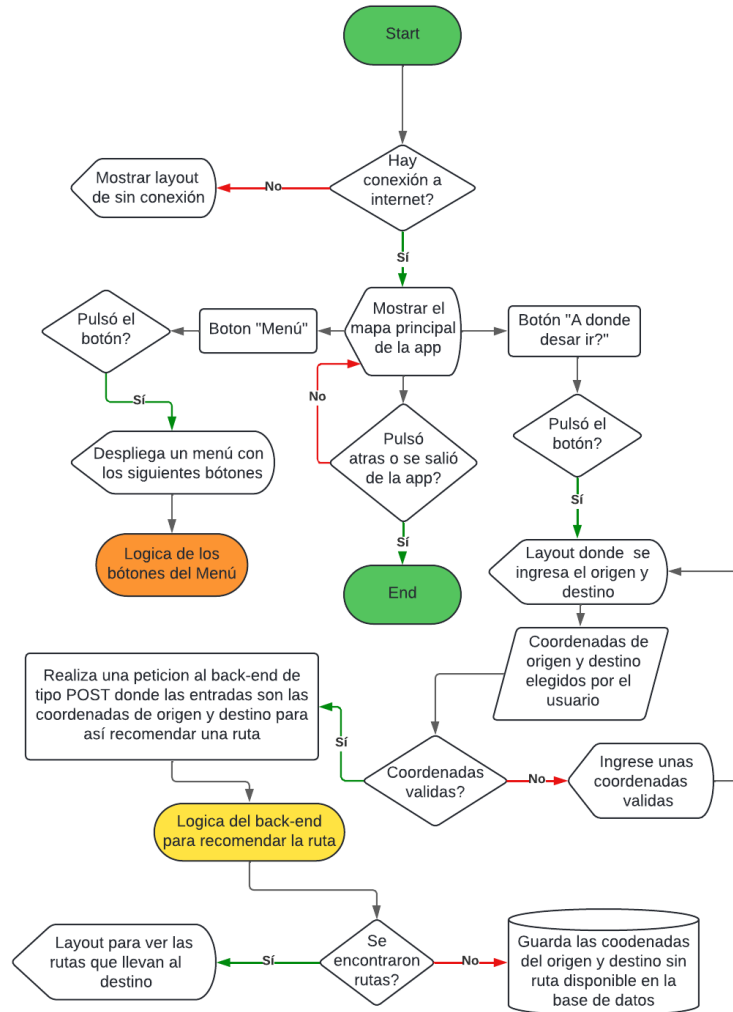


Figura 1. Diagrama de flujo front-end general. Autores

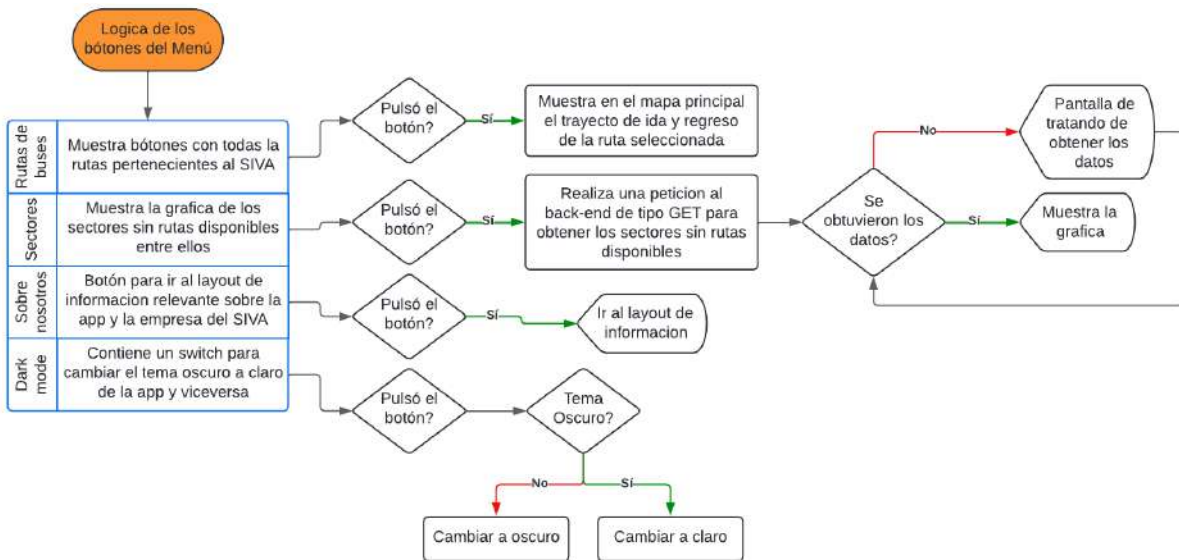


Figura 2. Diagrama de flujo de botones del menú. Autores

Su funcionamiento visual se puede observar a continuación mediante imágenes reales de la aplicación:



Figura 3. Pantalla al pulsar abrir app. Autores



Figura 4. Pantalla principal. Autores

Luego de que abre la aplicación, si pulsamos a desplegar menú:

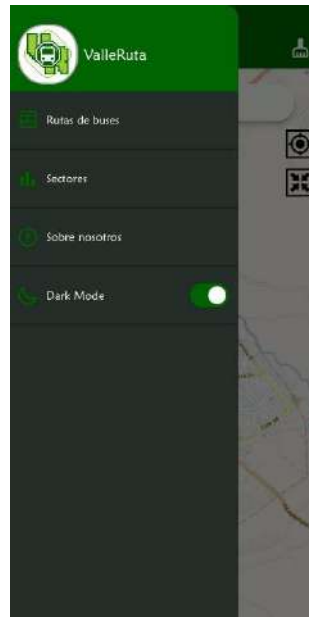


Figura 5. Menú desplegado (dark). Autores

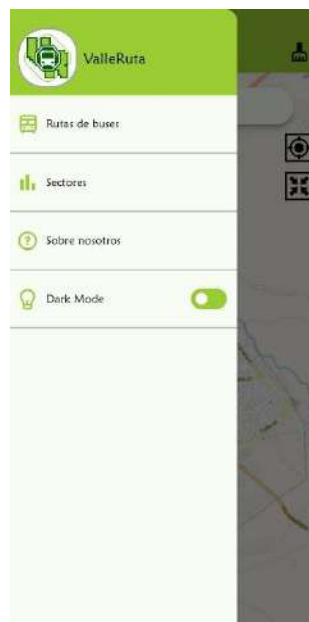


Figura 6. Menú desplegado. Autores

Si seleccionamos la opción de “Rutas de buses”:



Figura 7. Pantalla de lista de rutas. Autores

Si seleccionamos la opción de “Sectores”:



Figura 8. Pantalla de sectores en mapa. Autores

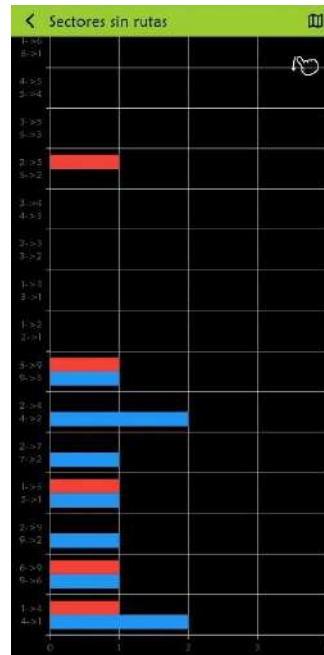


Figura 9. Gráfica estadística de relación de sectores sin rutas disponibles. Autores

Pantalla del buscador para recomendar rutas:



Figura 10. Pantalla de buscador. Autores

Realizamos un ejemplo para observar las diferentes pantallas al utilizar la app cuando hace búsqueda de recomendación de rutas:



Figura 11. Direcciones digitadas (1). Autores



Figura 12. Pantalla de carga (1). Autores



Figura 13. Recomendaciones (1). Autores

El usuario puede seleccionar las diferentes opciones de rutas recomendadas para tomar la mejor opción basado en su recorrido y el tiempo:



Figura 14. Recorrido de origen a destino de ruta seleccionada. Autores

Ahora realizamos un segundo ejemplo con una recomendación que involucra transbordo:



Figura 15. Direcciones digitadas (2). Autores



Figura 16. Pantalla de carga (2). Autores



Figura 17. Recomendaciones (2). Autores

Al seleccionar cualquiera de las opciones, le muestra el recorrido de ambas rutas, donde bajarse y subirse a la otra para un transbordo eficiente y que lo lleve al destino deseado.



Figura 18. Recorrido de origen a destino con transbordo de ruta. Autores

8.3 Desarrollo del código back-end

Una vez que se desarrolló la interfaz (front-end) de la aplicación, por consiguiente, se procedió a desarrollar el back-end, el cual se encarga de procesar los datos, conectar la base de datos con la aplicación y calcular las mejores rutas para recomendar al usuario.

Se definen las características que debe tener este código, las cuales son las siguientes:

- Recibe datos de entrada (coordenadas de origen y destino) desde la aplicación y encuentra las rutas más cercanas que pasen por dichos puntos.
- Busca si hay una ruta que pase por ambos puntos y si sí, la envía al front-end para ser recomendada.
- Si no hay ruta que pase por ambos puntos, compara una por una y encuentra puntos de intersección para recomendar transbordos.
- Si no encuentra rutas o intersecciones para transbordos, envía a la base de datos dicha información para ser almacenada y recopilada.
- El back-end accede a la base de datos donde se encuentran almacenados los puntos sin ruta disponible y encuentra a que sector pertenecen para mostrarlos gráficamente en la app.

Una vez se definieron las características y requisitos que debe tener la capa oculta de la aplicación, comienza la etapa de desarrollo del back-end, el cual se desarrolló en lenguaje de programación Python, el cual es un lenguaje de alto nivel ampliamente utilizado en la actualidad.

A continuación, se muestra el diagrama de flujo del back-end:

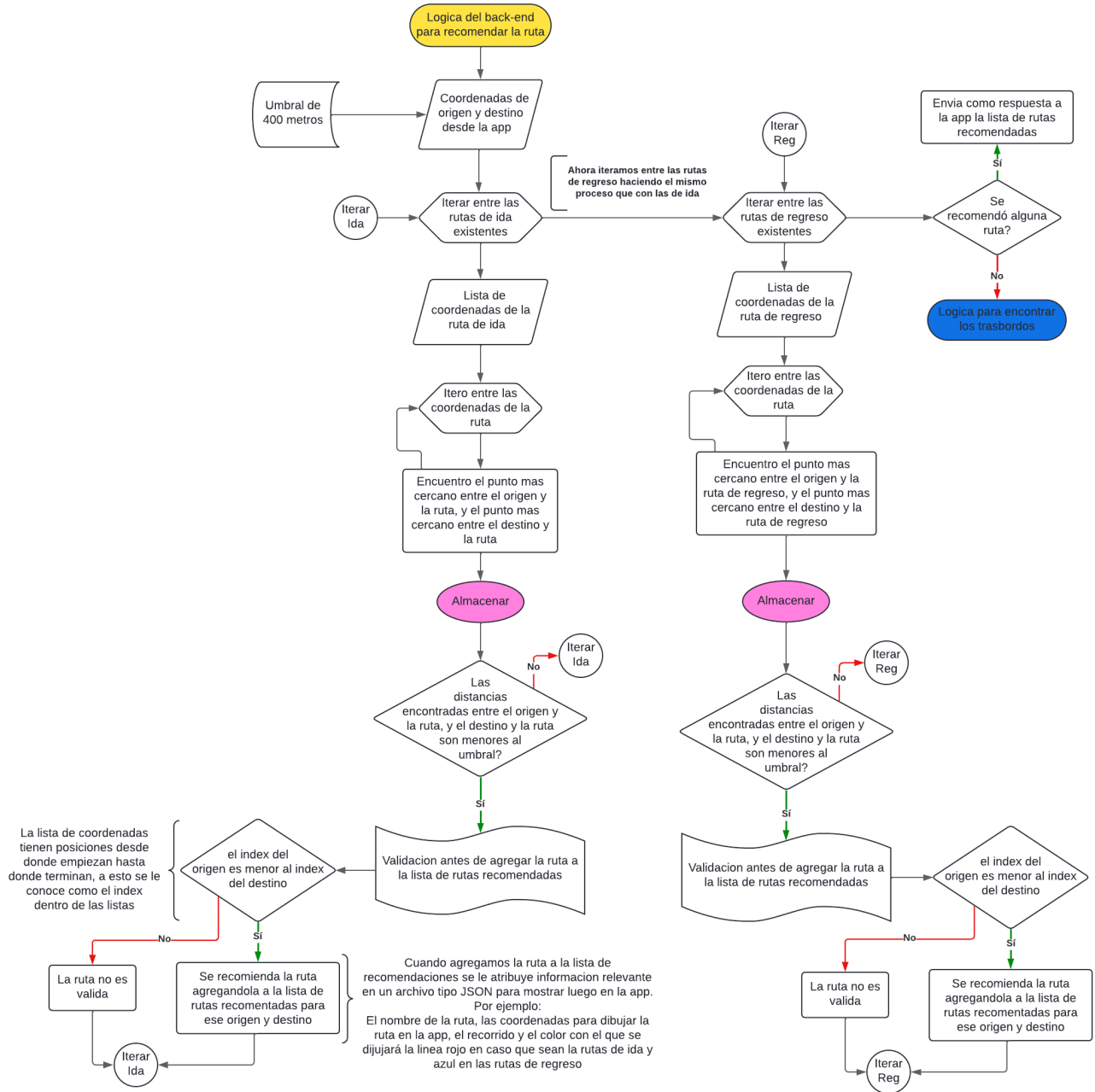


Figura 19. Diagrama back-end general. Autores

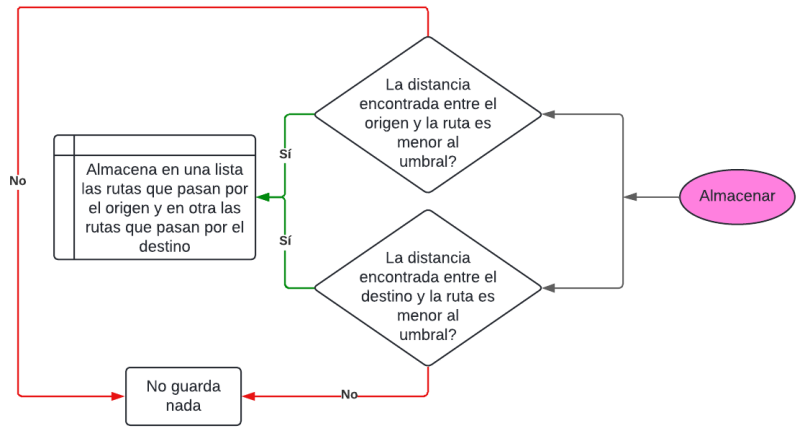


Figura 20. Diagrama de flujo de almacenamiento interno. Autores

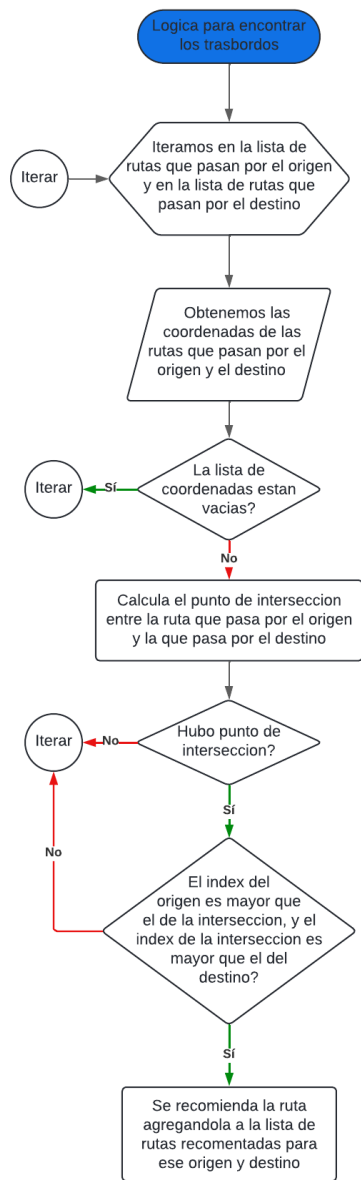


Figura 21. Diagrama de flujo para trasbordos. Autores

Una vez terminado el código back-end de debe proceder a desplegar en un servidor, para que la aplicación se pueda conectar a este mediante solicitudes http. Para ello se utilizó la herramienta “Render”, la cual ofrece alojamiento de servidores con planes gratuitos con ciertas limitaciones, pero que cuenta con lo necesario para poder desplegar el código del back-end.

Es importante tener en cuenta que una de las limitaciones de esta herramienta es que el servidor se apaga si no recibe solicitudes durante un determinado tiempo. Cuando llega una nueva solicitud de la aplicación, el servidor vuelve a activarse, lo cual toma un tiempo relativamente largo, aproximadamente un minuto.

8.4 OPTIMIZACIÓN Y MEJORAS

8.4.1 Pruebas para detectar errores o puntos a mejorar

Luego de tener la primera versión funcional de la app (ValleRuta), el siguiente paso fue realizar pruebas mediante el uso como un usuario real del día a día, se realizaron varios desplazamientos a lo largo de la ciudad en el sistema de transporte urbano de SIVA S.A.S, este trabajo se hizo durante 6 días, realizando 3 viajes diferentes por día, para un total de 18 desplazamientos en la prueba de la app.

De los 18 viajes que se realizaron, en 7 de ellos no se obtuvo la mejor recomendación, ya que recomendaba la ruta más rápida, pero guiando al usuario a un lugar donde debe caminar varias cuadras que se tardó de 14 a 20 minutos en recorrer para llegar al destino deseado en la prueba.

Siendo así que:

Porcentaje de error

$$\frac{7}{18} * 100\% = 38,9\%$$

Porcentaje de eficacia

$$100\% - 38,9\% = 61,1\%$$

Una vez detectamos el anterior punto a mejorar, nos percatamos que a pesar de ser un porcentaje de eficacia bajo y un porcentaje de error alto, la solución podría ser relativamente sencilla. Ya que analizando la lógica impartida para el desarrollo y revisando el código de recomendación de rutas (back-end), nos percatamos que teníamos un umbral que convertido a distancia era muy alto y esto ocasionaba que la app recomendara rutas retiradas de los puntos de origen y destino, haciendo que el usuario debiera caminar por varios minutos y cuadras.

Decidimos reducir el umbral en un 50% en el código, con el fin que no se recomendaran rutas donde el usuario debiera caminar más de 10min. Luego se realizaron nuevamente 10 viajes de prueba, donde se repitieron los 7 que no dieron la recomendación conforme en la prueba anterior y 3 viajes más diferentes a los de la primera prueba.

De la anterior y segunda prueba, se experimentó que, en los 10 viajes, la aplicación recomendó rutas que llevan al usuario a un destino donde debe caminar un máximo de 8min para llegar al lugar deseado.

8.4.2 Prueba de la aplicación por personas aparte de los autores

Una vez se realizaron pruebas y mejoras a la app por parte nuestra, procedimos a distribuirla a cinco personas interesadas en contribuir al desarrollo de esta. Les ensañamos a utilizar la aplicación y solicitamos que la usaran lo más que pudieran durante una semana y estos fueron los resultados:

Usuario 1:

La utilizó 4 veces en una semana, manifiesta que en todos los viajes la aplicación le proporcionó recomendaciones correctas.

$$Eficacia = 100\%$$

Del 0 al 100 el usuario califica con un 85 y aporta que la app tarda en mostrar las recomendaciones.

Usuario 2:

La utilizó 9 veces en una semana, manifiesta que en 7 de los viajes la aplicación le proporcionó recomendaciones correctas y en los otros 2 viajes no había rutas disponibles.

Se realizó una búsqueda por todas las rutas para encontrar una que pudiera llevar al usuario 2 de su origen al destino deseado y no se encontró alguna, por lo tanto, la app se comportó como se esperaba y guardó en la base de datos estos sectores sin rutas disponibles.

$$Eficacia = 100\%$$

Del 0 al 100 el usuario califica con un 70 porque no le dio recomendaciones en 2 viajes.

Usuario 3:

La utilizó 7 veces en una semana, manifiesta que en todos los 7 viajes la aplicación le proporcionó recomendaciones correctas.

$$Eficacia = 100\%$$

Del 0 al 100 el usuario califica con un 100 y opina que con más presupuesto se podría desarrollar una app para varias ciudades más.

Usuario 4:

La utilizó 11 veces en una semana, manifiesta que en todos los viajes la aplicación le recomendó las rutas adecuadas.

$$Eficacia = 100\%$$

Del 0 al 100 el usuario califica con un 90 la aplicación y avisa que la app demora en cargar las rutas recomendadas.

Usuario 5:

La utilizó 7 veces en la semana, manifiesta que en 6 de los viajes le dio la recomendación correcta y en el otro viaje tuvo que caminar alrededor de 12 minutos.

$$Eficacia = \frac{6}{7} 100\% = 86\%$$

Del 0 al 100 el usuario califica con un 95 la aplicación y dice que debería hacer recomendaciones donde el usuario no deba caminar.

En general, con la experiencia, opiniones y calificación de los usuarios que utilizaron la beta de la app se puede concluir los siguientes datos:

$$Eficacia_{promedio} = \frac{100\% + 100\% + 100\% + 100\% + 86\%}{5} = 97\%$$

$$Calificación_{promedio} = \frac{85 + 70 + 100 + 90 + 95}{5} = 88$$

9 RESULTADOS

Para analizar la eficiencia del producto final, procedemos a compararla con unas aplicaciones creadas con el mismo fin, analizando diferentes parámetros como lo son el tiempo de la ruta recomendada, transbordos, respuesta del servidor, entre otros.

Se realizó una misma búsqueda desde un origen hacia un destino en las tres aplicaciones diferentes (ValleRuta, SIVAlliu y Moovit).

Las búsquedas que se utilizaron para la comparación son las siguientes:

1. Universidad Popular del Cesar -> Residencia (Garupal).
2. Residencia (Garupal) -> Universidad Popular del Cesar.
3. Residencia (Garupal) -> Centro comercial guatapuri.
4. Centro comercial guatapuri -> Residencia (Garupal).
5. Barrio don Alberto -> Residencia (Garupal).
6. Residencia (Garupal) -> Barrio don Alberto.
7. Barrio don Alberto -> Centro comercial mayales.
8. Centro comercial mayales -> Barrio don Alberto.
9. Barrio don Alberto -> Barrio don Carmelo.
10. Barrio don Carmelo -> Barrio don Alberto.
11. Terminal de transporte -> Barrio don Alberto.
12. Barrio don Alberto -> Terminal de transporte.

Los resultados obtenidos de la aplicación ValleRuta fueron los siguientes:

ValleRuta						
Consultas	N° Rutas recomendadas	Tiempo promedio de Rutas recomendadas (min)	N° Trasbordos	Mayor tiempo (min)	Menor tiempo (min)	Respuesta del servidor (seg)
1	1	14	0	14	14	10,16
2	1	14	0	14	14	10,5
3	1	34	0	34	34	10,55
4	1	32	0	32	32	10,5
5	1	18	0	18	18	10,83
6	1	18	0	18	18	10,44
7	5	37	5	43	33	23,5
8	3	32,6	3	34	32	18,29
9	2	39	2	44	34	13,63
10	2	35,5	2	39	32	15,24
11	5	41,4	5	46	36	20,5
12	5	39,8	5	48	37	20,33

Tabla 1. Resultados de aplicación ValleRuta. Autores

Los resultados obtenidos de la aplicación SIVAlliu fueron los siguientes:

SIVAlliu						
Consultas	N° Rutas recomendadas	Tiempo promedio de Rutas recomendadas (min)	N° Trasbordos	Mayor tiempo (min)	Menor tiempo (min)	Respuesta del servidor (seg)
1	1	13	0	13	13	4,55
2	3	13	2	13	13	4,8
3	3	23,6	2	35	18	4,74
4	1	17	0	17	17	4,54
5	2	20,5	1	22	19	4,8
6	2	20,5	1	21	20	4,66
7	3	34	3	35	33	5,06
8	3	34	3	37	31	4,82
9	2	42	2	46	38	5,08
10	2	37,5	2	41	34	4,6
11	3	39	3	42	37	4,93
12	3	41,3	3	43	40	4,48

Tabla 2. Resultados de aplicación SIVAlliu. Autores

Los resultados obtenidos de la aplicación Moovit fueron los siguientes:

Moovit						
Consultas	N° Rutas recomendadas	Tiempo promedio de Rutas recomendadas (min)	N° Trasbordos	Mayor tiempo (min)	Menor tiempo (min)	Respuesta del servidor (seg)
1	2	17,5	1	20	15	0,5
2	3	19,3	0	24	15	0,7
3	3	28,3	1	33	21	0,9
4	3	24,3	1	33	20	0,5
5	1	25	0	25	25	1,52
6	3	26	1	31	20	0,5
7	4	42,25	3	47	38	1,91
8	4	39,25	3	42	36	1,5
9	3	51	3	53	48	0,7
10	3	51,6	3	51	52	0,5
11	3	46,3	3	47	46	1,7
12	3	51,6	3	58	45	1,2

Tabla 3. Resultados de aplicación Moovit. Autores

Para analizar los resultados más cómodamente, se grafican los diferentes datos para poder comparar las aplicaciones:

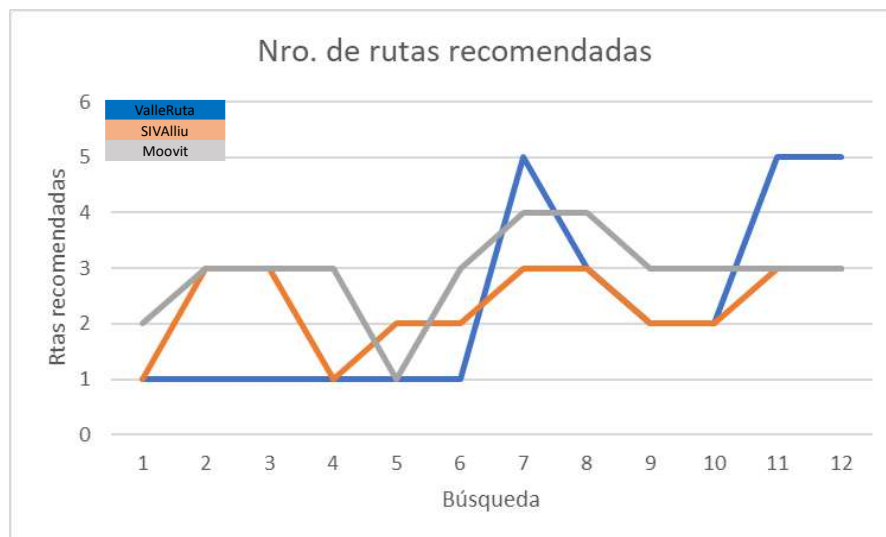


Figura 22. Gráfica comparativa de rutas recomendadas. Autores

En las rutas recomendadas vemos que hay una amplia diferencia entre las tres aplicaciones, siendo muy variable el delta entre los resultados y observando que ValleRuta es la que menos rutas sugiere. Esto se debe a que las demás aplicaciones sugieren transbordos, mientras que ValleRuta prioriza las recomendaciones donde el usuario va directo en una sola ruta, sin necesidad de cambiar de bus ni ruta.

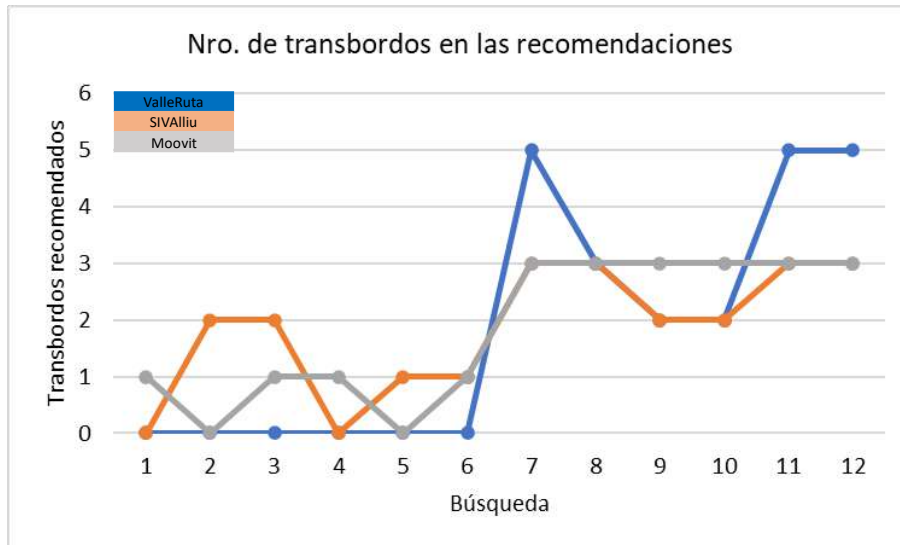


Figura 23. Gráfica comparativa de transbordos recomendados. Autores

Podemos observar que en la mayoría de las búsquedas, la aplicación ValleRuta es la que sugiere menos transbordos. Esto se traduce en un ahorro para el usuario, tanto en costos de pasaje como en tiempo, ya que una sola ruta lo llevará a su destino.

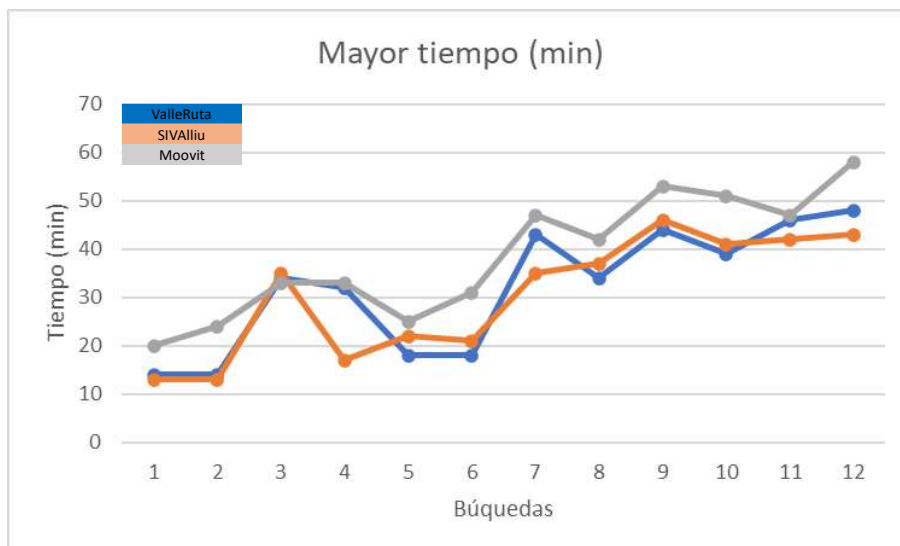


Figura 24. Gráfica comparativa de mayor tiempo de rutas recomendadas. Autores

En la comparación de los tiempos máximos de demora del recorrido observamos que no es mucha la diferencia entra las tres aplicaciones, siendo Moovit la que mayor tiempo le sugiere al usuario y entre ValleRuta y SIVAlliu sugieren casi el mismo tiempo máximo de recorrido.

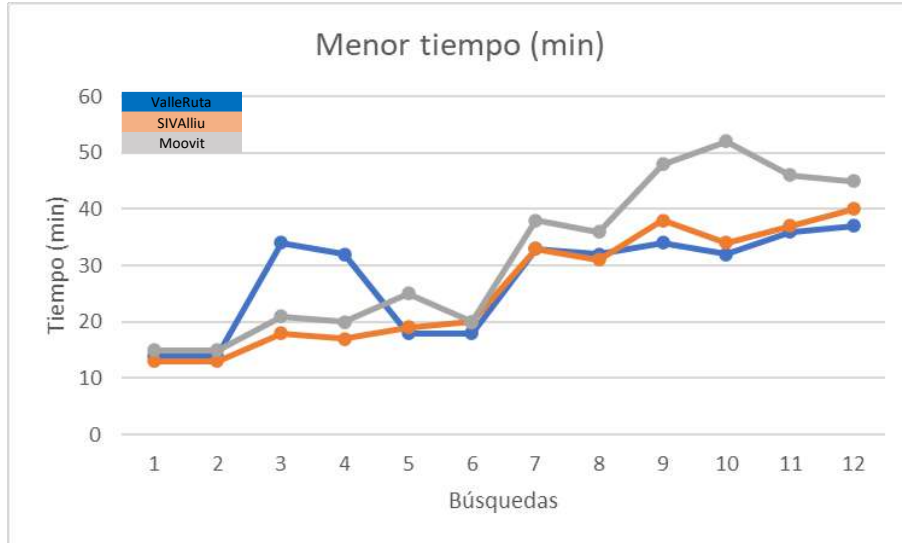


Figura 25. Gráfica comparativa de menor tiempo de rutas recomendadas. Autores

En el tiempo mínimo de recorrido de las rutas recomendadas se observa el mismo comportamiento en los datos resultados de las tres aplicaciones, siendo nuevamente Moovit la que recomienda un recorrido con mayor tiempo y ValleRuita junto a SIVAlliu siendo muy similares en las recomendaciones.

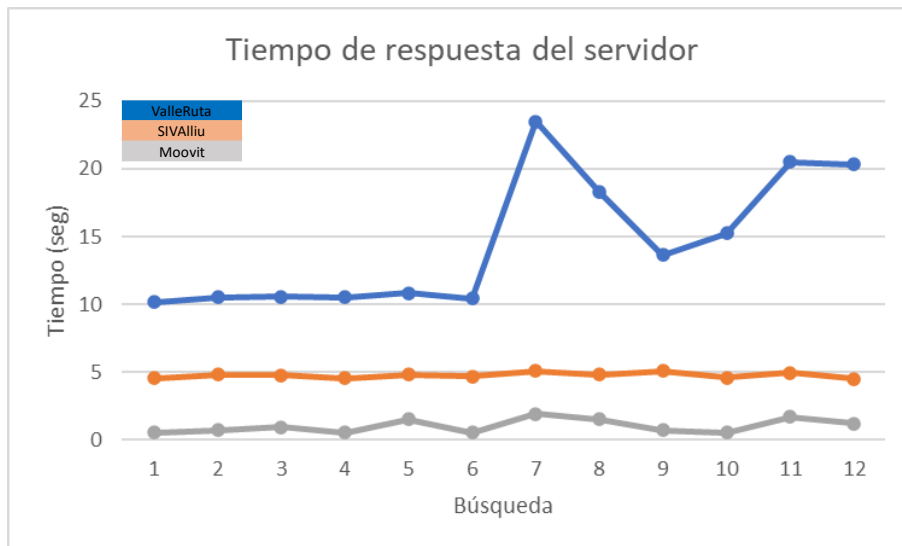


Figura 26. Gráfica comparativa de tiempo de respuesta del servidor. Autores

Sin duda, esta gráfica comparativa es la más estable y donde más diferencia se aprecia entre las aplicaciones, siendo Moovit la más destacada por su rapidez a la

hora de mostrar los resultados, siendo esta propiedad de Intel, quienes cuentan con gran presupuesto e infraestructura para poner a disposición servidores potentes. SIVAlliu no es tan rápida como Moovit, sin embargo, tiene una velocidad de respuesta aceptable para lo que requiere el aplicativo móvil. Y ValleRuta se con un tiempo de respuesta muy alto, siendo considerado ya como una aplicación lenta, esto a causa del servidor donde se despliega el back-end el cual es gratuito y por lo tanto no tan potente.

Al darle uso a las tres aplicaciones al mismo tiempo, se notó que a SIVAlliu le faltan varios puntos a mejorar sobre todo en los transbordos, ya que no indica de una manera clara al usuario donde debe hacer dicho transbordo. A continuación, se muestra el ejemplo y contraste con ValleRuta:



Figura 27. Transbordo app SIVAlliu. SIVAlliu

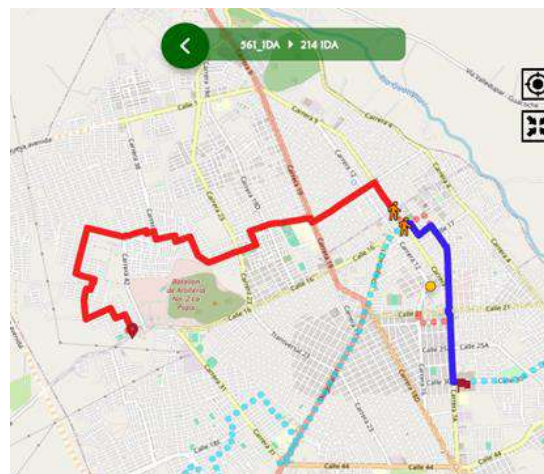


Figura 28. Transbordo app ValleRuta. Autores

Como se observa en la comparación anterior, SIVAlliu colapsa la interfaz de la app con mucha información de las rutas y hace poco entendible el recorrido que debe

hacer el usuario tanto en la ruta como al bajarse y transbordar. A diferencia de ValleRuta, el cual muestra la información más limpia, específica y optimizada para que el usuario entienda sin mayor esfuerzo.

10 CONCLUSIONES

A lo largo del desarrollo de la aplicación se observó diferentes puntos importantes en las diferentes fases, donde se pudieron corregir algunas falencias, implementar mejoras, detectar aspectos a mejorar en un futuro, tener retroalimentación de usuarios reales, calificación y opiniones de los mismos y entre otros puntos importantes que se lograron concluir son tales como:

- La idea de esta iniciativa es bien tomada y percibida como necesaria a gran escala tanto para ciudadanos del común, como para empresas operadoras y prestadoras del servicio urbano de la ciudad de Valledupar.
- La recopilación y tratamiento de datos es fundamental al momento de desarrollar un software que abarque un gran número de personas, en nuestro caso recopilando sugerencias de posibles rutas nuevas con el fin de facilitar el desarrollo del sistema de transporte público urbano.
- Durante la fase de pruebas y detección de errores se pudo notar que cuando el margen de distancia es mayor se pueden obtener más recomendaciones, pero el usuario deberá caminar más en algunas de estas opciones. Y a menor margen de distancia existirán menos sugerencias, pero estas serán más precisas para el usuario ya que estarán más cerca tanto al origen como al destino.
- Otras aplicaciones desarrolladas a nivel general y no específico para una ciudad, tiene mejor tiempo de respuesta, pero recomienda rutas con un tiempo mayor de recorrido y más lejanas del usuario.
- En las pruebas realizadas por diferentes usuarios durante una semana se logró obtener una eficacia del 97% de la aplicación, asimismo se obtuvo una calificación de satisfacción de 88 puntos de 100.
- Esta iniciativa contribuye al desarrollo tecnológico relacionado con el transporte, no solo haciéndolo más eficiente, sino también incentivar a la ciudadanía al uso de este sistema, aportando indirectamente a la economía y cuidado del medio ambiente.
- Para el desarrollo de una aplicación móvil no solo se debe considerar el hardware del equipo del usuario, sino también una infraestructura robusta de hardware capaz de gestionar eficientemente la cantidad de datos necesarios para un buen análisis de datos. Además, la disponibilidad de un servidor potente y dedicado mejora considerablemente la velocidad de respuesta de la aplicación.

- Se debe contar con un servicio de almacenamiento en la nube adecuado para almacenar de manera segura los datos recopilados de cada dispositivo de los usuarios.

11 RECOMENDACIONES

- En un futuro integrar más información, tales como los horarios de salida, paraderos, horario de servicio, etc. Los cuales no están fijados por el ente operador.
- Implementar la opción de que el usuario determine el margen de distancia que desea caminar, para mejorar la satisfacción de los ciudadanos, ya que algunos prefieren caminar más o caminar menos, dependiendo del estado físico, la preferencia y la ubicación del usuario.
- Incluir en el recorrido de las rutas sitios de interés para turistas, tales como hoteles, restaurantes y sitios turísticos de la ciudad.
- Contratar un servidor más potente para desplegar el back-end de la aplicación, esto permitiría un tiempo de respuesta más bajo y mayor satisfacción del usuario.
- Implementar dispositivos GPS en los vehículos tipo bus de la flota operativa y relacionarlos en la aplicación, para que el usuario pueda ver en tiempo real donde se encuentra el bus con la ruta que requiere para llegar a su destino.

12 COSTOS Y RECURSOS GASTADOS

Para estimar el costo del proyecto se debe tener en cuenta los recursos necesarios para el desarrollo del mismo. A continuación, en la tabla 2 se puede observar la estimación de costos.

Descripción		Cant.	Costo Unitario	Costo total
Materiales	Computador con al menos 8GB de RAM y windows 7 o superior	2	1.900.000,00 COP	3.800.000,00 COP
	Teléfono móvil con android 8 o superior	2	800.000,00 COP	1.600.000,00 COP
Mano de obra y servicios	Estudiantes de Ing. electrónica	2	1.000.000,00 COP	2.000.000,00 COP
	Transporte para pruebas de la App	10	4.000,00 COP	40.000,00 COP
Marketing	Lanzamiento de la App	1	300.000,00 COP	300.000,00 COP
Suma total				7.740.000,00 COP
Imprevistos (+20%)				1.548.000,00 COP
TOTAL				9.288.000,00 COP

Tabla 4. Análisis de costos y recursos. Autores

13 REFERENCIAS BIBLIOGRÁFICAS

- [1] "Sistema integrado de transporte de Valledupar - SIVA". SIVA. <https://siva.gov.co/>
- [2] "¿Cuándo entrarían en funcionamiento los nuevos buses en Valledupar? | Semanario La Calle". Semanario La Calle. <https://semanariolacalle.com/cuando-entrarian-en-funcionamiento-los-nuevos-buses-en-valledupar/>
- [3] "Q'ruta". Q'ruta. <https://m.queruta.com/>.
- [4] Ó. H. Cerquera Losada, C. F. Orjuela Yacue y S. E. Ferrari. "Mototaxismo y hurto a personas: Evaluación de impacto a partir del modelo de diferencias en diferencias*". Sistema de Información Científica Redalyc, Red de Revistas Científicas. <https://doi.org/10.18359/rfce.3788>
- [5] J. A. Valencia Basto, J. A. Molina Ruano, "Diseño de un aplicativo móvil para consultar las rutas de buses y sus recorridos dentro del AMCO", resumen extendido de trabajo de grado, Univ. Tecnol. Pereira, Pereira, 2019.
- [6] R. Fulgêncio, M. C. Ferreira, D. Abrantes, y M. Coimbra, «Restart: A Route Planner to Encourage the Use of Public Transport Services in a Pandemic Context», Transportation Research Procedia, vol. 62, pp. 123-130, 2022, doi: 10.1016/j.trpro.2022.02.016.
- [7] Google LLC. "Flutter - Build apps for any screen". Flutter - Build apps for any screen. Accedido el 7 de julio de 2023. Disponible: <https://flutter.dev/>
- [8] V. Diví. "¿Qué es el lenguaje de programación Dart?" inLab FIB. Accedido el 7 de julio de 2023. Disponible: <https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart>
- [9] Python Software Foundation. "Welcome to Python.org". Python.org. Disponible: <https://www.python.org/>
- [10] DesarrolloWeb. "JSON". Home de DesarrolloWeb.com. Disponible: <https://desarrolloweb.com/home/json>
- [11] Nominatim developer community. "Overview - Nominatim Manual". Nominatim. [En línea]. Disponible: <https://nominatim.org/release-docs/develop/api/Overview/>
- [12] M. T. Ashraf, K. Dey, A. Carrola, y X. Shi, «Impacts of real-time transit information on transit accessibility – A case study», J Public Trans, vol. 25, p. 100043, 2023, doi: 10.1016/j.jpubtr.2023.100043.

- [13] M. Strenitzerova y K. Stalmachova, «Customer requirements for urban public transport mobile application», *Transportation Research Procedia*, vol. 55, pp. 95-102, 2021, doi: 10.1016/j.trpro.2021.06.010.
- [14] I. Storesund Hesjevoll, A. Fyhri, y A. Ciccone, «App-based automatic collection of travel behaviour: A field study comparison with self-reported behaviour», *Transp Res Interdiscip Perspect*, vol. 12, p. 100501, 2021, doi: 10.1016/j.trip.2021.100501.
- [15] C. F. Libaque-Sáenz, S. F. Wong, Y. Chang, y E. R. Bravo, «The effect of Fair information practices and data collection methods on privacy-related behaviors: A study of Mobile apps», *Information & Management*, vol. 58, n.o 1, p. 103284, 2021, doi: 10.1016/j.im.2020.103284.
- [16] “FastAPI”. FastAPI. Disponible: <https://fastapi.tiangolo.com/es/>
- [17] Tutorial de KML. (2019). Keyhole Markup Language. Obtenido de la web: https://developers.google.com/kml/documentation/kml_tut?hl=es-419
- [18] Librería GSON. Github (2019). obtenido de la web: <https://github.com/google/gson>
- [19] MoviliXa. Aplicaciones móviles para optimizar el uso del transporte público. Obtenido de MoviliXa web: <http://movilixa.com/>
- [20] Moovit. La aplicación de transporte público # 1 en el mundo. Obtenido de Moovit web: <https://company.moovit.com/>
- [21] Heiss, Janice. The Advent of Kotlin: A Conversation with JetBrains’ Andrey Breslav. Oracle Technology Network. Obtenido de Oracle web: <https://www.oracle.com/technical-resources/articles/java/breslav.html>
- [22] Raygoza G. David. Convertir objetos Java a JSON y de regreso. Obtenido de la web: <https://medium.com/el-acordeon-del-programador/convertir-objetos-java-a-json-y-de-regreso-1077500d78f7>
- [23] “Definición y cómo funcionan las aplicaciones móviles”. Servicios Softcorp CA. [En línea]. Disponible: <https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>
- [24] OpenStreetMap. Disponible: <https://www.openstreetmap.org/>
- [25] “Bases de datos no relacionales | Bases de datos de gráficos | AWS”. Amazon Web Services, Inc. [En línea]. Disponible: <https://aws.amazon.com/es/nosql/>

- [26] Castañeda María Cabrera Alberto Navarro Yadira de Vries Wietse, Procesamiento de datos y análisis estadísticos utilizando SPSS: Un libro practico para investigadores y administradores educativos. Rio: EDIPUCRS, 2010.