

**Herramienta de consulta para la movilidad en transporte público en Neiva / Huila**

Walter Alejandro González Silva

Asesor

Jaime Rubiano Llorente

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería - ECBTI

Programa de Ingeniería de Sistemas

2025

## Tabla de Contenidos

Resumen.....	13
Abstract.....	14
Introducción .....	15
Planteamiento del Problema .....	17
Árbol Causa – Efecto del Problema.....	20
Justificación .....	21
Objetivos.....	23
Objetivo General.....	23
Objetivos Específicos.....	23
Marco Referencial.....	24
Marco Teórico:.....	24
Estado del Arte.....	24
Movilidad y Transporte Público en Entornos Urbanos .....	25
Movilidad Sostenible y sus Enfoques.....	26
Sostenibilidad y Transporte Inteligente .....	26
Marco Conceptual:.....	27
Scrum .....	27
Escala de Likert .....	27
Marco Legal:.....	28
Marco Tecnológico: .....	29
Metodología .....	30

	3
Análisis y diagnóstico del proceso investigativo .....	31
Metodología de desarrollo de software.....	31
Delimitaciones del Proyecto .....	36
Delimitación Geográfica.....	36
Delimitación Temporal .....	36
Delimitación Tecnológica.....	36
Delimitación de Usuarios.....	36
Delimitación Funcional.....	36
Análisis de Requerimientos .....	37
Requerimientos Funcionales:.....	37
Requerimientos no Funcionales:.....	37
Requerimientos del Hardware y Software: .....	38
Diagramas de Casos de Uso.....	39
Diagrama de Actividades.....	40
Diagrama de Clases.....	41
Diagrama de Secuencia.....	42
Cronograma de Actividades.....	43
Recursos Necesarios para la Implementación.....	46
Diseño de la Solución .....	48
Arquitectura del Sistema.....	48
Módulos del Sistema.....	48
Interfaz de Usuario (UI).....	49

	4
Flujo de Trabajo.....	49
Consideraciones de Seguridad.....	50
Implementación.....	51
Diseño de Interfaz de Usuario.....	51
Mockup MoviNeiva.....	53
Wireframe MoviNeiva.....	54
Desarrollo Prototipo Funcional del Aplicativo.....	55
Estructura y módulos.....	55
Servicios.....	55
Tipos de archivos manejados.....	56
Integración y UX.....	56
Interfaz y Navegación.....	57
Filtros dinámicos y comunicación entre componentes.....	58
Modales y formularios.....	60
Adaptabilidad y responsive design.....	62
Accesibilidad y simplicidad.....	63
Pruebas.....	64
Despliegue automático del instructivo.....	64
Pruebas del módulo de filtros.....	65
Evaluación con Usuarios.....	70
Objetivo de la Evaluación.....	70
Método de Evaluación.....	70

	5
Criterios de selección.....	70
Margen de Error.....	70
Resultados principales .....	71
Análisis de resultados por pregunta .....	71
Dificultades Encontradas por los Usuarios en el Primer Uso .....	84
Conclusiones.....	86
Referencias.....	88
Anexos .....	92
Anexo A - Manual técnico herramienta de consulta para la movilidad en transporte público en Neiva / Huila .....	92
Metodología de desarrollo .....	93
Roles Scrum.....	93
Estructura de los Sprints .....	93
Ceremonias Scrum Realizadas.....	93
Artefactos Scrum .....	94
Metodologías y Tecnologías Utilizadas.....	95
Framework y Estructura de Aplicación .....	95
Gestión Reactiva .....	95
Visualización Geoespacial .....	95
Análisis Geoespacial.....	95
Servicios.....	97
MapService .....	97

Principales Métodos y Funciones .....	97
NeighborhoodService .....	101
Métodos y Funciones .....	102
Archivo GeoJSON referenciado .....	102
RoutesService .....	102
Propiedades y Métodos .....	103
Archivos GeoJSON referenciados .....	105
Componentes .....	106
HeaderComponent .....	106
Análisis Visual y Funcional .....	106
MapComponent .....	106
Estructura del Componente .....	107
Propiedades .....	107
Métodos .....	108
Servicios utilizados .....	109
FilterComponent .....	110
Estructura del componente .....	110
Propiedades .....	111
Métodos .....	113
Eventos Emitidos al MapComponent .....	114
Servicios Utilizados .....	115
FooterComponent .....	115

Estructura del Componente.....	115
Dependencias .....	116
Funcionalidad Principal .....	116
Comportamiento de los Modales .....	116
Detalles de Cada Modal.....	117
Despliegue aplicativo web a GitHub Pages .....	118
Requisitos previos.....	118
Implementación de cambios en el aplicativo .....	118
Trabajar sobre la rama de desarrollo (master) .....	118
Sincronizar cambios en la rama principal (main) .....	118
Construir la aplicación para producción .....	118
Instalar y usar angular-cli-ghpages .....	119
Configurar GitHub Pages.....	119
Formulario de retroalimentación.....	120
Integración del formulario de retroalimentación .....	120
Interpretación general de las respuestas.....	121
Media ( $\mu$ ) .....	121
Desviación estándar ( $\sigma$ ) .....	121
Varianza y varianza ponderada.....	121
Pruebas y Validación .....	124
Pruebas de Usabilidad con Usuarios.....	124
Resumen de métricas obtenidas .....	124

Pruebas Manuales de Funcionalidad.....	125
Conclusión General.....	126
Pautas de Mantenimiento y Control de Versiones .....	127
Esquema de versionado (SemVer).....	127
Generación de changelogs .....	128
Reporte y corrección de errores (GitHub Issues).....	129
Plantilla sugerida de errores.....	129
Plantilla de solicitud de mejora.....	130
Buenas prácticas para futuras actualizaciones .....	131

### Lista de tablas

<b>Tabla 1</b> Cronograma de actividades.....	43
<b>Tabla 2</b> Recursos necesarios para la implementación.....	46
<b>Tabla 3</b> Análisis de resultados pregunta enfocada al acceso y apertura de la aplicación	71
<b>Tabla 4</b> Análisis de resultados pregunta enfocada a la facilidad de uso del mapa y filtros .....	73
<b>Tabla 5</b> Análisis de resultados pregunta enfocada al uso del botón de seguimiento de ubicación.....	74
<b>Tabla 6</b> Análisis de resultados pregunta enfocada a la búsqueda de rutas.....	76
<b>Tabla 7</b> Análisis de resultados pregunta enfocada con la utilidad del instructivo .....	77
<b>Tabla 8</b> Análisis de resultados pregunta enfocada con la claridad de la información visual (rutas, paraderos, colores).....	79
<b>Tabla 9</b> Análisis de resultados pregunta enfocada con el tiempo de respuesta para una consulta de ruta .....	80
<b>Tabla 10</b> Análisis de resultados pregunta enfocada con los niveles de recomendación. .	82
<b>Tabla 11</b> Análisis de resultados pregunta enfocada con el nivel de satisfacción.....	83
<b>Tabla 12</b> Análisis visual HeaderComponent.....	106
<b>Tabla 13</b> Propiedades MapComponent .....	107
<b>Tabla 14</b> Eventos recibidos desde el componente FilterComponent .....	109
<b>Tabla 15</b> Propiedades del componente FilterComponent. ....	111
<b>Tabla 16</b> Eventos emitidos al MapComponent .....	114
<b>Tabla 17</b> Interpretación de los resultados por tipo de pregunta. ....	122

**Tabla 18** Resumen de las métricas obtenidas..... 124

### Lista de figuras

<b>Figura 1</b> Árbol de problema.....	20
<b>Figura 2</b> Diagrama de caso de uso .....	39
<b>Figura 3</b> Diagrama de actividades .....	40
<b>Figura 4</b> Diagrama de clases.....	41
<b>Figura 5</b> Diagrama de secuencia.....	42
<b>Figura 6</b> Mockup MoviNeiva .....	53
<b>Figura 7</b> Wireframe MoviNeiva .....	54
<b>Figura 8</b> Vista Prototipo funcional en ordenadores. ....	57
<b>Figura 9</b> Vista Prototipo funcional desde dispositivos móviles.....	58
<b>Figura 10</b> Tarjeta de filtros .....	59
<b>Figura 11</b> Modal instructivo de uso de la aplicación. ....	60
<b>Figura 12</b> Modal sobre MoviNeiva.....	61
<b>Figura 13</b> Modal formulario de opinión sobre MoviNeiva.....	62
<b>Figura 14</b> Despliegue instructivo MoviNeiva.....	64
<b>Figura 15</b> Seleccionar inicio según ubicación .....	65
<b>Figura 16</b> Filtrado de ubicaciones por comuna.....	66
<b>Figura 17</b> Selección de rutas.....	66
<b>Figura 18</b> Ruta directa seleccionando punto de inicio y destino. ....	67
<b>Figura 19</b> Ruta combinada seleccionando punto de inicio y destino.....	68
<b>Figura 20</b> Opción de seguimiento en tiempo real.....	68
<b>Figura 21</b> Grafico pregunta sobre acceso y apertura de la aplicación. ....	72

	12
<b>Figura 22</b> Grafico pregunta relacionada con la facilidad de uso del mapa y filtros .....	73
<b>Figura 23</b> Grafico pregunta relacionada con el uso del botón de seguimiento de ubicación .....	75
<b>Figura 24</b> Grafico pregunta relacionada con la búsqueda de rutas .....	76
<b>Figura 25</b> Grafico pregunta relacionada con la utilidad del instructivo .....	78
<b>Figura 26</b> Grafico relacionado con la claridad de la información visual.....	79
<b>Figura 27</b> Grafico relacionada con el tiempo de respuesta para una consulta de ruta.....	81
<b>Figura 28</b> Grafico relacionado a la satisfacción del usuario al encontrar una ruta. ....	82
<b>Figura 29</b> Grafico relacionado a la satisfacción del usuario.....	84
<b>Figura 30</b> Sprint Backlog.....	94
<b>Figura 31</b> Función initMap .....	97
<b>Figura 32</b> Función getUserLocation. ....	98
<b>Figura 33</b> Función toggleUserTracking. ....	99
<b>Figura 34</b> Función cargarParaderos. ....	100
<b>Figura 35</b> Función resaltarBarrios .....	101
<b>Figura 36</b> Propiedad rutas.....	103
<b>Figura 37</b> Función cargarRuta. ....	105
<b>Figura 38</b> Modal formulario de opinión sobre MoviNeiva.....	120
<b>Figura 39</b> Archivo package.json en el atributo “version” .....	127
<b>Figura 40</b> Archivo CHANGELOG.mb .....	128
<b>Figura 41</b> Plantilla de reportes de bugs en GitHub Issues .....	130
<b>Figura 42</b> Plantilla de reportes de sugerencias en GitHub Issues .....	131

## Resumen

El presente proyecto tiene como objetivo desarrollar una aplicación web responsiva que centralice la información sobre rutas y paraderos del transporte público en Neiva, con el fin de mejorar la accesibilidad y eficiencia de la movilidad urbana. La iniciativa responde a las problemáticas derivadas del crecimiento de la ciudad y la falta de herramientas tecnológicas que faciliten la consulta de rutas, situación que afecta principalmente a trabajadores y estudiantes. El desarrollo de la aplicación se llevará a cabo bajo la metodología ágil Scrum, estructurada en cuatro sprints que incluyen levantamiento de requerimientos, desarrollo de servicios, implementación del frontend, pruebas y ajustes. Los requerimientos funcionales consideran la consulta de rutas y paraderos, así como la visualización en mapas interactivos con tecnologías como Leaflet. Los requerimientos no funcionales priorizan la usabilidad, el rendimiento, la seguridad de datos personales y la mantenibilidad del sistema. Tecnológicamente se empleará Angular y Bootstrap para la creación de una interfaz web responsiva y accesible desde cualquier dispositivo, además del despliegue en GitHub Pages para asegurar la disponibilidad del sistema. Se espera que esta herramienta tecnológica contribuya a una movilidad urbana más eficiente, promoviendo el uso del transporte público mediante una plataforma confiable, accesible y centrada en el usuario.

**Palabras clave:** Movilidad, Transporte público, Aplicación web responsiva

### **Abstract**

This Project has by objective to develop a responsive web application that centralizes information on public transportation routes and stops in Neiva, aiming to improve accessibility and efficiency of urban mobility. This initiative addresses issues stemming from the city's growth and the lack of technological tools that facilitate routes consultation, a situation that primarily affects workers and students of the city who haven't have a particular automobile to commute.

The application will be developed using the agile Scrum methodology, structured into four sprints that include requirements gathering, service development, frontend implementation, testing, and adjustments. The functional requirements include the consultation of routes and stops, as well as their visualization on interactive maps using technologies such as Leaflet. The non-functional requirements prioritize usability, performance, personal data security, and system maintainability.

Technologically, Angular and Bootstrap will be used to create a responsive web interface accessible from any device, and the system will be deployed on GitHub Pages to ensure availability. This technological tool is expected to contribute to more efficient urban mobility by promoting the use of public transportation through a reliable, accessible, and user-centered platform.

**Keywords:** Mobility, Public Transportation, Responsive Web Application.

## **Introducción**

La movilidad en las ciudades es un desafío creciente a medida que los centros urbanos experimentan un aumento en su población y en la demanda de servicios de transporte. La ciudad de Neiva, como otras áreas urbanas de rápido crecimiento, enfrenta una serie de problemas relacionados con la eficiencia y accesibilidad del transporte público. Estos problemas no solo afectan en la calidad de vida de los ciudadanos, sino que también ayudan al aumento de la congestión vehicular y las emisiones contaminantes. En este caso sería muy bueno contar con herramientas que faciliten la movilidad y mejore el uso del transporte público ya que esto se ha vuelto una necesidad urgente.

El presente proyecto propone el desarrollo de una herramienta digital que permita a los usuarios del transporte público en Neiva acceder a información centralizada sobre rutas, paraderos y puntos de origen y destino de los autobuses. Actualmente, los ciudadanos deben soportar y acoplarse a información fragmentada y poco accesible, lo que dificulta la planificación eficiente de sus desplazamientos. La falta de un sistema unificado afecta negativamente la experiencia de los usuarios y desincentiva el uso del transporte público, lo que contribuye a un aumento en la dependencia de vehículos privados.

La importancia de este proyecto está en su potencial para mejorar la movilidad urbana y hacer más eficiente el transporte público, promoviendo su uso entre los ciudadanos. Al centralizar la información de las rutas de autobuses y hacerla accesible a través de una interfaz amigable, se busca facilitar el desplazamiento de distintos grupos

de la población, incluidos estudiantes, trabajadores y turistas. Además, la herramienta contribuirá al desarrollo de una ciudad más sostenible al reducir el tráfico vehicular y con ello las emisiones de gases contaminantes.

El desarrollo de esta herramienta no solo representa una solución técnica al problema de la descoordinación en el transporte público, sino que también tiene un impacto social y ambiental significativo.

## **Planteamiento del Problema**

El problema identificado en este proyecto es la falta de un sistema unificado y accesible que permita a los ciudadanos de Neiva consultar de manera efectiva las rutas y paraderos del transporte público. Actualmente, la movilidad en la ciudad se ve seriamente afectada por la descoordinación entre las diversas compañías de transporte que operan sin un sistema de información unificado visible y eficiente para los ciudadanos. Como resultado, los usuarios se enfrentan a la incertidumbre y la falta de información ya sea porque no conocen el sistema de información actual o porque este no supe sus necesidades, lo que genera confusión, pérdida de tiempo, y un impacto negativo en la experiencia general de viaje.

Según cifras recientes, el uso del transporte público en Neiva ha disminuido drásticamente, pasando de 38.030 usuarios diarios en 2023 a solo 32.000 en 2025. Esta caída del 15.85% refleja una creciente insatisfacción con el sistema actual (López, J. E. R. 2025), agravada por la falta de herramientas digitales accesibles y confiables. Aunque el SETP lanzó en 2024 la plataforma [www.setpmigo.com](http://www.setpmigo.com) para consultar rutas (Alcaldía de Neiva, 2024), su bajo nivel de adopción evidencia la necesidad de una solución más visible, intuitiva y funcional..

Los principales afectados por esta situación son los ciudadanos que dependen del transporte público para sus desplazamientos diarios. Entre ellos se encuentran estudiantes, trabajadores y personas de la tercera edad, así como turistas que visitan la ciudad y no están familiarizados con las rutas de los buses. Para estos grupos, la falta de

una herramienta visible que centralice y facilite la consulta de información sobre el transporte público es una barrera significativa que afecta su movilidad y calidad de vida.

Este problema no solo está ocurriendo en Neiva, la capital del departamento del Huila, esta es una ciudad que, al igual que muchas otras ciudades en el país no cuenta con un sistema de transporte público bien coordinado y moderno. Las rutas de autobuses están concentradas en un sistema común, sin embargo esta no es de fácil acceso y sus funcionalidades son limitadas. La infraestructura tecnológica también es limitada, lo que agrava la situación.

El problema de la movilidad en el transporte público de Neiva ocurre de manera constante, pero se intensifica durante las horas pico, cuando el uso del transporte público alcanza su máximo. Durante estos momentos, la falta de información clara sobre las rutas y los paraderos genera una mayor frustración en los usuarios, que a menudo llegan tarde a sus destinos u hogares debido a la dificultad de planificar adecuadamente sus trayectos.

Este problema ocurre debido a varios factores. En primer lugar, la falta de visualización del sistema de información actual y sus funcionalidades limitantes. Además, no existe una plataforma digital que integre la información de forma centralizada de las rutas, paraderos y barrios, lo que obliga a los usuarios a depender de fuentes informales, como preguntas a conductores o la observación directa en los paraderos. La falta de una infraestructura tecnológica adecuada también son factores que contribuyen a este problema.

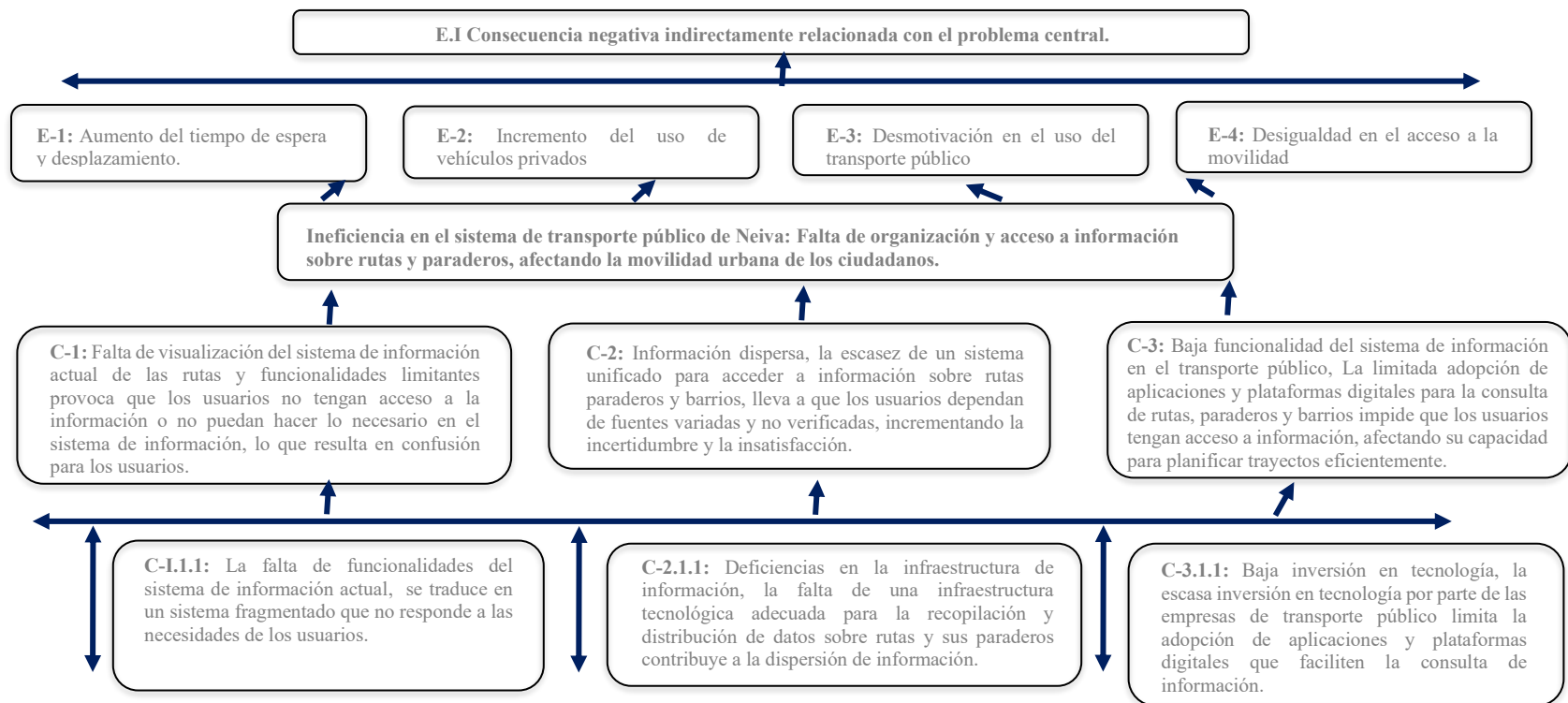
La falta de un sistema unificado de consulta de rutas de transporte público visible y con funcionalidades amplias tiene múltiples efectos negativos. Uno de los más

importantes es la pérdida de tiempo que sufren los usuarios al no poder planificar sus viajes de manera eficiente. Esto afecta directamente la productividad o descanso de los ciudadanos, quienes frecuentemente deben esperar largos periodos de tiempo para tomar un autobús, o tomar rutas menos eficientes debido a la falta de información. Además, esta situación genera una mayor congestión vehicular, ya que muchas personas prefieren usar transporte privado en lugar de depender de un sistema público poco eficiente. A nivel ambiental, el aumento del uso de vehículos privados contribuye a la contaminación y al incremento de las emisiones de gases de efecto invernadero.

El objetivo de este proyecto es desarrollar una herramienta digital que centralice la información sobre las rutas de autobuses en Neiva, facilitando su acceso a los usuarios a través de una interfaz amigable. Con esta herramienta, los ciudadanos podrán consultar las rutas, paraderos y barrios por donde pasan los autobuses de manera rápida y eficiente, mejorando la planificación de sus desplazamientos y contribuyendo a una mayor eficiencia del sistema de transporte público. Además, la herramienta fomentará el uso del transporte público, reduciendo la congestión vehicular y las emisiones de gases contaminantes.

## Árbol Causa – Efecto del Problema

**Figura 1**  
Árbol de problema



Nota. La figura muestra el árbol de causa y efecto que representa los factores que originan y resultan del problema de desinformación sobre rutas de transporte público en Neiva. Autoría propia

La justificación de este proyecto de investigación aplicado en ingeniería de sistemas se centra en la necesidad urgente de mejorar el transporte público en Neiva, un tema que impacta no solo a los usuarios, sino a toda la comunidad en términos de movilidad, calidad de vida y sostenibilidad ambiental. Actualmente, la falta de visibilidad del sistema unificado de información sobre rutas y paraderos crea inconvenientes significativos para los usuarios, quienes enfrentan incertidumbre y frustración al intentar planificar sus desplazamientos. Esta situación afecta a diversos grupos sociales, incluyendo estudiantes, trabajadores y personas mayores, quienes dependen del transporte público para acceder a oportunidades educativas y laborales.

Desde una perspectiva académica y disciplinaria, la solución propuesta se alinea con las competencias clave de la ingeniería de sistemas, que busca optimizar procesos mediante el uso de tecnología y metodologías ágiles. El desarrollo de una aplicación que centralice la información sobre rutas, paraderos y barrios permitirá aplicar conceptos de diseño de software, análisis de datos y experiencia del usuario, brindando al estudiante la oportunidad de enfrentar un desafío real y pertinente. Además, al implementar prácticas como Scrum, se promoverá el trabajo en equipo y la adaptabilidad, habilidades esenciales en el entorno laboral actual. ( MEDINA VELANDIA, L. N.; ANDRÉS GUTIÉRREZ, D. 2023)

En el ámbito social, la mejora en el transporte público tiene el potencial de transformar la calidad de vida de los habitantes de Neiva. Al facilitar el acceso a información precisa y confiable, se reducirán las frustraciones diarias de los usuarios,

promoviendo una movilidad más fluida y eficiente. Esto a su vez, puede contribuir a 22 una disminución en el uso de vehículos privados, lo que sería un paso importante hacia la sostenibilidad ambiental, ya que la congestión vehicular es uno de los principales contribuyentes a la contaminación urbana. (Sara C. Grisales Vargas 2020).

Por último, en lo personal, el compromiso con este proyecto representa una oportunidad para desarrollar habilidades técnicas y blandas que serán valiosas en el futuro profesional de los involucrados. La capacidad de diseñar y desarrollar soluciones tecnológicas que aborden problemas sociales es un aspecto clave en la formación de ingenieros de sistemas. La satisfacción de ver el impacto positivo de su trabajo en la comunidad puede ser una motivación poderosa para los estudiantes y profesionales que participan en este proyecto.

### **Objetivo General**

Desarrollar una aplicación web responsiva que centralice la información sobre rutas y paraderos del transporte público en Neiva, con el fin de mejorar la accesibilidad y eficiencia del sistema de movilidad urbana.

### **Objetivos Específicos**

Recopilar la información de rutas, paraderos y barrios del transporte público urbano en Neiva.

Organizar los datos recolectados en una estructura digital clara y reutilizable.

Diseñar una interfaz web responsiva para la consulta de rutas y paraderos.

Aplicar tecnologías de geolocalización y visualización cartográfica en el desarrollo del prototipo.

Validar la funcionalidad de la herramienta mediante pruebas con usuarios reales.

Analizar los resultados obtenidos en la evaluación de usabilidad y satisfacción.

Proponer estrategias de mejora y difusión basadas en la retroalimentación de los usuarios.

**Marco Teórico:***Estado del Arte*

El desarrollo de aplicaciones web para la gestión y visualización de rutas de transporte público ha sido ampliamente explorado en contextos urbanos a nivel mundial. Soluciones como Google Maps (Google, 2024), Moovit (Moovit App Global LTD, 2024) y Transit App (Transit, 2024) han demostrado ser herramientas efectivas para orientar a los usuarios en sistemas de transporte masivo, integrando geolocalización, tiempos estimados de llegada y visualización de rutas en tiempo real. Estas aplicaciones, sin embargo, requieren de una infraestructura tecnológica compleja y bases de datos en línea constantemente actualizadas por entidades oficiales.

A nivel nacional, se han desarrollado aplicaciones específicas para ciudades como Bogotá y Medellín. Por ejemplo, TransMiApp (TransMilenio S.A., 2024) y App Metro de Medellín (Metro de Medellín, 2024), desarrolladas con apoyo institucional, permiten consultar rutas, horarios y paraderos del transporte masivo. No obstante, en ciudades intermedias como Neiva, la disponibilidad de este tipo de herramientas es limitada o inexistente, lo cual genera una oportunidad importante para la implementación de soluciones tecnológicas locales.

Diversas investigaciones han demostrado que la incorporación de tecnologías como Leaflet, el cual es una biblioteca JavaScript de código abierto, especialmente diseñada para crear mapas interactivos de forma eficiente y compatible con dispositivos móviles (Agafonkin, 2024), OpenStreetMap, proporciona una base cartográfica de uso

libre y colaborativo que puede ser adaptada a las necesidades locales de visualización 25 urbana (OpenStreetMap Foundation, 2024), y el uso de archivos GeoJSON permite estructurar datos geográficos como puntos, líneas y polígonos en un archivo basado en JSON, lo que facilita su procesamiento y renderización en tiempo real dentro de entornos web. (Butler et al., 2016). La sinergia entre estas tecnologías ha demostrado ser especialmente útil en el desarrollo de aplicaciones ligeras para la visualización de datos geoespaciales.

En conclusión, el estado del arte evidencia una brecha tecnológica en ciudades como Neiva respecto al acceso a aplicaciones móviles que integren y faciliten el uso del transporte público. La aplicación *MoviNeiva* se enmarca dentro de estas iniciativas emergentes, utilizando tecnologías ligeras y de código abierto para brindar una solución contextualizada, accesible y de bajo costo para los ciudadanos.

### ***Movilidad y Transporte Público en Entornos Urbanos***

El crecimiento acelerado de las ciudades, como el observado en Recife y muchas otras áreas urbanas, ha generado importantes problemas de movilidad, especialmente en áreas densamente pobladas o en centros de generación de viajes, como universidades. Este crecimiento no planificado ha promovido el uso de transporte motorizado individual, que genera congestión urbana, contaminación y afecta negativamente la movilidad pública (Barros Carvalho Macedo, 2022). En muchas ciudades latinoamericanas, incluida Neiva, se observan patrones similares, donde la falta de planificación del transporte público genera una dependencia excesiva de los vehículos privados, afectando tanto a los usuarios del transporte público como al entorno urbano.

El transporte público juega un papel importante en la movilidad sostenible de las ciudades, pero su eficiencia depende de factores como la accesibilidad, la calidad del servicio, y la oferta de rutas. En el caso del estudio realizado en Recife, se evaluó el centro de generación de viajes conformado por las universidades UPE/UNICAP, el cual genera una alta demanda de transporte debido a la concentración de estudiantes, docentes y personal administrativo. Esto refleja una situación análoga a lo que sucede en la ciudad de Neiva donde se concentran centros educativos y otros puntos de interés que generan tráfico significativo. 26

### ***Movilidad Sostenible y sus Enfoques***

El concepto de movilidad sostenible no se limita únicamente a aspectos técnicos o de ingeniería relacionados con el transporte. Es esencial analizar la movilidad desde una perspectiva integral que incluya factores sociales, económicos y ambientales. (Prada, V.; Demoraes, F.; Sáenz Acosta, H. 2024) Este enfoque amplía la comprensión de la movilidad, reconociendo que las decisiones de transporte están influidas por comportamientos, condiciones socioeconómicas y trayectorias de vida. Así, la movilidad sostenible no solo implica cambiar los medios de transporte, sino también replantear la planificación urbana para integrar mejor el transporte con las necesidades de los ciudadanos.

### ***Sostenibilidad y Transporte Inteligente***

Por otro lado, el uso de tecnologías de comunicación e información (TIC) aplicadas al transporte inteligente ofrece oportunidades para mejorar la sostenibilidad en las ciudades. Según Wang, (Prada, V.; Demoraes, F.; Sáenz Acosta, H. 2024) estas

tecnologías pueden optimizar las rutas de transporte, reducir los tiempos de desplazamiento y disminuir las emisiones de carbono. Sin embargo, su implementación en los países en desarrollo está condicionada por las limitaciones de recursos económicos y la falta de infraestructura tecnológica adecuada.

### **Marco Conceptual:**

#### ***Scrum***

Scrum es una de las metodologías ágiles más utilizadas en la gestión de proyectos de software. Se basa en eventos que permiten la inspección y adaptación constante del progreso del proyecto. El proceso de Scrum se organiza en Sprints, que son ciclos cortos de trabajo de una a cuatro semanas, en los cuales se entregan incrementos del producto. Cada sprint está acompañado de una serie de eventos clave (*MEDINA VELANDIA, L. N.; ANDRÉS GUTIÉRREZ, D. 2023*):

- **Sprint Planning:** donde se definen los objetivos del Sprint y las tareas a desarrollar.
- **Daily Scrum:** reuniones diarias para revisar el progreso y los obstáculos.
- **Sprint Review:** revisión del incremento del producto.
- **Sprint Retrospective:** evaluación de las relaciones del equipo y los procesos para mejorar la productividad .

#### ***Escala de Likert***

La Escala de Likert es un método de medición en el que se pide a las personas que indiquen cuánto están de acuerdo o en desacuerdo con una afirmación. Es muy útil para entender opiniones, actitudes y comportamientos.

En lugar de limitar las respuestas a un simple "sí" o "no," esta escala permite 28 que los participantes expresen distintos niveles de acuerdo, lo que ofrece una perspectiva más rica y detallada de sus opiniones. (Muguira, A. 2024).

### **Marco Legal:**

El marco legal que sustenta el desarrollo de una aplicación para centralizar la información de rutas y paraderos del transporte público en Neiva debe estar alineado con varias normativas locales y nacionales. La Ley 105 de 1993 en el cual en su ARTÍCULO 3.- Principios del transporte público. El transporte público es una industria encaminada a garantizar la movilización de personas o cosas por medio de vehículos apropiados a cada una de las infraestructuras del sector, en condiciones de libertad de acceso, calidad y seguridad de los usuarios sujeto a una contraprestación económica (Gestor Normativo. 2022) y la Ley 336 de 1996 ya que nos indica que estas empresas que prestan el servicio publico tienen permisos los cuales están atados a rutas y horarios, como lo podemos evidenciar en el Capítulo Cuarto en su artículo 16 donde menciona De conformidad con lo establecido por el artículo 3, numeral 7 de la Ley 105 de 1993, sin perjuicio de lo previsto en tratados, acuerdos o convenios de carácter internacional, la prestación del servicio público de transporte estará sujeta a la habilitación y a la expedición de un permiso o a la celebración de un contrato de concesión u operación, según que se trate de rutas, horarios o frecuencias de despacho, o áreas de operación, servicios especiales de transporte, tales como: escolar, de asalariados, de turismo y ocasional (Gestor Normativo. 2022). son claves, ya que establecen principios para la regulación y operación eficiente del transporte público, promoviendo su accesibilidad y organización, lo que respalda la

importancia de una herramienta tecnológica en este sector. Además, la normativa local <sup>29</sup> del Sistema Estratégico de Transporte Público (SETP) de Neiva regula el uso de la infraestructura del transporte público, siendo crucial coordinar la aplicación con las directrices establecidas por las autoridades locales para garantizar la eficiencia del sistema.

Otro aspecto relevante es el cumplimiento de la Ley 1581 de 2012, que regula la protección de datos personales Gestor Normativo. (2023), ya que la aplicación podría recolectar información sobre los usuarios. Asimismo, la Ley 1341 de 2009 en el cual en su capítulo 1 y Artículo 3 menciona que el Estado reconoce que el acceso y uso de las Tecnologías de la Información y las Comunicaciones, el despliegue y uso eficiente de la infraestructura, el desarrollo de contenidos y aplicaciones, la protección a los usuarios, la formación de talento humano en estas tecnologías y su carácter transversal, son pilares para la consolidación de las sociedades de la información y del conocimiento (Gestor Normativo. 2023) e impulsa el acceso equitativo a las TIC, por lo que la aplicación debe ser accesible para todos los ciudadanos.

### **Marco Tecnológico:**

El marco tecnológico de este proyecto se basa en el uso de herramientas y plataformas digitales que permitan el desarrollo de una aplicación eficiente para la consulta de rutas y paraderos del transporte público en Neiva. Para el desarrollo del, se considera el uso del framework Angular, que permite manejar servicios y gestionar la lógica de negocio de forma escalable. Esta tecnología ofrece alta compatibilidad y capacidad de integración con otras plataformas, además de contar con una gran

comunidad de desarrolladores, lo que facilita el soporte técnico y las actualizaciones 30  
(Alonso Valencia .2023).

Se realizará la creación de interfaces de usuario interactivas. Esta tecnología permite el desarrollo de aplicaciones web accesibles desde cualquier dispositivo (Alonso Valencia .2023), lo que es esencial para los usuarios que dependen de la aplicación para transitar. Además, se complementará con el framework de diseño Bootstrap para asegurar una experiencia de usuario intuitiva y consistente.

Finalmente, se implementarán servicios en angular que harán las veces de bases de datos para almacenar y gestionar la información de rutas, paraderos y barrios, garantizando un acceso rápido y eficiente a los datos. Estos sistemas de gestión son conocidos por su fiabilidad y capacidad de manejo de grandes volúmenes de datos, fundamentales en un proyecto que busca integrar información de diferentes fuentes en una plataforma centralizada. Además, se aprovecharán tecnologías de geolocalización como Leaflet, para optimizar la búsqueda de rutas y brindar a los usuarios la mejor opción de transporte. Este marco tecnológico asegura que la aplicación sea eficiente, escalable y accesible para una amplia variedad de usuarios.

## **Metodología**

Durante la fase preliminar del proyecto, se llevó a cabo una observación directa del funcionamiento del transporte público en Neiva, enfocándose en aspectos como la disponibilidad de rutas, tiempos de espera y accesibilidad a la información. Esta observación fue complementada con testimonios obtenidos de manera informal a través de conversaciones con familiares, amigos y conocidos que son usuarios habituales del

sistema, quienes manifestaron dificultades comunes relacionadas con la falta de información clara sobre los recorridos, paraderos y horarios. La recopilación de estas percepciones permitió identificar un problema latente en la forma en que los ciudadanos acceden a la información sobre el servicio de transporte público, lo que sirvió como punto de partida para formular el objetivo del proyecto y establecer una propuesta de solución basada en el desarrollo de una aplicación informativa y accesible.

### ***Análisis y diagnóstico del proceso investigativo***

El diagnóstico se llevó a cabo a partir de la observación directa y conversaciones informales con amigos, familiares y conocidos que usan el transporte público en Neiva. A través de estos testimonios se identificaron patrones comunes en las quejas, percepciones y necesidades, lo que permitió tener una visión general sobre los principales problemas relacionados con la falta de información clara sobre rutas y paraderos. Esta aproximación permitió construir una base para enfocar el desarrollo de la aplicación.

### ***Metodología de desarrollo de software***

El desarrollo del software se implementará siguiendo la metodología Scrum, que divide el trabajo en ciclos de desarrollo cortos o sprints. A lo largo de 4 sprints, se construirán los diferentes módulos del proyecto, desde el análisis inicial hasta el lanzamiento de la aplicación. El primer sprint se enfocará en la observación y levantamiento de requisitos, mientras que los siguientes sprints cubrirán el desarrollo de los servicios, frontend, pruebas, y optimización del sistema, garantizando entregas continuas de funcionalidades. (MEDINA VELANDIA, L. N.; ANDRÉS GUTIÉRREZ, D. 2023).

## ***Sprint 1 – Preparación e investigación***

32

Tipo de Sprint: Exploratorio / Investigación

Duración estimada: Mes 1

Objetivo: Comprender el contexto del usuario y recopilar información clave del sistema actual.

Actividades:

Realización de un evento de observación para identificar el panorama actual del transporte público en Neiva y detectar necesidades de los usuarios.

Ejecución diaria de reuniones Daily Scrum para monitorear avances.

Revisión del sistema de información existente del Sistema Estratégico de Transporte Público (SETP) para identificar rutas principales y patrones.

Análisis de los datos recopilados (rutas, paraderos, cobertura) y mapeo de las rutas más representativas.

Consolidación y validación de la información recolectada con base en fuentes oficiales y observación directa.

## ***Sprint 2 – Diseño y definición***

Tipo de Sprint: Diseño

Duración estimada: Mes 2

Objetivo: Definir las funcionalidades clave de la aplicación y construir un prototipo navegable.

Actividades:

Realización de una Sprint Review al final para evaluar los avances de diseño y 33 definición.

Identificación y priorización de funcionalidades clave, además de establecer el alcance y los requerimientos funcionales y no funcionales del sistema.

Elaboración de wireframes y diseño del flujo de navegación de la interfaz, enfocado en la experiencia del usuario (UX/UI).

Presentación del prototipo inicial al director del proyecto aplicado para obtener retroalimentación y ajustar aspectos clave antes de iniciar el desarrollo.

### ***Sprint 3 – Desarrollo funcional***

Tipo de Sprint: Implementación técnica

Duración estimada: Mes 3

Objetivo: Desarrollar los servicios y la interfaz del sistema, e iniciar pruebas funcionales.

Actividades:

Implementación de los servicios backend y lógicos utilizando el framework Angular (por ejemplo, carga de datos GeoJSON, consultas de rutas y paraderos).

Desarrollo del frontend responsivo empleando Angular y Bootstrap para garantizar compatibilidad en múltiples dispositivos.

Reuniones Daily Scrum para el seguimiento constante del desarrollo.

Realización de una Sprint Review enfocada en la validación técnica de las funcionalidades desarrolladas.

Ejecución de pruebas de funcionamiento e integración entre los módulos frontend, con los ajustes necesarios. 34

#### *Sprint 4 – Validación y cierre*

Tipo de Sprint: Evaluación y cierre de proyecto

Duración estimada: Mes 4

Objetivo: Validar el prototipo con usuarios reales, recopilar retroalimentación y realizar mejoras finales.

Actividades:

Presentación del prototipo funcional a usuarios reales y al director del proyecto aplicado, con fines de validación en un entorno simulado de uso.

Recolección de datos mediante una encuesta digital estructurada en Google Forms, diseñada para evaluar la usabilidad, accesibilidad y funcionalidad de la aplicación.

Selección de participantes, se aplica un muestreo no probabilístico por conveniencia. Incluye usuarios mayores de edad con experiencia frecuente en el uso de transporte público urbano en Neiva y acceso a dispositivos con conexión a internet.

Criterios de inclusión:

Uso frecuente (diario o semanal) del transporte público en Neiva.

Acceso a un navegador web en computador o dispositivo móvil.

Voluntariedad y consentimiento informado para participar.

Criterios de exclusión:

Personas que no residen en Neiva o no utilizan transporte público regularmente.<sup>35</sup>

Personas con dificultades técnicas para acceder o interactuar con la aplicación.

Tamaño de la muestra: 34 usuarios (margen de error estimado del 16.8 % con un nivel de confianza del 95 %).

Análisis estadístico de los resultados obtenidos, incluyendo medidas como media, desviación estándar y varianza ponderada por ítem.

Aplicación de los ajustes finales al sistema según la retroalimentación obtenida.

Ejecución de la Sprint Retrospective para evaluar todo el proceso, identificar aprendizajes y documentar conclusiones y recomendaciones para futuras iteraciones.

### **Delimitación Geográfica**

El proyecto se desarrollará exclusivamente para la ciudad de Neiva, Huila, incluyendo únicamente las rutas y paraderos del sistema de transporte público urbano que opera dentro de su perímetro urbano y este suscrito al SEPT.

### **Delimitación Temporal**

El desarrollo del proyecto está planificado para ejecutarse en un periodo de cuatro meses, distribuidos en cuatro sprints, desde la recolección de datos hasta el despliegue de la aplicación.

### **Delimitación Tecnológica**

La aplicación será desarrollada utilizando Angular, Bootstrap y Leaflet, con servicios desplegados en GitHub Pages. No se contemplará la creación de versiones nativas para iOS o Android, sino una versión web adaptable a dispositivos móviles.

### **Delimitación de Usuarios**

La aplicación estará dirigida a ciudadanos residentes o visitantes frecuentes de Neiva que utilizan el transporte público, sin incluir operadores del sistema o autoridades reguladoras.

### **Delimitación Funcional**

La herramienta permitirá únicamente la consulta de rutas, paraderos y barrios asociados; no incluirá funciones de pago electrónico, compra de pasajes ni rastreo en tiempo real de vehículos.

El análisis de requerimientos es una fase fundamental en el desarrollo de cualquier proyecto de software, ya que define qué necesita el sistema para cumplir con los objetivos y expectativas de los usuarios. Para este proyecto, que se enfoca en el desarrollo de una aplicación para la consulta de rutas y paraderos del transporte público en Neiva, los requerimientos se pueden clasificar en funcionales y no funcionales (*Northware. 2022*).

### **Requerimientos Funcionales:**

- Consulta de rutas y paraderos, la aplicación debe permitir a los usuarios buscar rutas de autobuses, ingresando un punto de origen y un destino, mostrando la ruta, sus paraderos y barrios seleccionados.
- Visualización de mapas, el sistema debe mostrar un mapa interactivo que permita visualizar las rutas disponibles y la ubicación de los principales puntos de parada.

### **Requerimientos no Funcionales:**

- Usabilidad, la interfaz debe ser intuitiva y fácil de usar para personas de todas las edades, incluyendo adultos mayores, con accesos directos y menús simples.
- Rendimiento, el sistema debe ser capaz de procesar las consultas de rutas en menos de 2 minutos, con una capacidad de respuesta eficiente a las solicitudes de información.
- Seguridad, no se guardarán los datos de los usuarios.

- Mantenibilidad, el sistema debe estar diseñado para facilitar futuras actualizaciones y cambios, permitiendo la adición de nuevas rutas o funcionalidades sin afectar el rendimiento general.

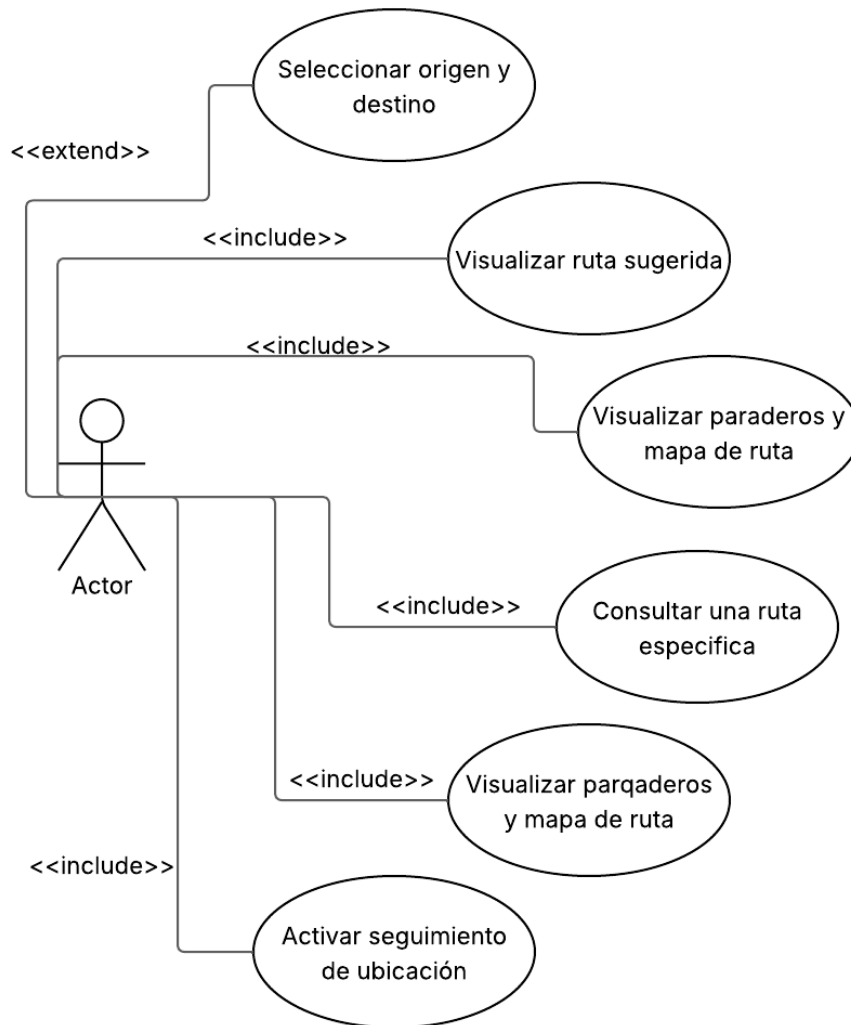
38

**Requerimientos del Hardware y Software:**

- Servidores, Realizara el despliegue de la aplicación en GitHub Pages que es un servicio gratuito de GitHub.
- Lenguaje de programación: Se utilizarán lenguajes como TypeScript/Angular para para el frontend y los servicios, facilitando la integración y el desarrollo ágil.

Figura 2

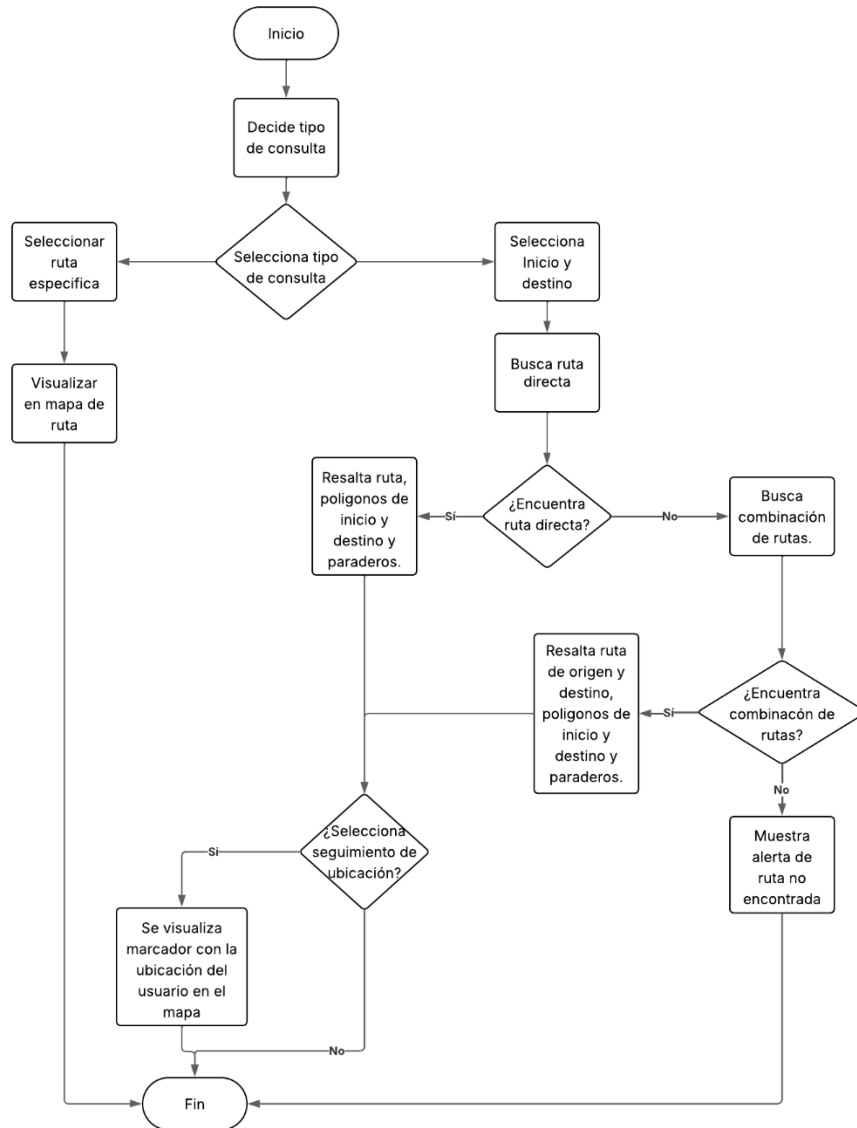
Diagrama de caso de uso



Nota. La figura muestra el diagrama de casos de uso que muestra las principales interacciones del usuario con la aplicación de rutas, como la consulta de rutas, selección de origen y destino, y seguimiento de ubicación. Autoría propia

Figura 3

Diagrama de actividades

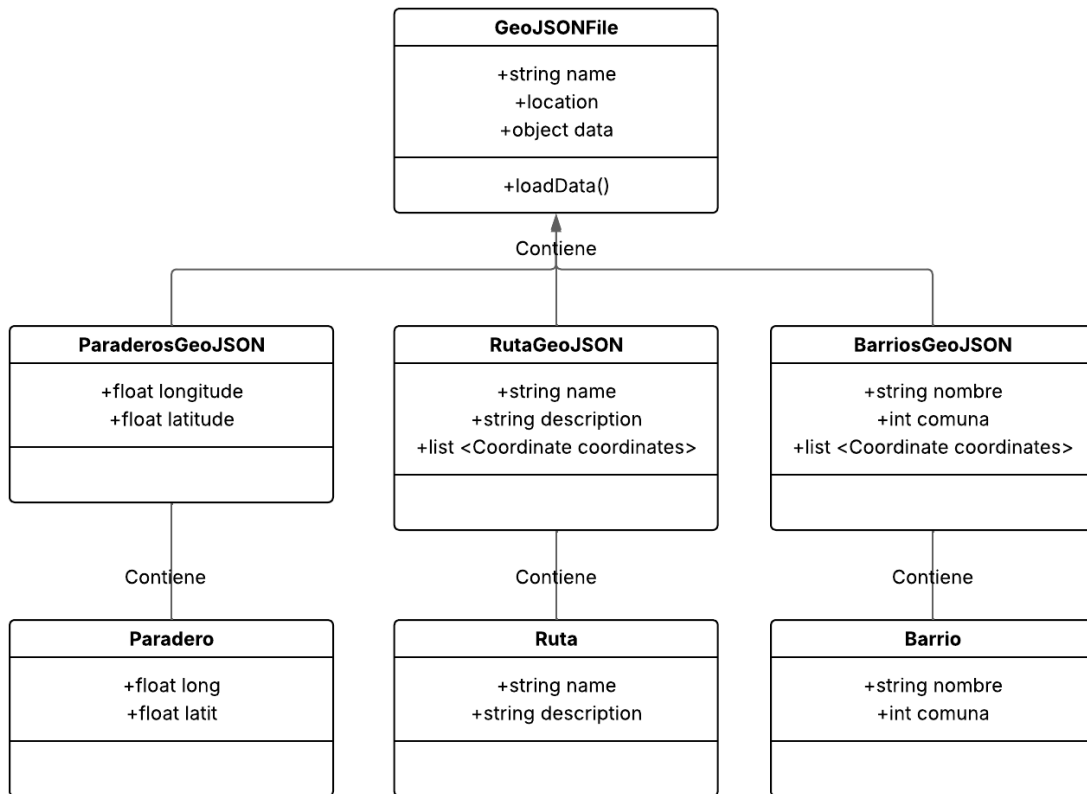


Nota. La figura muestra el diagrama de actividades que representa el flujo de acciones del usuario al interactuar con la aplicación para consultar rutas de transporte público en Neiva. Autoría propia

## Diagrama de Clases

**Figura 4**

*Diagrama de clases*

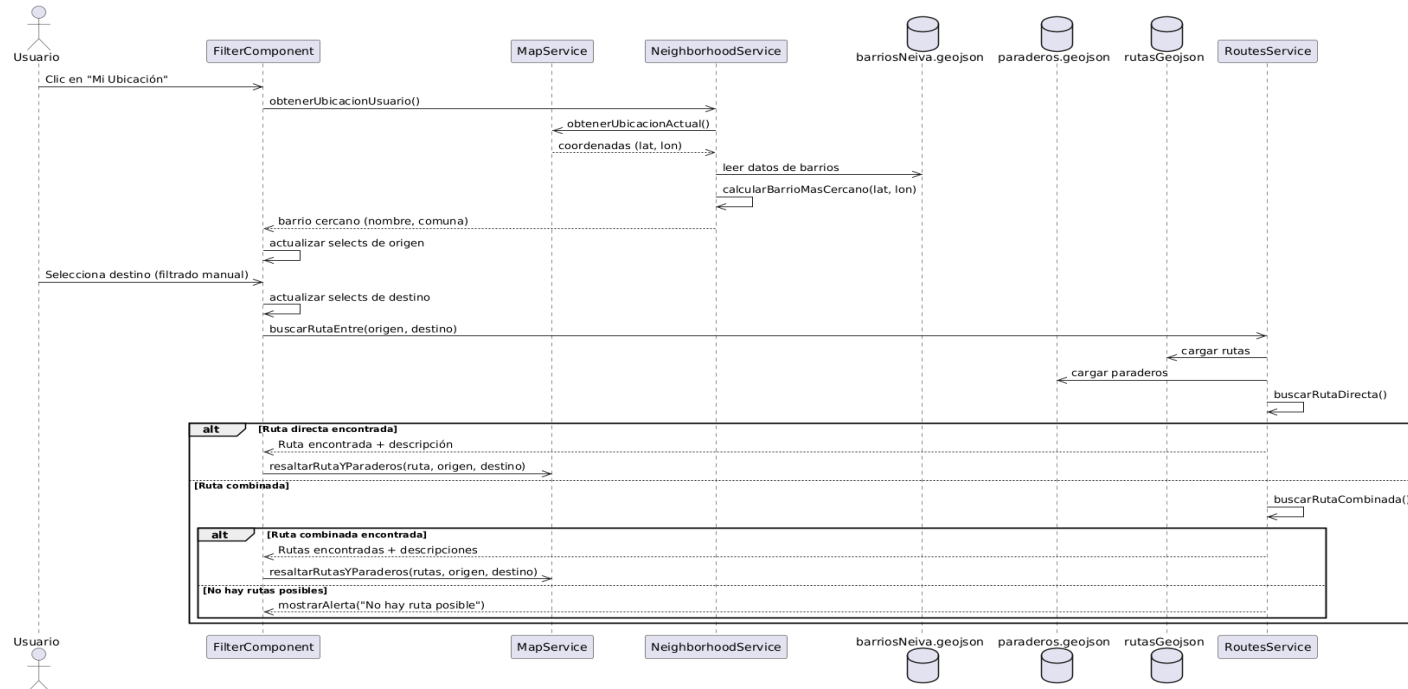


Nota. La figura muestra el diagrama de clases que representa la estructura de información utilizada por la aplicación MoviNeiva, modelando las entidades derivadas de los archivos GeoJSON empleados para la visualización y gestión de rutas, paraderos y barrios en la ciudad de Neiva. Autoría propia.

## Diagrama de Secuencia

**Figura 5**

Diagrama de secuencia.



Nota. La figura muestra el diagrama de secuencia que representa la interacción entre el usuario y los componentes de la aplicación Movineiva para la localización de barrios y la búsqueda de rutas de transporte público. El modelo describe el flujo de eventos desde la obtención de la ubicación del usuario hasta la visualización de rutas posibles, utilizando archivos GeoJSON como fuente de datos para barrios, paraderos y rutas. Autoría propia.

## Cronograma de Actividades

**Tabla 1**

*Cronograma de actividades*

Sprint	Tipo de Sprint	Duración estimada	Objetivo	Actividades
Sprint 1	Preparación / investigación	Mes 1	Entender el contexto del usuario y recopilar información clave del sistema actual.	<ol style="list-style-type: none"> <li>1. Evento de observación para identificar el panorama actual y posibles necesidades de los usuarios de transporte público.</li> <li>2. Daily Scrum para seguimiento del avance.</li> <li>3. Revisión del actual sistema de información del Sistema Estratégico de Transporte Público (SETP) para obtener datos de las rutas principales.</li> <li>4. Análisis de los datos recibidos y mapeo de las rutas más representativas.</li> <li>5. Revisión y análisis de la información.</li> </ol>

Sprint 2	Diseño y definición	Mes 2	Definir funcionalidades clave y crear un prototipo de la página web responsiva.	<p>6. Sprint Review al final para revisar avances.</p> <p>7. Definición de funcionalidades clave, alcance y requisitos de la aplicación.</p> <p>8. Maquetado de un prototipo funcional (wireframes y flujo de trabajo).</p> <p>9. Presentación del prototipo a director de proyecto aplicado para obtener su retroalimentación.</p>
Sprint 3	Desarrollo funcional	Mes 3	Desarrollar servicios y frontend inicial, realizar pruebas.	<p>10. Desarrollo de los servicios en Angular</p> <p>11. Desarrollo del Frontend (interfaz de usuario, experiencia de usuario, etc.)</p> <p>12. Daily Scrum para seguimiento.</p> <p>13. Sprint Review para revisar el desarrollo de las funcionalidades.</p> <p>14. Pruebas y ajustes en Servicios/frontend.</p>

Sprint 4	Validación y cierre	Mes 4	Validar el prototipo con usuarios reales y mejorar según el feedback.	<ul style="list-style-type: none"><li>15. Configurar y desplegar la versión funcional de MoviNeiva.</li><li>16. Demostración a usuarios representativos y al director del proyecto.</li><li>17. Creación de una encuesta estructurada en Google Forms.</li><li>18. Aplicación del formulario posterior al uso.</li><li>19. Aplicación de estadísticas descriptivas (media, desviación estándar, varianza).</li><li>20. Identificación de patrones, puntos críticos y fortalezas del sistema.</li><li>21. Identificación de logros, dificultades y recomendaciones futuras.</li><li>22. Documentación de aprendizajes clave del equipo.</li></ul>
----------	---------------------	-------	---	--

---

Nota. El cronograma muestra las actividades planificadas por sprint para el desarrollo y validación de la aplicación.

**Tabla 2**

*Recursos necesarios para la implementación.*

<b>Recurso</b>	<b>Descripción</b>	<b>Presupuesto</b>
<b>Equipo Humano</b>	Desarrollador del aplicativo (Proponente del proyecto)	0
<b>Equipos y Software</b>	Equipo desktop con Procesador AMD Ryzen 5 5600G with Radeon Graphics 3.90 GHz y 16 GB de Ram instaladas (Propiedad del Proponente del proyecto)Pantalla de 19 Marca LG (Propiedad del Proponente del proyecto)Uso de Visual Studio Code (licencia gratuita)	0
<b>Viajes y Salidas de Campo</b>	Transportes del Proponente del proyecto a los distintos puntos focales, paraderos y reuniones.	100.000
<b>Materiales y suministros</b>	Silla de oficina ergonómica, silla de escritorio de malla con respaldo alto con soporte lumbar y reposacabezas ajustable (Propiedad del Proponente del proyecto) Escritorio (Propiedad	0

	del Proponente del proyecto)	47
<b>Bibliografía</b>	Cursos digitales	0
<b>TOTAL</b>		100.000

---

Nota. La tabla presenta los recursos técnicos, humanos y financieros requeridos para la implementación del proyecto.

La solución propuesta es una aplicación web responsiva que permite a los usuarios del transporte público en la ciudad de Neiva consultar rutas y paraderos de buses de manera sencilla.

### **Arquitectura del Sistema**

La aplicación seguirá una arquitectura basada en cliente-servidor:

- Frontend: Será una aplicación web responsiva accesible desde navegadores, desarrollada con tecnologías como HTML5, Bootstrap y TypeScript (Angular). Proporcionará una interfaz intuitiva y amigable para el usuario, permitiendo consultar información sobre rutas y paraderos de buses.
- Servicios: Desarrollado con servicios de Angular gestionará la lógica del negocio, accediendo a archivos Geojson centralizados para manejar las rutas y Paraderos.
- Información: Se utilizarán archivos Geojson, que contendrá toda la información relacionada con las rutas, paraderos, y otros datos relevantes del sistema de transporte público.
- Servicios Externos: Se integrarán servicios de terceros para mapas y geolocalización, como Leaflet, para ofrecer a los usuarios una experiencia de navegación georreferenciada.

### **Módulos del Sistema**

La solución estará dividida en varios módulos principales:

- Módulo de Consulta de Rutas: Permitirá al usuario seleccionar un punto de origen y un destino, mostrando las rutas disponibles de los buses.

- Módulo de Geolocalización y Mapa: Usando servicios de mapas, este módulo 49 permitirá visualizar las rutas y paraderos en un mapa interactivo, mejorando la navegación y comprensión del sistema de transporte.
- Módulo de Feedback de Usuarios: Permitirá a los usuarios calificar las rutas y servicios, proporcionando retroalimentación por medio de un formulario de Google Forms que ayudará a mejorar la calidad del transporte público.

### **Interfaz de Usuario (UI)**

El diseño de la interfaz de usuario será intuitivo, accesible y funcional. Se implementarán los siguientes componentes:

- Pantalla principal: Mostrará las opciones para buscar rutas y podrá elegir entre buscar rutas que pasen por barrios, comunas o calle/carreteras representativas de la ciudad de Neiva.
- Mapa interactivo: Al seleccionar una ruta el usuario podrá observar la ruta y paraderos en el mapa junto con su trayectoria por la ciudad.

### **Flujo de Trabajo**

- El usuario abre la aplicación.
- Selección del tipo de consulta
- Opción A: Buscar una ruta personalizada
  - El usuario elige un punto de origen (puede ser manual o mediante su ubicación actual).
- El usuario selecciona un destino.

- La aplicación analiza las rutas y muestra:
  - Una ruta directa, si existe.
  - Una combinación de rutas, si se requiere hacer transbordo.
  - Se muestran detalles como paraderos, duración estimada, y mapa interactivo.
- Opción B: Consultar una ruta específica
  - El usuario selecciona directamente una ruta del listado.
  - Se muestra la ruta en el mapa con sus paraderos y barrios.
- Seguimiento en tiempo real (opcional)
  - El usuario puede activar la opción de seguimiento de ubicación.
  - La aplicación muestra su ubicación actual sobre la ruta para que sepa en qué punto del recorrido está.
- El usuario puede reiniciar la búsqueda o cerrar la aplicación.

### **Consideraciones de Seguridad**

- No se guardarán datos del usuario.

En esta fase se lleva a cabo el desarrollo práctico de la aplicación, transformando los resultados obtenidos en la etapa de análisis y diseño en código funcional mediante el uso del lenguaje de programación previamente definido (Angular/TypeScript). El sistema se estructura en módulos, como el servicio de rutas, la visualización de mapas y los filtros de búsqueda, los cuales se desarrollan de manera independiente y se integran progresivamente hasta obtener un prototipo funcional.

Inicialmente, se diseña un bosquejo de la interfaz de usuario que permite visualizar cómo será la interacción con la aplicación, considerando elementos como la selección de origen y destino, el uso de la ubicación actual y la consulta de rutas. A partir de este diseño, se implementan las funcionalidades básicas, asegurando que cada herramienta cumpla con los requerimientos definidos y sea accesible para los usuarios. Finalmente, se realizan pruebas exploratorias para validar el correcto funcionamiento de los módulos antes de su publicación.

### **Diseño de Interfaz de Usuario**

Como parte del diseño de la aplicación, se desarrollaron mockups y wireframes que permiten visualizar de forma preliminar la estructura y disposición de los elementos de la interfaz. El objetivo es garantizar una experiencia intuitiva, clara y accesible para el usuario desde el primer contacto con la aplicación.

El wireframe principal corresponde a la pantalla de inicio, la cual presenta una distribución funcional compuesta por los siguientes elementos:

Header: ubicado en la parte superior, contiene el botón “Migo”, que redirige al 52 usuario al repositorio externo de rutas del Sistema Estratégico de Transporte Público de Neiva (SETP).

Mapa Interactivo: ocupa el área central de la pantalla, y es el componente principal de visualización. Permite al usuario explorar las rutas disponibles en la ciudad, así como la ubicación de paraderos y barrios relevantes.

Botón de Filtros: permite al usuario seleccionar origen y destino para consultar la ruta directa o la combinación de rutas necesarias para llegar a su destino.

Botón de Activar Seguimiento: permite al usuario activar la función de geolocalización para ver en tiempo real su ubicación en la ruta seleccionada.

Footer: en la parte inferior de la pantalla, contiene tres enlaces que despliegan modales con información complementaria:

Instructivo: guía breve sobre cómo utilizar la aplicación.

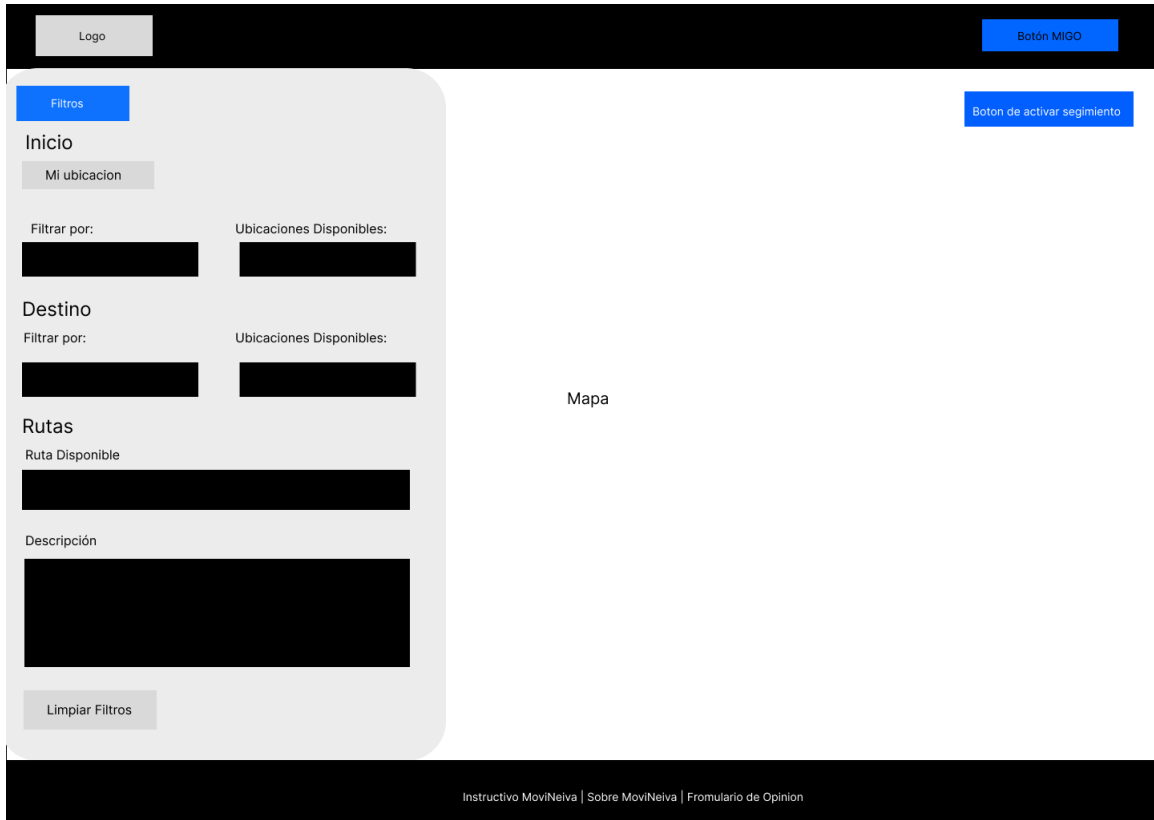
Sobre la aplicación: presenta una descripción general del propósito y alcance del proyecto.

Formulario de opinión: permite al usuario enviar sus comentarios o sugerencias para la mejora del sistema.

Este diseño busca facilitar la navegación del usuario y concentrar todas las funciones clave en una sola vista, optimizando el acceso a la información de rutas y paraderos del sistema de transporte.

Figura 6

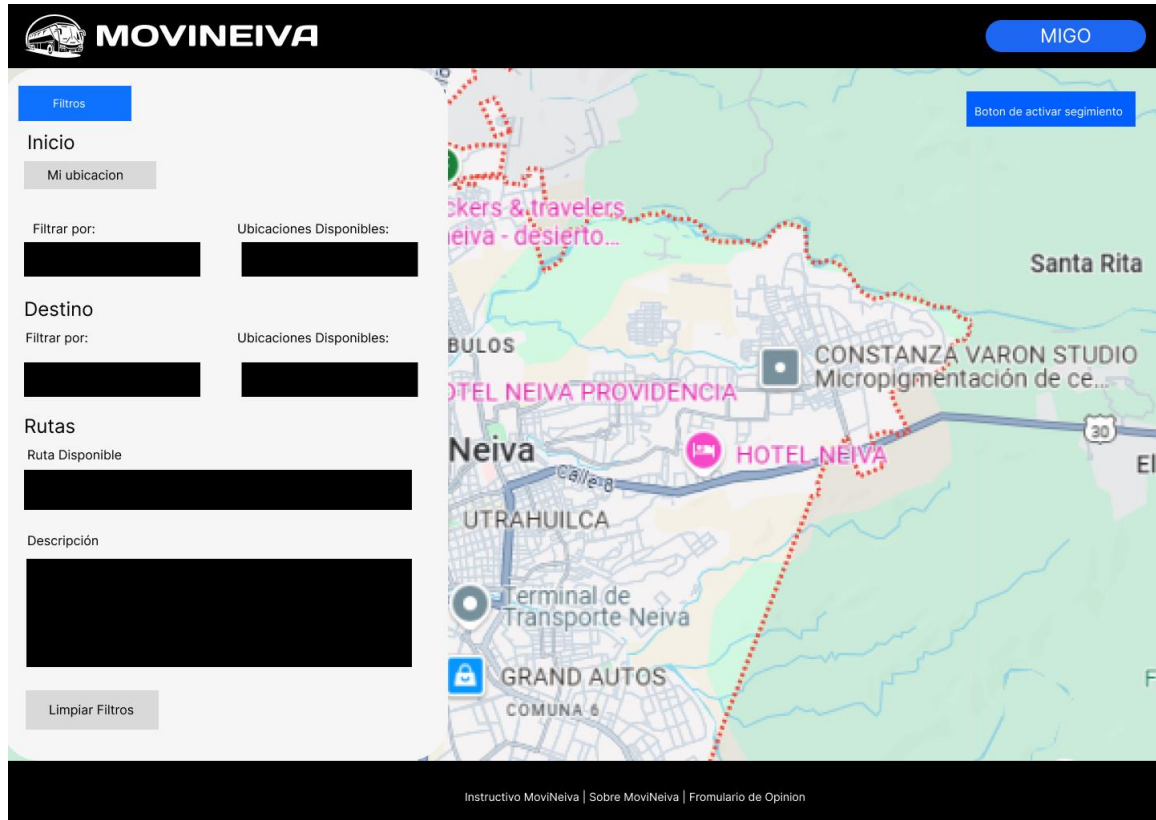
Mockup MoviNeiva



Nota. La figura muestra el mockup ilustrativo de la pantalla de inicio de la aplicación, donde se visualiza la distribución inicial de los elementos principales para la interacción del usuario. Autoría propia

Figura 7

Wireframe MoviNeiva



Nota. La figura muestra el wireframe de la pantalla de inicio que muestra la disposición estructural de los componentes clave de la interfaz, con detalles gráficos y estilos visuales. Autoría propia

El desarrollo del prototipo funcional se estructuró en base a una arquitectura modular utilizando Angular, permitiendo una separación clara de responsabilidades y facilitando el mantenimiento y la escalabilidad del aplicativo. El sistema se compone de múltiples componentes reutilizables, servicios inyectables y módulos específicos que organizan la lógica según su funcionalidad.

### Estructura y módulos

El aplicativo cuenta con varios módulos principales, entre ellos:

**HeaderModule:** El cual encapsula `<app-header>` el cual contiene el logo de la aplicación y un botón “MIGO” el cual redirecciona al repositorio oficial del SEPT.

**MapModule:** Encapsula el componente `<app-map>` que gestiona la visualización del mapa mediante la biblioteca Leaflet, permitiendo mostrar rutas, paraderos y la ubicación del usuario.

**FilterModule:** Contiene el componente `<app-filter>`, el cual maneja los filtros de búsqueda por comuna, barrio, avenidas, calles o carreras, así como la lógica de emisión de eventos hacia el mapa.

**FooterModule:** El cual encapsula `<app-footer>` el cual contiene enlaces de despliegan las modales de Instructivo, Sobre MoviNeiva y el Formulario de Opinión.

### Servicios

Se implementaron servicios Angular (`@Injectable`) para separar la lógica de datos del componente visual. Algunos de los servicios clave incluyen:

RoutesService: Encargado de cargar rutas en formato GeoJSON desde la carpeta assets/routes, diferenciando entre rutas individuales y generales.

NeighborhoodService: Carga y filtra los datos de barrios y comunas desde archivos GeoJSON, y permite obtener barrios por comuna o nombre.

MapService: Se encarga de manejar la lógica del mapa, por medio de este servicio se cargan los paraderos, barrios y ubicación del usuario.

### **Tipos de archivos manejados**

El sistema trabaja principalmente con archivos GeoJSON almacenados localmente en la carpeta assets/, clasificados de la siguiente manera:

routes/\*.geojson: Archivos individuales por cada ruta de transporte.

barrios.geojson: Archivo único que contiene los polígonos y metadatos de todos los barrios de Neiva.

paraderos.geojson: Archivo único que contiene la ubicación de todos los paraderos.

Los archivos son cargados y procesados en tiempo real por los servicios mediante HttpClient, y luego interpretados por Leaflet para su renderización sobre el mapa.

### **Integración y UX**

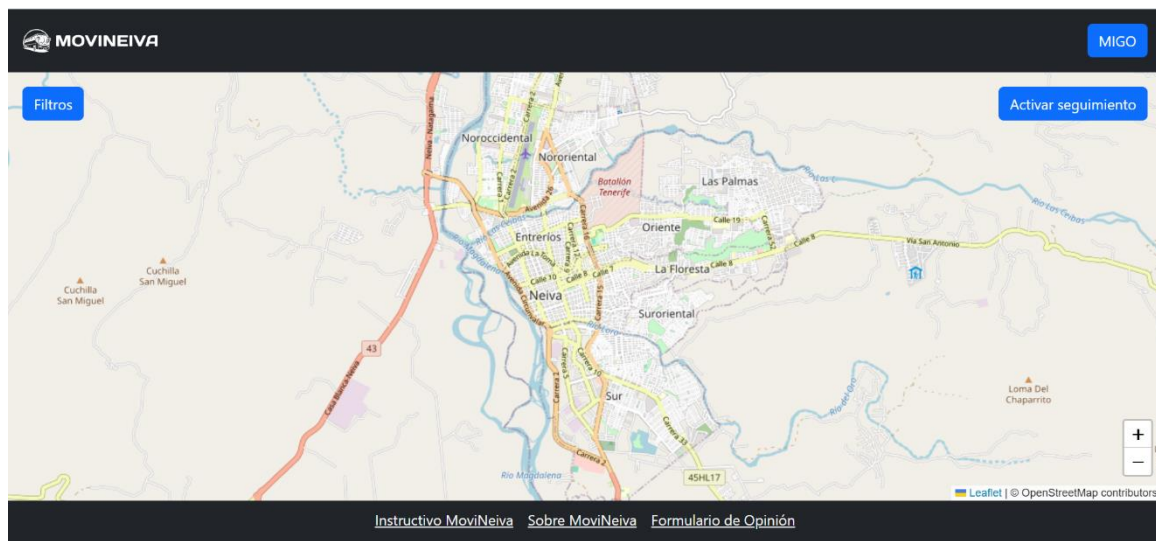
La integración de los distintos componentes del prototipo funcional se realizó de manera progresiva, manteniendo una alta cohesión entre los módulos del sistema y buscando siempre una experiencia de usuario fluida, intuitiva y adaptada a diferentes dispositivos.

El diseño de la interfaz se basa en una estructura vertical compuesta por tres bloques principales: un encabezado (<app-header>), un área central para el mapa (<app-map>) y un pie de página (<app-footer>). Esta disposición está contenida dentro de un div con clases de Bootstrap d-flex flex-column min-vh-100, lo cual garantiza que el contenido ocupe el 100% del alto de la pantalla, incluso en resoluciones pequeñas.

El componente de mapa ocupa la mayor parte de la pantalla y se adapta dinámicamente a diferentes resoluciones usando utilidades como flex-grow-1 y estilos CSS responsivos basados en calc(). Se cuidó especialmente que el pie de página no invadiera el espacio del mapa, incluso cuando este crece en pantallas pequeñas.

**Figura 8**

*Vista Prototipo funcional en ordenadores.*



Nota. La figura muestra la interfaz del aplicativo en ordenadores, destacando la disposición estructural del mapa, los botones flotantes y el acceso a filtros y modales, con estilo responsivo y visual limpio. Autoría propia

Vista Prototipo funcional desde dispositivos móviles.



Nota. La figura muestra la interfaz del aplicativo en dispositivos móviles, con una distribución adaptativa que mantiene accesibles el mapa, botones y filtros mediante un diseño responsivo y compacto. Autoría propia

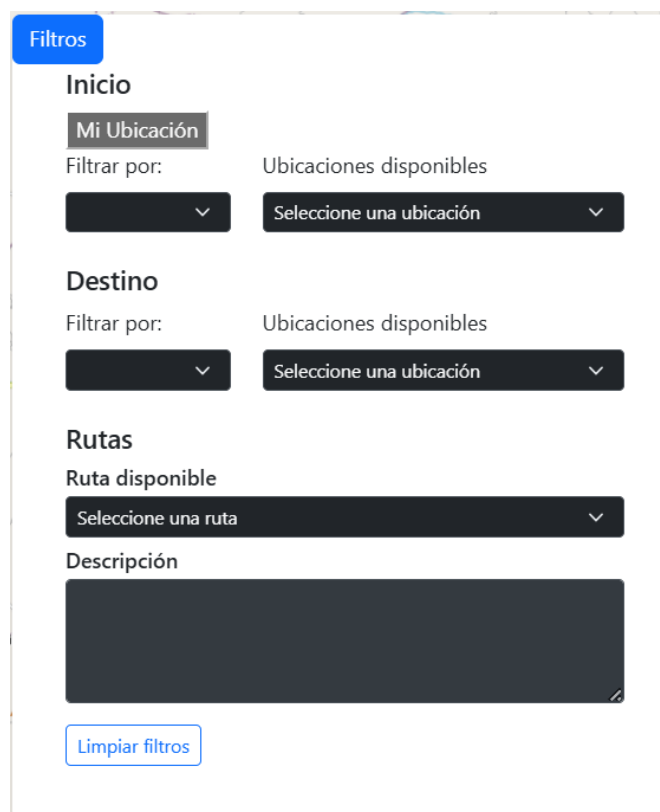
### ***Filtros dinámicos y comunicación entre componentes***

Desde el botón flotante ubicado en la esquina superior izquierda, el usuario puede acceder al componente <app-filter>, el cual se despliega como una tarjeta superpuesta. Este componente emite eventos al componente de mapa para mostrar rutas, resaltar barrios o cargar paraderos cercanos. Esta comunicación se realiza usando @Output y

EventEmitter, lo que permite una interacción fluida sin necesidad de recargar ni cambiar de página.

El sistema de filtros cuenta con lógica condicional: las listas de barrios se actualizan según la comuna seleccionada, y la selección del sitio representativo. Esto mejora la claridad del flujo de interacción y evita errores por selecciones inválidas.

**Figura 10**  
*Tarjeta de filtros*



Nota. La figura muestra la tarjeta de filtros, donde el usuario puede seleccionar rutas, comunas, barrios y sitios representativos, permitiendo una interacción dinámica con el mapa. Autoría propia

Se integraron modales usando Bootstrap para mostrar información adicional, como:

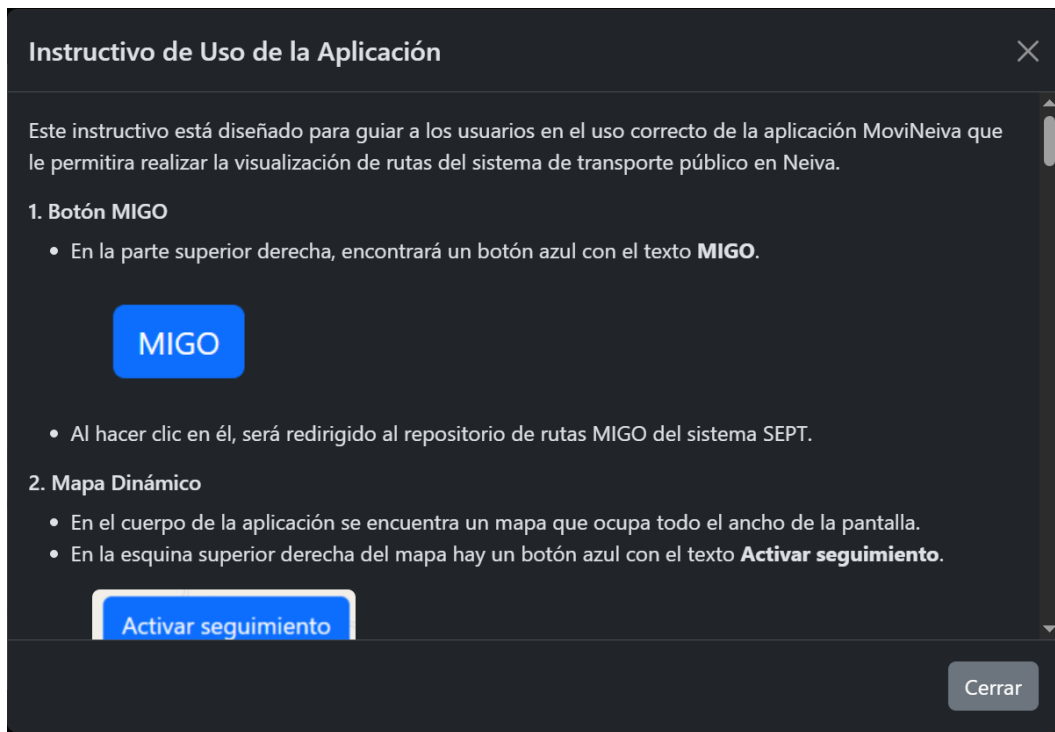
Instructivo MoviNeiva: ofrece una guía rápida sobre cómo utilizar el aplicativo, incluyendo cómo aplicar filtros, navegar el mapa y consultar rutas.

Acerca del Proyecto: presenta una descripción general del objetivo del proyecto.

Formulario de Opinión: permite a los usuarios nuevos dejar sus comentarios, sugerencias o calificaciones sobre su experiencia con la aplicación.

**Figura 11**

*Modal instructivo de uso de la aplicación.*



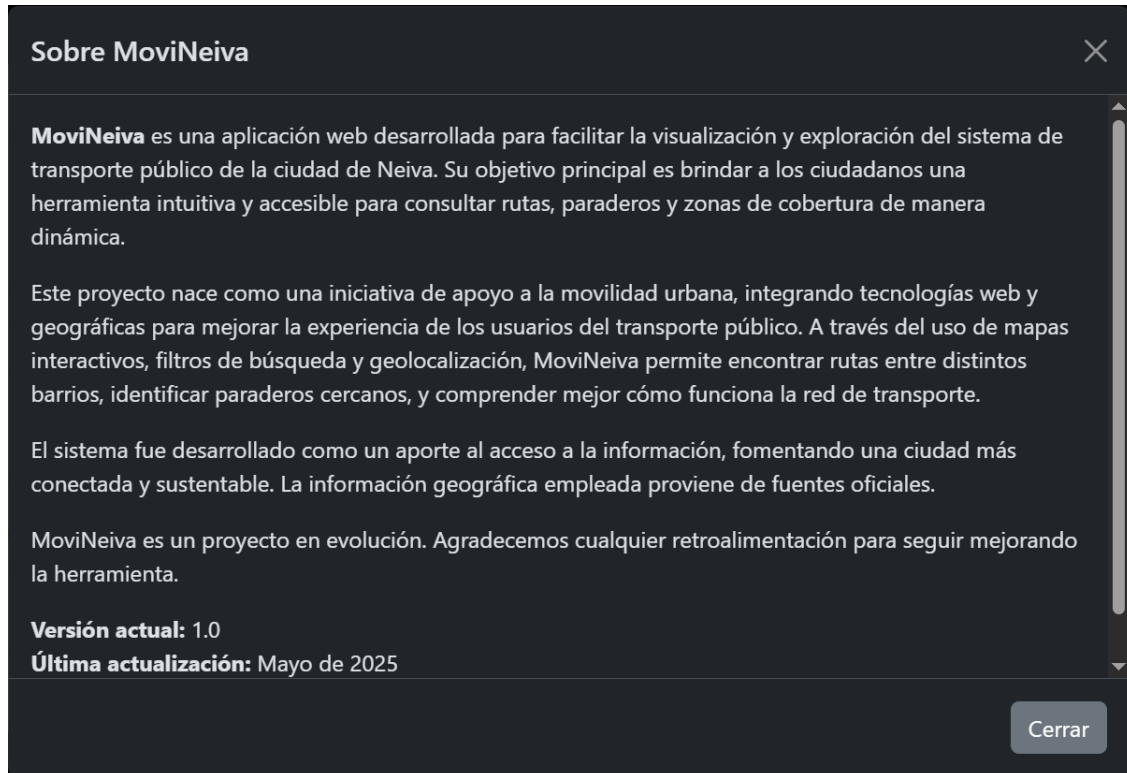
Nota. La figura muestra la ventana modal del instructivo, donde se explica el funcionamiento del aplicativo y cómo interactuar con sus principales funcionalidades.

61

Autoría propia

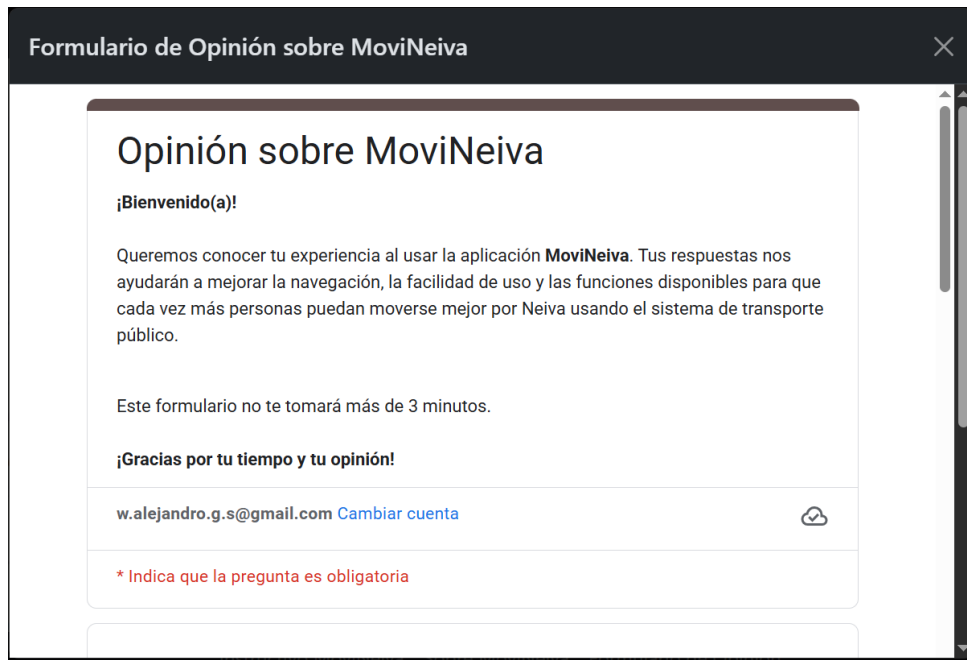
### **Figura 12**

*Modal sobre MoviNeiva*



Nota. La figura muestra la ventana modal con información sobre el proyecto MoviNeiva, incluyendo sus objetivos, enfoque y justificación dentro del contexto del transporte público en la ciudad. Autoría propia

*Modal formulario de opinión sobre MoviNeiva*



Nota. La figura muestra la ventana modal con el formulario de opinión, el cual permite a los usuarios nuevos compartir su experiencia y sugerencias sobre el uso de la aplicación MoviNeiva. Autoría propia

### ***Adaptabilidad y responsive design***

Toda la aplicación fue diseñada con un enfoque responsivo, usando el sistema de grillas de Bootstrap y clases utilitarias como container, row, col, text-center, p-3, m-auto, y overflow-auto. Se definieron ajustes específicos mediante media queries para pantallas menores a 500px, asegurando que:

Los botones flotantes no se superpongan con el mapa ni entre sí.

El mapa se redimensione correctamente para no desbordar su contenedor.

El footer reorganice sus enlaces sin desfigurar.

También se cuidó el uso de z-index para que elementos superpuestos (botones, 63 filtros, modales) no interfieran con el mapa ni con los popups de Leaflet.

### *Accesibilidad y simplicidad*

Se evitó sobrecargar al usuario con información innecesaria. Los botones y filtros fueron diseñados para ser grandes y fácilmente manipulables desde pantallas táctiles.

Además, se mantuvo un diseño de bajo contraste visual, con colores neutros y tipografía clara, enfocado en facilitar la lectura y la navegación para todos los perfiles de usuarios, incluyendo adultos mayores o personas no familiarizadas con tecnología.

Durante esta etapa, el prototipo funcional de la aplicación MoviNeiva se integró en un entorno de pruebas que simula las condiciones reales de uso. Se seleccionaron usuarios finales con el perfil objetivo del sistema, quienes interactuaron con la aplicación para verificar el cumplimiento de los requisitos funcionales y no funcionales previamente definidos. El objetivo principal de esta fase fue asegurar que la aplicación cumpla con los estándares mínimos de calidad, usabilidad y desempeño para una experiencia óptima en dispositivos móviles y de escritorio.

### Despliegue automático del instructivo

Al acceder por primera vez a la aplicación, debe abrirse automáticamente una ventana modal que presenta el instructivo de uso para facilitar la comprensión inicial del sistema.

### Figura 14

#### Despliegue instructivo MoviNeiva



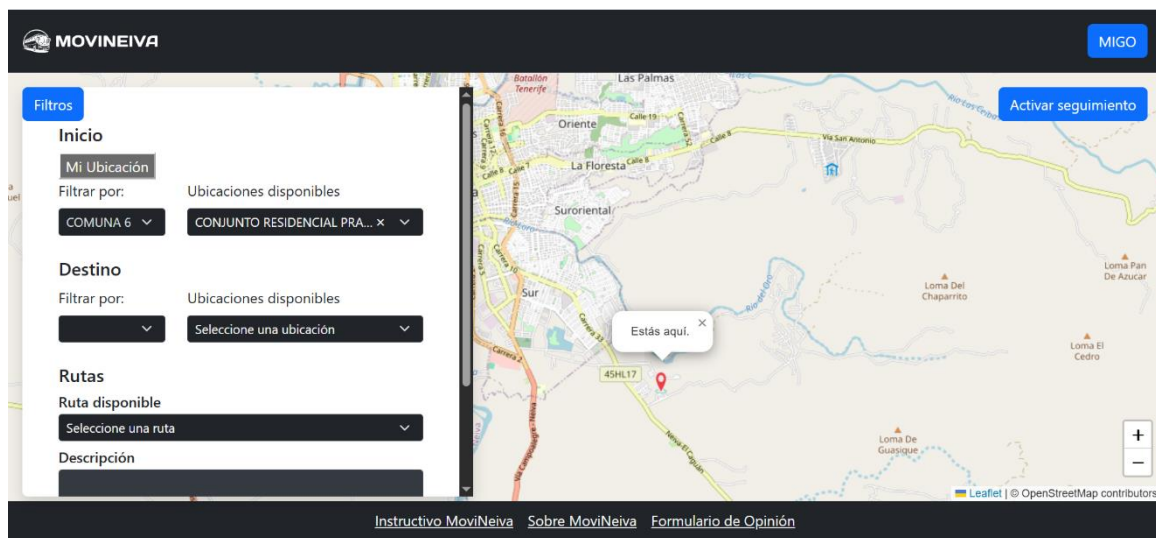
Nota. La figura muestra la ventana modal del instructivo, la cual se despliega al iniciar <sup>65</sup> la aplicación para orientar al usuario sobre el uso de los filtros y funcionalidades principales. Autoría propia.

### Pruebas del módulo de filtros

Uso del botón “Mi ubicación” para establecer automáticamente la ubicación del usuario y determinar su comuna y barrio actuales.

**Figura 15**

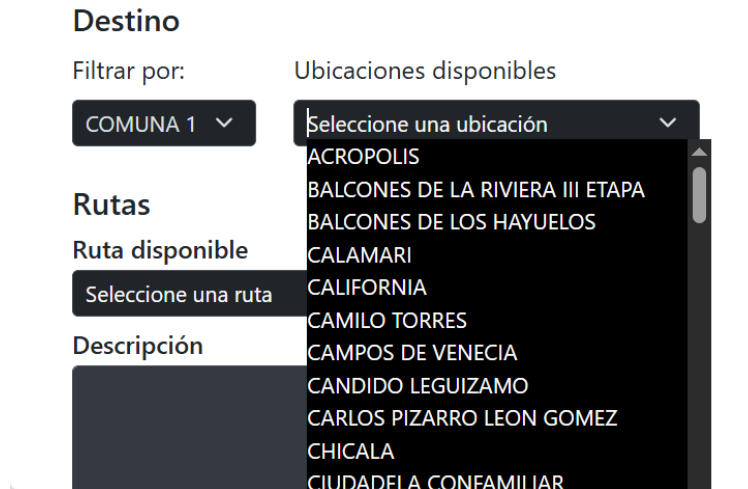
*Seleccionar inicio según ubicación*



Nota. La figura muestra la selección automática del punto de inicio utilizando la ubicación actual del usuario, lo cual permite identificar y cargar directamente la comuna y barrio correspondiente. Autoría propia.

Filtrado dinámico de barrios según la comuna seleccionada, permitiendo una navegación más precisa.

Filtrado de ubicaciones por comuna.



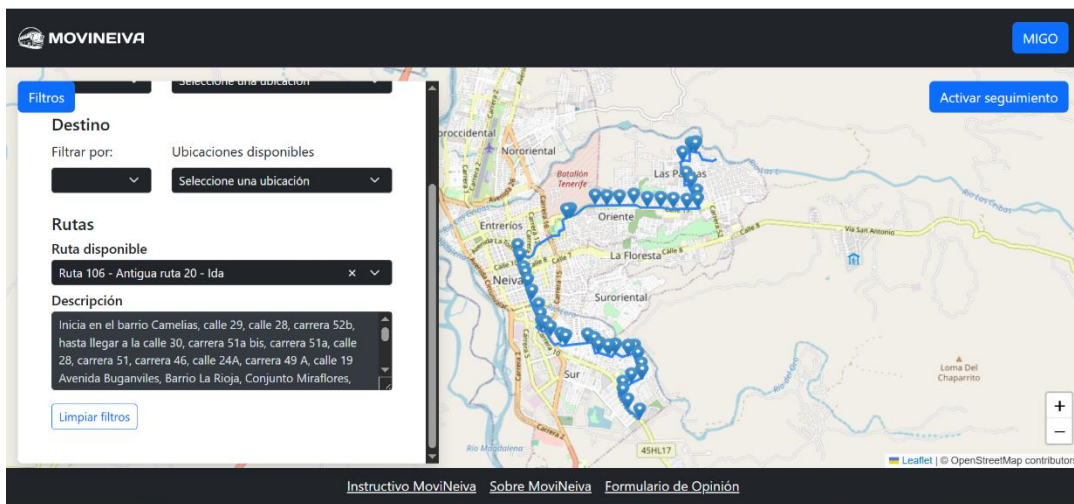
Nota. La figura muestra el filtrado de ubicaciones por comuna, permitiendo al usuario restringir la búsqueda de barrios y sitios representativos según la zona seleccionada.

Autoría propia.

Posibilidad de seleccionar una ruta específica sin necesidad de definir un punto de inicio ni de destino.

Figura 17

Selección de rutas.

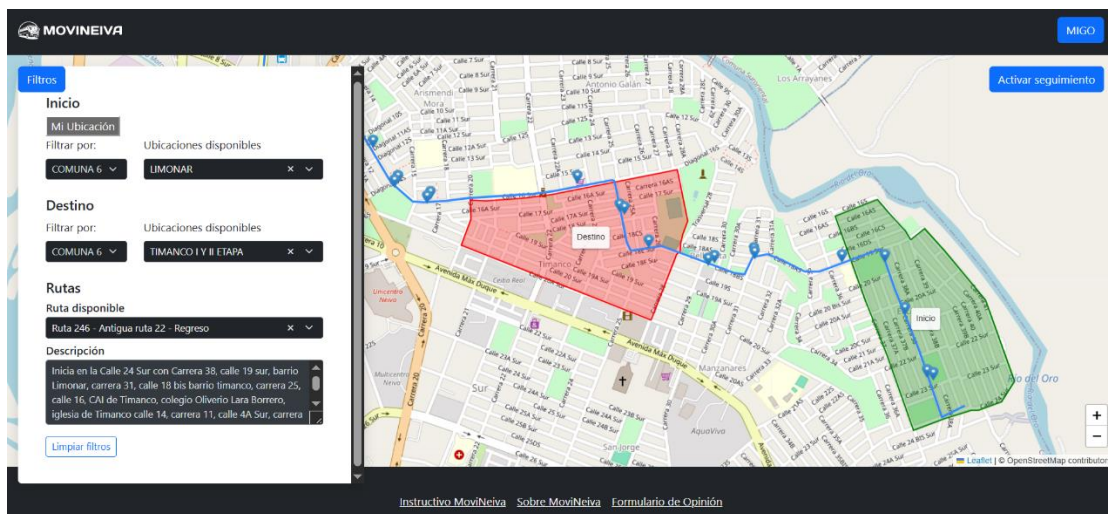


Nota: La figura muestra la selección directa de una ruta sin necesidad de especificar un <sup>67</sup> punto de inicio o destino, permitiendo al usuario explorar libremente las rutas disponibles. Autoría propia.

Selección de un punto de inicio y destino que permita visualizar una ruta directa en el mapa, junto con el resaltado de los barrios correspondientes.

### Figura 18

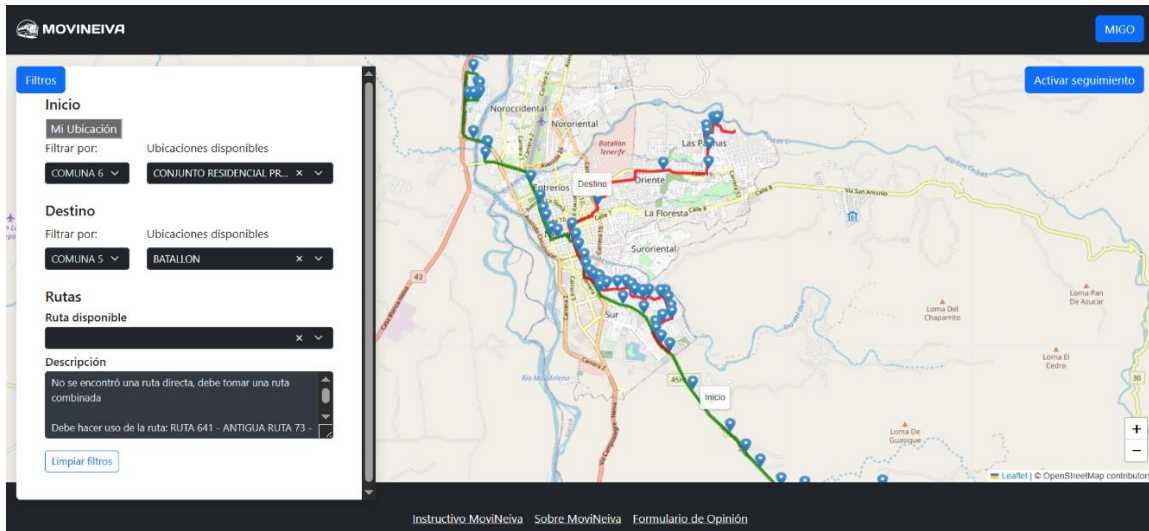
Ruta directa seleccionando punto de inicio y destino.



Nota: La figura muestra la visualización de una ruta directa obtenida al seleccionar un punto de inicio y destino, resaltando también los barrios correspondientes para mayor claridad. Autoría propia.

Selección de un punto de inicio y destino que arroje una ruta combinada, mostrando adecuadamente los tramos y barrios involucrados.

Ruta combinada seleccionando punto de inicio y destino.

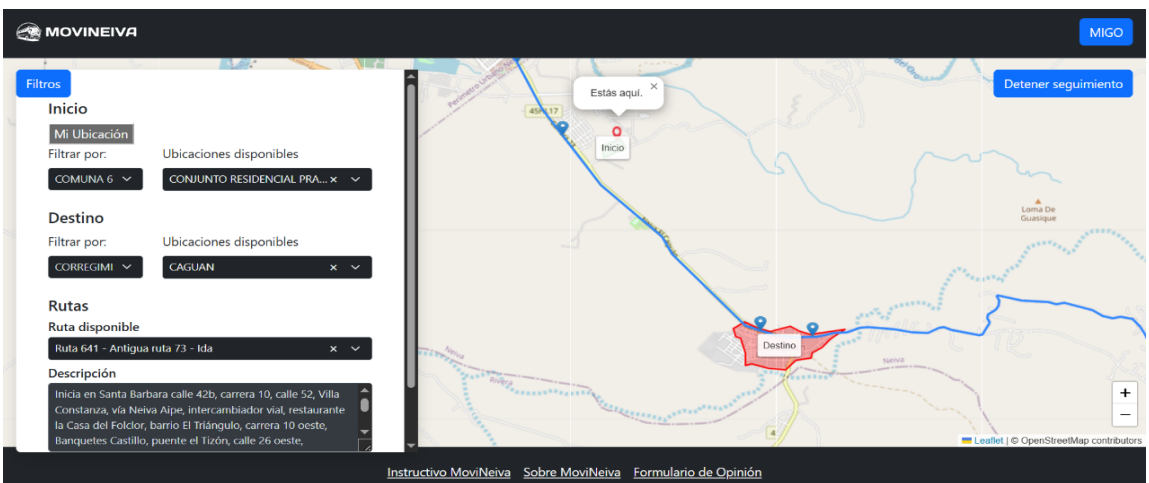


Nota: La figura muestra la visualización de una ruta combinada entre el punto de inicio y destino, donde se requiere hacer transbordo entre rutas distintas; los barrios seleccionados se resaltan para facilitar la interpretación. Autoría propia.

Seguimiento en tiempo real: Activación del seguimiento de ubicación del usuario, el cual debe actualizar su posición en el mapa conforme se desplaza.

Figura 20

Opción de seguimiento en tiempo real.



Nota: La figura muestra la funcionalidad de seguimiento en tiempo real, que permite al <sup>69</sup> usuario visualizar su ubicación actual sobre el mapa mientras se desplaza, mejorando la navegación y orientación dentro del sistema. Autoría propia.

Estas pruebas permitirán validar la funcionalidad del sistema desde la perspectiva del usuario y detectar posibles errores antes del despliegue definitivo.

### Objetivo de la Evaluación

Evaluar la usabilidad, accesibilidad y funcionalidad de la aplicación MoviNeiva, por tal motivo se publicó el prototipo en GitHub Pages, permitiendo que cualquier persona con el enlace accediera libremente. El enlace fue compartido con una muestra de 34 usuarios, quienes interactuaron con el aplicativo web y posteriormente respondieron un formulario de evaluación a través de Google Forms.

### Método de Evaluación

Tipo de instrumento: Encuesta digital con preguntas cerradas.

Población objetivo: Usuarios potenciales de transporte público urbano en Neiva.

Tamaño de la muestra: 34 personas.

Método de recolección: Aplicación de encuestas digitales.

### Criterios de selección

Los participantes fueron seleccionados por conveniencia, priorizando personas que utilizan transporte público en la ciudad de Neiva y que accedieran voluntariamente a probar la aplicación MoviNeiva.

### Margen de Error

Con un tamaño de muestra de 34 personas y un nivel de confianza del 95%, se calcula un margen de error aproximado de *Margen de error*  $\approx \pm 16.8\%$

Este valor refleja que los resultados permiten identificar tendencias generales, pero no pueden considerarse completamente representativos de toda la población de usuarios.

Se realizaron análisis básicos para cada pregunta, incluyendo:

Media ( $\mu$ ): Mide el valor promedio asignado por los usuarios.

Desviación estándar ( $\sigma$ ): Mide la dispersión de las respuestas.

Varianza: Cuadrado de la desviación estándar.

Varianza ponderada: Permite identificar la estabilidad general de cada ítem, considerando la frecuencia.

**Análisis de resultados por pregunta**

Acceso y apertura de la aplicación: El 94,1 % de los usuarios manifestó que fue muy fácil acceder y abrir la aplicación, mientras que solo un 2,9 % indicó que le tomó algo de tiempo o necesitó ayuda. La media de facilidad fue de 3,88, con una desviación estándar de 0,53, lo que refleja una ligera dispersión en las respuestas. La varianza ponderada de 0,28 muestra una mayor dispersión que en la pregunta anterior, pero sigue reflejando una clara preferencia hacia una experiencia de acceso positiva.

**Tabla 3**

*Análisis de resultados pregunta enfocada al acceso y apertura de la aplicación*

Pregunta	Respuesta	Código	Frecuencia	Porcentaje	Media	Desviación Estándar	Varianza Parcial
¿Fue fácil acceder y abrir la aplicación?	Sí, fue muy fácil	4	32	94,1%	3,88	0,53	0,44
	Sí, pero me tomó algo de tiempo	3	1	2,9%			0,78
	No, fue confuso	2	0	0,0%			0,00
	No, tuve que pedir	1	1	2,9%			8,31

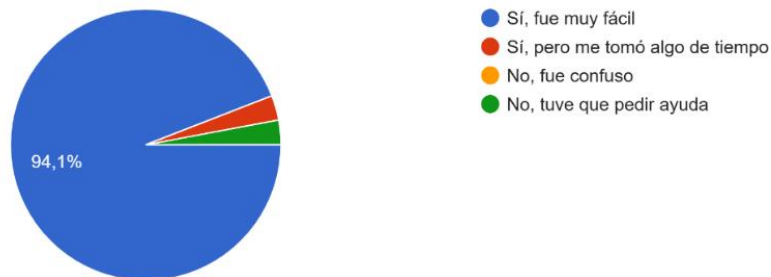
---

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

**Figura 21**

*Gráfico pregunta sobre acceso y apertura de la aplicación.*

¿Fue fácil acceder y abrir la aplicación?  
34 respuestas



Nota: Imagen tomada de los resultados de la encuesta realizada en Google Forms.

Facilidad de uso del mapa y filtros: Un 70,6 % de los encuestados afirmó haber comprendido el funcionamiento del mapa y los filtros de inmediato, mientras que el 29,4 % necesitó algo de exploración. No se registraron dificultades significativas en la comprensión. La media fue de 3,71, con una desviación estándar moderada (0,46), y una varianza ponderada de 0,21, que indica una buena concentración de respuestas positivas.

Análisis de resultados pregunta enfocada a la facilidad de uso del mapa y filtros

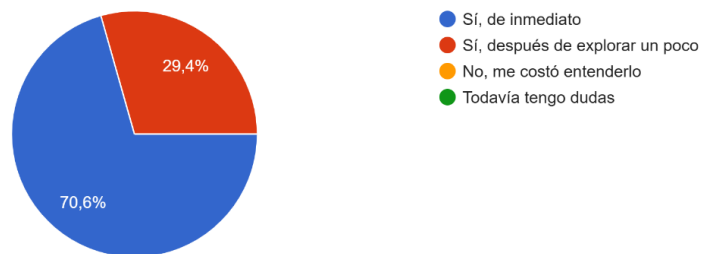
Pregunta	Respuesta	Código	Frecuencia	Porcentaje	Media	Desviación Estándar	Varianza Parcial
¿Pudo entender rápidamente cómo usar el mapa y los filtros?	Sí, de inmediato	4	24	70,6%	3,71	0,46	2,08
	Sí, después de explorar un poco	3	10	29,4%			4,98
	No, me costó entenderlo	2	0	0,0%			0,00
	Todavía tengo dudas	1	0	0,0%			0,00
Varianza Ponderada							0,21

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

Figura 22

Gráfico pregunta relacionada con la facilidad de uso del mapa y filtros

¿Pudo entender rápidamente cómo usar el mapa y los filtros?  
34 respuestas



Uso del botón de seguimiento de ubicación: Una amplia mayoría de los encuestados (94,1 %) afirmó que utilizó el botón de "Activar seguimiento" y que este funcionó correctamente. Solo un pequeño grupo (5,9 %) indicó no notar ningún cambio tras activarlo, lo cual puede relacionarse con aspectos de percepción del funcionamiento o falta de feedback visual claro. Es importante notar que ninguno manifestó no haberlo encontrado o desconocer su existencia.

La media alta (3,94) y la baja varianza ponderada (0,06) indican una percepción muy homogénea y positiva del funcionamiento de esta funcionalidad, lo que sugiere que fue bien implementada y comprendida por la mayoría de los usuarios.

**Tabla 5**

*Análisis de resultados pregunta enfocada al uso del botón de seguimiento de ubicación*

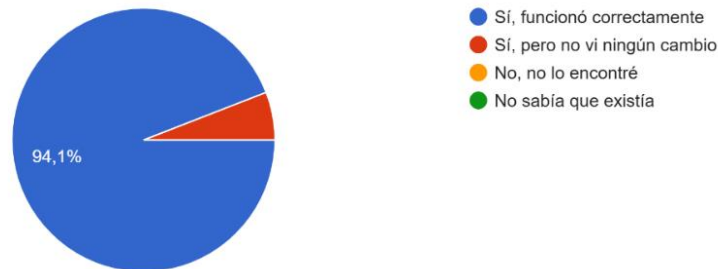
Pregunta	Respuesta	Código	Frecuencia	Porcentaje	Media	Desviación Estándar	Varianza Parcial
¿Usó el botón de "Activar seguimiento" para ver su ubicación en el mapa?	Sí, funcionó correctamente	4	32	94,1%	3,94	0,24	0,11
	Sí, pero no vi ningún cambio	3	2	5,9%			1,77
	No, no lo encontré	2	0	0,0%			0,00
	No sabía que existía	1	0	0,0%			0,00
Varianza Ponderada							0.06

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

*Gráfico pregunta relacionada con el uso del botón de seguimiento de ubicación*

¿Usó el botón de "Activar seguimiento" para ver su ubicación en el mapa?

34 respuestas



Nota: Imagen tomada de los resultados de la encuesta realizada en Google Forms.

Búsqueda de rutas: Una mayoría significativa (88,2 %) de los usuarios indicó que logró encontrar fácilmente una ruta desde su barrio, lo cual evidencia una experiencia satisfactoria en cuanto a la navegación de rutas. Sin embargo, un 5,9 % expresó que no comprendió los resultados, mientras que el 5,8 % restante manifestó dificultades para buscar rutas o se limitó a explorar el mapa.

La media de 3,79 refleja una valoración global bastante alta, aunque la varianza ponderada de 0,40 revela una ligera dispersión en las respuestas. Esto puede interpretarse como una oportunidad de mejora en la presentación de resultados o en la guía de uso para facilitar la comprensión del sistema de rutas.

Análisis de resultados pregunta enfocada a la búsqueda de rutas

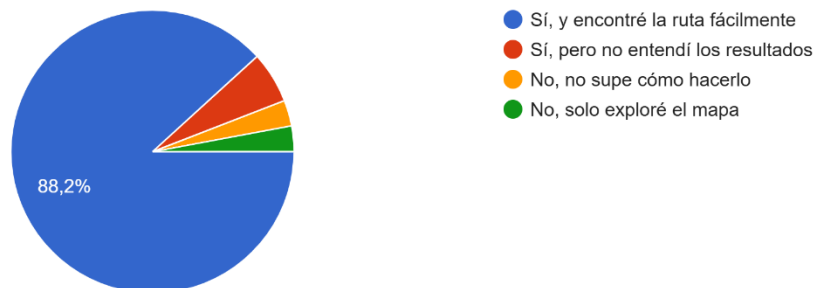
Pregunta	Respuesta	Código	Frecuencia	Porcentaje	Media	Desviación Estándar	Varianza Parcial
¿Probó buscar una ruta desde su barrio a otro destino?	Sí, y encontré la ruta fácilmente	4	30	88,2%	3,79	0,63	1,27
	Sí, pero no entendí los resultados	3	2	5,9%			1,26
	No, no supe cómo hacerlo	2	1	2,9%			3,22
	No, solo exploré el mapa	1	1	2,9%			7,81
Varianza Ponderada							0,40

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

Figura 24

Gráfico pregunta relacionada con la búsqueda de rutas

¿Probó buscar una ruta desde su barrio a otro destino?  
34 respuestas



Utilidad del instructivo: Un 94,1 % de los encuestados considera que el instructivo fue útil, lo que indica que la documentación integrada cumple eficazmente con su función de orientar al usuario. No obstante, un 5,8 % presentó dudas o no accedió al instructivo, lo cual sugiere que podría mejorarse su visibilidad o claridad.

La media de 3,91 es alta, con una varianza ponderada baja (0,14), lo que refleja consistencia en la percepción positiva sobre este recurso de ayuda.

**Tabla 7**

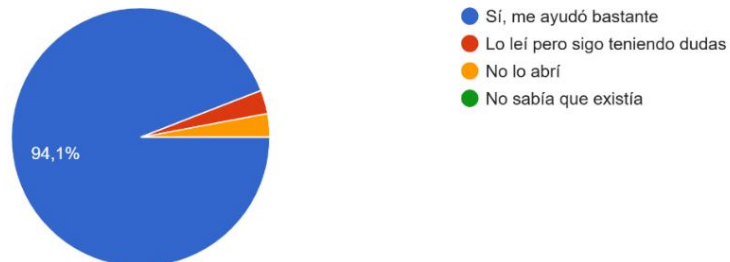
*Análisis de resultados pregunta enfocada con la utilidad del instructivo*

<b>Pregunta</b>	<b>Respuesta</b>	<b>Código</b>	<b>Frecuencia</b>	<b>Porcentaje</b>	<b>Media</b>	<b>Desviación Estándar</b>	<b>Varianza Parcial</b>
¿Le pareció útil el instructivo de la aplicación?	Sí, me ayudó bastante	4	32	94,1%	3,91	0,37	0,25
	Lo leí pero sigo teniendo dudas	3	1	2,9%			0,83
	No lo abrí	2	1	2,9%			3,65
	No sabía que existía	1	0	0,0%			0,00
Varianza Ponderada							0,14

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

*Gráfico pregunta relacionada con la utilidad del instructivo*

¿Le pareció útil el instructivo de la aplicación?  
34 respuestas



Nota: Imagen tomada de los resultados de la encuesta realizada en Google Forms.

Claridad de la información visual (rutas, paraderos, colores): La mayoría de los encuestados (70,6 %) consideró que la información mostrada en el mapa era muy clara, y un 29,4 % la consideró aceptable, sin registros en las opciones negativas. Esto indica una buena comprensión de los elementos visuales de la interfaz (como rutas, paraderos y colores), sin generar confusión significativa.

La media de 3,71 refuerza esta percepción favorable, mientras que la varianza ponderada de 0,21 señala una dispersión baja en las respuestas, es decir, hay consenso en la claridad de la información presentada en el mapa.

**Tabla 8**

*Análisis de resultados pregunta enfocada con la claridad de la información visual (rutas, paraderos, colores)*

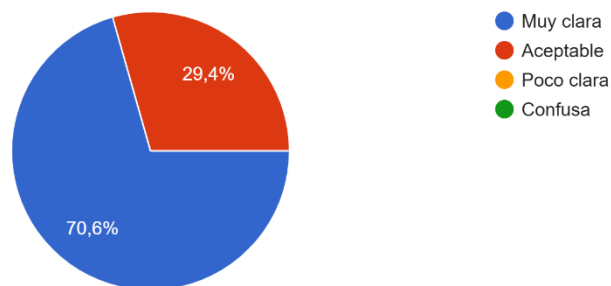
Pregunta	Respuesta	Código	Frecuencia	Porcentaje	Media	Desviación Estándar	Varianza Parcial
¿Le pareció clara la información en el mapa (rutas, paraderos, colores, etc.)?	Muy clara	4	24	70,6%	3,71	0,46	2,08
	Aceptable	3	10	29,4%			4,98
	Poco clara	2	0	0,0%			0,00
	Confusa	1	0	0,0%			0,00
Varianza Ponderada							0,21

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

**Figura 26**

*Gráfico relacionado con la claridad de la información visual*

¿Le pareció clara la información en el mapa (rutas, paraderos, colores, etc.)?  
34 respuestas



Nota: Imagen tomada de los resultados de la encuesta realizada en Google Forms.

Tiempo de respuesta para una consulta de ruta: El 91,2 % de los usuarios completó una consulta de ruta en menos de dos minutos, lo cual refleja un tiempo de respuesta ágil por parte de la aplicación. Solo el 8,8 % indicó que le tomó entre 2 y 3 minutos, y nadie reportó tiempos mayores a los 3 minutos.

La media de 3,41 muestra un desempeño eficiente en la experiencia de uso, mientras que la varianza ponderada de 0,42 evidencia una ligera dispersión en los tiempos, sin significar una problemática importante.

**Tabla 9**

*Análisis de resultados pregunta enfocada con el tiempo de respuesta para una consulta de ruta*

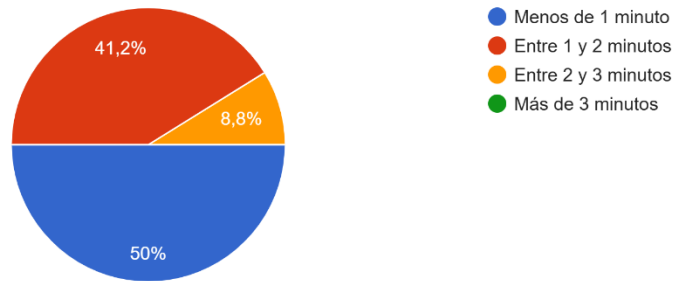
Pregunta	Respuesta	Código	Frecuencia	Porcentaje	Media	Desviación Estándar	Varianza Parcial
¿Cuánto tiempo le tomó, aproximadamente, completar una consulta de ruta en la aplicación?	Menos de 1 minuto	4	17	50,0%	3,41	0,65	5,88
	Entre 1 y 2 minutos	3	14	41,2%			2,37
	Entre 2 y 3 minutos	2	3	8,8%			5,98
	Más de 3 minutos	1	0	0,0%			0,00
Varianza Ponderada							0,42

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

Gráfico relacionado con el tiempo de respuesta para una consulta de ruta

¿Cuánto tiempo le tomó, aproximadamente, completar una consulta de ruta en la aplicación?

34 respuestas



Nota: Imagen tomada de los resultados de la encuesta realizada en Google Forms.

Satisfacción general y recomendación:

La gran mayoría de los usuarios (97,1 %) recomendaría sin reservas la aplicación MoviNeiva, y solo una persona (2,9 %) lo haría con observaciones o sugerencias de mejora. Ningún encuestado manifestó una opinión negativa o de incertidumbre.

Este resultado revela un alto nivel de satisfacción y confianza en la utilidad de la aplicación para terceros. La media de 3,97 y la varianza ponderada de 0,03 confirman una muy alta concentración de respuestas positivas, lo cual es un indicador claro del impacto favorable de la aplicación.

Análisis de resultados pregunta enfocada con los niveles de recomendación.

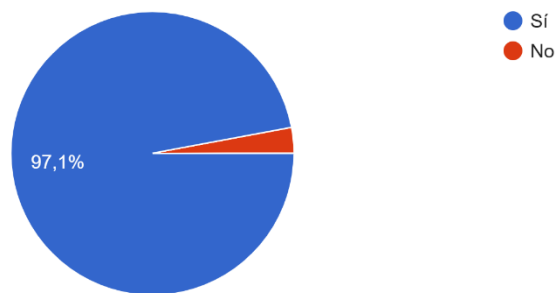
Pregunta	Respuesta	Código	Frecuencia	Porcentaje	Media	Desviación Estándar	Varianza Parcial
¿Recomendaría MoviNeiva a otras personas?	Sí, definitivamente	4	33	97,1%	3,97	0,17	0,03
	Sí, con algunas mejoras	3	1	2,9%			0,94
	No, aún no	2	0	0,0%			0,00
	No estoy seguro/a	1	0	0,0%			0,00
	Varianza Ponderada						

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

**Figura 28**

Gráfico relacionado a la satisfacción del usuario al encontrar una ruta.

¿Pudo encontrar satisfactoriamente una ruta que se ajustara a su origen y destino?  
34 respuestas



Nota: Imagen tomada de los resultados de la encuesta realizada en Google Forms.

La satisfacción general con la aplicación MoviNeiva es muy alta, con el 100 % 83 de los usuarios calificando su experiencia como "satisfecho" o "muy satisfecho". La mayoría absoluta (79,4 %) se ubicó en el nivel más alto de la escala.

El promedio de 4,79 sobre 5 refuerza esta percepción, mientras que la varianza ponderada de 0,16 indica que las respuestas están muy concentradas en las categorías positivas, con escasa dispersión y sin respuestas negativas. Esto sugiere una experiencia de usuario altamente positiva y consistente.

**Tabla 11**

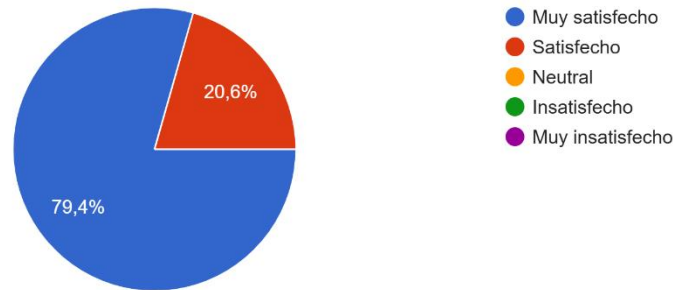
*Análisis de resultados pregunta enfocada con el nivel de satisfacción.*

<b>Pregunta</b>	<b>Respuesta</b>	<b>Código</b>	<b>Frecuencia</b>	<b>Porcentaje</b>	<b>Media</b>	<b>Desviación Estándar</b>	<b>Varianza Parcial</b>
¿Qué tan satisfecho(a) está con la aplicación en general?	Muy satisfecho	5	27	79,4%	4,79	0,40	1,14
	Satisfecho	4	7	20,6%			4,41
	Neutral	3	0	0,0%			0,00
	Insatisfecho	2	0	0,0%			0,00
	Muy insatisfecho	1	0	0,0%			0,00
Varianza Ponderada							0,16

Nota. La tabla resume los datos obtenidos lo que permite una interpretación detallada del comportamiento de las respuestas.

*Gráfico relacionado a la satisfacción del usuario.*

¿Qué tan satisfecho(a) está con la aplicación en general?  
34 respuestas



Nota: Imagen tomada de los resultados de la encuesta realizada en Google Forms.

Estos resultados evidencian que la aplicación cumple con los objetivos de diseño intuitivo, interfaz amigable y consulta rápida, siendo especialmente efectiva en dispositivos móviles y útil para usuarios diversos como adultos mayores y estudiantes.

### **Dificultades Encontradas por los Usuarios en el Primer Uso**

En la pregunta abierta "¿Qué dificultad o confusión encontró en su primer uso?", la mayoría de los usuarios indicó no haber tenido inconvenientes, lo que respalda la buena usabilidad de la aplicación. Las respuestas como "Ninguna", "Sin dificultad", "Todo fue muy bien explicado" se repitieron en más de la mitad de los casos.

Sin embargo, se identificaron algunos comentarios relevantes que ofrecen oportunidades de mejora:

Confusión inicial sobre el uso de filtros: Un usuario expresó duda sobre si debía completar todos los filtros para realizar una consulta, aunque luego comprendió la flexibilidad del sistema.

Sugerencias de mejora en la interacción: Se propuso que los filtros se minimicen automáticamente al activar el seguimiento de ubicación para mejorar la visualización del mapa.

Carga de rutas: Se mencionó que algunas rutas toman tiempo en cargar.

Navegación entre destinos: Algunos usuarios señalaron que cambiar de destino puede resultar confuso.

Deseos de funcionalidades adicionales: Se sugirió incluir localización de buses en tiempo real y mejorar la visualización en rutas combinadas sin conexión directa.

Estas observaciones permitirán ajustar la experiencia del usuario en futuras versiones, enfocándose en mejorar la fluidez del uso de filtros, tiempos de carga y funciones interactivas del mapa.

Enlace de la aplicación MoviNeiva: <https://walejandrog.github.io/movineiva/>

Enlace del Formulario: <https://forms.gle/5UUM5vg4PXrioHCUA>

El desarrollo del aplicativo web MoviNeiva representó un ejercicio exitoso de diseño, implementación y validación de una herramienta tecnológica pensada para mejorar la experiencia de los usuarios del transporte público en la ciudad de Neiva.

Se logró consolidar una base de datos estructurada en formato GeoJSON con rutas, paraderos y barrios representativos, resultado de la observación directa, revisión de fuentes oficiales y validación con usuarios.

Se estructuraron los datos en archivos reutilizables integrados en servicios de Angular, lo que permitió la carga dinámica de información en el mapa de la aplicación y su adaptación a futuros cambios.

Se implementó una UI funcional y responsiva usando Angular y Bootstrap, validada por usuarios en distintos dispositivos. La interfaz fue valorada como clara e intuitiva por el 94.1 % de los encuestados.

Se integró con éxito la biblioteca Leaflet para la visualización de mapas interactivos, paraderos y seguimiento de ubicación en tiempo real, cumpliendo así con los requerimientos técnicos definidos.

Durante la fase de pruebas, el prototipo fue desplegado en un entorno web (GitHub Pages) y se llevó a cabo una prueba con 34 usuarios reales, con un alto grado de satisfacción (97.1 % la recomendaría). Las pruebas permitieron detectar oportunidades de mejora y confirmar la usabilidad del sistema.

Se alcanzaron los objetivos propuestos: más del 80% de los participantes pudieron completar exitosamente una consulta de ruta en menos de dos minutos, los participantes

destacaron la claridad del mapa, la utilidad del instructivo y calificaron el sistema como fácil de usar y altamente recomendable. 87

Se identificaron sugerencias clave como mejorar la gestión de rutas combinadas y optimizar la visualización de filtros, las cuales serán contempladas en futuras versiones. Además, la aplicación fue difundida a través de GitHub Pages para libre acceso.

En términos de experiencia de usuario, accesibilidad y usabilidad, MoviNeiva demostró ser una solución funcional que aporta valor a los ciudadanos, especialmente a aquellos que no están familiarizados con el sistema de transporte urbano. Su diseño modular y la organización por componentes y servicios facilitaron el mantenimiento y la escalabilidad del proyecto.

En definitiva, MoviNeiva no solo cumplió con los requisitos funcionales y de usabilidad planteados al inicio, sino que también logró una fuerte validación social, consolidándose como una herramienta con potencial para ser implementada a mayor escala en beneficio de la movilidad urbana en Neiva.

- Agafonkin, V. (2024). Leaflet: An open-source JavaScript library for interactive maps.  
<https://leafletjs.com/>
- BARROS CARVALHO MACEDO, M. R. O. et al. Assessment of mobility and accessibility at the UPE / Unicap Trip Generation Hub in the city of Recife - PE. *Revista de Gestão Ambiental e Sustentabilidade (GeAS)*, [s. l.], v. 11, n. 1, p. 1–19, 2022. DOI 10.5585/geas.v11i1.19809. Disponível em: <https://research-ebSCO-com.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=0ea36471-3ef8-3b14-95b8-b755ee50b30d>. Acesso em: 23 set. 2024.
- Butler, T., Daly, M., Johnson, A., Dickinson, E., & Schaub, D. (2016). The GeoJSON format. RFC 7946. <https://www.rfc-editor.org/rfc/rfc7946>
- Crisis en el transporte urbano de Neiva: cada vez con menos pasajeros. *lanacion*. López, J. E. R. (2025, marzo 31). <https://www.lanacion.com.co/crisis-en-el-transporte-urbano-de-neiva-cada-vez-con-menos-pasajeros/>
- DE LOURDES VÁZQUEZ-ARANGO, M.; AMÍN RAMÍREZ-CASTILLO, E. Integración tecnológica y cambio organizacional: Caso de una empresa del Sistema de Transporte Público Urbano de la Zona Metropolitana de la Ciudad de Oaxaca. *CISTI (Iberian Conference on Information Systems & Technologies / Conferência Ibérica de Sistemas e Tecnologias de Informação) Proceedings*, [s. l.], n. 18, p. 1–6, 2023. Disponível em: <https://research-ebSCO-com.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=ec80cd50-fd4c-36c9-afe5-36c984320948>. Acesso em: 23 set. 2024.

emprendimientos tecnológicos y digitales: Un enfoque desde los derechos de los usuarios en el ciberespacio. Frónesis, [s. l.], v. 31, n. 1, p. 115–140, 2024.

Disponível em: <https://research-ebSCO-com.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=c7ef84e8-beed-3a7f-919a-f2f01b3605ee>. Acesso em: 23 set. 2024.

Google. (2024). Google Maps. <https://www.google.com/maps>

Ley 105 de 1993 - Gestor Normativo. (2022). Retrieved from

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=296>

Ley 336 de 1996 - Gestor Normativo. (2022). Retrieved from

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=346>

Ley 1341 de 2009 - Gestor Normativo. (2023). Retrieved from

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=36913>

Ley 1581 de 2012 - Gestor Normativo. (2023). Retrieved from

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>

Los top frameworks para Front End. - Alonso Valencia (2023) Retrieved from

<https://coderslink.com/talento/blog/los-top-frameworks-para-front-end/>

Metro de Medellín: transporte, servicios y cultura en Medellín. (s/f). Gov.co. de

<https://www.metrodemedellin.gov.co/>

MEDINA VELANDIA, L. N.; ANDRÉS GUTIÉRREZ, D. Pautas para optar por una

metodología ágil para proyectos de software. Revista Educación en Ingeniería, [s.

l.], v. 19, n. 37, p. 1–8, 2023. DOI 10.26507/rei.v19n37.1292. Disponível em: 90

<https://research-ebSCO-com.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=0ff79997-ac11-3ffb-a08f-ddef455806b7> Acceso em: 23 set. 2024.

Muguira, A. (2024). Escala de Likert: Qué es y cómo utilizarla en tus encuestas.

Retrieved from <https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/>

Northware. (2022). Requerimientos en el desarrollo de software y aplicaciones. Retrieved

from <https://www.northware.mx/blog/requerimientos-en-el-desarrollo-de-software-y-aplicaciones/>

OpenStreetMap Foundation. (2024). OpenStreetMap. <https://www.openstreetmap.org/>

Ortega, C. (2023). Investigación mixta. Qué es y tipos que existen. Retrieved from

<https://www.questionpro.com/blog/es/investigacion-mixta/>

PRADA, V.; DEMORAES, F.; SÁENZ ACOSTA, H. La movilidad sostenible:

conceptos, enfoques y realidad desde un sector periférico popular del área

metropolitana de Bogotá. Revista Ensayo, [s. l.], p. 33–52, 2024. Disponível em:

<https://research-ebSCO-com.bibliotecavirtual.unad.edu.co/linkprocessor/plink?id=8b50c843-5a94-3a0e-974c-013568a6af3b>. Acceso em: 23 set. 2024.

Sara C. Grisales Vargas (2020). Retrieved from [https://www.minambiente.gov.co/wp-](https://www.minambiente.gov.co/wp-content/uploads/2022/02/COL-evaluacion-de-las-medidas-de-mitigacion-de-contaminantes-climaticos-de-vida-corta-en-colombia.pdf)

[content/uploads/2022/02/COL-evaluacion-de-las-medidas-de-mitigacion-de-](https://www.minambiente.gov.co/wp-content/uploads/2022/02/COL-evaluacion-de-las-medidas-de-mitigacion-de-contaminantes-climaticos-de-vida-corta-en-colombia.pdf)

[contaminantes-climaticos-de-vida-corta-en-colombia.pdf](https://www.minambiente.gov.co/wp-content/uploads/2022/02/COL-evaluacion-de-las-medidas-de-mitigacion-de-contaminantes-climaticos-de-vida-corta-en-colombia.pdf)

Usuarios de transporte público podrán conocer las rutas a través de internet. Gov.co. 91

Alcaldía de Neiva. (2024, julio 18), de

<https://www.alcaldianeiva.gov.co/NuestraAlcaldia/SaladePrensa/Paginas/Usuarios>

-de-transporte-p%C3%BAblico-podr%C3%A1n-conocer-las-rutas-a-

trav%C3%A9s-de-internet.aspx

**Anexo A - Manual técnico herramienta de consulta para la movilidad en transporte público en Neiva / Huila**

Walter Alejandro González Silva

Asesor

Jaime Rubiano Llorente

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería - ECBTI

Programa de Ingeniería de Sistemas

2025

El proyecto se desarrolló utilizando la metodología ágil Scrum, permitiendo una entrega iterativa e incremental del producto. Este enfoque facilitó la retroalimentación continua, adaptación a cambios y colaboración efectiva entre los miembros del equipo.

### **Roles Scrum**

**Product Owner:** Responsable de definir y priorizar las funcionalidades del producto, gestionar el backlog y representar la voz del usuario.

**Scrum Master:** Facilitador del proceso Scrum, garantiza el cumplimiento de la metodología y elimina impedimentos.

**Equipo de desarrollo:** Encargado de diseñar, construir y probar el producto en cada sprint.

### **Estructura de los Sprints**

El proyecto se estructuró en 4 Sprints, cada uno con una duración de un mes. Cada sprint tuvo un objetivo claro y se enfocó en una fase específica del desarrollo.

### **Ceremonias Scrum Realizadas**

**Daily Scrum:** Reunión diaria para dar seguimiento al avance y detectar bloqueos.

**Sprint Review:** Al final de cada sprint, se presentaron los avances al Product Owner y se recogió retroalimentación.

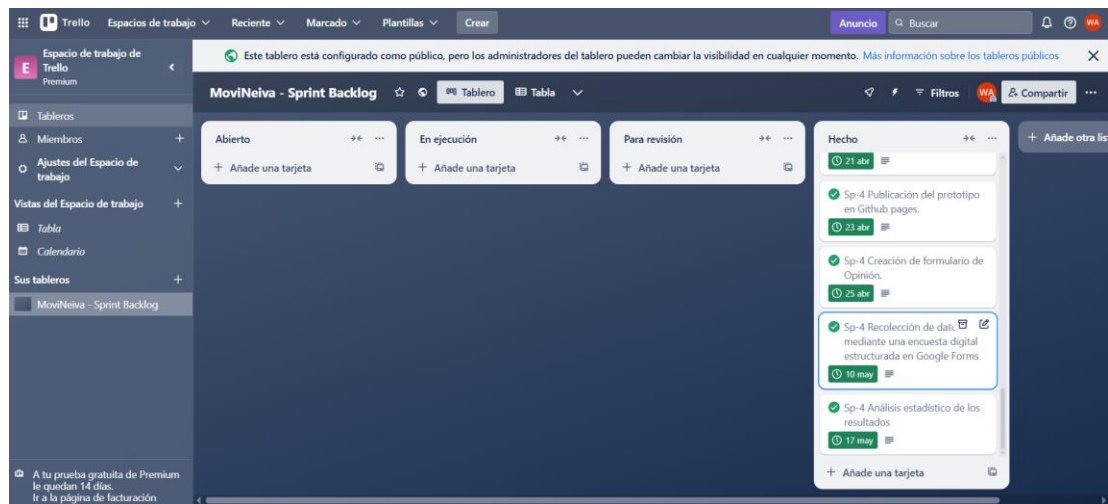
**Sprint Retrospective:** Reflexión final del equipo sobre lo que funcionó y lo que puede mejorar.

Sprint Backlog: Actividades a realizar durante cada sprint.

<https://trello.com/invite/b/6866ad7e392877e63651b870/ATTI5ad36b4fff2321ef89eb872d1fc0fe04B2294EEA/movineiva-sprint-backlog-1>

**Figura 30**

### *Sprint Backlog*



Nota: En la figura se muestra el Sprint Backlog el cual muestra el desarrollo de las tareas por cada uno de los Sprint. Autoría propia

Incremento: Producto funcional generado al final de cada sprint.

### ***Framework y Estructura de Aplicación***

***Angular (@Injectable, HttpClient)***: Utilizado como framework principal para el desarrollo de la aplicación.

**@Injectable**: Permite declarar servicios como inyectables, facilitando su uso a nivel global en la aplicación mediante el sistema de inyección de dependencias de Angular.

**HttpClient**: Cliente HTTP propio de Angular que permite realizar solicitudes HTTP para consumir recursos externos, como archivos GeoJSON locales.

### ***Gestión Reactiva***

***RxJS (Observable, of, map)***: Permite el manejo reactivo de datos a través de Observables, promoviendo un flujo de datos asíncronico y eficiente.

**Observable**: Utilizado para devolver datos de manera reactiva, especialmente aquellos cargados mediante HttpClient.

**of, map**: Utilizados para emitir y transformar datos dentro de flujos reactivos.

### ***Visualización Geoespacial***

***Leaflet (leaflet)***: Librería ligera y poderosa para la visualización de mapas interactivos. Se utiliza para renderizar mapas, añadir marcadores, cargar capas de rutas y polígonos, y gestionar interacciones del usuario con los elementos geoespaciales.

### ***Análisis Geoespacial***

***Turf.js (@turf/turf)***: Librería especializada en operaciones geoespaciales. Se emplea 96 para realizar cálculos como distancias entre puntos, detección de proximidad entre paraderos y rutas, y otras funciones relacionadas con análisis espacial.

## MapService

El MapService se encarga de la inicialización, visualización y manipulación del mapa interactivo en la aplicación. Provee funcionalidades relacionadas con:

Inicialización del mapa base usando Leaflet.

Geolocalización del usuario.

Seguimiento en tiempo real de la ubicación.

Mostrar y limpiar paraderos cercanos a rutas.

Resaltar polígonos de barrios seleccionados.

Limpieza de capas dinámicas en el mapa.

### *Principales Métodos y Funciones*

***initMap(mapId, coords, zoom)*** Inicializa el mapa de Leaflet en el contenedor con ID mapId, centrado en las coordenadas proporcionadas.

### *Figura 31*

#### *Función initMap*

```
initMap(mapId: string, coords: [number, number], zoom: number = 13): L.Map {
  this.map = L.map(mapId, {
    zoomControl: false
  }).setView(coords, zoom);
  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '&copy; OpenStreetMap contributors'
  }).addTo(this.map);
  L.control.zoom({
    position: 'bottomright'
  }).addTo(this.map);

  return this.map;
}
```

Nota: En la figura se muestra la función que inicializa el mapa de Leaflet en el contenedor con ID mapId. Autoría propia

***getUserLocation(callback)*** Obtiene la ubicación actual del usuario mediante la API de 98 geolocalización del navegador. Agrega un marcador en el mapa.

**Figura 32**

*Función getUserLocation.*

```
getUserLocation(callback?: (coords: [number, number]) => void): void {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition((position) => {
      const coords: [number, number] = [position.coords.latitude, position.coords.longitude];

      if (this.userMarker) {
        this.userMarker.setLatLng(coords);
      } else {
        this.userMarker = L.marker(coords, { icon: this.iconoUserLocation }).addTo(this.map).bindPopup('Estás aquí');
      }

      if (callback) callback(coords);
    }, () => {
      alert('No se pudo obtener la geolocalización.');
```

Nota: En la figura se muestra la función que obtiene la ubicación actual del usuario.

Autoría propia

***toggleUserTracking()*** Activa o desactiva el seguimiento continuo de la ubicación del usuario.

Función `toggleUserTracking`.

```
toggleUserTracking(): void {
  if (this.trackingEnabled) {
    navigator.geolocation.clearWatch(this.watchId!);
    this.watchId = null;
    this.trackingEnabled = false;
    console.log('Seguimiento detenido');
    this.clearUserLocation();
  } else {
    this.watchId = navigator.geolocation.watchPosition(
      (position) => {
        const coords: [number, number] = [position.coords.latitude, position.coords.longitude];

        if (this.userMarker) {
          this.userMarker.setLatLng(coords);
        } else {
          this.userMarker = L.marker(coords, { icon: this.iconoUserLocation })
            .addTo(this.map)
            .bindPopup('Estás aquí.')
            .openPopup();
        }
      },
      this.map.setView(coords, this.map.getZoom());
    );
    (error) => {
      alert('No se pudo obtener la ubicación.');
```

Nota: En la figura se muestra la función que Activa o desactiva el seguimiento continuo de la ubicación del usuario. Autoría propia

**`clearUserLocation()`** Elimina el marcador de la ubicación del usuario del mapa.

**`cargarParaderos(paraderos, rutaGeoJSON)`** Filtra los paraderos cercanos a una ruta (tipo `LineString`) usando `Turf.js`, y los muestra en el mapa con íconos personalizados.

Función *cargarParaderos*.

```

cargarParaderos(paraderos: any[], rutaGeoJSON: any) {
  const rutaFeature = rutaGeoJSON.features[0];

  if (rutaFeature.geometry.type !== 'LineString') {
    console.error('La ruta no es una LineString');
    return;
  }

  const linea = turf.lineString(rutaFeature.geometry.coordinates);

  const paraderosCercanos = paraderos.filter((paradero) => {
    const punto = turf.point(paradero.geometry.coordinates);
    const distanciaMaxima = 0.00010; // (~10 metros)
    const distancia = turf.pointToLineDistance(punto, linea, { units: 'degrees' });
    return distancia <= distanciaMaxima;
  });

  // Ya no eliminamos todos los paraderos anteriores.
  // Creamos una nueva capa para estos paraderos
  const nuevaCapaParaderos = L.geoJSON(paraderosCercanos, {
    pointToLayer: (feature, latlng) => {
      return L.marker(latlng, { icon: this.iconoParadero });
    },
    onEachFeature: (feature, layer) => {
      if (feature.properties && feature.properties.nombre) {
        layer.bindPopup(feature.properties.nombre);
      }
    }
  });

  nuevaCapaParaderos.addTo(this.map);

  // Guardar la capa para poder limpiarlas después
  this.paraderosLayers.push(nuevaCapaParaderos);
}

```

Nota: En la figura se muestra la función que Filtra los paraderos cercanos a una ruta.

Autoría propia

***clearParaderos()*** Elimina todas las capas de paraderos mostradas actualmente en el mapa.

***resaltarBarrios(barrios)*** Dibuja en el mapa los polígonos de los barrios seleccionados, aplicando un estilo visual para resaltarlos.

Función `resaltarBarrios`

```
resaltarBarrios(barrios: any[]) {  
  // Limpia las capas anteriores (si hay alguna)  
  if (this.barriosLayer) {  
    this.map.removeLayer(this.barriosLayer);  
  }  
  
  // Crear una nueva capa de barrios resaltados  
  this.barriosLayer = L.geoJSON(barrios, {  
    style: () => ({  
      color: 'green',  
      weight: 3,  
      opacity: 0.7,  
      fillOpacity: 0.3  
    })  
  }).addTo(this.map);  
}
```

Nota: En la figura se muestra la función que resalta en el mapa los polígonos de los barrios seleccionados. Autoría propia

**`clearMapLayers()`** Elimina todas las capas del mapa, excepto la capa base de tiles.

### **NeighborhoodService**

El `NeighborhoodService` es un servicio encargado de gestionar la carga de los datos geoespaciales relacionados con los barrios de la ciudad de Neiva. Su objetivo principal es proveer acceso al archivo `barriosNeiva.geojson`, el cual contiene los polígonos y atributos de cada barrio.

Este servicio facilita la reutilización de la lógica de carga de datos de barrios, centralizando su acceso y permitiendo una mejor separación de responsabilidades dentro de la aplicación.

**cargarBarrios(): Observable<any>** Este método realiza una solicitud HTTP GET para cargar el archivo GeoJSON ubicado en `assets/neighborhood/barriosNeiva.geojson`.

Tipo de retorno: `Observable<any>`, lo que permite suscribirse al resultado de manera reactiva desde cualquier componente.

Uso común: Es utilizado por componentes como `FilterComponent` y servicios como `MapService` para mostrar barrios en el mapa o filtrar por comuna y barrio.

### ***Archivo GeoJSON referenciado***

`barriosNeiva.geojson`: Archivo que contiene geometría en formato Polygon y propiedades como:

nombre del barrio

comuna a la que pertenece

### **RoutesService**

El `RoutesService` es responsable de gestionar la carga y administración de datos relacionados con las rutas del transporte público en Neiva. Se encarga de:

Proveer la lista de rutas disponibles.

Cargar archivos GeoJSON individuales que representan trayectos de ida y regreso.

Cargar el archivo de paraderos (`paraderos.geojson`).

Este servicio centraliza el acceso a los datos de rutas y paraderos, promoviendo la reutilización del código, la escalabilidad y la mantenibilidad del sistema. Facilita la carga

FilterComponent.

### *Propiedades y Métodos*

**Propiedad privada rutas:** `{ nombre: string, archivo: string }` Contiene una lista estática de rutas disponibles, cada una con un nombre descriptivo y su archivo GeoJSON correspondiente.

### *Figura 36*

*Propiedad rutas.*

```
private rutas = [
  { nombre: 'Ruta 106 - Antigua ruta 20 - Ida', archivo: 'ruta106Ida.geojson' },
  { nombre: 'Ruta 106 - Antigua ruta 20 - Regreso', archivo: 'ruta106Regreso.geojson' },
  { nombre: 'Ruta 110 - Antigua ruta 60 - Ida', archivo: 'ruta110Ida.geojson' },
  { nombre: 'Ruta 110 - Antigua ruta 60 - Regreso', archivo: 'ruta110Regreso.geojson' },
  { nombre: 'Ruta 112 - Antigua ruta 18+13 - Ida', archivo: 'ruta112Ida.geojson' },
  { nombre: 'Ruta 112 - Antigua ruta 18+13 - Regreso', archivo: 'ruta112Regreso.geojson' },
  { nombre: 'Ruta 146 - Antigua ruta 90 - Ida', archivo: 'ruta146Ida.geojson' },
  { nombre: 'Ruta 146 - Antigua ruta 90 - Regreso', archivo: 'ruta146Regreso.geojson' },
  { nombre: 'Ruta 210 - Antigua ruta 29 - Ida', archivo: 'ruta210Ida.geojson' },
  { nombre: 'Ruta 210 - Antigua ruta 29 - Regreso', archivo: 'ruta210Regreso.geojson' },
  { nombre: 'Ruta 246 - Antigua ruta 22 - Ida', archivo: 'ruta246Ida.geojson' },
  { nombre: 'Ruta 246 - Antigua ruta 22 - Regreso', archivo: 'ruta246Regreso.geojson' },
  { nombre: 'Ruta 247 - Antigua ruta 61 - Ida', archivo: 'ruta247Ida.geojson' },
  { nombre: 'Ruta 247 - Antigua ruta 61 - Regreso', archivo: 'ruta247Regreso.geojson' },
  { nombre: 'Ruta 248 - Antigua ruta 33 - Ida', archivo: 'ruta248Ida.geojson' },
  { nombre: 'Ruta 248 - Antigua ruta 33 - Regreso', archivo: 'ruta248Regreso.geojson' },
  { nombre: 'Ruta 641 - Antigua ruta 73 - Ida', archivo: 'ruta641Ida.geojson' },
  { nombre: 'Ruta 641 - Antigua ruta 73 - Regreso', archivo: 'ruta641Regreso.geojson' },
  { nombre: 'Ruta 910 - Antigua ruta 11 - Ida', archivo: 'ruta910Ida.geojson' },
  { nombre: 'Ruta 910 - Antigua ruta 11 - Regreso', archivo: 'ruta910Regreso.geojson' },
  { nombre: 'Ruta 936 - Antigua ruta 19 - Ida', archivo: 'ruta936Ida.geojson' },
  { nombre: 'Ruta 936 - Antigua ruta 19 - Regreso', archivo: 'ruta936Regreso.geojson' },
  { nombre: 'Ruta 938 - Antigua ruta 7 - Ida', archivo: 'ruta938Ida.geojson' },
  { nombre: 'Ruta 938 - Antigua ruta 7 - Regreso', archivo: 'ruta938Regreso.geojson' },
  { nombre: 'Ruta 941 - Antigua ruta 9 - Ida', archivo: 'ruta941Ida.geojson' },
  { nombre: 'Ruta 941 - Antigua ruta 9 - Regreso', archivo: 'ruta941Regreso.geojson' },
  { nombre: 'Ruta 946 - Antigua ruta 49 - Ida', archivo: 'ruta946Ida.geojson' },
  { nombre: 'Ruta 946 - Antigua ruta 49 - Regreso', archivo: 'ruta946Regreso.geojson' },
  { nombre: 'Ruta 947 - Antigua ruta 61 - Ida', archivo: 'ruta947Ida.geojson' },
  { nombre: 'Ruta 947 - Antigua ruta 61 - Regreso', archivo: 'ruta947Regreso.geojson' },
  { nombre: 'Ruta 949 - Antigua ruta 62 - Ida', archivo: 'ruta949Ida.geojson' },
  { nombre: 'Ruta 949 - Antigua ruta 62 - Regreso', archivo: 'ruta949Regreso.geojson' },
  { nombre: 'Ruta 999 - Antigua ruta 63 - Ida', archivo: 'ruta999Ida.geojson' },
  { nombre: 'Ruta 999 - Antigua ruta 63 - Regreso', archivo: 'ruta999Regreso.geojson' },
]
```

Nota: En la figura se muestra la propiedad ruta la cual es una lista estática de rutas disponibles Autoría propia

***cargarParaderosGeoJSON(): Observable<any>*** Carga el archivo GeoJSON que contiene los puntos de los paraderos disponibles en la ciudad.

Ruta del recurso: assets/routes/paraderos.geojson

Tipo de retorno: Observable<any>

Uso común: Mostrar paraderos en el mapa o filtrarlos según su cercanía a rutas seleccionadas.

***cargarTodasLasRutas(): Observable<{ nombre: string, archivo: string }[]>*** Devuelve la lista completa de rutas registradas en el arreglo interno rutas.

Tipo de retorno: Observable que emite un arreglo de objetos con nombre y archivo.

Uso común: Mostrar en un select o combo box la lista de rutas disponibles para que el usuario seleccione.

***cargarRuta(nombre: string): Observable<any>*** Carga un archivo GeoJSON específico con el trayecto de una ruta seleccionada.

Parámetro: nombre: nombre del archivo GeoJSON (por ejemplo, ruta106Ida.geojson)

Tipo de retorno: Observable<any>

Validación: Verifica que la respuesta sea un FeatureCollection válido.

Uso común: Mostrar la ruta seleccionada en el mapa mediante Leaflet.

Función *cargarRuta*.

```
// Carga un archivo específico
cargarRuta(nombre: string): Observable<any> {
  return this.http.get(`assets/routes/${nombre}`).pipe(
    map((data: any) => {
      // Verifica que los datos sean un GeoJSON válido
      if (data && data.type === 'FeatureCollection' && Array.isArray(data.features)) {
        return data;
      } else {
        throw new Error('El objeto no es un GeoJSON válido');
      }
    })
  );
}
```

Nota: En la figura se muestra la función *cargarRuta* la cual carga un archivo GeoJSON específico con el trayecto de una ruta seleccionada Autoría propia

#### *Archivos GeoJSON referenciados*

Rutas individuales: *rutaXXXIda.geojson*, *rutaXXXRegreso.geojson* (con *LineString*)

Paraderos: *paraderos.geojson* (con *Point* y propiedades como nombre del paradero)

## HeaderComponent

El HeaderComponent representa la barra superior (encabezado) de la aplicación MoviNeiva. Utiliza exclusivamente clases de Bootstrap 5 para su estilo visual, lo cual evita la necesidad de un archivo CSS personalizado.

Tiene un diseño sencillo y funcional que incluye:

Un logo en formato SVG (vectorial) alineado a la izquierda.

Un botón a la derecha que enlaza a la plataforma externa MIGO.

### *Análisis Visual y Funcional*

#### **Tabla 12**

*Análisis visual HeaderComponent.*

<b>Elemento HTML</b>	<b>Función</b>
<header>	Contenedor del encabezado con estilos Bootstrap (d-flex, p-3, etc.)
<svg>	Logo vectorial de la aplicación (blanco sobre fondo oscuro)
<button>	Botón que abre el sitio de <b>MIGO</b> en una nueva pestaña

Nota. En la tabla se presenta el análisis de los componentes HTML del componente HeaderComponent. Autoría propia

## MapComponent

El MapComponent es el componente principal de visualización geográfica en la aplicación MoviNeiva. Utiliza la biblioteca Leaflet para renderizar mapas interactivos, y está directamente vinculado al FilterComponent, que actúa como su componente hijo.

Este componente cumple múltiples funciones clave:

Inicializa el mapa centrado en Neiva.

107

Carga y muestra paraderos del transporte público desde un archivo GeoJSON.

Muestra rutas simples o combinadas (ida y vuelta) seleccionadas por el usuario.

Destaca visualmente barrios de origen y destino en el mapa.

Ofrece funciones de ubicación del usuario y seguimiento en tiempo real.

Se comunica con servicios como MapService y RoutesService para separar la lógica de datos del renderizado.

### ***Estructura del Componente***

Contenedor principal con posición absoluta para superponer el mapa, botones y filtro.

Botón para mostrar/ocultar el panel de filtros.

Botón para activar o detener el seguimiento de ubicación.

Contenedor del componente <app-filter>, que emite eventos al MapComponent.

Contenedor del mapa con id map.

### ***Propiedades***

***Tabla 13***

*Propiedades MapComponent*

<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
@ViewChild(FilterComponent)	FilterComponent	Referencia al componente hijo de filtros.
rutaActual	any	Capa Leaflet de la ruta actualmente visualizada.

Propiedad	Tipo	Descripción
mostrarFiltro	boolean	Controla la visibilidad del componente <app-filter>.
trackingStatus	boolean	Estado del seguimiento de la ubicación del usuario.

Nota. En la tabla se presenta las propiedades del componente MapComponent. Autoría propia

### ***Métodos***

***ngOnInit()*** Inicializa el mapa con MapService.initMap() y carga los paraderos con cargarParaderos.

***addRuta(nombreArchivo: string | string[])*** Carga una o dos rutas desde archivos GeoJSON usando RoutesService. Si es una ruta, elimina la anterior, la añade al mapa, ajusta el zoom y carga los paraderos cercanos. Si son dos rutas (origen y destino), las carga y visualiza ambas, además de mostrar la descripción en FilterComponent.

***mostrarRuta(ruta: any)*** Elimina cualquier ruta anterior, carga una nueva ruta desde un objeto GeoJSON, la añade al mapa, ajusta el zoom y carga paraderos cercanos.

***resaltarBarrios(barrios: any[])*** Elimina cualquier ruta anterior, obtiene los polígonos de los barrios seleccionados usando NeighborhoodService, y los pinta en verde (origen) y rojo (destino).

***toggleFiltro()*** Alterna la visibilidad del componente <app-filter>.

MapService.toggleUserTracking() y actualiza trackingStatus.

**Tabla 14**

*Eventos recibidos desde el componente FilterComponent*

<b>Evento</b>	<b>Método manejador</b>	<b>Descripción</b>
rutaSeleccionada	addRuta(\$event)	Carga una o dos rutas a partir de nombres de archivos.
rutaDesdeOrigenDestino	mostrarRuta(\$event)	Muestra una ruta generada entre origen y destino.
rutaDetectada	mostrarRuta(\$event)	Similar al anterior, muestra una ruta detectada por el filtro.
barriosSeleccionados	resaltarBarrios(\$event)	Resalta los barrios seleccionados.

Nota. En la tabla se presenta los eventos recibidos desde el componente

FilterComponent. Autoría propia

**Servicios utilizados**

**MapService:**

initMap(): Inicializa el mapa.

getMap(): Retorna la instancia de L.Map.

cargarParaderos(ruta: L.GeoJSON): Muestra los paraderos cercanos a una ruta.

toggleUserTracking(): Activa o desactiva el seguimiento del usuario.

`cargarRuta(nombre: string)`: Retorna un Observable con el contenido GeoJSON de la ruta solicitada.

***NeighborhoodService***

`getBarrio(nombre: string)`: Retorna el polígono GeoJSON de un barrio a partir de su nombre.

**FilterComponent**

El FilterComponent es el componente encargado de la gestión de filtros y parámetros de búsqueda en la aplicación Movineiva. Su principal función es permitir al usuario seleccionar una ruta, o bien definir un punto de origen y destino a partir de comunas, barrios y sitios representativos para mostrar la mejor ruta posible y sus barrios asociados.

Este componente se comunica directamente con el MapComponent a través de `@Output()` para emitir eventos con los datos seleccionados, y utiliza servicios para obtener las rutas disponibles y los barrios desde archivos GeoJSON.

***Estructura del componente******HTML (filter.component.html)***

Contenedor principal dividido en tres secciones: Inicio, Destino y Rutas.

Cada sección está separada visualmente y tiene su propio título y controles.

***Inicio***

Botón para usar la ubicación del usuario.

Combo para seleccionar comuna.

Combo (ng-select) para filtrar barrios según la comuna seleccionada.

111

### ***Destino***

Combo para seleccionar comuna.

Combo (ng-select) para filtrar barrios de destino.

### ***Rutas***

Selector de rutas predefinidas (ng-select).

Área de texto con la descripción de la ruta seleccionada.

Botón para limpiar todos los filtros y restaurar el estado inicial.

### ***Propiedades***

#### ***Tabla 15***

*Propiedades del componente FilterComponent.*

<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
@Output() rutaSeleccionada	EventEmitter<string>	Emite el nombre de la ruta seleccionada para que sea cargada en el mapa.
@Output() rutaDesdeOrigenDestino	EventEmitter<any>	Emite el objeto ruta generada a partir de origen y destino para ser renderizado.
@Output() barriosSeleccionados	EventEmitter<any[]>	Emite los nombres de los barrios seleccionados para ser destacados en el mapa.

<b>Propiedad</b>	<b>Tipo</b>	<b>Descripción</b>
@Output() rutaDetectada	EventEmitter<any>	Emite una ruta generada automáticamente desde los filtros.
rutasDisponibles	string[]	Lista de nombres de archivos GeoJSON de rutas cargadas desde assets/routes.
comunas	number[]	Lista única de comunas obtenidas desde el archivo barriosNeiva.geojson.
barrios	any[]	Lista completa de barrios con su información.
barriosOrigen / barriosDestino	any[]	Barrios filtrados según la comuna seleccionada para origen o destino.
sitiosOrigen / sitiosDestino	string[]	Sitios representativos extraídos de barrios seleccionados.
comunaOrigen / comunaDestino	number	Número de la comuna seleccionada como origen o destino.
barrioOrigen / barrioDestino	string	Nombre del barrio seleccionado como origen o destino.

Propiedad	Tipo	Descripción
sitioOrigen / sitioDestino	string	Nombre del sitio representativo seleccionado como origen o destino.
rutaSeleccionadaUsuario	string	Nombre de la ruta seleccionada desde el combo.

Nota. En la tabla se muestra la lista de propiedades del componente FilterComponent.

Autoría propia

### ***Métodos***

***ngOnInit()*** Carga las rutas disponibles desde assets/routes y obtiene los barrios para llenar las comunas únicas. Llama a obtenerBarrios() para extraer los datos de los barrios desde el servicio correspondiente.

***obtenerRutas()*** Utiliza RoutesService para buscar archivos de rutas disponibles y los almacena en rutasDisponibles.

***obtenerBarrios()*** Llama a NeighborhoodService.getBarrios() para cargar todos los barrios desde el archivo barriosNeiva.geojson. Extrae las comunas únicas y almacena la lista completa en barrios.

***filtrarBarrios(tipo: 'origen' | 'destino')*** Filtra los barrios disponibles según la comuna seleccionada (origen o destino). También limpia las selecciones previas de barrio y sitio.

***filtrarSitios(tipo: 'origen' | 'destino')*** Extrae los sitios representativos desde el barrio seleccionado (origen o destino). Los sitios están en la propiedad sitios del objeto del barrio.

***seleccionarRuta()*** Emite el evento `rutaSeleccionada` con el nombre de la ruta seleccionada por el usuario desde el combo. 114

***confirmarOrigenDestino()*** Emite dos eventos:

`rutaDesdeOrigenDestino` con los barrios de origen y destino.

`barriosSeleccionados` con un arreglo de los nombres de barrios para ser resaltados en el mapa.

***detectarRuta()*** Este método permite emitir el evento `rutaDetectada` con la información completa de la ruta detectada a partir de origen y destino, incluyendo comuna, barrio y sitio.

***limpiarFiltros()*** Reinicia todos los combos de filtros a su estado inicial, vacía listas dependientes y limpia las selecciones de ruta y ubicación.

### ***Eventos Emitidos al MapComponent***

***Tabla 16***

*Eventos emitidos al MapComponent*

<b>Evento</b>	<b>Método receptor en MapComponent</b>	<b>Descripción</b>
<code>rutaSeleccionada</code>	<code>addRuta(\$event)</code>	Solicita mostrar una ruta cargada desde su archivo GeoJSON.
<code>rutaDesdeOrigenDestino</code>	<code>mostrarRuta(\$event)</code>	Solicita mostrar una ruta calculada entre barrio origen y barrio destino.

Evento	Método receptor en MapComponent	Descripción
barriosSeleccionados	resaltarBarrios(\$event)	Envía los barrios seleccionados para ser destacados en el mapa.
rutaDetectada	mostrarRuta(\$event)	Similar al anterior, pero generado automáticamente desde filtros.

Nota. En la tabla se muestra los eventos emitidos al MapComponent. Autoría propia

### *Servicios Utilizados*

**RoutesService** getNombresRutas(): Retorna la lista de nombres de archivos de rutas .geojson disponibles.

**NeighborhoodService** getBarrios(): Retorna todos los barrios con su información, incluidos número de comuna y sitios representativos.

### **FooterComponent**

El componente FooterComponent representa el pie de página de la aplicación MoviNeiva. Contiene enlaces que abren modales informativos sobre el uso de la app, su propósito, y un formulario de retroalimentación. Estos modales utilizan Bootstrap 5 para su presentación y diseño.

### *Estructura del Componente*

Selector: app-footer

Archivo HTML: footer.component.html

Archivo TS: footer.component.ts

Bootstrap 5: Uso de modales con clases modal, modal-dialog, y comportamiento dinámico con Modal de Bootstrap JS.

ElementRef y ViewChild (Angular): Para capturar referencia al modal Instructivo y mostrarlo programáticamente tras la carga del componente.

**Funcionalidad Principal**

Mostrar un footer con enlaces interactivos.

Al hacer clic en los enlaces, se abren los siguientes modales:

Instructivo MoviNeiva: Guía de uso detallada de la aplicación.

Sobre MoviNeiva: Información general del propósito y visión del proyecto.

Formulario de Opinión: Espacio para que los usuarios dejen comentarios (HTML truncado).

En el método `ngAfterViewInit`, se inicializa y se muestra automáticamente el modal del instructivo (`#instructiveModal`).

**Comportamiento de los Modales**

Apertura automática: El modal de instructivo se abre automáticamente después de la carga del componente.

Control manual: Los otros dos modales (`#aboutModal` y `#feedbackModal`) se abren mediante los enlaces del footer.

**#instructiveModal** Contiene una guía paso a paso ilustrada sobre cómo utilizar la aplicación. Incluye:

Botones e imágenes clave del sistema (ubicación, seguimiento, filtros, rutas).

Explicaciones de cada sección: Inicio, Destino, Rutas, Búsqueda, Limpieza de filtros.

Scroll interno con altura máxima definida (max-height: 70vh).

**#aboutModal** Incluye una descripción general del objetivo de **MoviNeiva**, su versión actual, enfoque ciudadano y geográfico, así como un llamado a retroalimentación.

**#feedbackModal** Destinado a contener un formulario incrustado donde los usuarios pueden dejar sus comentarios. *(Nota: no compartiste el HTML completo de este modal.)*

### Requisitos previos

Tener una cuenta en GitHub y un repositorio creado

Proyecto Angular configurado correctamente

Git y Node.js instalados en tu máquina

### Implementación de cambios en el aplicativo

#### *Trabajar sobre la rama de desarrollo (master)*

Realizar todos los cambios necesarios y verificar que el proyecto funcione correctamente.

Guardar los cambios con Git:

```
git add .
```

```
git commit -m "Actualización de funcionalidad"
```

```
git push origin master
```

#### *Sincronizar cambios en la rama principal (main)*

Cambiar a la rama principal:

```
git checkout main
```

```
git merge master
```

```
git push origin main
```

#### *Construir la aplicación para producción*

Ejecutar el siguiente comando para compilar con configuración de producción y la ruta base correspondiente:

```
ng build --configuration production --base-href /movineiva/
```

Instalar la herramienta globalmente:

```
npm install -g angular-cli-ghpages
```

Publicar la app ejecutando:

```
npx angular-cli-ghpages --dir=dist/movineiva
```

Esto creará automáticamente una rama llamada gh-pages y subirá los archivos generados.

### ***Configurar GitHub Pages***

Dirigirse al repositorio en GitHub → Settings (Configuración) → Pages

En la sección Source (Fuente):

Seleccionar Deploy from a branch

Elegir la rama gh-pages

Carpeta raíz: / (root)

GitHub generará la URL pública del sitio, por ejemplo:

<https://walejandros.github.io/movineiva/>

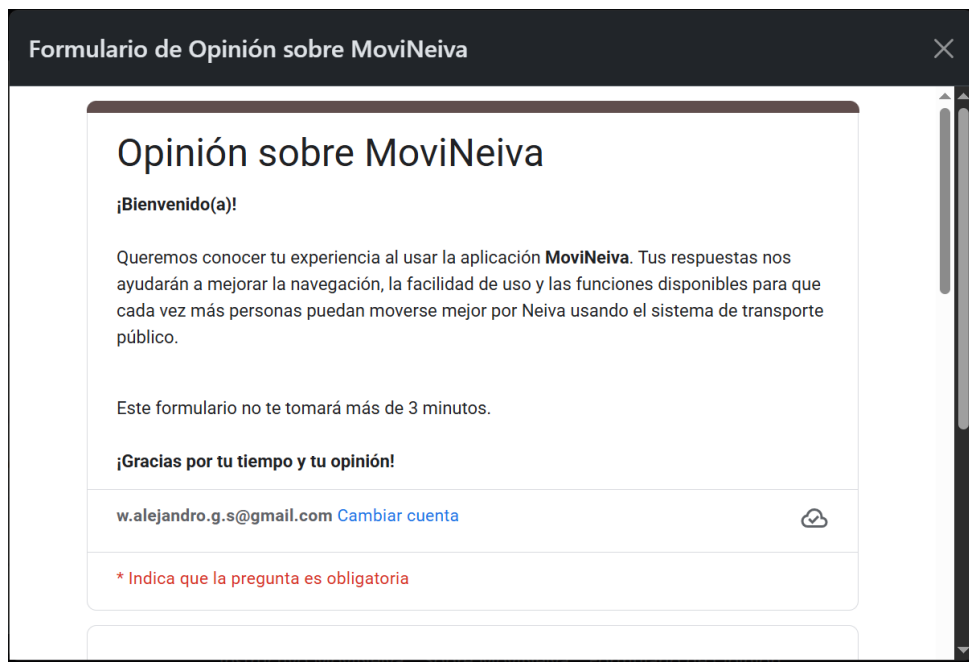
### Integración del formulario de retroalimentación

Se realizó la integración del formulario realizado en una modal del aplicativo web, el cual se puede encontrar en el footer, el cual permite a los usuarios nuevos dejar sus comentarios, sugerencias o calificaciones sobre su experiencia con la aplicación.

Enlace: <https://forms.gle/5UUM5vg4PXrioHCUA>

### Figura 38

Modal formulario de opinión sobre MoviNeiva



Nota. La figura muestra la ventana modal con el formulario de opinión, el cual permite a los usuarios nuevos compartir su experiencia y sugerencias sobre el uso de la aplicación MoviNeiva. Autoría propia

### ***Media ( $\mu$ )***

Sirve como indicador del valor promedio que los usuarios asignaron a una experiencia.

Una media alta (cercana al valor máximo posible en la escala) indica una valoración generalmente positiva.

Una media baja sugiere experiencias menos satisfactorias o confusión.

### ***Desviación estándar ( $\sigma$ )***

Mide qué tanto se alejan las respuestas individuales del promedio.

Desviación baja: Las respuestas están concentradas cerca del valor promedio → hay consenso.

Desviación alta: Las respuestas están más dispersas → hay opiniones variadas.

### ***Varianza y varianza ponderada***

Evalúan la dispersión de respuestas con más profundidad, considerando su frecuencia (ponderada).

Permiten detectar qué tan estables son las opiniones por ítem.

Una varianza ponderada baja indica que la percepción fue homogénea y consistente.

Una alta sugiere que algunos usuarios experimentaron situaciones distintas (positivas o negativas).

*Interpretación de los resultados por tipo de pregunta.*

<b>Tipo de pregunta</b>	<b>Cómo interpretar sus estadísticas</b>
<b>Acceso y apertura de la app</b>	Se analiza la facilidad de uso inicial. Una media alta y varianza baja indican una curva de aprendizaje rápida.
<b>Uso del mapa y filtros</b>	<p>Evalúa si los elementos interactivos del aplicativo son intuitivos. Una baja dispersión aquí sugiere buena usabilidad sin necesidad de mucha exploración.</p> <p>Se debe verificar si la funcionalidad técnica es reconocida y</p>
<b>Botón de seguimiento</b>	<p>comprendida. La varianza baja es clave para interpretar esto como éxito en implementación.</p> <p>Requiere observar si la interfaz guía de manera clara hacia</p>
<b>Búsqueda de rutas</b>	<p>resultados útiles. Si hay una desviación o varianza mayor, se detectan dificultades aisladas en interpretación.</p> <p>Se interpreta su utilidad percibida. Una media alta y consistencia en las respuestas validan su efectividad como material de apoyo.</p>
<b>Instructivo</b>	<p>Aquí importa cómo los elementos gráficos facilitan la</p>
<b>Claridad visual</b>	<p>comprensión. La dispersión baja refuerza que hay un diseño intuitivo.</p>

---

<b>Tipo de pregunta</b>	<b>Cómo interpretar sus estadísticas</b>
<b>Tiempo de respuesta</b>	Debe analizarse desde la percepción de eficiencia. Si hay dispersión, puede haber factores externos (como red o dispositivo) que influyen.
<b>Recomendación y satisfacción</b>	Estas preguntas resumen la experiencia global. Una concentración fuerte en las categorías más altas implica cumplimiento exitoso de expectativas.

---

Nota. En la tabla se muestra una guía de cómo se deben interpretar los resultados del formulario de opinión. Autoría propia

Evaluar la funcionalidad, usabilidad y desempeño de la aplicación MoviNeiva, considerando tanto la experiencia directa de usuarios reales como pruebas técnicas manuales en condiciones simuladas.

### **Pruebas de Usabilidad con Usuarios**

Estas pruebas se enfocaron en validar la experiencia del usuario final a través de un cuestionario posterior al primer uso. Se recopilieron datos cuantitativos y cualitativos con los siguientes fines:

Identificar fortalezas y oportunidades de mejora en la interfaz y navegación.

Medir tiempos de respuesta, tasa de éxito en tareas clave y nivel de satisfacción general.

Resumen de métricas obtenidas

#### ***Tabla 18***

##### *Resumen de las métricas obtenidas*

Métrica Evaluada	Resultado
Participantes	34 usuarios
Tasa de éxito promedio por tarea	90 %
Tiempo medio de consulta de ruta	< 2 minutos (en 91,2 % de los casos)
Nivel de recomendación	97,1 %
Satisfacción general	100 % satisfecho o muy satisfecho

Nota. En la tabla se muestra un resumen de las métricas obtenidas en los resultados del formulario de opinión. Autoría propia

Estas cifras evidencian una experiencia satisfactoria, intuitiva y accesible, con 125 una alta percepción de utilidad y facilidad de uso, especialmente desde dispositivos móviles.

### **Pruebas Manuales de Funcionalidad**

Se realizaron validaciones manuales del sistema en un entorno controlado con los siguientes objetivos:

- Verificar el cumplimiento de los requerimientos funcionales y no funcionales.
- Simular escenarios reales de uso y detectar posibles errores o barreras técnicas.
- Asegurar la integridad de las funcionalidades clave de navegación, visualización y seguimiento.

Aspectos evaluados:

- Despliegue automático del instructivo al acceder por primera vez.
- Uso del botón “Mi ubicación” para geolocalización automática.
- Filtrado dinámico de barrios según la comuna seleccionada.
- Selección de rutas específicas sin necesidad de ingreso de puntos de inicio/destino.
- Visualización de rutas directas y combinadas, con resaltado de barrios involucrados.
- Seguimiento en tiempo real de la posición del usuario sobre el mapa.

Estas pruebas permitieron comprobar la operatividad integral del sistema y garantizar una experiencia funcional fluida. Además, facilitaron ajustes técnicos previos al despliegue público.

La combinación de pruebas con usuarios reales y validaciones manuales asegura una evaluación integral de la aplicación MoviNeiva. Los resultados positivos en usabilidad y satisfacción, junto con el correcto funcionamiento técnico de los módulos probados, validan la preparación del sistema para su implementación y posible escalamiento.

Para asegurar la continuidad, trazabilidad y calidad del desarrollo de MoviNeiva, se establecen las siguientes pautas para el mantenimiento evolutivo, el control de versiones, y el manejo de incidencias:

### Esquema de versionado (SemVer)

Se adopta el esquema SemVer (Semantic Versioning) en el siguiente formato:

MAJOR.MINOR.PATCH:

MAJOR: cambios incompatibles con versiones anteriores (refactor completo, rediseño).

MINOR: nuevas funcionalidades compatibles con versiones previas.

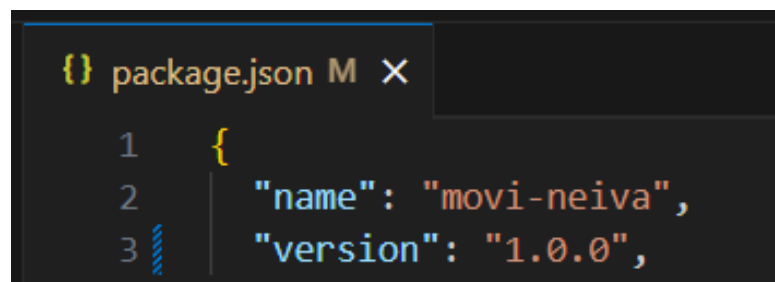
PATCH: correcciones de errores y mejoras menores sin cambios en funcionalidades.

Ejemplo: v1.2.3 → primera versión principal, dos nuevas funcionalidades agregadas, tres errores corregidos.

Este indicativo se usará en el archivo package.json en el atributo “version”

### **Figura 39**

*Archivo package.json en el atributo “version”*



```
{ } package.json M X
1  {
2    "name": "movi-neiva",
3    "version": "1.0.0",
```

Nota. La figura muestra el atributo “version” en la cual se maneja el versionamiento 128 de MoviNeiva. Autoría propia

### Generación de changelogs

Se utilizará el archivo CHANGELOG.md en la raíz del repositorio para documentar cambios relevantes. Su estructura seguirá el siguiente formato:

```
## [1.2.3] – 2025-07-03
```

```
### Added
```

- Funcionalidad de búsqueda por sitios representativos

```
### Fixed
```

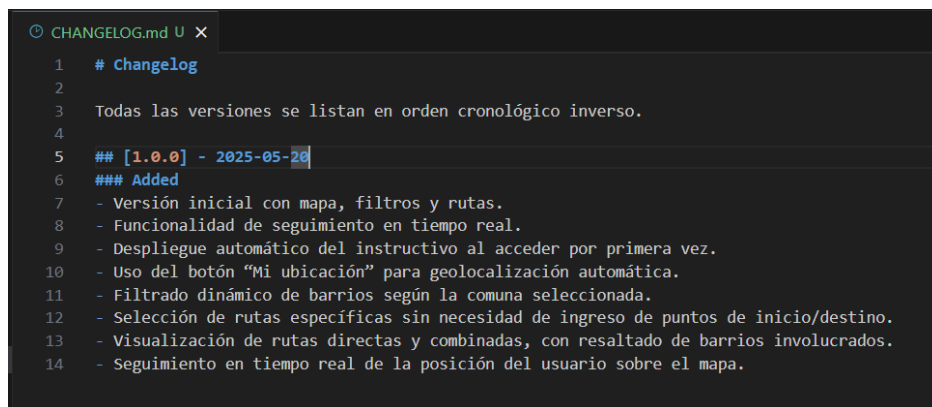
- Corrección en visualización de rutas combinadas

```
### Changed
```

- Ajuste en la ubicación automática del botón de seguimiento

### Figura 40

Archivo CHANGELOG.mb



```
CHANGELOG.md U X
1 # Changelog
2
3 Todas las versiones se listan en orden cronológico inverso.
4
5 ## [1.0.0] - 2025-05-20
6 ### Added
7 - Versión inicial con mapa, filtros y rutas.
8 - Funcionalidad de seguimiento en tiempo real.
9 - Despliegue automático del instructivo al acceder por primera vez.
10 - Uso del botón “Mi ubicación” para geolocalización automática.
11 - Filtrado dinámico de barrios según la comuna seleccionada.
12 - Selección de rutas específicas sin necesidad de ingreso de puntos de inicio/destino.
13 - Visualización de rutas directas y combinadas, con resaltado de barrios involucrados.
14 - Seguimiento en tiempo real de la posición del usuario sobre el mapa.
```

Nota. La figura muestra el archivo CHANGELOG.md . Autoría propia

Se configurará el repositorio de GitHub con plantillas (issue templates) para facilitar el reporte estructurado de errores y mejoras.

### *Plantilla sugerida de errores*

## Descripción del error

Breve descripción del comportamiento inesperado.

## Pasos para reproducir

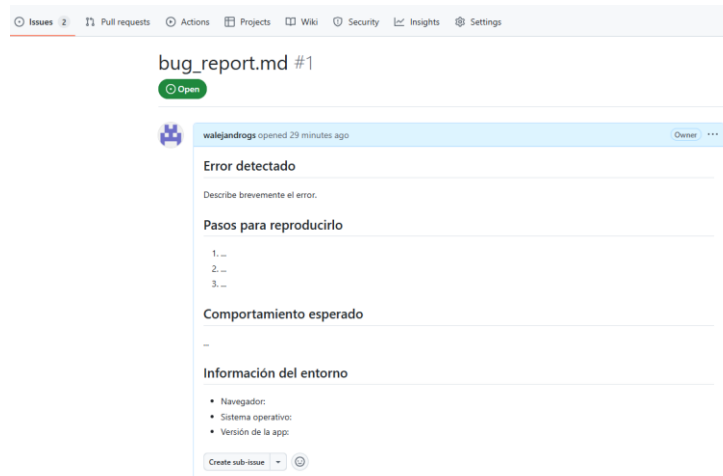
1. Ir a '...'
2. Hacer clic en '...'
3. Ver error

## Comportamiento esperado

Describe qué esperabas que sucediera.

## Información del entorno

- Navegador:
- Sistema operativo:
- Versión de la app:

**Figura 41****Plantilla de reportes de bugs en GitHub Issues**

Nota. La figura muestra la plantilla de reportes de bugs en GitHub Issues. Autoría propia

**Plantilla de solicitud de mejora**

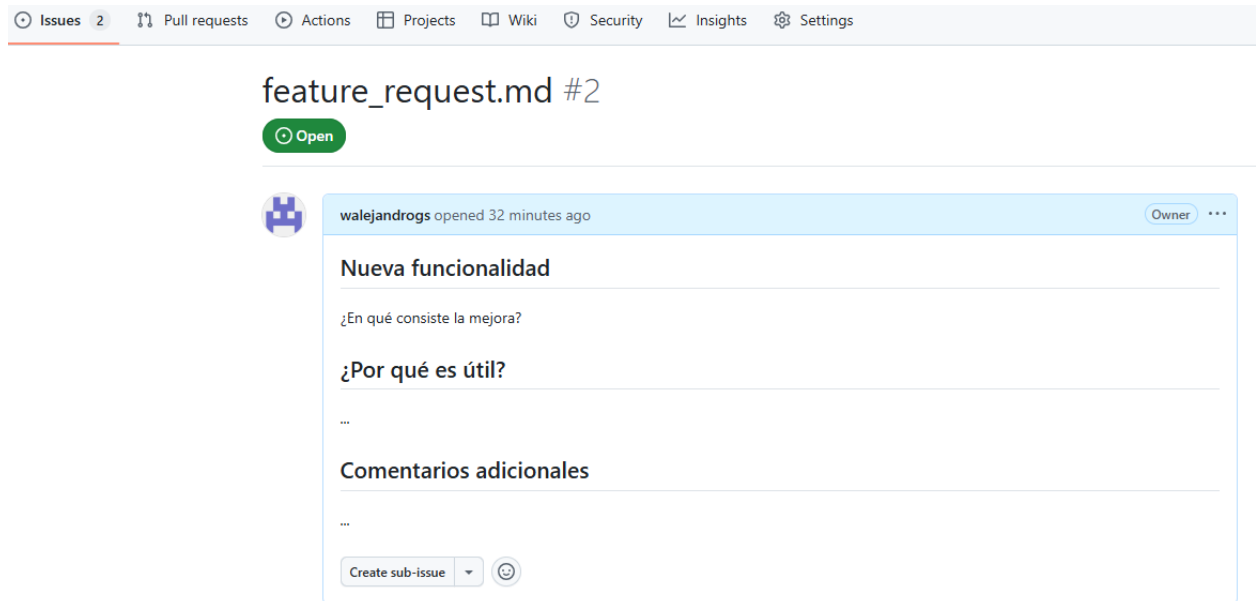
## Nueva funcionalidad

¿En qué consiste la mejora?

## ¿Por qué es útil?

...

## Comentarios adicionales

*Plantilla de reportes de sugerencias en GitHub Issues*

Nota. La figura muestra la plantilla de sugerencias en GitHub Issues. Autoría propia

**Buenas prácticas para futuras actualizaciones**

Cada nueva función o corrección que subas debe estar asociada a una rama:

feature/<nombre> para mejoras nuevas

bugfix/<nombre> para errores

Antes de hacer merge a la rama main, asegúrate de:

Probarlo localmente

Actualizar el CHANGELOG.md

Cambiar la versión en package.json