

**APLICATIVO MÓVIL PARA GUIAR A UNA PERSONA CON DISCAPACIDAD
VISUAL EN AMBIENTES EXTERIORES USANDO REALIMENTACIÓN DE
AUDIO**

DANIEL GONZÁLEZ HERNÁNDEZ

Código: 201925283

**Universidad del Valle
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica y Electrónica
Santiago de Cali
2025**

**APLICATIVO MÓVIL PARA GUIAR A UNA PERSONA CON DISCAPACIDAD
VISUAL EN AMBIENTES EXTERIORES USANDO REALIMENTACIÓN DE
AUDIO**

DANIEL GONZÁLEZ HERNÁNDEZ

Código: 201925283

**Proyecto de grado presentado al programa de
Ingeniería Electrónica (3744) para optar por el
título de: Ingeniero Electrónico**

Director:

Eval Bladimir Bacca Cortés, Ph.D.

**Grupo de Investigación Percepción y Sistemas
Inteligentes**

**Universidad del Valle
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica y Electrónica
Santiago de Cali
2025**

Nota de aceptación:

Firma del director del trabajo

Firma del evaluador

Firma del evaluador

RESUMEN

En el mundo, se estima que cientos de millones de personas viven con discapacidad visual, y una parte importante enfrenta barreras para desplazarse de forma autónoma en entornos urbanos complejos. La movilidad independiente en exteriores exige soluciones de guiado robustas y accesibles, especialmente en escenarios de baja cobertura o cañón urbano, donde los errores de GNSS aumentan de forma significativa y pueden comprometer la seguridad del desplazamiento.

Este trabajo presenta el desarrollo de GuiaMe, una aplicación móvil accesible para Android que proporciona navegación paso a paso en rutas peatonales mediante retroalimentación de audio y vibración. A partir de los requerimientos funcionales definidos, se implementaron tres módulos principales: configuración (selección de idioma, tipo de mapa online/offline y tipo de retroalimentación), gestión de destinos (creación, edición, almacenamiento y selección de puntos de interés) y navegación (seguimiento de la ruta en tiempo real, indicaciones giro a giro y alertas ante desvíos). La app incorpora un modo de acompañante para la configuración inicial (destino y preferencias) y un modo de usuario con interfaz simplificada y compatible con lector de pantalla.

Para atenuar saltos de posición y mejorar la continuidad del guiado se evaluaron distintos métodos de filtrado sobre la señal de localización (filtro de Kalman lineal, filtro de Kalman extendido y filtro de partículas), seleccionándose este último por su mejor desempeño. En las pruebas de campo realizadas en rutas peatonales internas de la Universidad del Valle se registraron trayectorias GNSS crudas y filtradas y se aplicaron encuestas de usabilidad a los participantes. El análisis punto a punto del RMSE del error entre la posición GNSS cruda y la posición estimada por el filtro muestra que, en la mayor parte del recorrido, las correcciones se mantienen en el orden de 1–2 m, aunque existen tramos específicos donde el error se incrementa debido a la variabilidad de la señal GNSS. Además, la mayoría de los usuarios valoró positivamente la claridad de las indicaciones y la facilidad de uso de la aplicación, lo que valida el prototipo y señala la necesidad de continuar mejorando la navegación en escenarios con mayor degradación de la señal, en particular en el modo offline.

Palabras clave: discapacidad visual; navegación asistida; retroalimentación de audio; GNSS; cañón urbano; filtro de partículas; accesibilidad.

ABSTRACT

Worldwide, hundreds of millions of people live with visual impairments, and many of them face significant barriers to moving independently in complex urban environments. Independent outdoor mobility requires robust and accessible guidance solutions, especially in scenarios with low satellite visibility or urban canyons where GNSS errors increase substantially and may compromise safe navigation.

This work presents the development of GuiaMe, an accessible Android mobile application that provides step-by-step pedestrian navigation through audio and vibration feedback. Based on the functional requirements defined, three main modules were implemented: configuration (selection of language, online/offline map mode, and type of feedback), destination management (creation, editing, storage, and selection of points of interest), and navigation (real-time route tracking, turn-by-turn instructions, and deviation alerts). The app also includes a companion mode for initial setup (destination and user preferences) and a user mode with a simplified interface compatible with screen readers.

To reduce position jumps and improve the continuity of guidance, different filtering methods were evaluated (linear Kalman filter, extended Kalman filter, and particle filter), the latter being selected due to its superior performance. Field tests were conducted on pedestrian routes inside the Universidad del Valle campus, recording both raw and filtered GNSS trajectories and collecting usability feedback from participants. The point-by-point RMSE analysis of the error between the raw GNSS position and the position estimated by the filter shows that, along most of the route, corrections remain within 1–2 m, although specific segments exhibit higher error due to GNSS variability. Additionally, participants positively rated the clarity of the instructions and the overall ease of use, validating the prototype and highlighting the need to further improve navigation in scenarios with degraded signal conditions, particularly in offline mode.

Keywords: visual impairment; assisted navigation; audio feedback; GNSS; urban canyon; particle filter; accessibility.

Tabla de contenido

ÍNDICE DE FIGURAS.....	viii
ÍNDICE DE TABLAS.....	ix
1 CAPÍTULO 1: INTRODUCCIÓN	1
1.1 INTRODUCCIÓN	1
1.2 PLANTEAMIENTO DEL PROBLEMA	3
1.3 OBJETIVOS.....	4
1.3.1 OBJETIVO GENERAL.....	4
1.3.2 OBJETIVOS ESPECIFICOS.....	4
1.4 SOLUCIÓN PROPUESTA	5
1.5 ESTRUCTURA DEL LIBRO.....	6
2 CAPÍTULO 2: ANTECEDENTES Y MARCO TEÓRICO.....	7
2.1 INTRODUCCIÓN	7
2.2 ANTECEDENTES	7
2.3 MARCO TEÓRICO	15
2.3.1. Discapacidad Visual	15
2.3.2. Tecnologías de asistencia para la navegación de personas con discapacidad visual.....	16
2.3.3. Tecnologías de geolocalización	19
2.3.4. Métodos de filtrado de señales GPS/GPRS.....	22
2.3.5. Aplicaciones móviles accesibles y retroalimentación de audio.....	25
2.3.6. Metodología de desarrollo de software SCRUM	26
2.4 OBSERVACIONES FINALES.....	26
3 CAPÍTULO 3: DESARROLLO DEL APLICATIVO MÓVIL	28
3.1 INTRODUCCIÓN	28
3.2 DISEÑO DEL APLICATIVO MÓVIL GUIAME.....	29
3.2.1 Requerimientos funcionales y no funcionales	29
3.2.2 Diagrama de conceptos.....	31
3.2.3 Diagrama de clases	32
3.3 IMPLEMENTACIÓN DEL APLICATIVO MÓVIL GUIAME	34
3.3.1 Diagrama de bloques del aplicativo	35
3.3.2 Módulo la interfaz inicial y de configuración.....	35
3.3.3 Módulo de gestión de destinos	39
3.3.4 Módulo de adquisición y preprocesamiento de la señal GPS/IMU	45

3.3.5	Módulo de procesamiento y filtrado de la navegación.....	48
3.3.6	Módulo de realimentación. (TurnCueManager, FeedbackManager, tipos de feedback: voz y pitidos)	53
3.3.7	Integración de módulos y flujo completo de navegación (NavigationActivity”) 59	
3.4	CONCLUSIONES Y OBSERVACIONES FINALES	63
4	CAPÍTULO 4: RESULTADOS Y ANÁLISIS.....	65
4.1	INTRODUCCIÓN	65
4.2	PRUEBAS DE INTEGRACIÓN	65
4.2.1	Módulo de configuración.....	66
4.2.2	Módulo de gestión de destinos	69
4.2.3	Módulos de adquisición y preprocesamiento de la señal GPS/IMU y de procesamiento y filtrado de la navegación.	72
4.2.4	Módulos de navegación y de realimentación auditiva	74
4.3	PRUEBAS DE CAMPO.....	77
4.3.1	Resultados cuantitativos	79
4.3.2	Resultados de satisfacción	81
4.4	DISCUSIÓN DE RESULTADOS FRENTE A LOS ANTECEDENTES	86
4.5	CONCLUSIONES Y OBSERVACIONES	87
5	CAPÍTULO 5: CONCLUSIONES Y TRABAJOS FUTUROS	89
5.1	CONCLUSIONES GENERALES	89
5.2	TRABAJOS FUTUROS	91
6	REFERENCIAS	92
7	ANEXOS	95

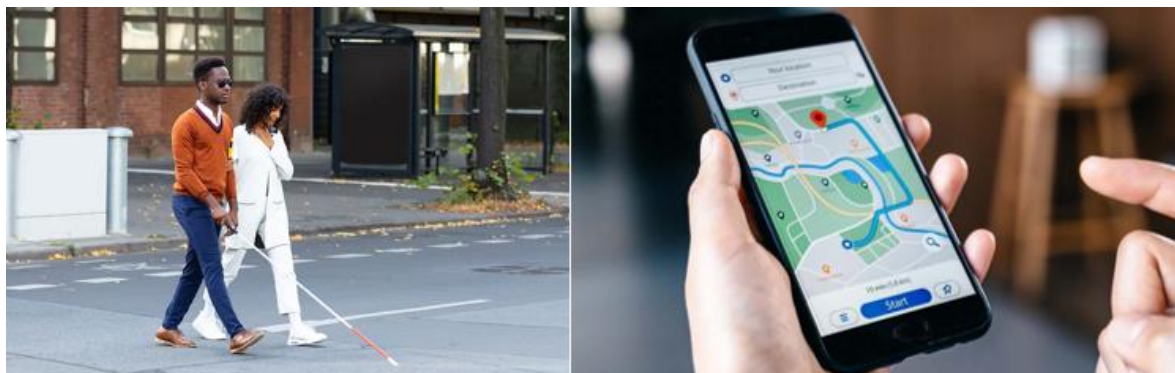
ÍNDICE DE FIGURAS

FIGURA 1.1. INTERFACES CORRESPONDIENTES A LAS APLICACIONES DE ACCESIBILIDAD LAZARILLO [3] Y SEEING ASSISTANT MOVE [4] RESPECTIVAMENTE.....	2
FIGURA 1.2 ILUSTRACIÓN DE LA SOLUCIÓN PROPUESTA	5
FIGURA 2.1 HERRAMIENTAS DESARROLLADAS PARA LA GUÍA Y ASISTENCIA DE PERSONAS CON DISCAPACIDAD VISUAL. A LA IZQUIERDA UNA INTERFAZ DE UNA APLICACIÓN MÓVIL PARA BRINDAR AYUDA GENERAL [10] Y A LA DERECHA UN PROTOTIPO PARA LA DESCRIPCIÓN AUDITIVA DEL ESCENARIO [11].	7
FIGURA 2.2 GRADOS DE DISCAPACIDAD VISUAL SEGÚN LA ORGANIZACIÓN MUNDIAL DE LA SALUD, RESUMIDOS EN [31].	16
FIGURA 2.3 ESQUEMA DE TECNOLOGÍAS TÍPICAS ETAS RESUMIDAS EN [35].	17
FIGURA 2.4 INTERFAZ CORRESPONDIENTE A LA APLICACIÓN LAZARILLO (EOA) [3].	18
FIGURA 2.5 INTERFAZ DE UNA APLICACIÓN MÓVIL PARA BRINDAR AYUDA GENERAL INCLUYENDO MÓDULOS EXTERNOS DE DETECCIÓN, CONSTRUIDA EN [10].	19
FIGURA 2.6 SEGMENTOS ESPACIO, CONTROL Y USUARIO. PARTES DEL GNSS ELABORADA EN [37].	20
FIGURA 2.7 ALGORITMO DEL FILTRO DE KALMAN [40].	22
FIGURA 2.8 FASES DE LA METODOLOGÍA SCRUM [42].	26
FIGURA 3.1 INTERFAZ DE LA APLICACIÓN ANDROID STUDIO PARA DESARROLLAR APLICACIONES MÓVILES.	28
FIGURA 3.2 DIAGRAMA DE CONCEPTOS.	32
FIGURA 3.3 DIAGRAMA DE CLASES SIMPLIFICADO DEL APLICATIVO GUIAME	33
FIGURA 3.4 DIAGRAMA DE BLOQUES DEL APLICATIVO	35
FIGURA 3.5 PANTALLA INTERFAZ INICIAL.	36
FIGURA 3.6 PANTALLA DE CONFIGURACIÓN.	38
FIGURA 3.7 PANTALLA DE GESTIÓN DE DESTINOS.	40
FIGURA 3.8 PANTALLA DE ADICIÓN/EDICIÓN DE DESTINOS.	42
FIGURA 3.9 LISTA DE DESTINOS PARA INICIAR LA NAVEGACIÓN	44
FIGURA 3.10 SOLICITUD DE PERMISOS DE UBICACIÓN.	45
FIGURA 3.11 RUTA CALCULADA EN EL MAPA ONLINE.	51
FIGURA 3.12 SELECTOR DEL TIPO DE REALIMENTACIÓN.	55
FIGURA 3.13 PANTALLA DE NAVEGACIÓN EN EJECUCIÓN.	60
FIGURA 4.1 PUESTA EN MARCHA DEL APLICATIVO.	65
FIGURA 4.2 PRUEBAS DE CAMPO DEL APLICATIVO GUIAME REALIZADAS EN LA UNIVERSIDAD DEL VALLE.	78
FIGURA 4.3 RUTA DEFINIDA PARA LA PRUEBA DE CAMPO DEL MAPA ONLINE (IZQUIERDA) Y OFFLINE (DERECHA).	78
FIGURA 4.4 HISTOGRAMA DEL ERROR ENTRE LA POSICIÓN GPS CRUDA Y LA POSICIÓN ESTIMADA POR EL FILTRO.	79
FIGURA 4.5 RMSE DEL ERROR CRUDO–FILTRADO EN FUNCIÓN DE LA DISTANCIA A LO LARGO DE LA RUTA PARA TRAYECTORIAS DE IDA Y DE VUELTA.	80
FIGURA 4.6 RESUMEN DE RESULTADOS DE LA ENCUESTA DE EVALUACIÓN DEL APLICATIVO GUIAME.	83
FIGURA 7.1 ÁRBOL DE DIRECTORIO DE LOS ANEXOS.	95

ÍNDICE DE TABLAS

TABLA 2-1 ANTECEDENTES.	8
TABLA 3-1 REQUERIMIENTOS FUNCIONALES.	29
TABLA 3-2 REQUERIMIENTOS NO FUNCIONALES.	30
TABLA 4-1 PRUEBA DE INTEGRACIÓN - MÓDULO INICIO Y CONFIGURACIÓN.	66
TABLA 4-2 PRUEBA DE INTEGRACIÓN - MÓDULO GESTIÓN DE DESTINOS.	69
TABLA 4-3 PRUEBA DE INTEGRACIÓN - MÓDULO DE ADQUISICIÓN Y PREPROCESAMIENTO DE LA SEÑAL GPS/IMU Y DE PROCESAMIENTO Y FILTRADO DE LA NAVEGACIÓN.	72
TABLA 4-4 PRUEBA DE INTEGRACIÓN – MÓDULO DE NAVEGACIÓN Y REALIMENTACIÓN.	74
TABLA 4-5 RESUMEN ERROR CRUDO VS FILTRADO.	79
TABLA 4-6 PREGUNTAS DE LA ENCUESTA.	81
TABLA 4-7 RESUMEN DE LA ENCUESTA SOBRE LAS PRUEBAS DE CAMPO.	83

1 CAPÍTULO 1: INTRODUCCIÓN



1.1 INTRODUCCIÓN

Según datos del Departamento Administrativo Nacional de Estadística (DANE) [1], en Colombia hay aproximadamente 1.948.332 personas con discapacidad visual, lo que representa el 62,17% de la población con discapacidad en el país. Esta condición repercute de manera directa en su autonomía y movilidad cotidiana, tanto en entornos urbanos como rurales, al enfrentar barreras físicas (infraestructura peatonal discontinua, señalización limitada), informacionales (ausencia de señales audibles y cartografía accesible) y sociales (dependencia de terceros y estigmas). Investigaciones realizadas por la Universidad del Valle han caracterizado las condiciones socioeconómicas y de participación social de personas con discapacidad visual, resaltando la importancia de la tecnología en la mejora de su calidad de vida al potenciar la independencia, la orientación espacial y la toma de decisiones durante el desplazamiento [2].

En la actualidad, la tecnología de geolocalización es un soporte clave para la movilidad de la población en general y, en particular, de las personas con discapacidad visual. Estas, además de depender de la asistencia de una persona vidente, recurren a aplicaciones de navegación como Lazarillo [3], Seeing Assistant Move [4] y Google Maps [5] mostradas en la Figura 1.1, que ofrecen funcionalidades de accesibilidad mejorada (anuncios por voz, puntos de interés accesibles, rutas peatonales) para desplazarse con mayor seguridad.

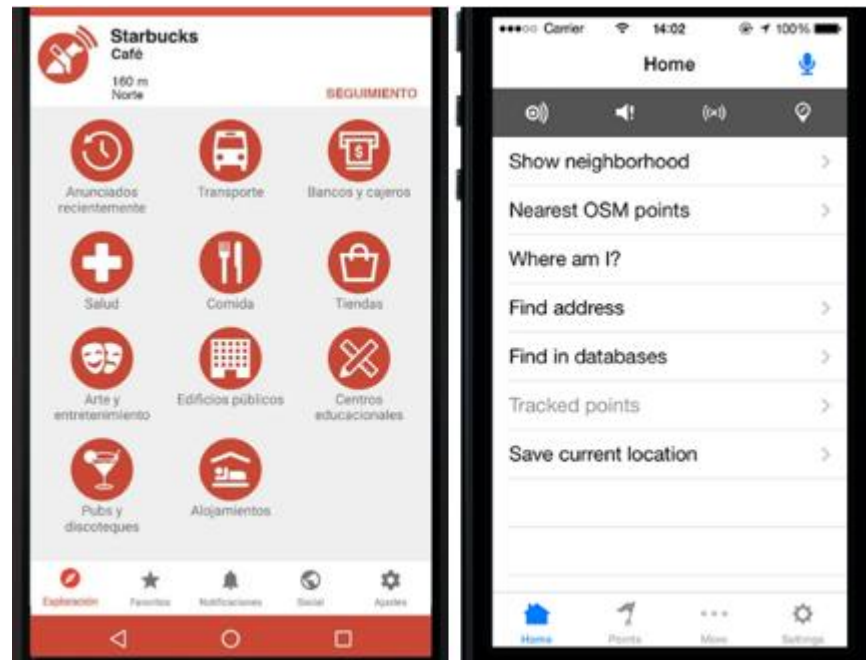


Figura 1.1. Interfaces correspondientes a las aplicaciones de accesibilidad Lazarillo [3] y Seeing Assistant Move [4] respectivamente.

No obstante, la experiencia real de navegación no depende solo de la interfaz: está condicionada por la calidad de la señal, la disponibilidad de datos móviles, el estado de la infraestructura (andén continuo, rampas, señalización audible) y el contexto ambiental (tráfico, lluvia, ruido), todos factores que pueden comprometer la orientación. Específicamente, en muchos entornos urbanos y rurales, la señal GPS/GPRS presenta deficiencias debido a la baja cobertura [6], cañones urbanos [7], vegetación densa o interferencias ambientales, generando sobresaltos en la localización que pueden afectar la precisión de la asistencia brindada por estos sistemas, cuya eficacia radica en mantener una retroalimentación continua y confiable al usuario.

Con esta propuesta de trabajo de grado se busca mediante el desarrollo de una aplicación móvil, implementar un sistema de filtrado de la señal GPS/GPRS que permitirá mejorar la estabilidad de la localización, corrigiendo errores y sobresaltos para ofrecer una navegación más precisa y constante para los usuarios con discapacidad visual. Además, se explorará la integración con sensores de movimiento (IMU) para mejorar la estimación de posición cuando la señal GPS/GPRS sea deficiente.

Este trabajo de grado surge del interés por contribuir al bienestar de las personas mediante la implementación de soluciones tanto en software como en hardware, con el propósito de mejorar su calidad de vida. En este sentido, la propuesta no solo

busca desarrollar una herramienta tecnológica, sino también aportar al fortalecimiento de la inclusión social y la autonomía de las personas con discapacidad visual, generando un impacto positivo en su interacción con el entorno.

Con el fin de tener un punto de referencia en el desarrollo de esta propuesta, se realiza una revisión detallada de antecedentes relacionados con el desarrollo de aplicaciones para usuarios con discapacidad visual y de filtrado y corrección de lecturas de señales GPS/GPRS. De igual forma, en el marco teórico se presenta un resumen de los conceptos relacionados con el filtrado de señales de GPS/GPRS, esto es los métodos generales usados para su resolución, así como la metodología SCRUM para el desarrollo de software, con el propósito de sentar las bases conceptuales requeridas para el desarrollo del trabajo de grado.

Este documento empieza por el planteamiento del problema para contextualizar la problemática a resolver y mejorar, posteriormente se presenta el objetivo general y los específicos. Seguido a los objetivos, se plantea una metodología en donde se propone un conjunto de actividades distribuidas en un periodo de 9 meses (2 semestres académicos), que aporte al cumplimiento de cada objetivo específico y se resume de forma gráfica en el cronograma. Finalmente, se presenta un estimado de los costos para la ejecución del proyecto.

1.2 PLANTEAMIENTO DEL PROBLEMA

Actualmente, las personas con discapacidad visual enfrentan grandes desafíos al desplazarse en entornos urbanos y rurales. Para mejorar su movilidad y autonomía como parte de su bienestar, se han desarrollado diversas aplicaciones móviles basadas en GPS/GPRS que proporcionan asistencia en la navegación en espacios exteriores. Sin embargo, estas aplicaciones como Lazarillo, Seeing Assistant Move y Google Maps con accesibilidad mejorada dependen en gran medida de la precisión y estabilidad de la señal GPS/GPRS, lo que representa una limitación significativa en regiones con baja cobertura, interferencias ambientales o entornos urbanos con estructuras que contribuyen a la obstrucción de la señal [8].

El problema radica en que la señal GPS/GPRS no siempre es precisa debido a diversos factores como la multipropagación (multipath), la interferencia de edificios altos, el bloqueo de señal en túneles o zonas con densa vegetación, la baja cobertura en áreas rurales y la deficiente señal en el territorio. Estas condiciones generan fluctuaciones en la señal, provocando sobresaltos en la ubicación estimada, cambios abruptos en la posición del usuario y errores en la dirección sugerida por la aplicación. Para una persona con discapacidad visual,

estos errores pueden representar un riesgo significativo, ya que una indicación incorrecta puede llevarlos a tomar rutas peligrosas, cruzar calles en puntos no seguros o perder la referencia de su ubicación real [9].

Los métodos tradicionales de corrección de señal de GPS/GPRS, pueden no ser suficientes para mitigar los errores, ya que no tienen en cuenta las zonas de sombra de la señal ni la fusión de información con otros sensores disponibles en los dispositivos móviles. Es por ello que en este trabajo se propone desarrollar una aplicación móvil para la navegación en exteriores considerando un filtro para la señal de GPS/GPRS, el cual permitirá realizar estimaciones más estables al combinar los datos del GPS/GPRS con sensores inerciales como las IMU [9].

A partir de lo expuesto anteriormente se plantea resolver la siguiente pregunta de ingeniería:

¿Cómo desarrollar un aplicativo móvil para guiar a una persona con discapacidad visual en ambientes exteriores usando realimentación de audio filtrando la señal GPS/GPRS en zonas donde ésta no sea estable?

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar un aplicativo móvil para guiar a una persona con discapacidad visual en ambientes exteriores usando realimentación de audio filtrando la señal GPS/GPRS en zonas donde ésta no sea estable.

1.3.2 OBJETIVOS ESPECIFICOS

1. Sintetizar un conjunto de referencias bibliográficas sobre los métodos y sistemas de filtrado de señales GPS/GPRS y de creación de aplicaciones móviles para asistencia con personas con discapacidad visual con el fin de definir los requerimientos funcionales y no funcionales del aplicativo a desarrollar.
2. Diseñar el aplicativo móvil para guiar a una persona con discapacidad visual en ambientes exteriores usando realimentación de audio y filtrando la señal GPS/GPRS en zonas donde ésta no sea estable.
3. Implementar el aplicativo móvil para guiar a una persona con discapacidad visual en ambientes exteriores usando realimentación de audio y filtrando la señal GPS/GPRS en zonas donde ésta no sea estable.
4. Elaborar un protocolo de pruebas de integración, cuantitativas y de campo para la validación del aplicativo móvil para guiar a una persona con discapacidad visual en ambientes exteriores usando realimentación de audio.

1.4 SOLUCIÓN PROPUESTA

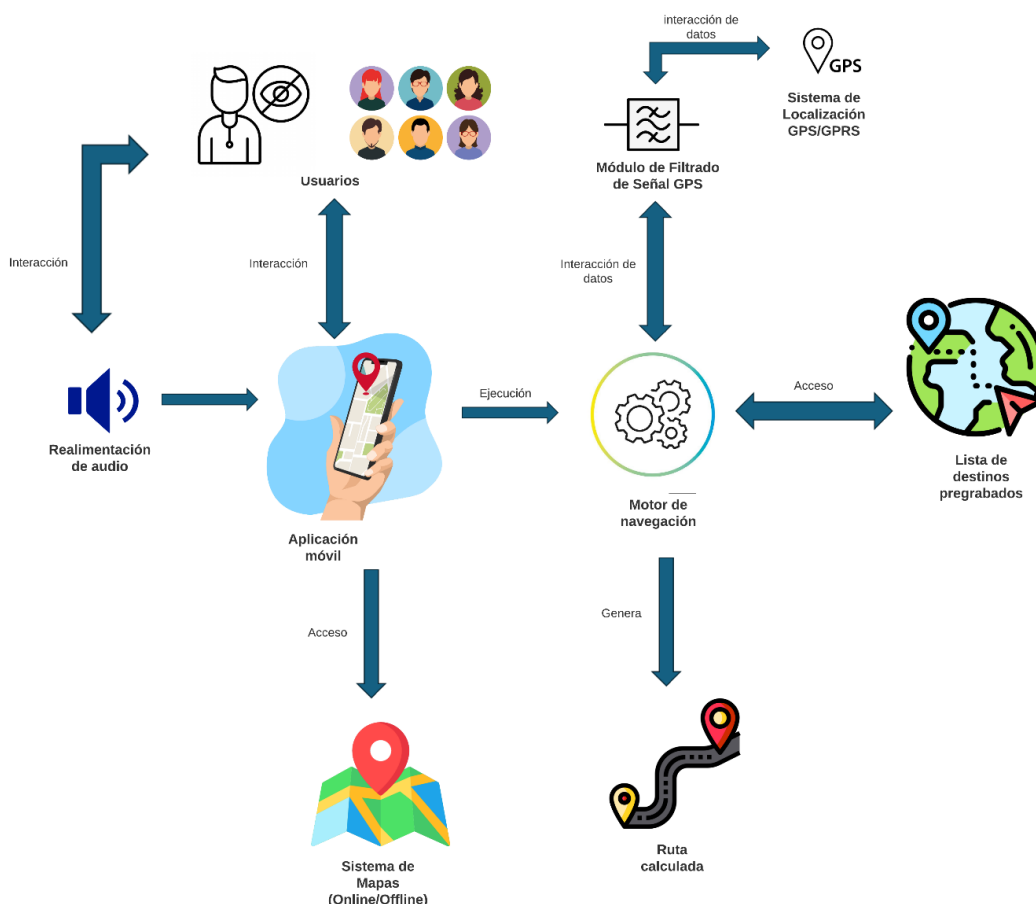


Figura 1.2 Ilustración de la solución propuesta

El diagrama presentado en la Figura 1.2 ilustra el funcionamiento de la solución propuesta. Consiste en el desarrollo de una aplicación móvil para dispositivos con sistema operativo Android, orientada a brindar asistencia en la movilidad de personas con discapacidad visual. En una primera etapa, la configuración de la aplicación será realizada por un acompañante o persona de apoyo, quien podrá ajustar parámetros como el idioma, la estatura del usuario, el volumen de la retroalimentación, entre otros aspectos necesarios para personalizar la experiencia de uso. Posteriormente, este acompañante podrá añadir destinos a una lista que quedarán almacenados en el sistema.

Una vez definida esta configuración inicial, la persona con discapacidad visual podrá seleccionar un destino mediante distintos mecanismos de accesibilidad, tales como comandos de voz, la función TalkBack o gestos de interacción propios de la aplicación. La guía hacia el destino se realizará mediante un sistema de

retroalimentación configurable, que podrá ser de tipo auditivo (voz), sonoro (pitidos) o háptico (vibración), de acuerdo con la preferencia establecida previamente. Así, la aplicación ofrecerá una experiencia adaptable y segura, favoreciendo la autonomía y confianza del usuario en entornos exteriores.

Para garantizar la precisión de la navegación, la aplicación procesará la señal de posicionamiento empleando algoritmos de filtrado, específicamente un Filtro de Partículas, el cual permitirá corregir errores y sobresaltos ocasionados por baja cobertura o interferencias en la señal GPS/GPRS, contribuyendo a mantener una localización estable y confiable en todo momento.

1.5 ESTRUCTURA DEL LIBRO

En el transcurso de este documento se detallan los capítulos que abordan los objetivos específicos planteados, describiendo el proceso mediante el cual fueron alcanzados y se presenta una explicación general que permite al lector obtener una visión clara del enfoque adoptado para el desarrollo de la solución propuesta.

- En el **primer capítulo** se presenta la introducción, cuyo propósito es contextualizar al lector mediante la exposición del problema a abordar, las motivaciones que dieron origen al proyecto, los objetivos planteados, la solución propuesta y una breve descripción de la estructura del documento.
- En el **capítulo 2**, se presenta la revisión bibliográfica que reúne las definiciones y fundamentos teóricos que sustentan este trabajo. Se abordan de manera detallada los conceptos clave que integran el proyecto, así como el análisis y la discusión de trabajos previos y aplicaciones similares, los cuales sirven de base para la elaboración de la propuesta actual.
- En el **capítulo 3**, se detalla el proceso de construcción de la solución, iniciando con la definición de los requisitos funcionales y no funcionales, junto con sus respectivos diagramas de conceptos. Asimismo, se describe el diseño de la interfaz de usuario bajo la metodología Scrum y, finalmente, la implementación de la aplicación en sus diferentes módulos.
- En el **capítulo 4**, se describen los escenarios de prueba aplicados al aplicativo, incluyendo las pruebas de integración y de campo. Asimismo, se presentan las métricas de evaluación utilizadas, los resultados obtenidos y el análisis de estos, junto con sus respectivas conclusiones.
- En el **capítulo 5**, se presentan las reflexiones finales sobre los aprendizajes y logros alcanzados durante el proceso de desarrollo del proyecto. Asimismo, se plantean observaciones generales y se proponen posibles mejoras o líneas de investigación que permitan ampliar el alcance de la solución en el futuro.

2 CAPÍTULO 2: ANTECEDENTES Y MARCO TEÓRICO

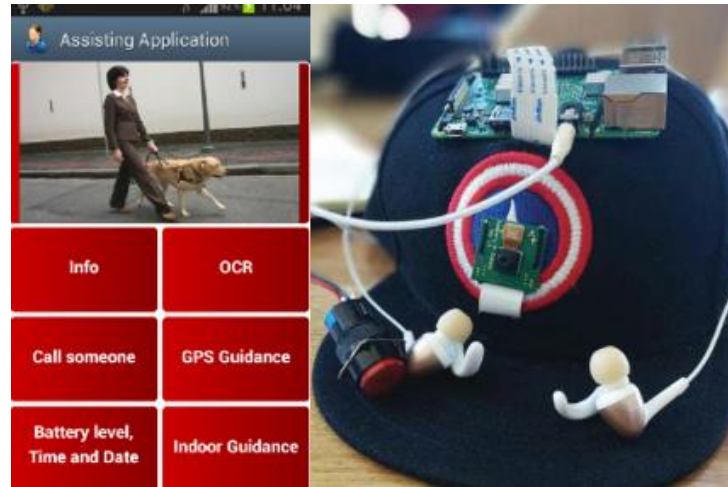


Figura 2.1 Herramientas desarrolladas para la guía y asistencia de personas con discapacidad visual. A la izquierda una interfaz de una aplicación móvil para brindar ayuda general [10] y a la derecha un prototipo para la descripción auditiva del escenario [11].

2.1 INTRODUCCIÓN

En este capítulo se abordarán los distintos trabajos relacionados con el área de la asistencia para la navegación de personas con discapacidad visual. Estos se organizaron en una tabla en la que se comparan las tecnologías usadas, los métodos de detección y el tipo de entorno al que se orienta su uso, entre otros aspectos. Asimismo, se discuten con mayor profundidad cada uno de los trabajos. Además, en este capítulo se introducen los conceptos necesarios para entender la temática abordada en los capítulos posteriores, con el fin de que el lector pueda seguir correctamente el desarrollo del trabajo. El marco teórico trata aspectos teóricos fundamentales involucrados en el proyecto, como lo son las tecnologías de asistencia para personas con discapacidad visual, las tecnologías de geolocalización, los filtros Kalman, Kalman extendido, de partículas, la accesibilidad en Android, entre otras.

2.2 ANTECEDENTES

En este apartado, se presentan trabajos que proponen soluciones a la navegación de personas con discapacidad visual con el objetivo de asistir y mejorar la eficacia e independencia de su movilidad en espacios interiores y exteriores mediante el uso de dispositivos electrónicos y la aplicación de técnicas de adquisición de imágenes, sensores, filtros entre otros. En la Tabla 2.1 se presenta un compendio de 16 referencias las cuales constituyen los antecedentes de este proyecto y en esta se resumen las principales características de cada documento relacionadas a la identificación de diferencias y similitudes entre estos.

Tabla 2-1 Antecedentes.

Ref.	Tipo de tecnología	Plataforma	Tipo de Asistencia	Entorno	Sensores/Dispositivos	Dependencia de internet	Nivel de autonomía
[10]	Smartphone + (Arduino Nano, Raspberry Pi)	Android	Navegación lectura OCR, llamadas, estado del sistema con TTS	Interior y exterior	GPS, cámara, orientación, acelerómetro, luz, ultrasonidos, Raspberry Pi con OpenCV; comunicación BT/Wi-Fi	Media–Alta (Mapas de Google y OCR en servidor externo)	Media
[11]	Dispositivo embebido con cámara +	Raspberry Pi (Raspbian)	Descripción de escenas/objetos con salida por voz	Interior y exterior	Raspberry Pi 3, Pi Camera, botón físico, auriculares;	Alta (la conectividad es la restricción principal)	Media
[12]	Smartphone + Sensores IMU	Android	Asistencia de navegación	Interior	IMU	Baja (funciona offline)	Alta
[13]	Filtro de partículas	Android	Asistencia de navegación	Exterior	Sensor GPS	Media (usa GPS, optimiza datos locales)	Alta
[14]	Integración Google Maps	Android	Lectura y navegación	Mixto	Micrófono, GPS	Alta (requiere acceso a internet)	Media
[15]	Visión por computadora	Android	Detección de obstáculos	Exterior	Cámara	Media (depende de procesamiento local)	Alta
[16]	Tensor Flow + Visión Artificial	Raspberry Pi + App	Detección de objetos	Exterior	Cámara, Acelerador Coral	Baja (procesamiento local)	Alta
[17]	GPS + Auto posicionamiento	Android	Guía por voz	Exterior	GPS	Media (dependencia moderada del	Alta

						GPS	
[18]	Realidad Aumentada Auditiva	Android	Navegación asistida por voz	Exterior	GPS, Wi-Fi	Alta (requiere internet para mapas)	Media
[19]	GPS + Mapas Digitales	Android	Guía por voz	Exterior	GPS, Wi-Fi	Alta (conexión requerida)	Alta
[20]	Sensores ultrasónicos + GPS	IoT + App móvil	Detección de obstáculos + GPS	Mixto	Sensores ultrasónicos, GPS	Alta (requiere red IoT)	Alta
[21]	Modelo de Evaluación Funcional	General	Evaluación de apps	Mixto	N/A	No aplica	No aplica
[22]	Beacons Bluetooth + GPS	iOS	Navegación contextual	Interior	Beacons Bluetooth	Media (uso offline limitado)	Alta
[23]	IoT + Sensores inteligentes	Android/iOS	Navegación mixta	Mixto	Beacons, RFID	Alta (infraestructura IoT)	Alta
[24]	Visión por Computadora	Android	Detección de objetos	Mixto	Cámara	Baja (procesamiento local)	Alta
[25]	Asistencia de voluntarios	Android/iOS	Asistencia comunitaria	Mixto	Cámara, micrófono	Alta (transmisión de video)	Media

Los trabajos revisados fueron extraídos de las bases de datos IEEE Xplorer y Scopus, reconocidas por su amplio contenido en ingeniería. Estos reúnen trabajos recientes sobre asistencia a la movilidad para personas con discapacidad visual que van desde apps Android que operan solo con el teléfono hasta soluciones con hardware complementario (Raspberry Pi, ultrasonidos, beacons). Los enfoques cubren guiado interior y exterior, descripción del entorno por visión artificial y fusión sensorial (GPS-IMU, PF/KF/EKF), con distintos grados de dependencia de servicios en la nube.

En [10] se presenta una aplicación móvil en Android con interfaz accesible y TTS que integra módulos de asistencia (llamadas por voz, OCR, guiado GPS, guiado interior). Para navegación interior usa ultrasonidos en cabeza/torso/piernas vía Arduino, y para refinar detección de objetos emplea Raspberry Pi con OpenCV. La comunicación con el teléfono es por Bluetooth/Wi-Fi. El guiado exterior se apoya en Google Maps; el OCR se resuelve en un servidor externo por sus mejores tasas de acierto. Ensayos preliminares reportan usabilidad adecuada, con necesidad de seguir ajustando distancias óptimas de detección.

En [11] se desarrolla un prototipo “wearable” con Raspberry Pi y cámara que captura imágenes al presionar un botón, las envía a Microsoft Cognitive Services para generar una descripción textual y la narra al usuario mediante gTTS. Se valida la tubería completa (captura-API-narración) y se destaca que la principal limitación es la dependencia de Internet; el trabajo plantea como futuro integrar sensores ultrasónicos y capacidades de navegación, así como multilenguaje.

En [12] se aborda la movilidad de personas con discapacidad visual mediante el uso exclusivo de teléfonos inteligentes, sin depender de sensores adicionales. Esto se justifica por la creciente capacidad computacional de los smartphones y su accesibilidad económica en comparación con dispositivos especializados. Se propone una arquitectura de navegación en interiores, detallando sus componentes, interfaces y algoritmos clave. El sistema se basa en módulos de software, siendo la aplicación móvil el núcleo del procesamiento de localización y navegación. Para garantizar la continuidad del servicio, la aplicación almacena en caché información relevante del entorno, permitiendo la navegación incluso sin conexión a datos. Además, aprovecha los sensores IMU del dispositivo para monitorear el movimiento del usuario, promoviendo un nivel de autonomía alto. Se analizan los requisitos específicos de un sistema de navegación en interiores para personas con discapacidad visual, presentando soluciones de diseño que

abordan estos desafíos y la funcionalidad de los módulos que componen el sistema.

En [13] se analiza un enfoque basado en filtros de partículas para mejorar la precisión de localización de usuarios utilizando únicamente los datos proporcionados por teléfonos inteligentes, sin necesidad de hardware adicional. Los resultados indican que esta técnica puede ofrecer una solución eficiente y accesible para personas con discapacidad visual, permitiéndoles desplazarse con mayor autonomía. Se describe la implementación del algoritmo dentro de una aplicación de navegación personalizable y se presentan pruebas preliminares que evalúan su desempeño. Este sistema no depende de conexión constante a Internet, ya que opera principalmente con datos locales. También se examina la posibilidad de integrar otros métodos de búsqueda para optimizar la precisión de la localización. Finalmente, el estudio aborda las limitaciones y mejoras del enfoque, destacando su impacto potencial en la movilidad asistida.

En [14] se centró en el desarrollo e implementación de Voice Helper (VH), un sistema de asistencia para usuarios con discapacidad visual en dispositivos Android. VH integra diversas funciones de accesibilidad, incluyendo lectura de mensajes, archivos de texto y documentos OCR, así como un marcador de voz diseñado específicamente para mejorar la interacción del usuario. Además, ofrece un sistema de navegación asistida para peatones y ciclistas basado en Google Maps, proporcionando indicaciones de voz detalladas sobre distancias y direcciones. El sistema ha sido probado por personas con discapacidad visual para evaluar su usabilidad y efectividad en la vida cotidiana.

En [15] se propone un sistema de navegación que combina tecnologías de detección de imágenes y traducción de voz para asistir a personas con discapacidad visual en entornos interiores y exteriores. Su objetivo principal es proporcionar información en tiempo real para mejorar la autonomía del usuario y su confianza al desplazarse. La aplicación, desarrollada en Android, emplea algoritmos de aprendizaje automático para identificar obstáculos y puntos de referencia como puertas y escaleras. También incorpora una función de traducción de voz para hacer las instrucciones de navegación más accesibles a usuarios con dificultades de lectura. Gracias a esto, funciona en gran medida de forma local con una dependencia de Internet baja y otorga a los usuarios un nivel de autonomía alto.

En [16] se presenta un sistema basado en Visión Artificial, Aprendizaje Automático y TensorFlow para mejorar la movilidad de personas con

discapacidad visual. Mediante el reconocimiento e identificación de objetos en tiempo real, el sistema permite una navegación independiente. Se implementó el hardware en una Raspberry Pi 4 y un servidor con una GPU NVIDIA Titan X, complementado con el acelerador Google Coral para mejorar el procesamiento. Además, se monitorea el rendimiento a través de un servidor dedicado, asegurando la eficiencia del sistema en entornos reales.

En [17] se describe un sistema de navegación para personas con discapacidad visual que calcula la mejor ruta basándose en la posición del usuario y bases de datos cartográficas digitales. Se emplea GPS y un método de auto posicionamiento para garantizar una localización precisa, especialmente en entornos donde la señal GPS es deficiente. El sistema proporciona información del área mediante una voz artificial en puntos de referencia clave. Además, mediante técnicas de coincidencia de mapas, se minimizan errores en la estimación de la posición, mejorando la precisión en la navegación.

En [18] se presenta una aplicación móvil basada en realidad aumentada auditiva, diseñada para mejorar la movilidad de personas con discapacidad visual. Esta solución integra tecnologías como GPS, Wi-Fi y redes móviles para determinar la ubicación del usuario (alta dependencia de Internet). A través de su conexión con Google Maps, la aplicación móvil proporciona instrucciones de navegación en tiempo real mediante comandos de voz y señales sonoras. Informa al usuario sobre calles, intersecciones y puntos de interés cercanos, facilitando así un desplazamiento más seguro y autónomo. La interfaz está optimizada para reducir al mínimo la interacción táctil, permitiendo el control por voz o mediante gestos. Sin embargo, la aplicación enfrenta limitaciones como la baja precisión del GPS en zonas urbanas densas y la ausencia de sensores que detecten obstáculos físicos en tiempo real.

En [19] se expone el desarrollo de una aplicación Android destinada a asistir a personas con discapacidad visual en su movilidad cotidiana. Al igual que otras soluciones, emplea GPS, redes móviles y servicios de mapas digitales para determinar la ubicación y ofrecer instrucciones auditivas en tiempo real, incluyendo alertas sobre calles, cruces e información contextual del entorno. La interfaz está pensada para minimizar la dependencia de la interacción táctil y puede ser controlada por voz o gestos. No obstante, los autores identifican desafíos similares, como la inexactitud del GPS en entornos urbanos complejos y la falta de sensores que detecten obstáculos inmediatos.

En [20] se propone un sistema de navegación inteligente basado en Internet de las Cosas (IoT), específicamente diseñado para mejorar la seguridad y movilidad de personas con discapacidad visual. El sistema combina sensores ultrasónicos, módulos GPS, microcontroladores como Arduino o Raspberry Pi, y una aplicación móvil que interpreta los datos del entorno (baja dependencia de Internet). Los sensores permiten la detección de obstáculos físicos, mientras que el GPS aporta información sobre ubicación. Estos datos se procesan y se transforman en comandos de voz comprensibles para el usuario, brindándole orientación en tiempo real. Además, el sistema puede enviar alertas o ubicaciones a familiares o cuidadores, siendo una solución portátil, de bajo costo y accesible para usuarios en distintos contextos.

En [21] se presenta un enfoque metodológico centrado en la evaluación cualitativa de aplicaciones de navegación para personas con discapacidad visual. La propuesta se basa en un modelo funcional que considera no solo la precisión técnica, sino también la experiencia del usuario. Se valoran aspectos como la claridad de las instrucciones, el tiempo de respuesta del sistema, la toma de decisiones del usuario ante situaciones diversas y el grado de confianza generado por la aplicación. El estudio subraya la importancia de comprender cómo interactúan realmente las personas ciegas con estas tecnologías para desarrollar soluciones más accesibles, efectivas y seguras.

En [22] se habla sobre BlindNavi, una aplicación móvil diseñada para facilitar la navegación independiente de personas con discapacidad visual, combinando principios de orientación y movilidad (O&M) con tecnología accesible. A diferencia de otras aplicaciones móviles de mapas, BlindNavi utiliza una interfaz simplificada compatible con lectores de pantalla como VoiceOver y permite realizar búsquedas de direcciones y números telefónicos en solo tres pasos. Ofrece retroalimentación auditiva basada en puntos de referencia fácilmente reconocibles, y emplea tecnología de micro-localización, como beacons Bluetooth, para brindar precisión en interiores donde el GPS no funciona bien. Al funcionar principalmente offline su dependencia de Internet es baja-media y facilita la navegación independiente en interiores, promoviendo un nivel de autonomía alto. Evaluada con usuarios reales, la aplicación demostró mejoras significativas en confianza, autonomía y exploración de espacios nuevos.

En [23] se explora cómo las aplicaciones móviles pueden integrarse con infraestructura IoT para mejorar la movilidad y autonomía de personas con discapacidad visual. La propuesta combina sensores ultrasónicos, cámaras, beacons Bluetooth, sistemas RFID y módulos GPS conectados a una red

inteligente que recopila datos del entorno. Estos datos se procesan en tiempo real para ofrecer retroalimentación auditiva o háptica sobre obstáculos, rutas seguras y ubicación actual. El artículo detalla cómo esta arquitectura supera las limitaciones del GPS tradicional, permitiendo una cobertura más precisa tanto en interiores como en exteriores. También se abordan desafíos técnicos como la sincronización de datos, duración de batería y diseño de interfaces accesibles, destacando el potencial de estas soluciones para construir entornos urbanos más inclusivos.

En [24] se propone una solución de navegación móvil que combina visión por computadora con traducción de voz para asistir a personas con discapacidad visual. Utilizando la cámara del teléfono, la aplicación móvil captura imágenes del entorno en tiempo real y las analiza para identificar objetos, obstáculos y puntos relevantes, como escaleras o puertas. La información visual se convierte en instrucciones auditivas mediante un sistema de voz, enriquecido con procesamiento de lenguaje natural (NLP) para que las indicaciones sean comprensibles y contextualizadas. Como la mayoría de procesamiento se realiza en el dispositivo móvil su dependencia de Internet es baja. Este sistema permite una navegación segura sin depender exclusivamente del GPS, y demostró en pruebas una mejora significativa en la autonomía y confianza de los usuarios en entornos complejos.

En [25] se presenta una aplicación que permite a personas con discapacidad visual recibir asistencia en tiempo real mediante la ayuda de una comunidad de usuarios voluntarios. Usando la cámara del smartphone, el usuario transmite su entorno a otros usuarios que proporcionan instrucciones verbales en vivo. Este enfoque de navegación colaborativa, o "crowdsourcing", ofrece una alternativa flexible y accesible a los sistemas automáticos tradicionales. La aplicación móvil incorpora funciones de privacidad, selección de asistentes y almacenamiento de rutas frecuentes, permitiendo mejorar progresivamente la experiencia del usuario. Su enfoque comunitario destaca por su capacidad de adaptación, su accesibilidad y su potencial para complementar otras tecnologías de asistencia, sin embargo, implica una fuerte dependencia de Internet y ofrece un nivel de autonomía bajo debido a la necesidad de asistencia humana constante.

El análisis de los antecedentes demuestra la importancia de desarrollar tecnologías de asistencia para la navegación de personas con discapacidad visual, destacando la necesidad de mejorar la precisión en la estimación de la posición del usuario. Diferentes estudios han propuesto enfoques basados en algoritmos de filtrado, fusión de sensores y visión artificial para optimizar la

navegación en entornos tanto interiores como exteriores. Estos métodos han demostrado ser eficaces en la reducción de errores en la señal GPS y en la integración con sensores inerciales (IMU) para compensar la pérdida de señal en áreas de baja cobertura.

Los estudios revisados también evidencian que la efectividad de los sistemas de navegación asistida depende no solo de la precisión de los algoritmos de localización, sino también de factores como la calidad de los sensores, la iluminación del entorno y la capacidad de procesamiento del dispositivo. En este sentido, la combinación de GPS, IMU y visión artificial se presenta como una solución integral que podría superar las limitaciones de los métodos tradicionales.

2.3 MARCO TEÓRICO

En el siguiente apartado se definen conceptos fundamentales presentes en este trabajo de grado, que permiten abordar el contexto de la investigación de una mejor manera. Se presentan algunos conceptos necesarios para comprender de manera correcta la discapacidad visual, los tipos y las características de tecnologías para brindar ayudas. Además, se presentan las técnicas más utilizadas para el procesamiento de las señales GPS/GPRS, junto los métodos más usados para el desarrollo de aplicaciones accesibles.

2.3.1. Discapacidad Visual

La discapacidad visual se refiere a cualquier tipo de pérdida de visión, total o parcial [8]. Se refiere a personas que han perdido la visión lo suficiente como para tener que acercarse a los objetos para verlos con claridad, mientras que otras son completamente ciegas [26]. Según las cifras de la Organización Mundial de la Salud (OMS), hay 2200 millones de personas con discapacidad visual en todo el mundo [27]. En Colombia estudios más recientes, como el publicado por el Fondo de Población de las Naciones Unidas (UNFPA) en 2023, estiman que la población con discapacidad en Colombia es de aproximadamente 2,9 millones de personas, de las cuales el 54,9% son mujeres y el 45,1% son hombres [28].

La severidad de la discapacidad se cuantifica principalmente con: agudeza visual para lejos [29] (p. ej., escala Snellen o logMAR) en el mejor ojo con corrección, y campo visual [30] (diámetro del campo central en grados). Estas métricas permiten clasificar a la persona y comparar resultados entre estudios y contextos

clínicos. La Figura 2.2 resume los grados de discapacidad visual establecidos por la Organización Mundial de la Salud, previamente descritos.

Grado de discapacidad visual	Agudeza visual Snell	Campo visual (grados °)
0 Leve o nula	Mejor a 20/70	120
1 Moderada	20/70 o mejor a 20/200	50-60
2 Severa (dificultad para leer aún con ayudas)	20/200 o mejor a 20/400	20
3 Ceguera	20/400 o mejor a 20/1200	10
4 Ceguera	20/1200 a percepción de luz	2
5 Ceguera total	No percepción de luz	

Figura 2.2 Grados de discapacidad visual según la organización mundial de la salud, resumidos en [31].

La severidad y el tipo de pérdida (reducción de agudeza, túnel visual, deslumbramiento, escotomas) [26] condicionan estrategias de orientación y movilidad y la selección de tecnologías de asistencia: quienes mantienen parte del campo periférico suelen beneficiarse de guiado auditivo y map-matching [32], mientras que constricciones marcadas del campo o multitrayectoria urbana exigen mayor dependencia de ayudas hápticas/voz y fusión sensorial (IMU+GNSS+visión) [33]. Estas decisiones deben alinearse con la clasificación clínica y las metas de rehabilitación.

2.3.2. Tecnologías de asistencia para la navegación de personas con discapacidad visual

Las personas con discapacidad visual o ceguera enfrentan diversos desafíos en su movilidad diaria lo que limita su autonomía y acceso a espacios urbanos e interiores. Para esto, los dispositivos de asistencia desempeñan un papel vital en áreas desconocidas, ya que permiten a estas personas navegar y moverse de un lugar a otro. Tradicionalmente, han utilizado herramientas como bastones y perros guía; sin embargo, los avances tecnológicos han permitido el desarrollo

de soluciones más sofisticadas basadas en sensores y dispositivos inteligentes [34].

En términos funcionales, las herramientas y soluciones para la navegación de personas con discapacidad visual, se agrupan en tres categorías [33] según el tipo de información que entregan al usuario:

A) Electronic Travel Aids (ETAs).

Las ayudas electrónicas de viaje (ETA) entregan principalmente información de detección y alerta de obstáculos —bordes, huecos, escalones y mobiliario urbano— usando sensores como ultrasonido, LiDAR y cámaras (visión por computador), y en algunos casos radar; en configuraciones portátiles suelen integrarse en el bastón, en un wearable o aprovechar la cámara del smartphone.

La retroalimentación al usuario se da por audio (TTS o señales sonoras) y/o háptico (vibración). Su ventaja es reducir colisiones y “sorpresas” locales, especialmente en interiores o zonas donde el GNSS es inviable; sus limitaciones incluyen rango corto y oclusiones, sensibilidad a iluminación (visión), ruido (ultrasonido) o textura de superficies, y la necesidad de calibrar la latencia para no sobrecargar al usuario. En la Figura 2.3 se sintetizan las tecnologías típicas empleadas en las ETAs, lo que permite visualizar de forma esquemática los componentes y flujos de información más comunes.

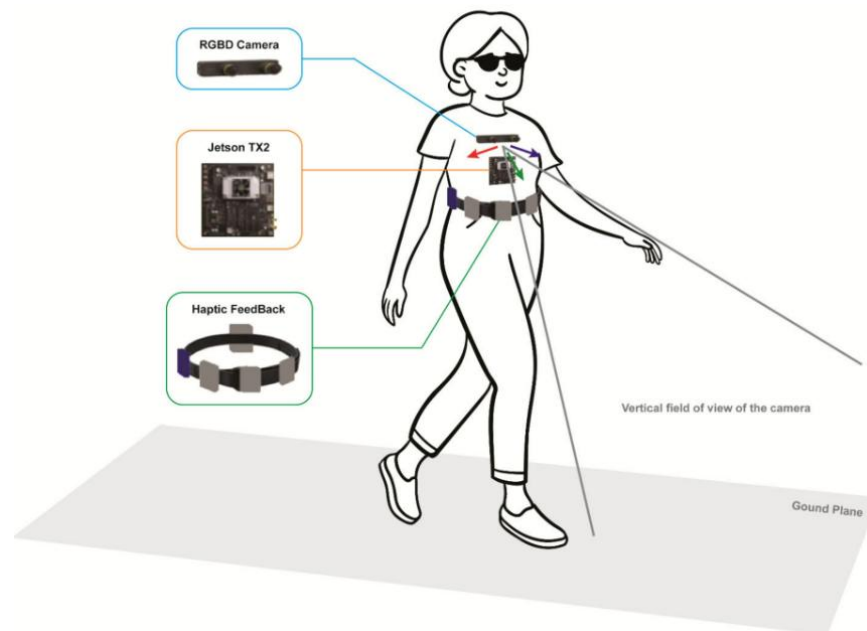


Figura 2.3 Esquema de tecnologías típicas ETAs resumidas en [35].

B) Ayudas electrónicas de orientación (EOAs).

Las ayudas electrónicas de orientación (EOA) proveen posicionamiento y navegación con indicaciones paso a paso, uso de puntos de referencia y apoyo en decisiones en intersecciones o puertas, combinando fuentes como GNSS (GPS/Galileo), IMU del teléfono para PDR/odometría, Wi-Fi/BLE en interiores y map-matching sobre redes peatonales, con apoyo ocasional de visión para landmarks.

La salida es principalmente por voz (TTS) y, como complemento, háptico; algunos sistemas incorporan sonido espacial. Entre sus ventajas están cubrir trayectos largos y la toma de decisiones, además de poder operar offline con mapas en caché y fusión GNSS+IMU; sus limitaciones incluyen la degradación del GNSS en cañón urbano, las transiciones interior-exterior que afectan continuidad/integridad del posicionamiento y, para mayor precisión en interiores, la dependencia de infraestructura. En la Figura 2.4 se muestra la interfaz de la aplicación Lazarillo como ejemplo de ayuda electrónica de orientación orientada a la navegación accesible.



Figura 2.4 Interfaz correspondiente a la aplicación Lazarillo (EOA) [3].

C) Binary Electronic Mobility Systems (BEMS).

En la práctica, ETAs y EOAs se combinan para obtener las ventajas de ambos: las ETAs resuelven la seguridad inmediata (evitar choques) mientras las EOAs sostienen la orientación global (saber por dónde ir). que integran detección de obstáculos y guiado en un mismo flujo interactivo, junto con fusión multisensor (GNSS+IMU+visión) para robustecer precisión, continuidad y disponibilidad en entornos complejos. Un ejemplo de este tipo de solución integrada se presenta en la Figura 2.5, donde se observa una aplicación móvil que incorpora módulos externos de detección conectados al teléfono inteligente.



Figura 2.5 Interfaz de una aplicación móvil para brindar ayuda general incluyendo módulos externos de detección, construida en [10].

2.3.3. Tecnologías de geolocalización

GNSS son las siglas en inglés de Sistema Global de Navegación por Satélite (Global Navigation Satellite System). Es un término genérico que agrupa distintos sistemas de navegación por satélite que entregan posicionamiento, navegación y temporización a escala mundial, ya sea de forma autónoma o con sistemas de aumentación [36]. Consiste en una red de satélites que transmiten información temporal y orbital utilizada para mediciones de navegación y posicionamiento. A través de una compleja serie de cálculos de trilateración, la tecnología calcula la ubicación basándose en la posición del receptor en relación con al menos cuatro satélites [37].

El GNSS es más que los satélites que orbitan la Tierra. Los múltiples grupos de satélites, conocidos como constelaciones, transmiten señales a estaciones maestras de control y a los usuarios de GNSS en todo el planeta. Estos tres segmentos —espacio, control y usuario— se consideran todos parte del GNSS, aunque con mayor frecuencia el término se usa para describir a los satélites en el espacio [37]. En la Figura 2.6 se ilustran los 3 segmentos.

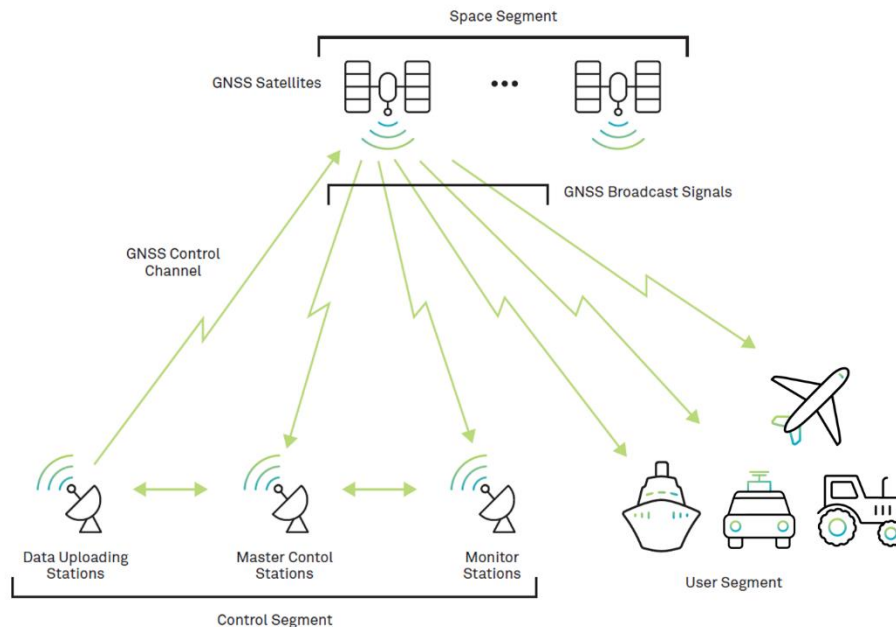


Figura 2.6 Segmentos espacio, control y usuario. Partes del GNSS elaborada en [37].

El segmento espacial describe las constelaciones GNSS que orbitan entre 20.000 y 37.000 kilómetros sobre la Tierra. Estos satélites transmiten señales que identifican qué satélite está transmitiendo y su hora, órbita y estado o salud. El segmento de control es una red de estaciones maestras de control, de carga de datos y de monitoreo ubicadas alrededor del mundo. Estas estaciones reciben la señal de un satélite y comparan dónde dice el satélite que está con modelos orbitales que muestran dónde debería estar. El segmento de usuario incluye el equipo que recibe las señales de los satélites y produce una posición basada en la hora y la ubicación orbital de al menos cuatro satélites. Este segmento incluye las antenas del usuario para identificar y recibir señales de buena calidad, así como receptores de alta precisión y motores de posicionamiento que procesan las señales y resuelven posibles errores de temporización [37].

Hay cuatro constelaciones principales en órbita así como dos sistemas regionales [36] y [37]. Cada una está gestionada por un país diferente:

- El sistema global GPS (Global Position System): EEUU, altura de 20.200 Km. Operado por la U.S. Space Force; primer satélite lanzado en 1978 y primera serie plenamente operativa en 1993. Actualmente cuenta con 31 satélites en órbita.
- El sistema global Glonass (Global Orbiting Navigation Satellite System): Rusia, 24 satélites, altura de 25.500 Km. Desarrollado en la URSS en los años 70 y operado hoy por Roscosmos; primer lanzamiento en 1982, constelación completada en 1995 y cobertura plena en Rusia desde 2010.
- El sistema global Galileo (ESA): Europa, altura de 23.200 Km. Primera misión lanzada en 2011; operado por la European GNSS Agency. Actualmente 26 satélites en órbita con plan de alcanzar 30.
- El sistema global BeiDou (BDS): China, altura entre 21.600 Km. y 35.800 km. Primer lanzamiento en 2000; operado por la CNSA. En las últimas dos décadas ha desplegado 48 satélites en órbita.
- El sistema regional QZSS (Japan's Quasi-Zenith Satellite System): Mantiene la cobertura regional de Asia–Oceanía entre Japón y Australia; 4 satélites en órbita. Operado por JAXA, con primer lanzamiento en 2010 y con la intención de agregar +3 satélites.
- El sistema regional IRNSS (Indian Regional Navigation Satellite System): La cobertura se centra en India, extendiéndose hacia el oeste (Arabia Saudita), norte y este (toda China) y sur (Mozambique y Australia Occidental); 8 satélites. Operado por ISRO.

2.3.3.1. Problemas de precisión en entornos con baja cobertura

En condiciones de baja cobertura (cañón urbano, interiores, túneles o vegetación densa), la visibilidad de satélites disminuye y la geometría se degrada, lo que eleva el error de posicionamiento, aunque el “señal-en-espacio” sea nominalmente preciso. A ello se suman multi-trayectoria y NLOS por bloqueo/reflexiones en fachadas, que introducen sesgos en los pseudorangos y generan colas pesadas en el error, especialmente en “cañones profundos” y de “dos lados” (fachadas a ambos lados). Estas condiciones producen estimaciones inestables, saltos de posición y pérdidas de continuidad [38].

2.3.4. Métodos de filtrado de señales GPS/GPRS

2.3.4.1. Filtro de Kalman (KF)

El filtro de Kalman es un algoritmo que estima el estado de un sistema a partir de datos medidos. Fue desarrollado originalmente por el ingeniero húngaro Rudolf Kalman, de quien toma su nombre. El algoritmo del filtro es un proceso de dos pasos: el primer paso predice el estado del sistema, mientras que el segundo utiliza las mediciones de ruido para ajustar la estimación del estado del sistema [39].

En la actualidad existen diversas variantes del filtro de Kalman original. Estos filtros son muy utilizados para aplicaciones que dependen de la estimación, como la visión artificial, los sistemas de guía y navegación, la econometría y el procesamiento de señales. En la siguiente figura 2.7 se puede ver el algoritmo del filtro Kalman descrito en [40]:

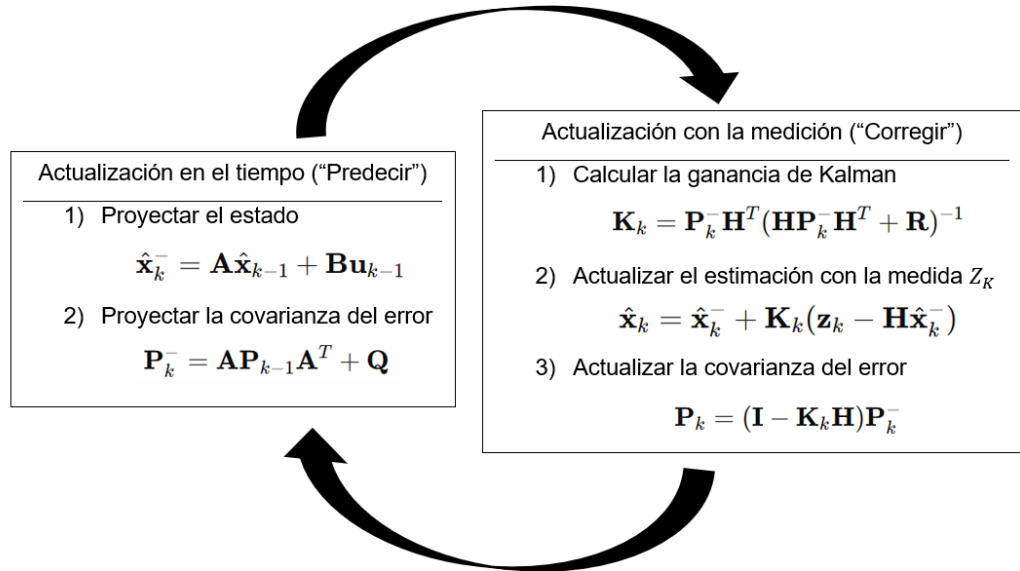


Figura 2.7 Algoritmo del filtro de Kalman [40].

A continuación, se definen las matrices y demás variables que aparecen en la Figura 2.7:

Sea n la dimensión del estado, m la dimensión de la medición y l la dimensión de la entrada de control.

- $\hat{\mathbf{x}}_{k|k-1}, \hat{\mathbf{x}}_{k|k}$ son estimaciones a priori y a posteriori del estado en el instante k (vectores $n \times 1$).
- $\mathbf{u}_{k-1}$: entrada/acción de control aplicada entre $k - 1$ y k (vector $l \times 1$).

- $\mathbf{z}_k$: medición provista por los sensores en el instante k (vector $m \times 1$).
- $\mathbf{A}$: matriz de transición del modelo ($n \times n$).
- $\mathbf{B}$: matriz de control que acopla la entrada al estado ($n \times l$).
- $\mathbf{H}$: matriz de observación que proyecta el estado al espacio de medición ($m \times n$).
- $\mathbf{P}_{k|k-1}$, $\mathbf{P}_{k|k}$ covarianzas del error de estimación a priori y a posteriori ($n \times n$).
- $\mathbf{Q}$: covarianza del ruido de proceso ($n \times n$), modela la incertidumbre del modelo dinámico.
- $\mathbf{R}$: covarianza del ruido de medición ($m \times m$), modela la incertidumbre del sensor.
- $\mathbf{v}_k$: innovación o residuo de medición $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}$ (vector $m \times 1$)
- $\mathbf{S}_k$: covarianza de la innovación $\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}$ ($m \times m$)
- $\mathbf{K}_k$: ganancia de Kalman ($n \times m$), pondera predicción vs. medición.
- $\mathbf{I}$: matriz identidad ($n \times n$); $(\cdot)^T$: transposición.

2.3.4.2. Filtro de Kalman extendido (EKF)

El filtro de Kalman originalmente solo puede operar con sistemas lineales. Cuando la dinámica y/o la medición son no lineales, se usan las funciones $f(\cdot)$ y $h(\cdot)$ y sus jacobianos evaluados en la estimación corriente, dando como resultado la variación del filtro de Kalman extendido [40]. En las ecuaciones 2.1, 2.2 y 2.3 se muestran de forma esquemática los jacobianos empleados en el EKF para linealizar las funciones de estado y de medición alrededor de la estimación actual.

$$(2.1) \quad \mathbf{A}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}}, \quad (2.2) \quad \mathbf{B}_k = \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}},$$

$$(2.3) \quad \mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}$$

Así, reemplazando en la figura 2.7 del algoritmo del filtro de kalman $\mathbf{A} \rightarrow \mathbf{A}_k$, $\mathbf{B} \rightarrow \mathbf{B}_k$, y $\mathbf{H} \rightarrow \mathbf{H}_k$ en la figura 2.7 se obtiene el ciclo predicción–corrección del EKF.

2.3.4.3. Filtro de partículas (PF)

El filtro de partículas (PF) es un estimador bayesiano no paramétrico que se utiliza cuando el sistema es no lineal y/o el ruido no es gaussiano, de

modo que representar el estado solo con una media y una covarianza no es suficiente. En lugar de una única estimación, el PF aproxima la distribución posterior $p(x_k|z_{1:k})$ mediante un conjunto finito de muestras (partículas) $\{x_k^{(i)}\}_{i=1}^N$ y pesos asociados $\{w_k^{(i)}\}_{i=1}^N$. Cada partícula representa una hipótesis de dónde podría estar el usuario; las mediciones van reforzando las hipótesis más plausibles y descartando las demás. Esto lo hace especialmente adecuado para navegación asistida cuando hay ambigüedad de posición, multitrayectoria, pasos por interior–exterior, o cuando se quiere forzar que la solución permanezca sobre un mapa peatonal [41].

A continuación, se presenta el algoritmo básico del filtro ejemplificado en [41]:

(1) Inicialización.

Se generan N partículas desde una distribución inicial $p(x_0)$ (posición conocida con ruido, o zona probable) y se asignan pesos iguales $w_0^{(i)} = 1/N$.

(2) Predicción (propagación dinámica).

Para cada partícula se aplica el modelo de movimiento, como se muestra en la Ecuación 2.4:

$$(2.4) \mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_{k-1})$$

Aquí se introduce el ruido de proceso: cada partícula avanza como lo haría el peatón/usuario, pero con una dispersión controlada.

(3) Ponderación con la medición.

Llegada la medición z_k , se calcula para cada partícula su verosimilitud, como se muestra en la Ecuación 2.5:

$$(2.5) w_k^{(i)} \propto p(\mathbf{z}_k | \mathbf{x}_k^{(i)})$$

Las partículas que están en una posición compatible con la medición (GNSS, UWB, landmark) reciben más peso; las demás, menos. Luego se normalizan los pesos para que $\sum_i w_k^{(i)} = 1$.

(4) Remuestreo (resampling).

Si pocos pesos concentran casi toda la probabilidad (degeneración), se vuelve a muestrear N partículas según los pesos actuales (remuestreo multinomial, sistemático o estratificado). Así se mantienen partículas “buenas” y se eliminan las de peso casi nulo.

(5) Estimación. La estimación puntual puede obtenerse como media ponderada, como se muestra en la Ecuación 2.6:

$$(2.6) \mathbf{x}_k = \sum_{i=1}^N w_k^{(i)} \mathbf{x}_k^{(i)}$$

A continuación, se definen las variables descritas se en el algoritmo:

- $x_k^{(i)}$: partícula i-ésima (hipótesis de estado) en el tiempo k.
- $w_k^{(i)}$: peso de la partícula i, proporcional a la probabilidad de que ese estado explique la medición.
- N: número de partículas (compromiso entre precisión y costo computacional).
- $p(x_k|x_{k-1}, u_{k-1})$: modelo de transición o dinámica.
- $p(z_k|x_k)$: modelo de medición (cómo debería verse el sensor si el estado fuera x_k).
- Re muestreo: paso que evita que el filtro quede con una sola partícula “viva”.

2.3.5. Aplicaciones móviles accesibles y retroalimentación de audio

Las aplicaciones móviles para personas con discapacidad visual priorizan interacción accesible y guiado auditivo consistente en distintos contextos (interior/exterior). En la práctica, las funciones incluyen: navegación paso a paso con instrucciones por voz, notificación de intersecciones/POI, y opciones de personalización (volumen, velocidad, idioma, tipo de alerta) para adecuar la carga cognitiva del usuario [32].

Ejemplos comerciales documentados [32] muestran este patrón:

Lazarillo entrega indicaciones de audio y admite personalizar volumen/velocidad/idioma para navegación exterior y, con beacons BLE y mapas internos, también en interiores, manteniendo la interfaz simple para un uso directo por personas no expertas. Ariadne GPS ofrece instrucciones y retroalimentación auditiva, con ajustes de salida y funciones específicas como “situational awareness”, favoritos y exploración táctil del mapa con anunciado por voz de los elementos tocados. Nav by ViaOpta combina voz giro a giro y retroalimentación háptica en una interfaz simplificada, también con planificación de ruta y gestión de POI, disponible en múltiples idiomas. BlindSquare, por su parte, emite la información mediante lector de pantalla (voiceover) y permite filtros de información para evitar sobrecarga, lo que mejora la experiencia de uso.

En estas soluciones, la salida auditiva mezcla; síntesis de voz (TTS) para instrucciones contextuales, señales sonoras o tonos para eventos breves (p. ej.,

confirmaciones, avisos de proximidad), y en algunos casos, sonidos espaciales/háptica para reforzar orientación o dirección de giro [32].

2.3.6. Metodología de desarrollo de software SCRUM

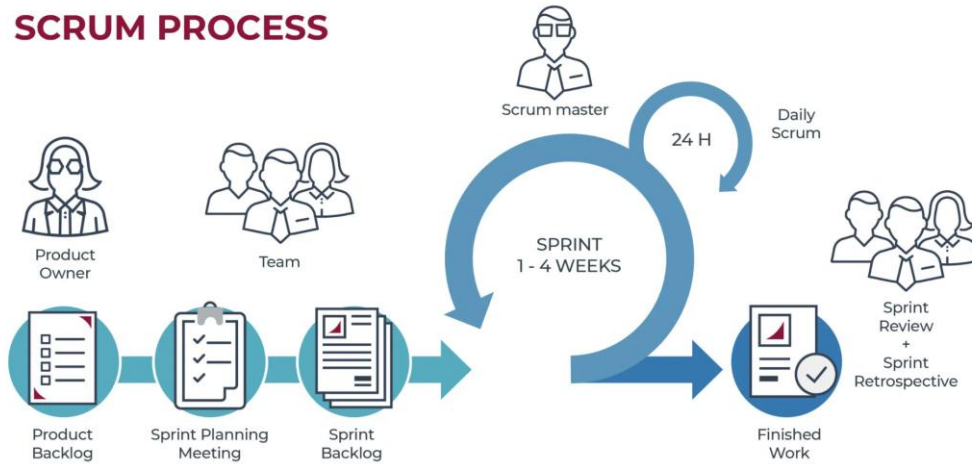


Figura 2.8 Fases de la metodología SCRUM [42]

Scrum es una metodología ágil de gestión de proyectos que organiza el trabajo en ciclos cortos llamados "sprints", generalmente de 2 a 4 semanas. Cada sprint culmina con un incremento del producto, permitiendo entregas frecuentes y adaptaciones rápidas a los cambios. Las reuniones diarias, conocidas como "Scrum Daily", de no más de 15 minutos, facilitan la comunicación y la identificación temprana de obstáculos. Al finalizar cada sprint, se realiza la "Sprint Review" para evaluar el trabajo realizado y la "Sprint Retrospective" para identificar oportunidades de mejora en el proceso. Los roles clave en Scrum incluyen el Product Owner, que representa al cliente y prioriza las tareas; el Scrum Master, que facilita el proceso y elimina impedimentos; y el equipo de desarrollo, encargado de implementar las tareas planificadas. Empresas como Spotify, Apple, Google y Amazon han adoptado Scrum para mejorar su eficiencia y capacidad de adaptación en entornos cambiantes [37]. Las fases principales del ciclo de trabajo en Scrum se resumen en la Figura 2.8, donde se ilustran los eventos y artefactos que se repiten en cada sprint.

2.4 OBSERVACIONES FINALES

Del análisis de los antecedentes revisados se concluye que existe un interés creciente por desarrollar soluciones de navegación que mejoren la autonomía de

las personas con discapacidad visual, tanto en espacios interiores como exteriores. Los trabajos estudiados exploran un amplio espectro de tecnologías: desde aplicaciones que se apoyan únicamente en el smartphone hasta sistemas que integran sensores externos. En conjunto, estos aportes muestran que la precisión en la estimación de la posición, la disponibilidad de la señal en entornos complejos y el tipo de retroalimentación entregada al usuario (voz, sonidos, vibración) son factores determinantes para lograr una navegación segura y eficaz.

Por otra parte, el marco teórico permitió organizar y profundizar los conceptos clave que sustentan este trabajo de grado. Se revisaron las principales tecnologías de asistencia para la navegación, los sistemas GNSS y sus limitaciones en cañón urbano o entornos con baja cobertura, así como los métodos de filtrado (KF, EKF y PF) empleados para mitigar errores de posicionamiento mediante fusión de sensores.

En síntesis, las observaciones de este capítulo permiten delimitar las necesidades que aborda la propuesta de este trabajo: contar con una aplicación móvil accesible que integre técnicas de filtrado y fusión sensorial para mejorar la estimación de posición en exteriores, reduzca la dependencia de conectividad permanente y proporcione retroalimentación auditiva y háptica adecuada para la navegación de personas con discapacidad visual. Estos elementos constituyen la base conceptual sobre la cual se estructura el diseño e implementación del sistema presentado en los capítulos siguientes.

3 CAPÍTULO 3: DESARROLLO DEL APLICATIVO MÓVIL

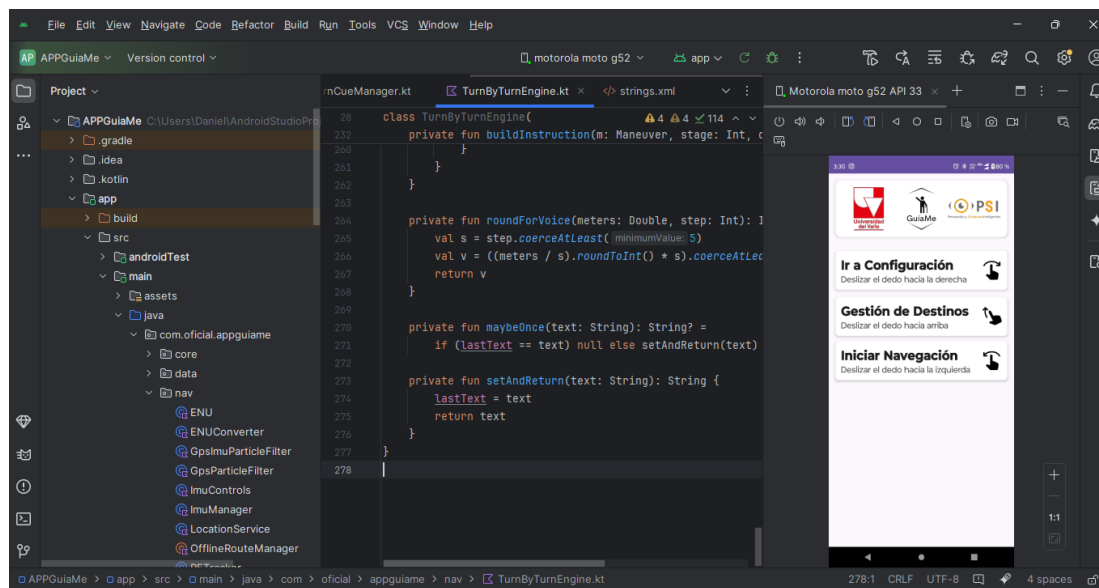


Figura 3.1 Interfaz de la aplicación Android Studio para desarrollar aplicaciones móviles.

3.1 INTRODUCCIÓN

Este capítulo presenta el desarrollo del aplicativo Guíame, diseñado para asistir a personas con discapacidad visual en la navegación en exteriores mediante realimentación auditiva basada en voz sintética y pitidos 2D lateralizados. El aplicativo integra el uso de mapas en línea y mapas offline, técnicas de filtrado sobre la señal de GPS apoyadas en datos de la IMU y diferentes modos de retroalimentación sonora, con el fin de mejorar la precisión en la localización y la comprensión de las indicaciones durante el desplazamiento. A partir de los requerimientos funcionales y no funcionales definidos previamente, se describe cómo se estructuró la solución propuesta y cómo se materializó en los distintos componentes de software.

En primer lugar, se explican los requerimientos del sistema y el diagrama conceptual que muestra la relación entre las funciones principales del aplicativo. A continuación, se presenta el diagrama de clases, en el que se detalla el rol de cada clase, sus responsabilidades y las relaciones entre los módulos de interfaz, almacenamiento de datos y navegación. Finalmente, se describe la implementación del aplicativo organizada en varios módulos: el módulo de configuración e interfaz inicial, que permite ajustar las preferencias de conectividad, idioma y tipo de retroalimentación; el módulo de gestión de destinos, donde se registran, consultan y editan los puntos de interés; el módulo de adquisición y preprocesamiento de la señal GPS/IMU; el

módulo de procesamiento y filtrado de la navegación, responsable del cálculo de ruta y del seguimiento de la trayectoria; el módulo de realimentación auditiva, que transforma las instrucciones en voz y pitidos 2D lateralizados; y, por último, la integración completa de estos módulos en la pantalla de navegación, donde se coordina el flujo de información que recibe la persona usuaria.

3.2 DISEÑO DEL APLICATIVO MÓVIL GUIAME

Para desarrollar el aplicativo se usó la metodología SCRUM, con la cual se definieron los requerimientos funcionales y no funcionales, lo que permitió elaborar diagramas de concepto y clases, facilitando una planificación y entrega organizada del software. Otros entregables como las historias de usuario y sprints se incluyen en los anexos.

3.2.1 Requerimientos funcionales y no funcionales

Los requerimientos funcionales definen las características que el software debe cumplir según los objetivos del usuario. Para el desarrollo de la aplicación, cubriendo los objetivos planteados en este documento, se diseñan los siguientes requerimientos funcionales, y se categorizan en dos vertientes: Esenciales (E) y Opcionales (O). Esto se observa en la tabla 3-1.

Tabla 3-1 Requerimientos funcionales.

Ref #	Funciones	Categoría
1.0	El aplicativo debe permitir seleccionar entre uso de mapas en línea y mapas offline desde el módulo de configuración, antes de iniciar la navegación.	E
2.0	El aplicativo debe permitir configurar los parámetros por defecto de la navegación (idioma, volumen, tipo de realimentación, altura del usuario y uso de pasos en las instrucciones) desde la pantalla de configuración.	E
3.0	El aplicativo debe permitir seleccionar el destino deseado a partir de una lista de destinos registrados, tanto desde el módulo de gestión de destinos como desde la pantalla de navegación.	E
4.0	El aplicativo debe permitir que una persona de apoyo configure y almacene destinos, incluyendo crear, editar y eliminar destinos pregrabados con nombre y coordenadas.	E
5.0	El aplicativo debe captar y monitorear continuamente la señal de	E

	localización del usuario mediante GPS (y, cuando esté disponible, sensores inerciales), durante una sesión de navegación activa.	
6.0	El aplicativo debe filtrar y corregir la señal de localización mediante un filtro de partículas, reduciendo el ruido de la trayectoria antes de usarla para el cálculo de ruta y generación de instrucciones.	E
7.0	El aplicativo debe guiar a la persona hacia su destino mediante realimentación auditiva, utilizando instrucciones de voz sintética y/o pitidos 2D lateralizados que indiquen dirección y proximidad a los giros.	E
8.0	El aplicativo debe permitir seleccionar el tipo de realimentación (solo voz, solo pitidos lateralizados, voz + pitidos lateralizados) desde la pantalla de configuración, y aplicar esta preferencia durante toda la navegación.	E
9.0	El aplicativo debe generar una alerta clara de llegada al destino mediante realimentación auditiva, informando al usuario que ha alcanzado el punto objetivo y deteniendo la navegación.	E
10.0	El aplicativo debe recalcular la ruta cuando detecte que el usuario se ha desviado significativamente del recorrido previsto, e informar de este recálculo mediante realimentación auditiva.	E

Tabla 3-2 Requerimientos no funcionales.

Ref #	Descripción	Categoría
1.0	El aplicativo debe ser compatible con dispositivos móviles Android con versión 8.0 (Oreo) o superior.	E
2.0	El tiempo de respuesta del sistema ante cambios de ubicación debe ser menor a 2 segundos.	E
3.0	La realimentación de audio (voz o pitidos) debe activarse con una latencia menor o igual a 1 s desde que el módulo de navegación genera una nueva instrucción.	E
4.0	El aplicativo debe ser accesible para usuarios con discapacidad visual, utilizando TalkBack u otros lectores de pantalla, etiquetas de accesibilidad (contentDescription), tamaños de texto adecuados y contrastes suficientes, siguiendo las pautas de	E

	accesibilidad de Android e inspirándose en los principios de WCAG 2.1.	
5.0	El aplicativo debe ser capaz de mantener la navegación cuando se pierda temporalmente la conectividad de datos, utilizando el modo offline si está disponible, y debe informar al usuario cuando la precisión del GPS sea insuficiente para generar instrucciones fiables.	E
6.0	La información de ubicación y los destinos registrados deben almacenarse localmente en el dispositivo y no deben enviarse a servidores de terceros. El aplicativo debe solicitar consentimiento explícito para el uso de la ubicación y permitir al usuario eliminar sus destinos guardados.	E

Los requerimientos no funcionales son aquellos que describen los atributos del sistema que no están directamente relacionados con las funciones y capacidades del software, sino con su rendimiento, calidad, seguridad, usabilidad y otros aspectos que no se refieren a las funcionalidades específicas del software. Estos se muestran en la Tabla 3-2.

3.2.2 Diagrama de conceptos

En la sección 1.4 se habló de la solución propuesta y en la Figura 3.2, se muestra el diagrama de conceptos, en este se muestra de forma general las funciones del sistema y el actor principal de las mismas que es el aplicativo móvil.

El aplicativo tiene cuatro pantallas: Inicial, configuración, gestión de destinos y navegación. En la pantalla inicial se puede acceder a las demás pantallas por medio de gestos definidos y explicados. En la pantalla de configuración, el usuario establece los parámetros como uso de mapas online-offline, idioma, altura, tipo de realimentación. En la pantalla de gestión de gestión de destinos se definen y configuran los destinos a visitar. En la pantalla de navegación se escoge un destino, generando de forma automática una ruta y comenzando la navegación a partir de instrucciones accesibles definidas previamente en la configuración, filtrando constantemente la señal GPS con un filtro de partículas.

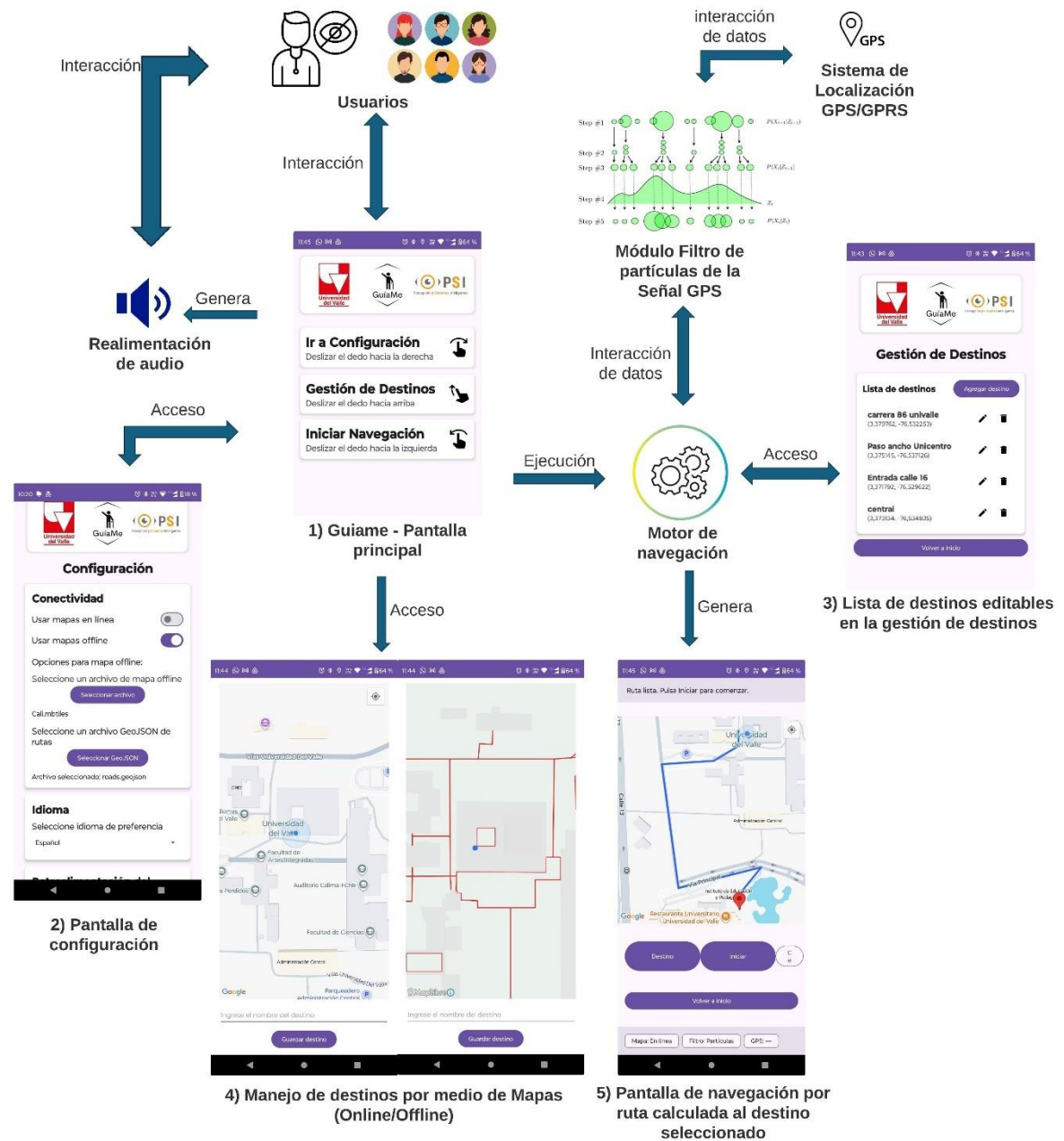


Figura 3.2 Diagrama de conceptos.

3.2.3 Diagrama de clases

La Figura 3.3 presenta una versión simplificada del diagrama de clases del aplicativo GuíaMe, organizada en tres grupos principales: interfaz de usuario (UI), gestión de datos y mapas y navegación.

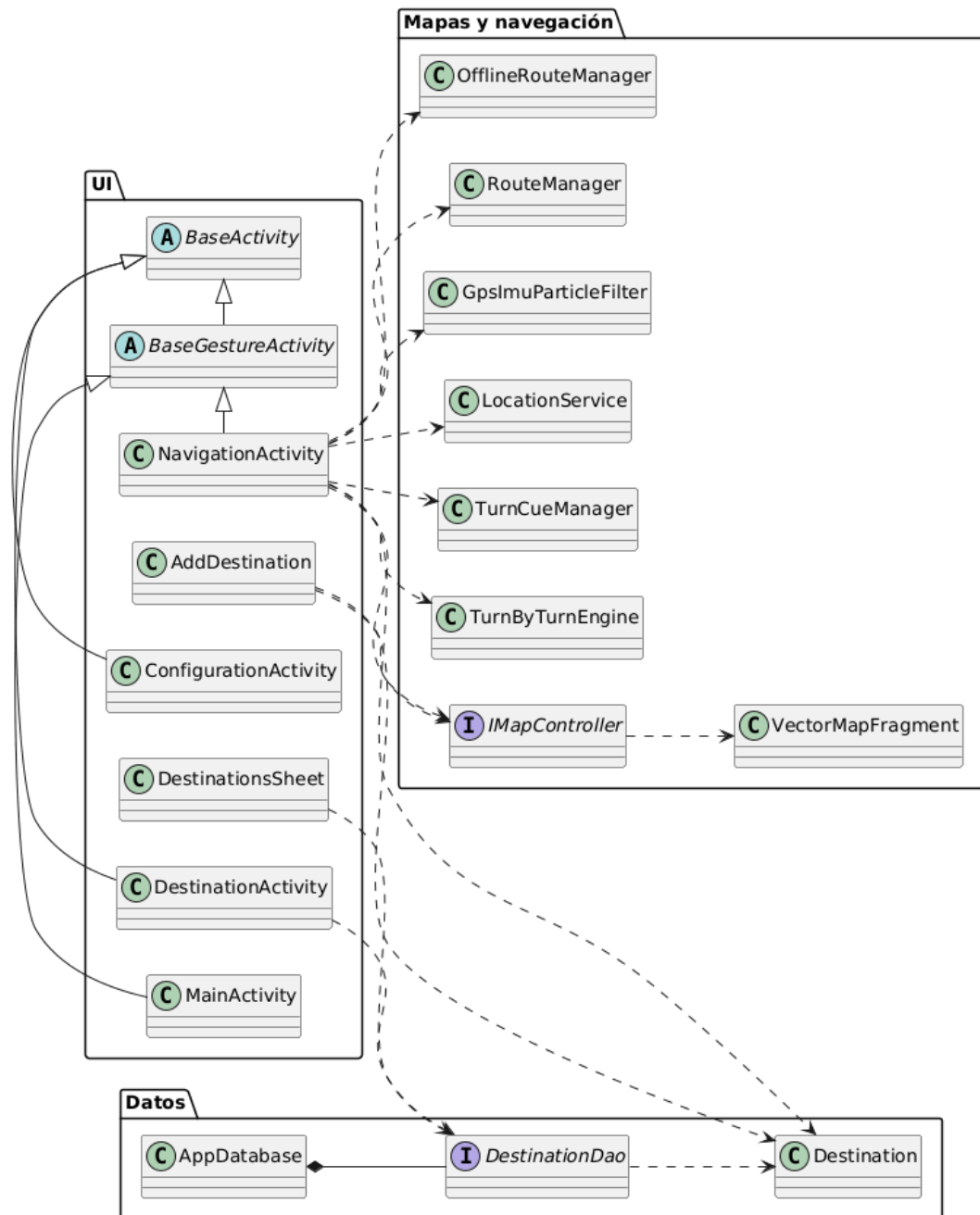


Figura 3.3 Diagrama de clases simplificado del aplicativo GuiaMe

En la capa de interfaz de usuario se encuentran las actividades y fragmentos que el usuario utiliza de forma directa. Las clases **BaseActivity** y **BaseGestureActivity** concentran el comportamiento común de las pantallas (ciclo de vida, gestión de idioma y, en el segundo caso, detección de gestos de deslizamiento). A partir de

ellas se derivan las actividades principales: MainActivity (pantalla inicial), ConfigurationActivity (configuración de mapas e idioma y tipo de realimentación), DestinationActivity y AddDestination (gestión de destinos) y NavigationActivity, que reúne la lógica de navegación paso a paso. El fragmento DestinationsSheet se utiliza desde la navegación para seleccionar rápidamente un destino almacenado.

En la capa de datos se resume el uso de Room para almacenar los destinos de forma local. La entidad Destination representa cada punto de interés, DestinationDao define las operaciones de acceso (consultar, insertar, actualizar y eliminar) y AppDatabase centraliza el acceso a la base de datos. Tanto DestinationActivity como DestinationsSheet y AddDestination interactúan con estas clases para presentar y modificar la lista de destinos.

El grupo de mapas y navegación recoge las clases que soportan la representación cartográfica y el guiado. La interfaz IMapController abstrae las operaciones básicas sobre el mapa, que se implementan mediante VectorMapFragment (modo offline). LocationService proporciona las mediciones de posición, GpsImuParticleFilter genera una trayectoria filtrada a partir de GPS/IMU, RouteManager y OfflineRouteManager calculan las rutas en modo en línea u offline y TurnByTurnEngine convierte la ruta en instrucciones de giro. Finalmente, TurnCueManager transforma esas instrucciones en realimentación auditiva para el usuario.

El diagrama completo, con todas las clases auxiliares (por ejemplo, manejo detallado de sensores, conversión de coordenadas y utilidades de permisos y audio), se incluye la sección 7 de Anexos para consulta detallada, mientras que la Figura 3.3 resume únicamente los componentes principales necesarios para entender la arquitectura general del aplicativo.

3.3 IMPLEMENTACIÓN DEL APLICATIVO MÓVIL GUIAME

En esta sección se describe la implementación del aplicativo Guíame, organizada en varios módulos que facilitan la comprensión y la futura modificación del software. En primer lugar, En la sección 3.3.1 el diagrama de bloque del aplicativo donde se resume el flujo general del aplicativo, luego en la sección 3.3.2 se presentan los módulos de configuración e interfaz inicial, desde los cuales el usuario o su acompañante ajustan los parámetros de funcionamiento y acceden a las demás pantallas. A continuación, en la sección 3.3.3 se detalla el módulo de gestión de destinos, encargado de registrar, editar y consultar los puntos a los que se puede navegar. Posteriormente, en la sección 3.3.4 se describen los módulos de adquisición y preprocesamiento de la señal GPS/IMU y en la sección 3.3.5 de procesamiento y filtrado de la navegación, responsables de obtener la posición del

usuario, estabilizar la trayectoria y calcular la ruta. Finalmente, en la sección 3.3.6 se expone el módulo de realimentación auditiva y en la sección 3.3.7 la integración completa de los módulos en la pantalla de navegación, donde se coordinan las instrucciones paso a paso que recibe la persona usuaria.

3.3.1 Diagrama de bloques del aplicativo

La Figura 3.4 muestra el diagrama de bloques del procesamiento del aplicativo, cuyo funcionamiento general está descrito en el diagrama conceptual de la figura 3.2.

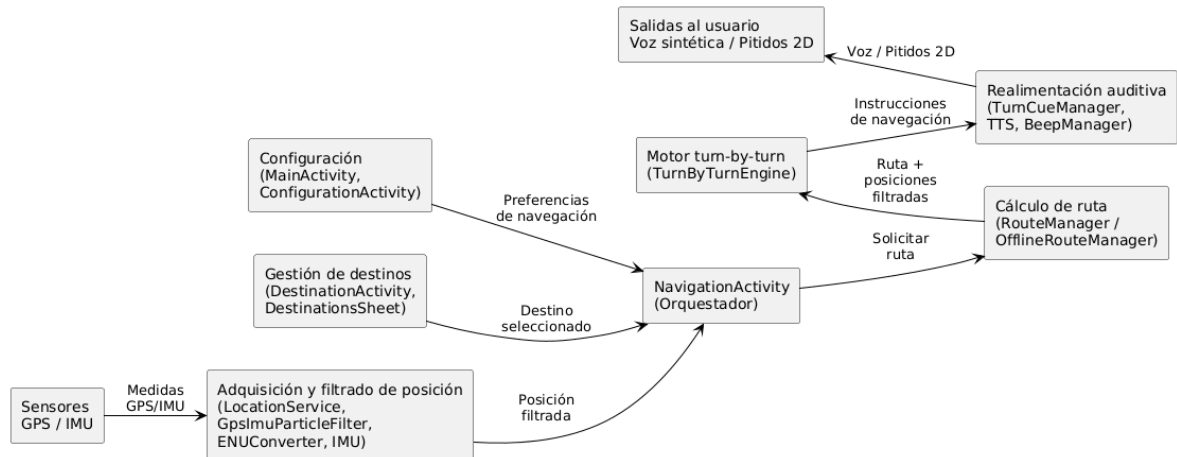


Figura 3.4 Diagrama de bloques del aplicativo

En el centro se ubica **NavigationActivity**, que actúa como orquestador del sistema: recibe las preferencias de navegación (**Configuración**), el destino seleccionado (**Gestión de destinos**) y las posiciones filtradas (**Adquisición y filtrado de posición**), delega el cálculo de ruta a **RouteManager / OfflineRouteManager** y entrega la ruta al motor turn-by-turn (**TurnByTurnEngine**), cuyas instrucciones se envían al módulo de realimentación auditiva y a las salidas al usuario (voz sintética y pitidos 2D). En la parte izquierda del esquema se representan **Configuración** y **Gestión de destinos**, mientras que en la parte superior derecha se agrupan el cálculo de ruta, el motor turn-by-turn y la realimentación auditiva, que transforman la ruta y la posición actual en indicaciones comprensibles para el usuario. De este modo, el diagrama resume cómo **NavigationActivity** coordina el intercambio de información entre los distintos módulos que se detallan en las secciones 3.3.2 a 3.3.7.

3.3.2 Módulo la interfaz inicial y de configuración

Considerando el diagrama de navegación mostrado en la figura 3.2 del diagrama de conceptos, las actividades 1 y 2 referentes a las pantallas de inicio y configuración respectivamente, agrupan a los módulos de interfaz inicial (**MainActivity**) y configuración (**ConfigurationActivity**). Estas pantallas permiten preparar el aplicativo

antes de iniciar el proceso de navegación y, al mismo tiempo, ofrecen una interacción sencilla para el usuario con discapacidad visual y su acompañante.

3.3.2.1 Interfaz inicial (MainActivity)

La Figura 3.5 presenta con más detalle la actividad 1 de diagrama de conceptos; la pantalla inicial. En la parte superior los logotipos de la Universidad del Valle, el proyecto GuíaMe y el grupo de investigación, seguidos de tres tarjetas que indican las acciones principales del sistema: Ir a Configuración, Gestión de destinos e Iniciar navegación. Cada tarjeta incluye un texto descriptivo y un ícono que ilustra el gesto correspondiente (deslizar hacia la derecha, hacia arriba o hacia la izquierda).



Figura 3.5 Pantalla interfaz inicial.

Desde el punto de vista de implementación, MainActivity hereda de BaseGestureActivity. De esta forma, toda la pantalla actúa como una superficie de gestos apto para una persona con discapacidad visual:

- Un deslizamiento hacia la derecha abre la pantalla de configuración,
- Un deslizamiento hacia arriba abre la pantalla de gestión de destinos,
- Un deslizamiento hacia la izquierda abre la pantalla de navegación.
- Opcionalmente, también se puede acceder mediante toques sobre cada tarjeta(para el funcionamiento con Talkback).

Esta decisión de diseño reduce la necesidad de pulsar botones pequeños, lo cual favorece el uso conjunto con lectores de pantalla. Las tarjetas visibles funcionan como apoyo visual para usuarios videntes o acompañantes, mientras que el gesto ocupa toda la superficie táctil.

El funcionamiento general de esta pantalla se resume en el Pseudocódigo 3.1.

Pseudocódigo 3- 1

```
01 Algoritmo InterfazInicial_MainActivity
02 Inicio
03   mostrarLogos()
04   mostrarTarjetasDeAcceso() // Configuración, Destinos, Navegación
05   al detectarGestoDeslizamiento(dir) hacer
06     si dir = DERECHA entonces abrir(ConfigurationActivity)
07     si dir = ARRIBA entonces abrir(DestinationActivity)
08     si dir = IZQUIERDA entonces abrir(NavigationActivity)
09   fin al
10 Fin
```

Este procedimiento modela el comportamiento de la pantalla inicial de Guíame. Al comenzar, se muestran los logotipos y las tarjetas que describen las tres acciones principales (configuración, gestión de destinos y navegación). La interacción central se basa en la detección de gestos de deslizamiento sobre la pantalla completa: al deslizar hacia la derecha se abre la pantalla de configuración, al deslizar hacia arriba se accede a la gestión de destinos y al deslizar hacia la izquierda se inicia la pantalla de navegación. De esta forma, la lógica de la interfaz inicial se simplifica en un conjunto de gestos globales que facilitan el acceso a las funciones principales del aplicativo.

3.3.2.2 Pantalla de configuración (ConfigurationActivity)

La Figura 3.6 presenta con más detalle la actividad 2 de diagrama de conceptos; la pantalla de configuración. Esta permite ajustar los parámetros principales del aplicativo antes de iniciar la navegación. Esta actividad se debe configurar con ayuda de la persona de apoyo. La pantalla se encuentra organizada en tarjetas temáticas con accesibilidad integrada:

- **Conectividad:** selección entre uso de mapas en línea y mapas offline.
- **Idioma:** elección del idioma de la interfaz (por ejemplo, español, inglés o francés).
- **Retroalimentación del sistema:** tipo de retroalimentación (solo voz, solo pitidos lateralizados o voz + pitidos lateralizados).
- **Altura (cm) y uso de pasos:** registro de la altura del usuario y activación de la opción para expresar las distancias en número de pasos durante las indicaciones de navegación.

Configuración

Conectividad

Usar mapas en línea ☒

Usar mapas offline ☐

Idioma

Seleccione idioma de preferencia

Español ▼

Retroalimentación del sistema

Tipo de retroalimentación

Voz ▼

Altura (cm)

175

Rango sugerido: 50–200 cm

Figura 3.6 Pantalla de configuración.

Al crear la actividad, se leen los valores almacenados en SharedPreferences y se actualiza cada control visual (interruptores, lista desplegable, barra de volumen y campo de texto). Cualquier cambio realizado por el usuario se guarda inmediatamente, de manera que la configuración se mantenga, aunque el aplicativo se cierre. De forma general, la lógica de esta pantalla se resume en el Pseudocódigo 3.2.

Pseudocódigo 3- 2

01 Algoritmo ModuloConfiguracion_ConfigurationActivity 02 Inicio 03 // Cargar y reflejar preferencias 04 prefs ← leerSharedPreferences() 05 aplicarPrefsEnInterfaz(prefs) 06 // Conectividad 07 al cambiar switchesOnlineOffline hacer 08 validarConexionSiEsOnline()	16 // Idioma y tipo de retroalimentación 17 al cambiar idioma o tipoRetroalimentacion en los spinners hacer 18 guardarPrefsIdiomaYRetro() 19 si cambialdioma entonces recargarActividad() 20 fin al 21 // Volumen, altura y uso de pasos 22 al mover barraVolumen o cambiar switchUsarPasos hacer
---	---

09 guardarPrefsConectividad()	23 guardarPrefsVolumenYPasos()
10 fin al	24 fin al
11 // Selección de archivos offline	25 al confirmarAltura(valor) hacer
12 al pulsar "Seleccionar MBTiles" o "Seleccionar GeoJSON" hacer	26 si valor válido entonces guardarAltura(valor)
13 uri ← abrirSelectorArchivo()	27 sino mostrarMensajeError()
14 si uri ≠ nulo entonces guardarUriEnPrefs(uri)	28 fin al
15 fin al	29 al pulsar "Volver a inicio" hacer
	30 cerrarActividad()
	31 fin al
	32 Fin

Este pseudocódigo describe el comportamiento principal de la pantalla de configuración. Al iniciar, la actividad lee las preferencias almacenadas y las refleja en los controles de la interfaz (líneas 3–5). Luego, se gestionan los distintos grupos de opciones mediante manejadores de eventos: al cambiar entre mapas en línea y offline se valida la conectividad y se guardan las preferencias de conectividad (líneas 7–10); al seleccionar archivos MBTiles o GeoJSON se abre un selector y, si el usuario elige un archivo válido, se guarda su URI en las preferencias (líneas 12–15).

De forma similar, los cambios de idioma y tipo de retroalimentación se registran en las preferencias y, si se modifica el idioma, se recarga la actividad para aplicar los textos actualizados (líneas 17–20). El volumen, la altura del usuario y el uso de pasos en las instrucciones se actualizan cada vez que el usuario ajusta los controles correspondientes (líneas 22–27). Finalmente, el botón “Volver a inicio” cierra la actividad y retorna a la pantalla principal (líneas 29–31). Con ello, la pantalla de configuración actúa como punto central para establecer todos los parámetros que utilizarán los módulos de navegación y realimentación.

3.3.3 Módulo de gestión de destinos

El módulo de gestión de destinos representado en el diagrama de conceptos de la Figura 3.2 como las actividades 3 y 4, permite registrar, consultar, modificar y eliminar los puntos a los que el usuario desea navegar. Está constituido principalmente por las clases `DestinationActivity`, `AddDestination` y `DestinationsSheet`, todas apoyadas en una base de datos local SQLite gestionada mediante el framework Room, descrita en la sección 3.2.3 (clases `AppDatabase` y `DestinationDao`).

3.3.3.1 Pantalla de lista de destinos (DestinationActivity)

DestinationActivity presenta la lista de destinos almacenados en la entidad Destination. La Figura 3.7 muestra la interfaz de la pantalla de gestión de destinos, donde se observa el listado de destinos y los botones de edición y eliminación.



Figura 3.7 Pantalla de gestión de destinos.

La interfaz se compone de:

- Un título (“Gestión de destinos”) que indica el propósito de la pantalla.
- Un RecyclerView donde cada fila corresponde a un destino y se muestra su nombre y coordenadas.
- Íconos de edición y eliminación en cada card para modificar o borrar el destino seleccionado.
- Un botón o gesto para agregar destino, que abre la actividad AddDestination.

Al crearse la actividad, se obtiene una referencia a AppDatabase y a DestinationDao, y se suscribe a los cambios en la tabla de destinos. Cada vez que la base de datos se actualiza, el adaptador (DestinoAdapter/DestinationAdapter) refresca la lista mostrada en pantalla. Esta suscripción garantiza que las operaciones de agregar, editar o eliminar se reflejen de forma inmediata. En la siguiente sección 3.3.3.2, se detalla a profundidad la explicación del funcionamiento de la edición de destinos. El funcionamiento principal de esta pantalla se resume en el Pseudocódigo 3.3.

Pseudocódigo 3- 3

01 Algoritmo ModuloGestionDestinos_DestinationActivity 02 Inicio 03 dao ← AppDatabase.get(context).destinationDao() 04 configurarRecyclerViewCon(adaptador) 05 observar dao.observeAll() y cuando cambie hacer 06 adaptador.actualizarLista(destinos) 07 fin observar 08 al pulsar "Agregar destino" hacer 09 abrirActividad(AddDestination, modo = NUEVO) 10 fin al	11 al pulsar íconoEditar(dest) en un ítem hacer 12 abrirActividad(AddDestination, modo = EDITAR, destino = dest) 13 fin al 14 al pulsar íconoEliminar(dest) en un ítem hacer 15 si usuarioConfirma() entonces dao.delete(dest) 16 fin al 17 Fin
--	--

Este pseudocódigo resume el comportamiento principal de la pantalla de gestión de destinos. Al iniciarse, la actividad obtiene una referencia al DestinationDao y configura el RecyclerView con su adaptador (líneas 3–4). Luego se suscribe al flujo observeAll() de la base de datos: cada vez que cambia la lista de destinos, el adaptador actualiza el contenido mostrado en pantalla (líneas 5–7).

Las acciones del usuario se gestionan mediante eventos: al pulsar “Agregar destino” se abre AddDestination en modo NUEVO (líneas 8–10); al pulsar el ícono de edición de un ítem se abre la misma actividad en modo EDITAR, enviando el destino seleccionado (líneas 11–13); y al pulsar el ícono de eliminación se solicita confirmación al usuario y, si la respuesta es afirmativa, se elimina el destino de la base de datos (líneas 14–16).

3.3.3.2 Registro y edición de destinos (AddDestination)

La actividad AddDestination se encarga de registrar un nuevo destino o editar uno existente. Su interfaz combina:

- Un campo de texto para ingresar o modificar el nombre del destino.
- Un mapa (en línea u offline, según la configuración previa) que permite seleccionar el punto geográfico.
- La creación de un pin al mantener presionado un punto del mapa por un tiempo prolongado.
- Un botón para guardar los datos y regresar a la pantalla anterior.

En la Figura 3.8 se presenta la pantalla de adición/edición de destino, donde el usuario selecciona el punto en el mapa y asigna un nombre.

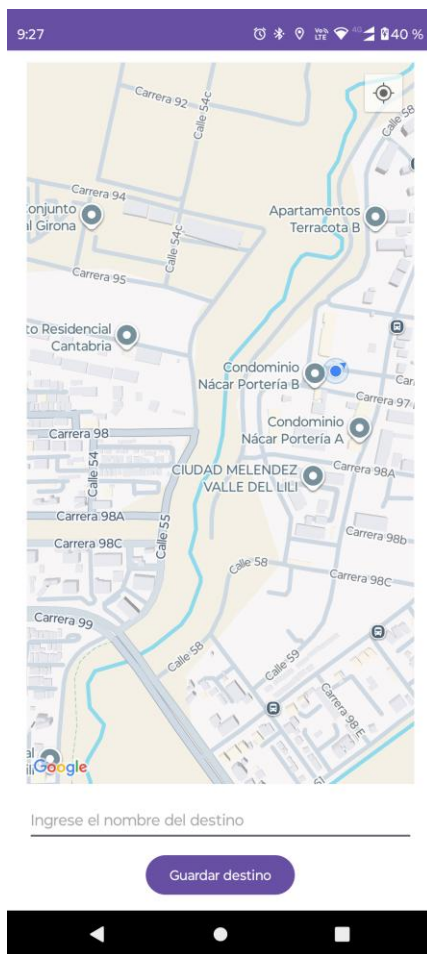


Figura 3.8 Pantalla de adición/edición de destinos.

Cuando la actividad se crea en modo “nuevo”, el mapa se centra en la posición actual del dispositivo y espera a que el usuario seleccione el punto del destino mediante un toque. En modo “editar”, se muestran el nombre y la ubicación previamente almacenados. Al confirmar, se devuelve el resultado a DestinationActivity, que se encarga de insertar o actualizar la entidad en la base de datos. El funcionamiento de esta pantalla se resume en el Pseudocódigo 3.4.

Pseudocódigo 3- 4

01 Algoritmo ModuloGestionDestinos_AddDestination 02 Inicio	14 al pulsar "Guardar" hacer 15 nombre ← textoDe(etNombre) 16 si nombre es vacío o destinoSeleccionado = nulo entonces
---	--

03 leerParametrosIntent(modo, destinoInicial)	17 mostrarMensaje("Ingrese un nombre y seleccione un punto")
04 si modo = EDITAR entonces	18 sino
05 etNombre ← destinoInicial.nombre	19 destinoConstruido ←
06 colocarMarcador(destinoInicial.lat, destinoInicial.lng)	construirDestino(nombre, destinoSeleccionado, modo)
07 fin si	20 devolverResultadoOK(destinoConstruido)
08 usarOnline ← leerPreferencia("usarMapasOnline")	21 cerrarActividad()
09 inicializarMapa(usarOnline)	22 fin si
10 al tocarMapa(posicion) hacer	23 fin al
11 actualizarMarcador(posicion)	24 Fin
12 destinoSeleccionado ← posicion	
13 fin al	

Este pseudocódigo describe cómo la actividad AddDestination maneja tanto la creación como la edición de destinos. Primero lee los parámetros recibidos en el Intent y, si el modo es EDITAR, carga el nombre y la posición inicial del destino, mostrando el marcador correspondiente en el mapa (líneas 3–7). Luego decide si el mapa se inicializa en modo en línea u offline según la preferencia almacenada (líneas 8–9).

Durante la interacción, cada toque sobre el mapa actualiza la posición del marcador y guarda la coordenada seleccionada como destino candidato (líneas 10–13). Al pulsar el botón “Guardar”, se verifica que exista un nombre y que se haya seleccionado un punto; si falta alguno, se muestra un mensaje de error (líneas 15–17). En caso contrario, se construye el objeto destino, se devuelve como resultado a la actividad que llamó (por ejemplo, DestinationActivity) y se cierra la pantalla (líneas 19–21).

3.3.3.3 Registro y edición de destinos (AddDestination)

Aunque la gestión completa se realiza en DestinationActivity, durante la navegación resulta útil seleccionar un destino sin abandonar la pantalla principal. Para ello se utiliza el fragmento DestinationsSheet, que implementa un panel tipo “bottom sheet” con un listado compacto de los destinos almacenados.

La Figura 3.9 muestra el panel DestinationsSheet utilizado durante la navegación para seleccionar rápidamente un destino.

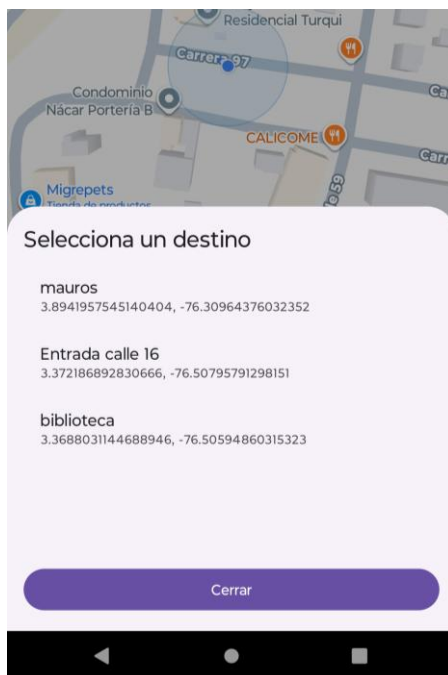


Figura 3.9 Lista de destinos para iniciar la navegación

DestinationsSheet reutiliza AppDatabase y DestinationDao para cargar la lista de destinos y utiliza DestinationAdapter para mostrarla. Al seleccionar un destino, el fragmento lo envía de vuelta a NavigationActivity, que lo emplea para calcular la ruta. De esta forma, la misma base de datos de destinos sirve tanto para la gestión previa como para la selección dinámica durante el recorrido. El funcionamiento de esta pantalla se resume en el Pseudocódigo 3.5.

Pseudocódigo 3- 5

01 Algoritmo ModuloGestionDestinos_DestinationsSheet 02 Inicio 03 dao ← AppDatabase.get(context).destinationDao() 04 configurarRecyclerViewCon(adaptador) 05 observar dao.observeAll() y cuando cambie hacer 06 adaptador.actualizarLista(destinos) 07 fin observar	08 al seleccionarItem(dest) en el listado hacer 09 enviarDestinoSeleccionadoANavigationActivity(dest) 10 cerrarSheet() 11 fin al 12 Fin
--	--

Este pseudocódigo corresponde al fragmento DestinationsSheet, que permite seleccionar un destino rápidamente sin salir de la pantalla de navegación. Al iniciarse, el fragmento obtiene el DestinationDao e inicializa el RecyclerView con su

adaptador (líneas 3–4). Luego se suscribe a `dao.observeAll()`, de modo que cualquier cambio en la tabla de destinos (creación, edición o eliminación) se refleja automáticamente en la lista mostrada (líneas 5–7).

Cuando el usuario selecciona un elemento del listado, el fragmento envía el destino elegido de vuelta a `NavigationActivity` y se cierra a sí mismo (líneas 8–11). Así, `DestinationsSheet` reutiliza la misma base de datos de destinos, pero en un formato compacto y accesible durante la navegación.

3.3.4 Módulo de adquisición y preprocesamiento de la señal GPS/IMU

En la figura 3.2 correspondiente al diagrama de conceptos, se presenta la actividad correspondiente al motor de navegación, representando el módulo de adquisición y preprocesamiento, encargado de obtener la posición del dispositivo de forma continua, verificar los permisos necesarios y preparar las mediciones para su posterior filtrado y uso en la navegación. Este módulo está compuesto principalmente por el servicio de localización `LocationService`, el gestor de permisos (`PermissionHelper`) y las clases `ENU`, `ENUConverter` y `StepEstimator`, que participan en la representación geométrica de la trayectoria y en la adaptación de las distancias a la forma en que se presentarán al usuario. En este contexto, `ENU` (East–North–Up) corresponde a un sistema de coordenadas local en el que cada posición se expresa mediante sus componentes este y norte respecto a un punto de referencia, lo que facilita el cálculo de distancias y desplazamientos sobre el recorrido.

3.3.4.1 Gestión de permisos y arranque del servicio de localización

La figura 3.10 muestra la pantalla de solicitud de permisos al iniciar el aplicativo por primera vez.



Figura 3.10 Solicitud de permisos de ubicación.

Antes de iniciar la adquisición de datos de GPS, la aplicación debe comprobar que cuenta con los permisos de ubicación en primer plano y, cuando aplica, en segundo plano. Esta tarea se centraliza en `PermissionHelper`, el cual se invoca desde `MainActivity` y desde la pantalla de navegación. Si los permisos no están concedidos, se solicita al usuario mediante los diálogos estándar de Android; solo cuando la autorización es otorgada se permite iniciar `LocationService` en primer plano.

Una vez verificados los permisos, `NavigationActivity` ordena el arranque del servicio de localización. `LocationService` configura las peticiones de ubicación con prioridad alta (alta precisión) y una frecuencia aproximada de actualización de 1 Hz, de modo que se disponga de una trayectoria suficientemente densa para el módulo de filtrado sin comprometer en exceso el consumo energético del dispositivo.

El flujo general del módulo, desde la verificación de permisos hasta la preparación de las posiciones para la navegación, se resume en el Pseudocódigo 3.6.

3.3.4.2 Adquisición de posiciones con `LocationService`

`LocationService` se ejecuta como servicio en primer plano, lo que garantiza que la captura de posiciones continúe incluso si la actividad de navegación pasa a segundo plano mientras el usuario camina. Cada vez que el sistema de localización entrega una nueva estimación puntual de la posición formada por latitud, longitud y una medida de precisión, el servicio obtiene estos valores y los empaqueta en un mensaje que se envía a `NavigationActivity` mediante un mecanismo de difusión (broadcast o callback).

De esta forma, la actividad de navegación no interactúa directamente con las API de localización de bajo nivel, sino que recibe una secuencia de eventos de posición procedentes de `LocationService` y, a partir de ellos, aplica el filtrado de la trayectoria y el cálculo de ruta. Esto facilita la separación de responsabilidades y permite mantener un único punto de configuración de la frecuencia y prioridad de las medidas.

3.3.4.3 Transformación a sistema local ENU y estimación de pasos

Las coordenadas de GPS se entregan en términos de latitud y longitud sobre el elipsoide WGS84, lo que no resulta práctico para calcular distancias y desplazamientos locales sobre el recorrido. Para abordar esto, el aplicativo utiliza un sistema de coordenadas local ENU (Este–Norte), representado por la clase `ENU`, que almacena las componentes *x* y *y* respecto a un punto de referencia. La conversión entre coordenadas geográficas (LatLng) y coordenadas locales se realiza mediante `ENUConverter`, el cual es utilizado internamente por el filtro de partículas (`GpsImuParticleFilter`) para trabajar en un sistema métrico más adecuado para el seguimiento de la trayectoria.

Adicionalmente, se emplea la clase StepEstimator para vincular las distancias en metros con el número de pasos del usuario, a partir de la altura registrada previamente en el módulo de configuración. Este componente aplica una relación aproximada entre altura y longitud de paso y ofrece una función metersToSteps(meters, heightMeters) que convierte directamente la distancia en metros al número de pasos. De esta manera, cuando el usuario así lo haya configurado, las indicaciones de navegación pueden expresarse en forma más natural (“en 20 pasos gire a la derecha”) en lugar de únicamente en metros. El flujo integrado del módulo de adquisición y preprocesamiento de la señal GPS/IMU se presenta de forma resumida en el Pseudocódigo 3.6.

Pseudocódigo 3- 6

01	Algoritmo	□ // 3. En LocationService: configurar actualizaciones GPS
02	ModuloAdquisicionPreprocesamiento	
03	Inicio	13 configurarProveedorUbicacion(prioridadAlta, frecuenciaAprox1Hz)
04	// 1. Verificar permisos de ubicación	14 al recibirNuevaUbicacion(lat, lng, acc) hacer
05	si no hayPermisoUbicacion() entonces	15 enviarUbicacionANavigationActivity(lat, lng, acc)
06	solicitarPermisoUbicacion()	16 fin al
07	si usuarioNoAutoriza() entonces	17 // 4. En NavigationActivity: recibir posición y delegar al módulo de filtrado/navegación
08	mostrarMensaje("No se puede iniciar la navegación sin permisos de ubicación")	18 al recibirUbicacion(lat, lng, acc) hacer
09	terminarProceso()	19 rawLatLng ← construirLatLng(lat, lng)
10	fin si	20 delegarAlModuloDeFiltradoY Navegacion(rawLatLng, acc)
11	fin si	21 fin al
12	// 2. Iniciar servicio de localización en primer plano	22 Fin
	iniciarServicio(LocationService)	

Este pseudocódigo resume el funcionamiento del módulo de adquisición y preprocesamiento. En primer lugar, se verifica que la aplicación cuente con los permisos de ubicación necesarios; si no es así, se solicitan al usuario y, en caso de que no sean concedidos, se muestra un mensaje y se detiene el proceso de navegación (líneas 3–10).

Si los permisos están disponibles, se inicia el servicio de localización en primer plano (LocationService), donde se configuran las actualizaciones de GPS con alta prioridad y una frecuencia aproximada de 1 Hz (líneas 11–14). Cada nueva posición

obtenida por el servicio (latitud, longitud, precisión) se envía a NavigationActivity mediante un mecanismo de comunicación interno (líneas 15–17).

Finalmente, en NavigationActivity, cada ubicación recibida se transforma en un objeto LatLng y se delega al módulo de filtrado y navegación, que será el encargado de suavizar la trayectoria, calcular la ruta y generar las instrucciones correspondientes (líneas 19–21). De esta forma, el módulo separa claramente la captura de datos de su procesamiento posterior.

3.3.5 Módulo de procesamiento y filtrado de la navegación

En la figura 3.2 correspondiente al diagrama de conceptos, se presenta el flujo a partir de la actividad correspondiente al motor de navegación, la adquisición y filtrado de la señal GPS/GPRS representando el módulo de procesamiento y filtrado de la navegación. Este toma como entrada las mediciones de GPS proporcionadas por LocationService, junto con la información derivada de la IMU, y genera una trayectoria estabilizada sobre la cual se calculan rutas e instrucciones de giro. Internamente trabaja en coordenadas locales ENU mediante ENUConverter, pero hacia el resto de la aplicación expone posiciones filtradas en forma de LatLng. Este módulo está conformado principalmente por las clases Tracker, PFTracker, GpsImuParticleFilter, RouteManager, OfflineRouteManager y TurnByTurnEngine.

3.3.5.1 Selección del filtro de navegación

La elección de PFTracker / GpsImuParticleFilter como filtro principal de navegación no fue arbitraria. Previamente se desarrolló una aplicación de pruebas independiente en Android, que utilizó Google Maps y un servicio de localización en primer plano (≈ 1 Hz, prioridad alta) para comparar tres enfoques de filtrado sobre varias trayectorias reales: filtro de Kalman clásico (KF), filtro de Kalman extendido (EKF) y filtro de partículas (PF). En cada trayectoria se definió una ruta de referencia (bien sea consultando Google Directions o grabando el recorrido real), se transformaron las posiciones GNSS a un sistema local ENU y se midió, en cada fix, la distancia ortogonal punto–ruta como error de navegación.

Para cada filtro se registraron las posiciones crudas y filtradas y se calcularon métricas como el error cuadrático medio global (RMSE_total_global) y los percentiles p95/p99 de la distancia al recorrido de referencia, tanto por trayectoria como en forma agregada. Los resultados mostraron un empate muy estrecho en precisión global entre KF, EKF y PF, con una ligera ventaja del filtro de partículas (por ejemplo, un RMSE_total_global $\approx 7,81$ m, frente a $\approx 7,85$ m para KF y $\approx 7,88$ m para EKF), y una pequeña mejora adicional de PF en los indicadores de cola (p95/p99), es decir, en el control de los desvíos más grandes.

Dado que, para la aplicación Guíame, resulta especialmente importante limitar los errores grandes que afectan la experiencia de guiado (más que ganar unas décimas

en el RMSE medio), se seleccionó PF como filtro por defecto por ofrecer el mejor compromiso entre reducción del ruido instantáneo y control de colas. No obstante, el EKF se mantiene como alternativa viable en trabajos futuros si se desea explorar configuraciones con menor complejidad de implementación o validar su desempeño en escenarios GNSS más adversos. El diseño experimental detallado, junto con las tablas de resultados y las figuras comparativas por trayectoria, se recoge en la sección 7 de Anexos. Informe de selección de filtros, al que se remite al lector interesado en el análisis completo. El funcionamiento integrado del módulo de procesamiento y filtrado de navegación se resume en el Pseudocódigo 3-7.

Pseudocódigo 3- 7

01 Algoritmo ModuloProcesamientoFiltradoNavegacion 02 Inicio 03 // 1. Selección de destino y cálculo de ruta 04 al seleccionarDestino(destino) hacer 05 origen ← posicionFiltradaActual() // última estimación conocida 06 si modoOnline entonces 07 ruta ← RouteManager.getRouteOnline(origen, destino) 08 sino 09 ruta ← OfflineRouteManager.getRouteOffline(origen, destino) 10 fin si 11 stepLengthEstimado ← calcularLongitudPaso(alturaUsuario) 12 TurnByTurnEngine.setRoute(ruta, stepLengthEstimado) 13 fin al 14 // 2. Procesamiento de datos IMU (estimación de controles de movimiento) 15 al recibirDatosIMU(ax, ay, az) hacer 16 ImuControls.updateFromImu(ax, ay, az) 17 controles ← ImuControls.obtenerControles() // p.ej. (speedMps, yawRateRad)	□ // 3. Filtrado de la trayectoria y generación de instrucciones □ al recibirMedidaGPS(rawLatLng, acc) hacer 20 posFiltrada ← GpsImuParticleFilter.updateGps(rawLatLng, acc) 21 instruccion ← TurnByTurnEngine.updatePosition(posFiltrada) 22 si instruccion ≠ nulo entonces 23 enviarInstruccionAlModulo DeRealimentacion(instruccion) 24 fin si 25 si TurnByTurnEngine.hasArrived() entonces 26 enviarInstruccionAlModulo DeRealimentacion("Has llegado al destino") 27 detenerNavegacion() 28 fin si 29 fin al 30 Fin
---	---

18 GpsImuParticleFilter.actualizarControles(controles)	
19 fin al	

En las líneas 4–13 se describe cómo, cuando el usuario selecciona un destino, el sistema toma la última posición filtrada como origen, decide si usar ruteo en línea u offline (líneas 6–10), calcula la ruta correspondiente y configura el motor turn-by-turn con dicha ruta y con la longitud de paso estimada a partir de la altura del usuario (líneas 11–12).

En las líneas 15–19 se procesa la información procedente de la IMU: se actualizan los controles de movimiento (por ejemplo, velocidad y velocidad de giro) y estos se entregan al filtro de partículas para mejorar la predicción del movimiento del usuario.

Finalmente, en las líneas 21–30 cada nueva medición de GPS se filtra mediante GpsImuParticleFilter para obtener una posición suavizada, que se pasa a TurnByTurnEngine. Este componente genera, cuando corresponde, una instrucción de navegación (líneas 23–26) y comprueba si el usuario ha llegado al destino, en cuyo caso emite el mensaje final de llegada y detiene la navegación (líneas 27–30).

3.3.5.2 Adecuación de datos y filtrado de la trayectoria (Tracker, PFTracker, GpsImuParticleFilter)

La interfaz Tracker define las operaciones básicas de un filtro de seguimiento de trayectoria (reset, predict, update, estimate), mientras que PFTracker implementa dichas operaciones mediante un filtro de partículas tipo SIR (Sequential Importance Resampling) o bootstrap particle filter. En esta implementación, cada partícula representa una hipótesis del estado del usuario en coordenadas locales ENU, considerando la posición (x, y) , la velocidad lineal v , la orientación ψ y la velocidad angular w .

Durante la etapa de predicción, las partículas se propagan con un modelo cinemático tipo CTRV (velocidad y velocidad angular aproximadamente constantes), al cual se añade ruido de proceso en velocidad, orientación, velocidad angular y posición. Posteriormente, en la etapa de actualización, cada nueva medición GNSS se transforma a coordenadas ENU y se utiliza para calcular la verosimilitud de cada partícula en función de su distancia a la observación. Los pesos se normalizan y, cuando el número efectivo de partículas cae por debajo de un umbral, se aplica remuestreo sistemático, seguido de una etapa de roughening para mantener la diversidad muestral. La estimación puntual de la trayectoria se obtiene como la media ponderada de las partículas.

Sobre esta base se construye la clase GpsImuParticleFilter, que encapsula a PFTracker y gestiona la conversión entre coordenadas geográficas y coordenadas

locales ENU mediante ENUConverter. Además, recibe información procedente de ImuManager e ImuControls, a partir de la cual se estiman una velocidad lineal y una velocidad de giro (speedMps, yawRateRad). En la implementación actual, estos controles inerciales no reemplazan directamente el estado interno de las partículas, sino que se utilizan para modular la propagación temporal de la etapa de predicción antes de corregir con la medición GPS. Finalmente, la estimación filtrada se transforma nuevamente a LatLng, de modo que el resto de la aplicación opera con una posición suavizada y consistente con la navegación paso a paso.

3.3.5.3 Cálculo de rutas en línea y offline (RouteManager, OfflineRouteManager)

Para llevar al usuario desde su posición actual hasta un destino registrado, el aplicativo ofrece dos mecanismos de cálculo de rutas:

- RouteManager se encarga de solicitar rutas en línea, apoyándose en servicios de ruteo externos cuando hay conectividad.
- OfflineRouteManager calcula rutas sobre los datos de mapa almacenados localmente (modo offline), utilizando la información del archivo vectorial.

En ambos casos, el resultado se representa como una secuencia de puntos geográficos (LatLng) que describe el recorrido previsto entre el origen y el destino. Esta secuencia se entrega a TurnByTurnEngine para su posterior análisis.

La figura 3.11 muestra una ruta calculada en el mapa online al seleccionar un destino guardado.

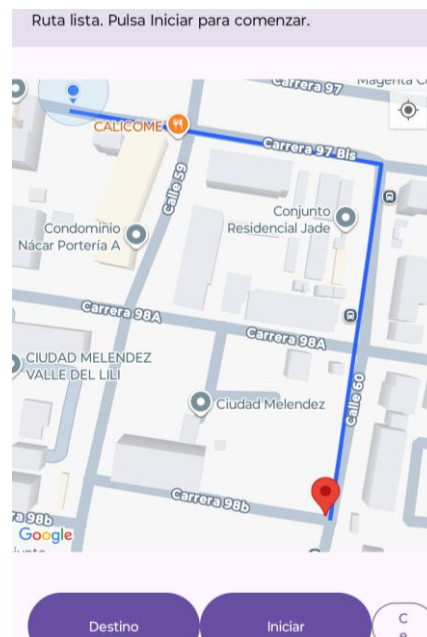


Figura 3.11 Ruta calculada en el mapa online.

3.3.5.4 Generación de instrucciones de navegación (TurnByTurnEngine)

TurnByTurnEngine recibe la ruta como una lista de puntos y la procesa para identificar los tramos rectos y los giros relevantes. A partir de esta información:

- Genera una lista de maniobras con su tipo (giro a la derecha, izquierda, seguir recto) y su posición dentro de la ruta,
- Calcula en cada instante la distancia desde la posición filtrada del usuario hasta el siguiente giro,
- Y, con base en umbrales predefinidos (por ejemplo, 120 m, 60 m, 25 m y “ahora”), decide cuándo emitir cada instrucción.

Cuando se supera el vértice de un giro, el motor avanza a la siguiente maniobra de la lista; al aproximarse suficientemente al destino, genera un mensaje de llegada. El texto generado (en metros o pasos, según la configuración) se entrega al módulo de realimentación (sección 3.3.5), que lo convierte en instrucciones de voz y/o secuencias de pitidos 2D lateralizados, de acuerdo con el modo de feedback seleccionado por el usuario que se explicará en la sección 3.3.5.

El funcionamiento integrado del módulo de procesamiento y filtrado de navegación se resume en el Pseudocódigo 3-8.

Pseudocódigo 3- 8

01 Procedimiento InicializarTurnByTurn(rutaPuntos)	22 // 2. Seleccionar umbral y mensaje según la distancia
02 Inicio	23 si dist > 120 m entonces
03 // 1. Analizar la ruta y detectar giros	24 instrucion ← nulo // aún es muy pronto para avisar
04 listaManiobras ← []	25 sino si dist > 60 m entonces
05 para cada segmento consecutivo (p_i , p_{i+1} , p_{i+2}) en rutaPuntos **hacer**	26 instrucion ← ConstruirMensaje("lejos", maniobra) // p.ej. "En 120 m gire a la derecha"
06 angulo ← CalcularCambioDeDireccion(p_i , p_{i+1} , p_{i+2})	27 sino si dist > 25 m entonces
07 si angulo ≥ umbralGiro entonces	28 instrucion ← ConstruirMensaje("cerca", maniobra) // p.ej. "En 60 m gire a la derecha"
08 tipo ← ClasificarGiro(angulo) // derecha, izquierda, seguir recto	29 sino si dist > 0 m **entonces**
09 posicionGiro ← p_{i+1}	30 instrucion ← ConstruirMensaje("ahora", maniobra) // p.ej. "Ahora gire a la derecha"
10 añadir (tipo, posicionGiro) a listaManiobras	31 sino
11 fin si	
12 fin para	

13 indiceManiobraActual ← 0	32 instruccion ← nulo
14 Fin Procedimiento	33 fin si
15 Procedimiento ActualizarInstrucciones(posFiltrada)	34 // 3. Avanzar a la siguiente maniobra cuando se pasa el vértice
16 Inicio	35 si dist ≤ 0 m **entonces**
17 si indiceManiobraActual ≥ tamaño(listaManiobras) entonces	36 indiceManiobraActual ← indiceManiobraActual + 1
18 retornar nulo // no hay más giros	37 fin si
19 fin si	38 retornar instruccion
20 maniobra ← listaManiobras[indiceManiobraActual]	39 Fin Procedimiento
21 dist ← CalcularDistancia(posFiltrada, maniobra.posicionGiro)	

En el procedimiento InicializarTurnByTurn (líneas 1–14) la ruta se recorre para detectar cambios significativos de dirección entre tramos consecutivos; cada cambio que supera un umbral angular se registra como una maniobra con su tipo (giro a la derecha, izquierda o seguir recto) y su posición dentro de la ruta. En el procedimiento ActualizarInstrucciones (líneas 16–44), a partir de la posición filtrada actual se calcula la distancia al siguiente giro (líneas 22–23) y, usando umbrales predefinidos (por ejemplo, 120 m, 60 m, 25 m y “ahora”), se decide si generar una instrucción lejana, cercana o inmediata, o no emitir nada todavía (líneas 26–36). Cuando el usuario sobrepasa el vértice del giro, se avanza a la siguiente maniobra (líneas 38–41).

3.3.6 Módulo de realimentación. (TurnCueManager, FeedbackManager, tipos de feedback: voz y pitidos)

En la figura 3.2 correspondiente al diagrama de conceptos, se presenta la actividad correspondiente la realimentación del aplicativo, mostrando un flujo de interacción directo entre el aplicativo y los usuarios. Este módulo se encarga de transformar las indicaciones de navegación generadas por el motor turn-by-turn en señales de realimentación auditiva adaptadas a las necesidades del usuario con discapacidad visual. Para ello se implementan dos tipos de feedback: voz sintética y pitidos 2D lateralizados, que pueden seleccionarse de forma independiente o combinada desde la pantalla de configuración de la aplicación.

En etapas iniciales de diseño se consideró también el uso de vibración como canal háptico adicional. Lo cual se realizó en el marco del proyecto de investigación previo a este trabajo de grado. Sin embargo, tras las pruebas preliminares y con el fin de simplificar la interacción y concentrar la atención del usuario en el canal auditivo, la

vibración se descartó en la versión final del prototipo, manteniendo únicamente voz y pitidos.

La lógica central de la realimentación auditiva se concentra en la clase TurnCueManager. Esta clase no calcula la ruta, sino que recibe como entrada el resultado del motor de navegación: una instrucción textual ya formateada (por ejemplo, “En 20 pasos, gire a la izquierda”) y, cuando está disponible, un ángulo relativo que indica en qué dirección se encuentra el siguiente punto de la ruta con respecto al cuerpo del usuario (izquierda, frente o derecha). A partir de estos elementos, TurnCueManager decide qué canales de salida activar (síntesis de voz y/o pitidos) y, en el caso de los pitidos, cómo distribuir el sonido entre los canales izquierdo y derecho de los audífonos o auriculares, asignándolo a una de cinco regiones del espacio frontal (izquierda, diagonal izquierda, frente, diagonal derecha o derecha). De este modo, el módulo traduce una instrucción abstracta (“gire a la izquierda”) en una señal auditiva concreta y espacializada, coherente con la situación de navegación del usuario.

Por su parte, el componente FeedbackManager se utiliza principalmente desde la pantalla de configuración para inicializar y probar la síntesis de voz y los sonidos de ejemplo, mientras que la lógica de navegación en tiempo real se apoya directamente en TurnCueManager y BeepManager para generar las indicaciones auditivas durante el recorrido.

3.3.6.1 Selección del tipo de feedback en la pantalla de configuración

La pantalla de configuración ofrece al acompañante o al propio usuario la posibilidad de elegir el tipo de realimentación que desea utilizar durante la navegación. Esta selección se realiza mediante un Spinner que presenta opciones como:

- **Solo voz:** el sistema únicamente enuncia instrucciones habladas, por ejemplo: “En 10 pasos, gire a la derecha”.
- **Solo pitidos lateralizados:** el sistema no habla, sino que utiliza secuencias de pitidos 2D que suenan más fuerte por el canal izquierdo o derecho según la dirección del giro, por ejemplo, varios pitidos hacia el oído derecho para indicar “corrija hacia la derecha”.
- **Voz + pitidos lateralizados:** el sistema combina ambos canales; por ejemplo, la voz indica “En 10 pasos, corrija ligeramente hacia la izquierda”, y al mismo tiempo se reproducen pitidos predominantes en el canal izquierdo que refuerzan esa corrección.

La opción elegida se almacena en SharedPreferences bajo la clave tipoRetroalimentacion, como un índice entero que luego se mapea internamente a un enum (FeedbackMode). De esta manera, cada vez que se dispara una indicación

de navegación, TurnCueManager consulta este valor y decide si debe hablar, emitir pitidos o combinar ambas modalidades.

La selección del tipo de feedback (voz, pitidos lateralizados o ambos) se observa en la Figura 3.12.

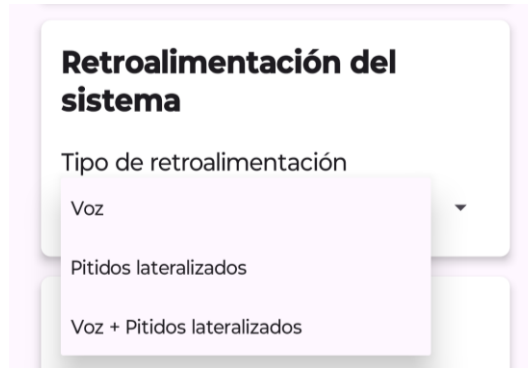


Figura 3.12 Selector del tipo de realimentación.

Aunque durante el diseño se evaluó el uso de vibración como canal adicional, este modo fue descartado en la versión final del sistema para concentrar la interacción en señales auditivas coherentes (voz y pitidos 2D lateralizados) y evitar sobrecargar al usuario con múltiples tipos de estímulos simultáneos.

3.3.6.2 Cálculo del ángulo relativo ruta–usuario

Para poder lateralizar los pitidos, el sistema necesita saber en qué dirección está el siguiente punto de la ruta con respecto al cuerpo del usuario. Este cálculo se realiza en la actividad de navegación, a partir de tres elementos:

1. Posición actual filtrada (lastFiltered)
Es la posición GPS ya suavizada por el filtro de partículas.
2. Dirección de la ruta hacia adelante (routeForwardBearingDeg)
A partir de la polilínea de la ruta, se calcula el bearing (azimut) desde la posición actual hacia el siguiente punto significativo del recorrido.
3. Heading del usuario (lastHeadingRad)
La IMU del dispositivo proporciona la orientación del teléfono respecto al norte. Esta orientación se convierte a grados (userHeadingDeg) y se toma como referencia del eje frontal del usuario (0°).

Con estos valores se calcula el ángulo relativo entre la ruta y el usuario. En primer lugar, se obtiene la diferencia entre el rumbo de la ruta y el heading del usuario según la ecuación 3.1:

$$(3.1) \theta_{diff} = bearing_{ruta} - heading_{usuario}$$

A continuación, esta diferencia se normaliza al rango $[-180^\circ, 180^\circ]$ mediante una función de normalización angular, como se observa en la Ecuación 3.2:

$$(3.2) \theta_{rel} = NormalizarAngulo(\theta_{diff})$$

De esta forma, un valor $\theta_{rel} \approx 0^\circ$ indica que la ruta está justo al frente del usuario, valores negativos indican que el siguiente punto de la ruta se encuentra hacia la izquierda y valores positivos indican que se encuentra hacia la derecha.

Este ángulo relativo se pasa como parámetro opcional a `TurnCueManager.cue(...)`, para que el módulo de pitidos pueda determinar en qué región del espacio debe ubicarse el sonido.

Este funcionamiento se resume en el pseudocódigo 3-9.

Pseudocódigo 3- 9

01 Procedimiento CalcularAnguloRelativoRutaUsuario() 02 Inicio 03 // 1. Obtener estado actual de navegación 04 lastFiltered ← ObtenerUltimaPosicionFiltrada() 05 routeForwardBearingDeg ← CalcularBearingRuta(lastFiltered) 06 lastHeadingRad ← ObtenerHeadingIMU() 07 // 2. Verificar que los datos sean válidos 08 **si** lastFiltered es NULO o routeForwardBearingDeg es NULO o lastHeadingRad es NULO entonces 09 retornar NULO // no hay ángulo fiable 10 fin si	11 // 3. Convertir heading del usuario a grados 12 userHeadingDeg ← ConvertirRadAGrados(lastHeadingRad) 13 // 4. Calcular diferencia de ángulos ($\theta_{diff} =$ bearing_ruta – heading_usuario) 14 thetaDiff ← routeForwardBearingDeg – userHeadingDeg 15 // 5. Normalizar al rango $[-180^\circ, 180^\circ]$ ($\theta_{rel} = NormalizarAngulo(\theta_{diff})$) 16 thetaRel ← NormalizarAngulo(thetaDiff) 17 // 6. Devolver ángulo relativo ruta–usuario 18 retornar thetaRel 19 Fin Procedimiento
--	---

3.3.6.3 Discretización en cinco regiones y generación de pitidos 2D lateralizados

Una vez obtenido el ángulo relativo ruta–usuario, el módulo discretiza el semiplano frontal en cinco regiones:

- **Izquierda (LEFT):** objetivo claramente hacia la izquierda.
- **Diagonal izquierda (DIAG_LEFT):** objetivo ligeramente desviado hacia la izquierda.

- **Frente (FRONT):** objetivo alineado con el eje frontal del usuario.
- **Diagonal derecha (DIAG_RIGHT):** objetivo ligeramente desviado hacia la derecha.
- **Derecha (RIGHT):** objetivo claramente hacia la derecha.

Los límites angulares se definen de forma empírica, buscando un compromiso entre precisión y facilidad de interpretación para el usuario. Un ejemplo de configuración es:

- **LEFT:** ángulos entre $[-140^\circ, -60^\circ]$.
- **DIAG_LEFT:** ángulos entre $(-60^\circ, -25^\circ]$.
- **FRONT:** ángulos entre $(-25^\circ, 25^\circ)$.
- **DIAG_RIGHT:** ángulos entre $[25^\circ, 60^\circ)$.
- **RIGHT:** ángulos entre $[60^\circ, 140^\circ]$.

Si el ángulo cae fuera del rango $[-140^\circ, 140^\circ]$ se considera que el objetivo se encuentra significativamente por detrás del usuario, por lo que el sistema puede optar por no emitir pitidos lateralizados y recurrir únicamente a instrucciones de voz específicas (por ejemplo, “gire en U” o “dé la vuelta”).

Una vez identificada la región, el módulo asocia a cada una un par de ganancias estéreo (gananciaIzquierda, gananciaDerecha) para reproducir el mismo pitido con distinta intensidad en cada canal de audio:

- **LEFT** $\rightarrow (1.0, 0.0) \rightarrow$ solo canal izquierdo.
- **DIAG_LEFT** $\rightarrow (1.0, 0.4) \rightarrow$ izquierda fuerte, derecha suave.
- **FRONT** $\rightarrow (1.0, 1.0) \rightarrow$ ambos canales equilibrados.
- **DIAG_RIGHT** $\rightarrow (0.4, 1.0) \rightarrow$ derecha fuerte, izquierda suave.
- **RIGHT** $\rightarrow (0.0, 1.0) \rightarrow$ solo canal derecho.

La reproducción del pitido se realiza mediante un gestor de audio (BeepManager) basado en SoundPool, que permite especificar las ganancias de cada canal de manera independiente.

3.3.6.4 Integración de voz y pitidos 2D lateralizados

La función de alto nivel TurnCueManager.cue(...) integra la información de navegación (texto base, etapa de la indicación y ángulo relativo) con la preferencia de usuario almacenada en tipoRetroalimentacion. El flujo básico es el siguiente:

- En modo “Solo voz”, la instrucción se adapta según la región angular (izquierda, diagonal izquierda, frente, diagonal derecha, derecha) y se enuncia mediante TTS, incorporando también la distancia en pasos o metros.

- En modo “Solo pitidos lateralizados”, se determina la región a partir del ángulo relativo y se reproducen secuencias de pitidos 2D con las ganancias estéreo y el número de repeticiones asociados a la etapa de la indicación.
- En modo “Voz + pitidos lateralizados”, se ejecutan ambos procesos: la voz describe la acción y, en paralelo, los pitidos se lateralizan hacia la región correspondiente (por ejemplo, DIAG_RIGHT), reforzando la misma dirección.

Esta integración garantiza que una única estimación angular y una única discretización en cinco regiones alimenten tanto la salida verbal como la salida de pitidos, evitando contradicciones entre los canales y manteniendo la consistencia del mensaje para la persona usuaria.

El funcionamiento integrado del módulo de realimentación se resume en el Pseudocódigo 3-10.

Pseudocódigo 3- 10

01 Pseudocódigo 3.8. Módulo de realimentación (voz y pitidos 2D lateralizados)	21 // 4. Leer modo de feedback seleccionado (voz, pitidos, voz+pitidos)
02 Procedimiento GestionarRealimentacion(textoBase, etapaCue)	22 modo ← LeerPreferencia("tipoRetroalimentacion")
03 // 1. Obtener estado de navegación	23 // 5. Activar salidas según modo
04 pos ← ObtenerUltimaPosicionFiltrada()	24 si modo incluye VOZ entonces
05 heading ← ObtenerHeadingIMU()	25 ReproducirTTS(textoVoz)
06 bearingRuta ← CalcularBearingRuta(pos)	26 fin si
07 si pos, heading o bearingRuta son inválidos **entonces**	27 si modo incluye PITIDOS entonces
08 anguloRelativo ← NULO	28 (gL, gR) ← GananciasStereo(region)
09 si no	29 repeticiones ← RepeticionesPorEtapa(etapaCue)
10 anguloRelativo ← NormalizarAngulo(bearingRuta - heading)	30 intervalo ← IntervaloPorEtapa(etapaCue)
11 fin si	31 para i desde 1 hasta repeticiones **hacer**
12 // 2. Determinar región direccional (izq, diag izq, frente, diag der, der)	32 ReproducirPitidoStereo(gL, gR)
13 si anguloRelativo es válido entonces	33 Esperar(intervalo)
14 region ← ClasificarEnCincoRegiones(anguloRelativo)	34 fin para
15 si no	35 fin si
	36 Fin Procedimiento

<pre> 16 region ← InferirRegionDesdeGiro(textoBase) // aproximación cuando no hay ángulo fiable 17 fin si 18 // 3. Construir texto final de voz consistente con la región 19 textoVoz ← FormatearInstruccion(textoBase, region, etapaCue) 20 MostrarEnPantalla(textoVoz) </pre>	
--	--

En GestionarRealimentacion primero se obtiene el estado actual de la navegación: posición filtrada, orientación del usuario y dirección de la ruta (líneas 3–5), con lo que se calcula el ángulo relativo ruta–usuario o se marca como nulo si faltan datos (6–10). Luego se determina la región direccional (izquierda, frente, derecha, etc.), usando el ángulo cuando es fiable o, en su defecto, una inferencia a partir del texto de la instrucción (12–16).

Con esa región se genera el texto final de voz coherente con la situación y se consulta el modo de realimentación elegido en la configuración (18–21). Finalmente, según ese modo, se reproducen la voz por TTS y/o los pitidos 2D lateralizados con el patrón adecuado de repeticiones e intervalos (23–34), de manera que ambos canales indiquen de forma consistente qué debe hacer el usuario y con qué urgencia.

3.3.7 Integración de módulos y flujo completo de navegación (NavigationActivity)

La NavigationActivity actúa como “orquestador” del aplicativo Guíame, integrando los módulos de configuración, gestión de destinos, adquisición y preprocesamiento de la señal GPS/IMU, procesamiento y filtrado de la trayectoria, y realimentación auditiva. Desde esta pantalla se selecciona el destino, se inicia o detiene la navegación y se actualiza de forma continua tanto el mapa como las indicaciones que recibe el usuario.

La Figura 3.13 muestra la pantalla de navegación del aplicativo Guíame, donde se integran la visualización del mapa, la selección de destino y las indicaciones de navegación.

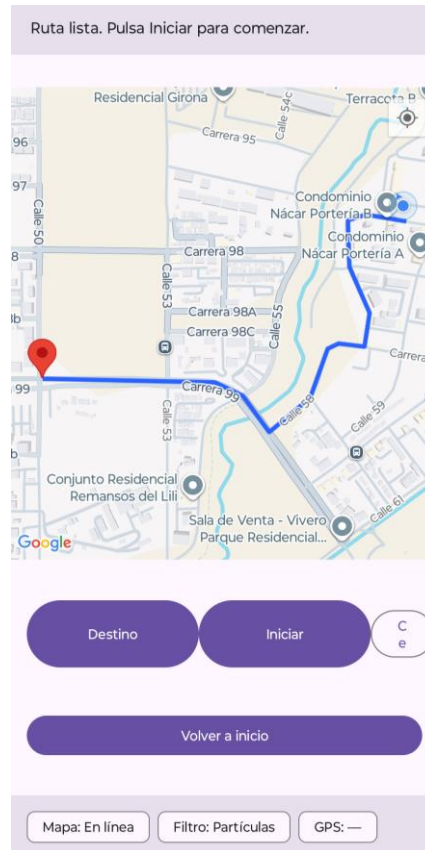


Figura 3.13 Pantalla de navegación en ejecución.

Al abrir la actividad, se leen las preferencias almacenadas (tipo de mapa, idioma, tipo de feedback, uso de pasos e información de altura) y se inicializan los componentes principales: el controlador de mapa (IMapController en modo en línea u offline), el estimador de longitud de paso (StepEstimator), el filtro de partículas (GpsImuParticleFilter, que internamente utiliza ENUConverter para operar en coordenadas locales), el motor turn-by-turn (TurnByTurnEngine) y el módulo de realimentación (TurnCueManager). La interfaz muestra, además, controles para seleccionar un destino y un botón para iniciar/detener la navegación.

La selección de destino puede realizarse de dos formas:

- Recibiendo un destino previamente creado en DestinationActivity
- Utilizando el fragmento DestinationsSheet, que despliega una lista compacta de destinos sobre la propia pantalla de navegación. En ambos casos, NavigationActivity almacena el destino actual como punto objetivo para el cálculo de ruta.

Cuando el usuario pulsa el botón de iniciar navegación, la actividad verifica que exista un destino válido y que los permisos de ubicación estén concedidos. A

continuación, se arranca LocationService en primer plano y se espera a la primera posición fiable del GPS. Una vez obtenida, se calcula la ruta desde la posición filtrada inicial hasta el destino empleando RouteManager (modo en línea) u OfflineRouteManager (modo offline), y se pasa el resultado a TurnByTurnEngine, que construye la secuencia de maniobras.

A partir de ese momento, la navegación se desarrolla como un bucle continuo:

Si el sistema detecta que el usuario se ha desviado de la ruta más allá de un umbral permitido, se dispara un proceso de recálculo: se solicita una nueva ruta al gestor correspondiente (en línea u offline) y se notifica al usuario mediante una combinación de voz y patrones de pitidos específicos para la etapa de recálculo. Cuando el motor turn-by-turn determina que se ha alcanzado el destino, se generan los mensajes finales de llegada, se detiene la navegación y se detiene también LocationService, volviendo la interfaz a un estado de espera para una nueva sesión de navegación.

El flujo integrado de la NavigationActivity, que articula todos los módulos descritos, se resume en el Pseudocódigo 3-11.

Pseudocódigo 3- 11

01 Algoritmo FlujoCompletoNavegacion_NavigationActivity 02 Inicio 03 // 1. Inicialización 04 prefs ← leerSharedPreferences() 05 configurarMapa(usrOnline = prefs.usarMapasOnline) 06 stepLen ← StepEstimator.estimateStepLength(prefs.alturaUsuario) 07 filtro ← nuevo GpsImuParticleFilter() 08 tbtEngine ← nuevo TurnByTurnEngine() 09 cueManager ← TurnCueManager.inicializar(prefs.tipoRetroalimentacion) 10 □ mostrarControlesSelecciónDestinoYBotónNavegación() 11 // 2. Selección de destino 12 al pulsar "Seleccionar destino" hacer 13 abrirDestinationsSheet()	38 // 4. Bucle principal: procesamiento de ubicaciones 39 al recibirUbicación(lat, lng, acc) desde LocationService hacer 40 rawLatLng ← construirLatLng(lat, lng) 41 posFiltrada ← filtro.updateGps(rawLatLng, acc) 42 □ actualizarMarcadorEnMapa(posFiltrada) 43 si estado = ESPERANDO_PRIMERA_POSICION entonces 44 ruta ← obtenerRuta(origen = posFiltrada, destino = destinoActual, prefs) 45 tbtEngine.setRoute(ruta, stepLen) 46 estado ← NAVEGANDO 47 cueManager.cue("Ruta calculada, inicio de navegación", INICIO, NULO)
---	--

14	al elegirDestino(dest) en sheet hacer	48	fin si
15	destinoActual ← dest	49	si estado = NAVEGANDO
16	mostrarDestinoEnUI(destinoActual)	entonces	
17	fin al	50	instruccion ←
18	fin al	tbtEngine.updatePosition(posFiltrada)	
19	// 3. Inicio / parada de la navegación	51	si instruccion ≠ nulo entonces
20	al pulsar botonNavegacion hacer	52	GestionarRealimentacion(instruccion.texto,
21	si estado = DETENIDO entonces	instruccion.etapa)	
22	si destinoActual es nulo entonces	53	fin si
23	mostrarMensaje("Seleccione un destino antes de navegar")	54	si tbtEngine.requiereRecalculo()
24	*sino si no hayPermisoUbicacion() entonces	entonces	
25	solicitarPermisoUbicacion()	55	ruta ← obtenerRuta(origen =
26	sino	posFiltrada, destino = destinoActual, prefs)	
27	estado ←	56	tbtEngine.setRoute(ruta,
ESPERANDO_PRIMERA_POSICION		stepLen)	
28	iniciarServicio(LocationService)	57	cueManager.cue("Recalculando ruta",
29	cambiarTextoBoton("Detener navegación")	RECASC, NULO)	
30	fin si	58	fin si
31	sino si estado = NAVEGANDO o estado =	59	si tbtEngine.hasArrived()
ESPERANDO_PRIMERA_POSICION entonces		entonces	
32	detenerServicio(LocationService)	60	cueManager.cue("Has llegado
33	estado ← DETENIDO	al destino", LLEGADA, NULO)	
34	cambiarTextoBoton("Iniciar navegación")	61	detener
35	cueManager.cue("Navegación detenida", etapa = LLEGADA, anguloRel = NULO)	Servicio(LocationService)	
36	fin si	62	estado ← DETENIDO
37	fin al	63	cambiarTextoBoton("Iniciar navegación")
		64	fin si
		65	fin si
		66	fin al
		67	Fin

La NavigationActivity comienza leyendo las preferencias de usuario y configurando el mapa, la longitud de paso, el filtro de partículas, el motor turn-by-turn y el gestor

de realimentación (líneas 4–10). Después, al pulsar “Seleccionar destino”, se abre el panel de destinos y, cuando el usuario elige uno, se guarda como destinoActual y se muestra en la interfaz (líneas 12–18).

El botón de navegación alterna entre iniciar y detener: si el estado es DETENIDO, se verifica que haya destino y permisos de ubicación; si todo es correcto, se pasa al estado ESPERANDO_PRIMERA_POSICION y se arranca LocationService (líneas 21–30). Si ya se está navegando (o esperando la primera posición), pulsar el botón detiene el servicio, devuelve el estado a DETENIDO y anuncia que la navegación se ha detenido (líneas 31–36).

Cada vez que llega una nueva ubicación desde LocationService, se construye la posición, se filtra y se actualiza el mapa (líneas 39–42). Si aún se esperaba la primera posición, con esa ubicación se calcula la ruta, se inicializa TurnByTurnEngine y se cambia al estado NAVEGANDO, informando al usuario que la ruta ha sido calculada (líneas 43–47). Mientras el estado es NAVEGANDO, cada posición filtrada se pasa al motor turn-by-turn: si se genera una nueva instrucción, se envía al módulo de realimentación (líneas 49–53); si el motor detecta que la ruta debe recalcularse, se obtiene una nueva ruta y se notifica el recálculo (líneas 54–57); y si se detecta la llegada al destino, se anuncia, se detiene el servicio de localización y se restaura el estado a DETENIDO (líneas 59–63).

3.4 CONCLUSIONES Y OBSERVACIONES FINALES

El desarrollo del capítulo 3 permitió materializar la propuesta de guía para personas con discapacidad visual en un aplicativo móvil funcional, organizado en módulos claramente definidos. A partir de los requerimientos funcionales y no funcionales establecidos previamente, se consolidó una arquitectura que integra el uso de mapas en línea y mapas offline, la fusión de datos GPS/IMU y distintos modos de realimentación auditiva. El diseño apoyado en Scrum, junto con los diagramas de conceptos, clases y bloques, facilitó la planificación del software y la transición ordenada desde el análisis hasta la implementación de cada componente.

Cada uno de los módulos implementados cumple un rol específico dentro del sistema. El módulo de configuración e interfaz inicial permite ajustar parámetros clave de navegación —tipo de mapa, idioma, tipo de retroalimentación, volumen y uso de pasos—, garantizando que el aplicativo pueda adaptarse a las necesidades y preferencias de cada usuario. El módulo de gestión de destinos ofrece herramientas para crear, editar y eliminar puntos de interés, así como para seleccionarlos de manera ágil antes de iniciar el recorrido, constituyendo la base de los escenarios de navegación que se evalúan en los capítulos posteriores.

En cuanto al procesamiento de la información de posición, el módulo de adquisición y preprocesamiento organiza el flujo de datos desde los sensores GPS/IMU,

transformándolos a un sistema de coordenadas local y asociándolos con la estimación de longitud de paso del usuario. Sobre esta base, el módulo de procesamiento y filtrado de la navegación, soportado en el filtro de partículas GpsImuParticleFilter y en los gestores de rutas online/offline, genera una trayectoria estabilizada y rutas peatonales adecuadas para el contexto del usuario. Finalmente, el módulo de realimentación auditiva, articulado por TurnCueManager y el motor turn-by-turn, convierte las instrucciones de navegación en mensajes de voz y patrones de pitidos 2D lateralizados, diseñados para facilitar la comprensión de la dirección y proximidad de los giros.

En síntesis, el capítulo 3 demuestra que el aplicativo Guíame alcanza un nivel de integración suficiente para cubrir el flujo completo de navegación: desde la configuración inicial y la selección de destinos hasta la generación de instrucciones auditivas durante el recorrido. La modularidad de la solución, la posibilidad de operar con conectividad variable y la incorporación de técnicas de filtrado sensorial sientan las bases para las pruebas de funcionamiento y análisis de desempeño que se presentan en el capítulo 4, donde se evaluará la capacidad del sistema para corregir el error de navegación y la percepción de satisfacción de los usuarios.

4 CAPÍTULO 4: RESULTADOS Y ANÁLISIS

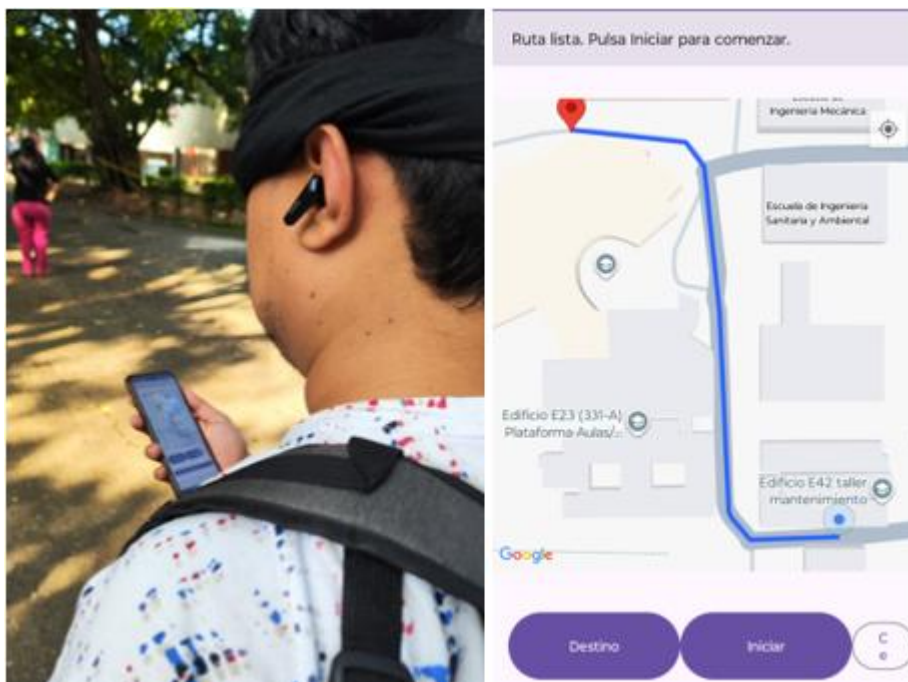


Figura 4.1 Puesta en marcha del aplicativo.

4.1 INTRODUCCIÓN

En este capítulo se describe el procedimiento de evaluación del aplicativo Guíame, con el propósito de verificar el cumplimiento de los requerimientos, así como la aceptabilidad de la solución por parte de los usuarios. Para ello se realizaron pruebas de integración entre los módulos descritos en el capítulo anterior y pruebas de campo con usuarios videntes vendados, que simulan el uso por personas con discapacidad visual y permiten analizar la principal motivación de este trabajo de grado, corregir la señal ruidosa de la señal GPS/GPRS en zonas de sombra muy comunes en nuestros países.

Cada prueba se documenta en tablas donde se registran el requerimiento evaluado, el tipo de prueba, el entorno de ensayo, el procedimiento seguido, el resultado esperado y el obtenido, junto con observaciones relevantes. A partir de esta información se presentan y discuten los resultados, con el fin de valorar el desempeño global del aplicativo Guíame durante el proceso de navegación asistida enfocada en la corrección de la señal GPS/GPRS.

4.2 PRUEBAS DE INTEGRACIÓN

Las pruebas de integración tienen como objetivo verificar el funcionamiento conjunto de los distintos módulos del aplicativo Guíame, asegurando que la interacción entre ellos sea coherente, estable y acorde con los requerimientos funcionales definidos.

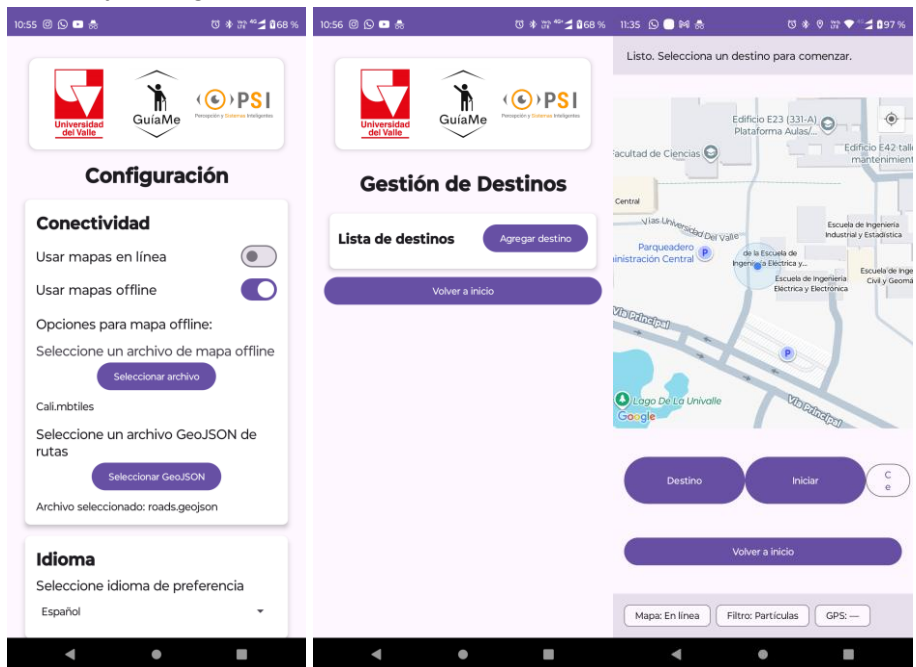
Para ello se realizó una sesión de pruebas centrada en la comunicación entre la interfaz de usuario (módulos de configuración, gestión de destinos y navegación), el servicio de localización y filtrado de posición, el cálculo de ruta (en línea y offline) y el subsistema de realimentación auditiva. Cada prueba se documenta en una tabla estructurada donde se registran el requerimiento evaluado, el tipo de prueba, el hardware y software involucrados (dispositivo Android, servicios de mapas, filtros, etc.), el objetivo de la prueba, los datos necesarios para su ejecución, el procedimiento seguido, el resultado esperado, el resultado obtenido y los comentarios adicionales relevantes para el análisis del comportamiento del sistema.

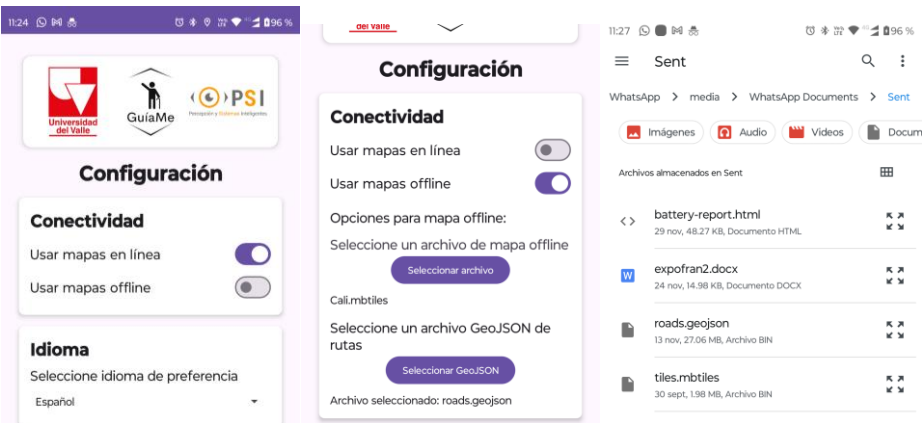
4.2.1 Módulo de configuración

Esta subsección presenta la prueba de integración destinada a validar la navegación entre las distintas pantallas del aplicativo y el funcionamiento de la pantalla de configuración. Se puede observar en la Tabla 4-1 el procedimiento y los resultados obtenidos.

Tabla 4-1 Prueba de integración - Módulo inicio y configuración.

PRUEBA DE INTEGRACIÓN – MÓDULO INICIO Y CONFIGURACIÓN	
• Iniciar la aplicación y mediante los gestos definidos acceder a las pantallas del aplicativo. • Acceder a la pantalla de configuración y modificar los parámetros allí establecidos: Selección de mapa online y selección de mapa offline con carga de los archivos correspondientes, selección de idioma, selección de la retroalimentación del sistema, registro de la altura.	
Tipo de Prueba	Integración
Hardware Requerido	Dispositivo móvil con sistema operativo Android.
Software Requerido	Aplicativo móvil GuiaMe. Android 8.0 o superior.
Objetivo	Probar que el aplicativo inicia correctamente y que se puede acceder mediante gestos a las diferentes pantallas. En la pantalla de configuración verificar el funcionamiento de todas las opciones configurables y asegurar que mantenga los cambios una vez ingresados.
Iniciar la aplicación y mediante los gestos definidos acceder a las pantallas del aplicativo.	
Objetivo	Validar la correcta funcionalidad de los gestos de la pantalla principal
Datos de la Prueba	En esta prueba se necesita que el usuario realice el desplazamiento a las 3 pantallas del aplicativo desde la pantalla de inicio.

Procedimiento	1. Abrir el aplicativo en el dispositivo móvil. 2. Deslizar el dedo hacia la derecha para acceder a la pantalla de configuración, y desde esta pantalla deslizar a la izquierda para regresar a la pantalla de inicio. 3. Deslizar el dedo hacia arriba para acceder a la pantalla de gestión de destinos, y desde esta pantalla deslizar hacia abajo para regresar a la pantalla de inicio. 4. Deslizar el dedo hacia la izquierda para acceder a la pantalla de navegación, y desde esta pantalla deslizar a la derecha para regresar a la pantalla de inicio
Resultado Esperado	1. Se acceden fácilmente a las pantallas mediante los gestos. 2. Se regresa a la pantalla de inicio mediante el gesto contrario previamente usado.
Resultado Obtenido	Prueba satisfactoria. A continuación, se presentan las imágenes correspondientes a los resultados. a. Los gestos y sus contrapartes son detectados con éxito por el aplicativo para navegar por el mismo. b. Se accede a las pantallas de Configuración, gestión de destinos y navegación respectivamente: 
Acceder a la pantalla de configuración y modificar los parámetros allí establecidos.	
Objetivo	Validar la correcta funcionalidad de la pantalla de configuración para seleccionar el tipo de mapa preferido: online y offline, seleccionar entre los idiomas disponibles, seleccionar la retroalimentación del sistema y registrar la altura.

	Verificar que los cambios ingresados se guardan en el aplicativo.
Datos de la Prueba	En esta prueba se necesita que el usuario acceda a la pantalla de configuración, y modifique las opciones a su preferencia.
Procedimiento	1. Abrir el aplicativo en el dispositivo móvil y acceder a la pantalla de configuración. 2. Seleccionar mapa online verificando que mapa offline no se active al tiempo. 3. Seleccionar mapa offline verificando que mapa online no se active al tiempo. Realizar la carga de los archivos correspondientes al mapa offline y la rutas. 4. Seleccionar entre los idiomas disponibles. 5. Seleccionar entre los tipos de realimentación disponibles. 6. Registrar la altura del usuario. 7. Cerrar el aplicativo e ingresar nuevamente verificando que los cambios hechos previamente están guardados y aplicados.
Resultado Esperado	1. Se accede a la pantalla de configuración. 2. Se registran las configuraciones preferidas. 3. La configuración se guarda con éxito y el aplicativo las mantiene después de cerrado.
Resultado Obtenido	Prueba satisfactoria. A continuación, se presentan las imágenes correspondientes a los resultados. a. Se accede a la pantalla de configuración y se modifican las opciones allí establecidas. b. Configuración de los mapas online y offline. En mapa offline acepta la carga de archivos MBtiles para el mapa y Geojson para las rutas.  c. Permite la selección de idioma y retroalimentación del sistema.

	Language Select preferred language Spanish English French Retroalimentación del sistema Tipo de retroalimentación Voz Pitidos lateralizados Voz + Pitidos lateralizados d. Configuración de la altura. Altura (cm) 175 e. Al cerrar y abrir el aplicativo los cambios se mantienen.
--	---

En la tabla 4-1 se registran las capturas de pantallas respectivas al procedimiento realizado de acceso a las pantallas del aplicativo y del manejo de la configuración del mismo. En las capturas se evidencian la modificación de las configuraciones a las preferencias del usuario, que se deberán realizar por el acompañante, y estas quedarán disponibles para el usuario con discapacidad visual en cualquier momento facilitando el uso futuro del aplicativo.

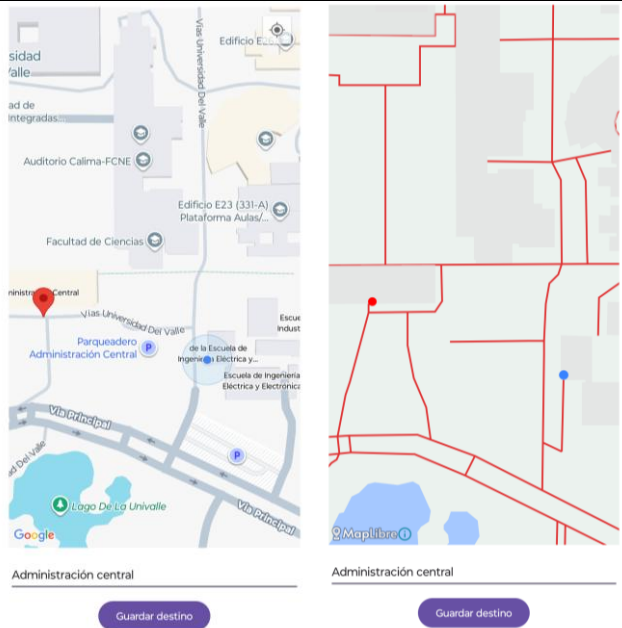
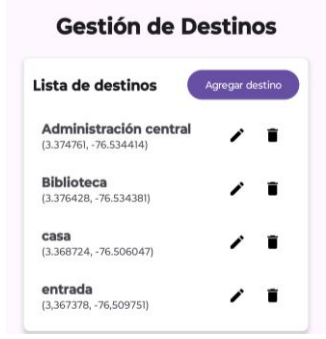
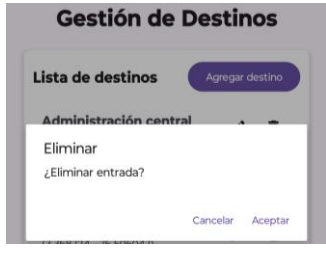
4.2.2 Módulo de gestión de destinos

Esta subsección presenta la prueba de integración destinada a validar la gestión de destinos del aplicativo. Se puede observar en la Tabla 4-2 el procedimiento y los resultados obtenidos.

Tabla 4-2 Prueba de integración - Módulo gestión de destinos.

PRUEBA DE INTEGRACIÓN – MÓDULO GESTIÓN DE DESTINOS	
	- Acceder a la pantalla de gestión de destinos del aplicativo. - Agregar, editar y eliminar destinos con mapa online. - Agregar, editar y eliminar destinos con mapa offline.
Tipo de Prueba	Integración
Hardware Requerido	Dispositivo móvil con sistema operativo Android.
Software Requerido	Aplicativo móvil GuiaMe. Android 8.0 o superior.
Objetivo	Validar la correcta funcionalidad de la pantalla de gestión de destinos en las modalidades de mapa online y offline, para agregar, editar y eliminar

	los destinos. Verificar que los cambios queden guardados en el aplicativo.
Adicionar, editar y eliminar destinos.	
Datos de la Prueba	Crear, editar y eliminar destinos, asignando un nombre y guardando las coordenadas mediante la asignación de un pin en el mapa.
Procedimiento	1. Abrir el aplicativo en el dispositivo móvil configurar el mapa online en la pantalla de configuraciones y posteriormente acceder a la pantalla de gestión de destinos. 2. Presionar el botón de agregar destinos y seleccionar un pin en el mapa online que se despliega, asignarle un nombre y guardarlo. 3. Agregar 2 o más destinos. 4. Editar al menos 1 de los destinos cambiando el nombre y la ubicación del pin. 5. Eliminar al menos 1 de los destinos. 6. Cerrar y abrir el aplicativo. 7. Verificar la existencia de la lista creada de los destinos. 8. Repetir los pasos cambiando al mapa offline en la pantalla de configuración, verificando que los destinos ya agregados en el mapa online también se puedan gestionar a partir de allí.
Resultado Esperado	1. Se accede a la pantalla de gestión de destinos. 2. Al presionar agregar destino se despliega el mapa y permite agregar un destino seleccionando un pin y nombrarlo. 3. Al presionar editar se abre el mapa mostrando el nombre y el pin del destino guardado para editarlo. 4. Al presionar borrar permite borrar el destino deseado. 5. Los destinos agregados se guardan con éxito. 6. La lista de destinos puede manejarse en ambas modalidades de los mapas online y offline.
Resultado Obtenido	Prueba satisfactoria. A continuación, se presentan las imágenes correspondientes a los resultados. a. El mapa online se despliega después de presionar el botón de agregar destino y permite guardar el destino deseado. El mapa offline funciona de igual forma.

	 b. Se crea la lista de destinos agregando los que se requieran.  c. Se editan y eliminan los destinos que se deseen.  d. Al cambiar a mapa offline, la pantalla sigue funcionando y los cambios quedan registrados, como se muestra en el inciso a.
--	---

En la tabla 4-2 se registran las capturas de pantallas respectivas al procedimiento realizado para gestionar los destinos a los que el usuario con discapacidad visual desee ir. En las capturas se evidencian la adición, edición y eliminación de destinos funcional en las dos modalidades de mapas. Esta gestión se deberá realizar por el acompañante, y estos destinos quedaran disponibles para el usuario con discapacidad visual en cualquier momento facilitando el uso futuro del aplicativo.

4.2.3 Módulos de adquisición y preprocesamiento de la señal GPS/IMU y de procesamiento y filtrado de la navegación.

Esta subsección presenta la prueba de integración destinada a validar la adquisición de y preprocesamiento de la señal por parte del aplicativo y de la generación de rutas. Se puede observar en la Tabla 4-3 el procedimiento y los resultados obtenidos.

Tabla 4-3 Prueba de integración - Módulo de adquisición y preprocesamiento de la señal gps/imu y de procesamiento y filtrado de la navegación.

PRUEBA DE INTEGRACIÓN – MÓDULO DE ADQUISICIÓN Y PREPROCESAMIENTO DE LA SEÑAL GPS/IMU Y DE PROCESAMIENTO Y FILTRADO DE LA NAVEGACIÓN	
	• Verificar la posición en tiempo real del GPS en el mapa. • Verificar el cálculo y creación de rutas en el aplicativo. • Verificar la activación del filtro de partículas. • Verificar funcionamiento en mapas online y offline.
Tipo de Prueba	Integración
Hardware Requerido	Dispositivo móvil con sistema operativo Android.
Software Requerido	Aplicativo móvil GuiaMe. Android 8.0 o superior.
Objetivo	Verificar el acceso a la ubicación del dispositivo por parte del aplicativo, la creación de rutas y la activación del filtro de partículas.
Acceso a la ubicación, creación de rutas y filtrado del GPS/GPRS.	
Datos de la Prueba	En esta prueba el aplicativo accede de forma permanente a los servicios de ubicación del dispositivo y genera las rutas hacia el destino deseado.
Procedimiento	1. Iniciar el aplicativo por primera vez y dar permisos de ubicación. 2. Acceder a la pantalla de navegación del aplicativo. 3. Verificar la posición en tiempo real del GPS en el mapa. 4. Seleccionar un destino y verificar el cálculo y creación de rutas en el aplicativo. 5. Iniciar la navegación. 6. Verificar la activación del filtro de partículas. 7. Repetir el procedimiento con el mapa offline.
Resultado Esperado	1. Se piden los permisos de ubicación y se otorgan. 2. Se accede a la pantalla de navegación. 3. Se selecciona un destino y el aplicativo genera la ruta. 4. Se inicia la navegación y el filtro comienza su funcionamiento.

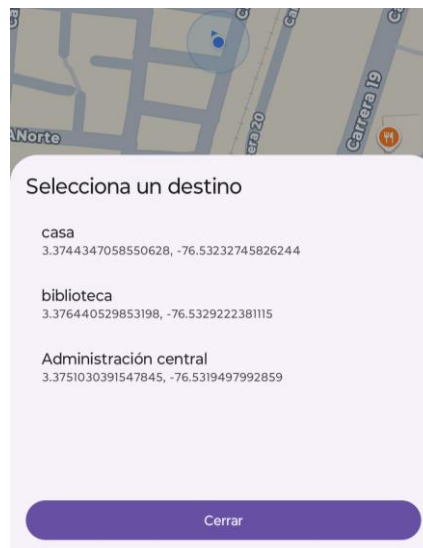
Resultado Obtenido

Prueba satisfactoria. A continuación, se presentan las imágenes correspondientes a los resultados.

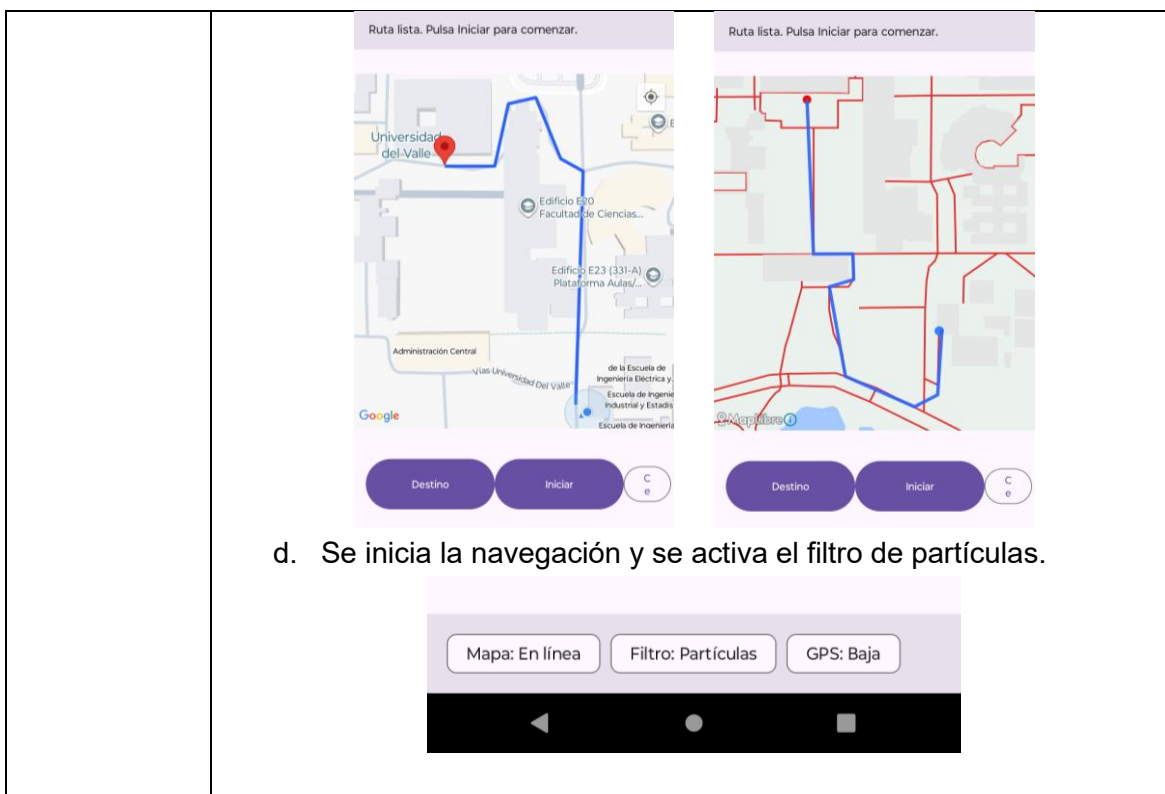
- a. El aplicativo solicita los permisos de ubicación.



- b. En la pantalla de navegación se observa la ubicación del dispositivo, posteriormente se escoge un destino al que se quiera ir.



- c. Luego de la selección, se genera la ruta a partir de la ubicación del dispositivo. Funciona de igual forma en el mapa offline.



En la tabla 4-3 se registran las capturas de pantallas respectivas al procedimiento realizado para que el aplicativo pueda acceder a la ubicación del dispositivo, y para que el usuario con discapacidad visual seleccione un destino guardado al que desee ir y generar la ruta de desplazamiento. Este procedimiento es compatible con la accesibilidad del dispositivo para que el usuario con discapacidad visual pueda realizarlo de forma independiente.

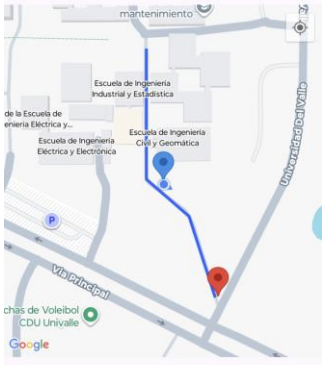
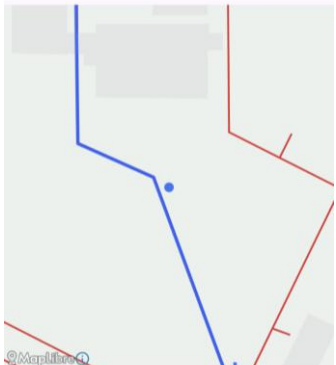
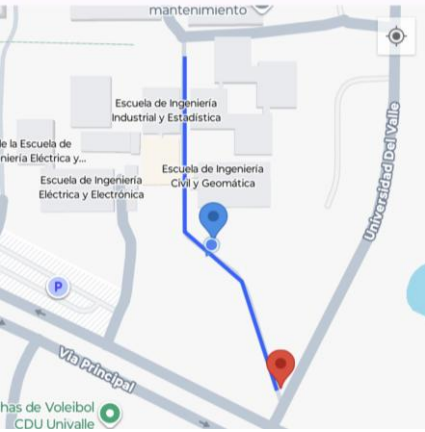
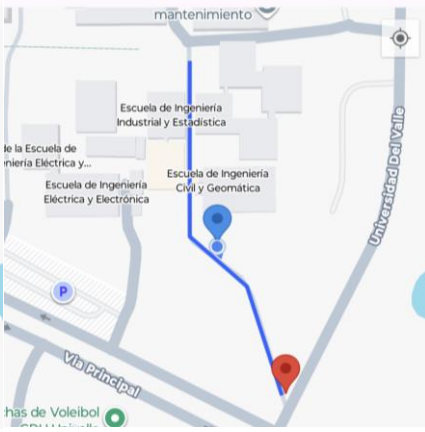
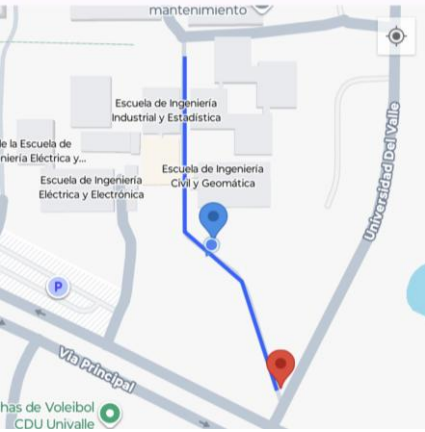
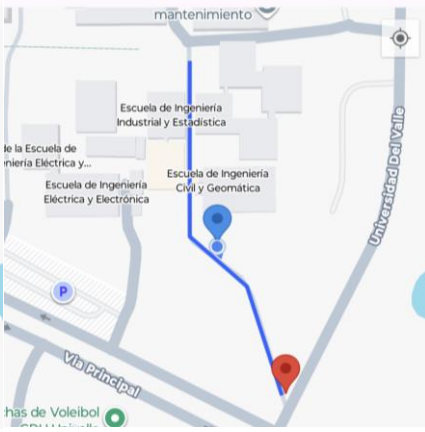
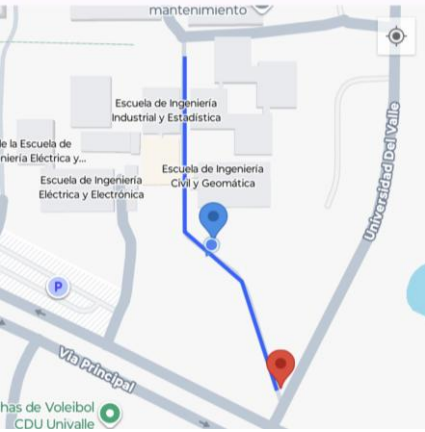
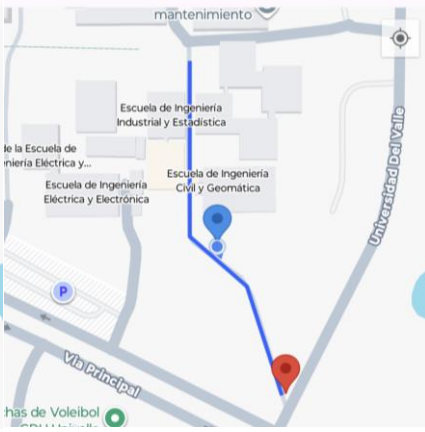
4.2.4 Módulos de navegación y de realimentación auditiva

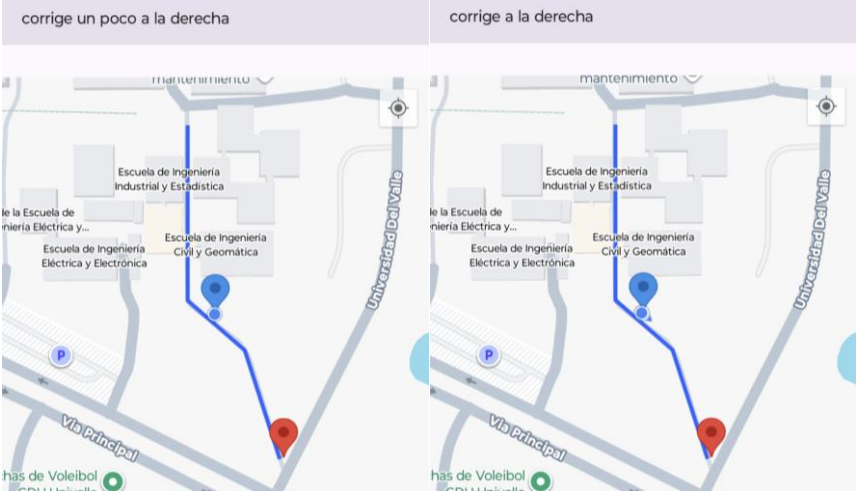
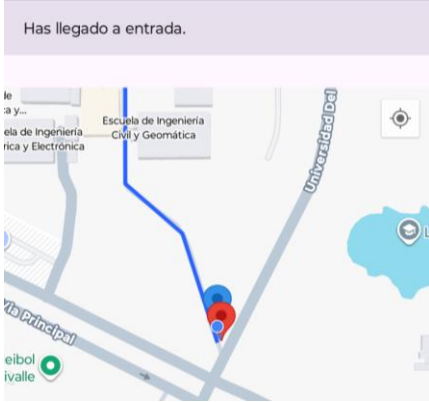
Esta subsección presenta la prueba de integración destinada a la navegación y la realimentación por parte del aplicativo. Se puede observar en la Tabla 4-4 el procedimiento y los resultados obtenidos.

Tabla 4-4 Prueba de integración – Módulo de navegación y realimentación.

PRUEBA DE INTEGRACIÓN – MÓDULO DE NAVEGACIÓN Y DE REALIMENTACIÓN
• Verificar la navegación hacia un destino deseado. • Verificar las diferentes modalidades de realimentación. • Verificar la generación de instrucciones según la discretización en 5 regiones. • Verificar funcionamiento en mapas online y offline.

Tipo de Prueba	Integración
Hardware Requerido	Dispositivo móvil con sistema operativo Android.
Software Requerido	Aplicativo móvil GuiaMe. Android 8.0 o superior.
Objetivo	Probar que el aplicativo navega efectivamente al destino seleccionado dando las indicaciones configuradas durante el trayecto. Al llegar al destino, verificar el aviso de llegada y la opción de volver a iniciar otra navegación.
Realizar la navegación a un destino y verificar la realimentación.	
Datos de la Prueba	En esta prueba el aplicativo realiza la navegación hacia el destino seleccionado, respetando las configuraciones establecidas: tipo de mapa, idioma y realimentación,
Procedimiento	1. Seleccionar en la pantalla de configuración una modalidad de realimentación. 2. Acceder a la pantalla de navegación del aplicativo. 3. Seleccionar un destino e iniciar la navegación. 4. Realizar la ruta descrita por el aplicativo siguiendo las indicaciones generadas. 5. Verificar la generación de instrucciones según la discretización en 5 regiones girando y desviándose de la ruta. 6. Repetir el procedimiento con los otros dos tipos de realimentación. 7. Repetir el procedimiento con el mapa offline.
Resultado Esperado	1. Se inicia la navegación 2. Se generan instrucciones de navegación de acuerdo con la ruta. 3. El aplicativo corrige desvíos e integra la realimentación de forma efectiva. 4. Al llegar al destino se notifica y reinicia para poder hacer otra navegación. 5. La navegación puede manejarse en ambas modalidades de los mapas online y offline.
Resultado Obtenido	Prueba satisfactoria. A continuación, se presentan las imágenes correspondientes a los resultados. a. El aplicativo inicia la navegación con las indicaciones generadas y usando la realimentación escogida.

	Sigue recto, En 41 pasos, gira a la derecha.  Destino Detener C e	corrige a la derecha  Destino Detener C e		
	b. La discretización en 5 sentidos funciona correctamente y corrige la orientación de la navegación según el grado de desvío de la ruta. <table border="1"> <tr> <td data-bbox="560 856 982 1369"> corrige a la izquierda  </td> <td data-bbox="982 856 1404 1369"> corrige un poco a la izquierda  </td> </tr> </table>		corrige a la izquierda 	corrige un poco a la izquierda 
corrige a la izquierda 	corrige un poco a la izquierda 			

	 c. La realimentación por voz se basa en la lectura TTS de las indicaciones, y los pitidos lateralizados coinciden efectivamente con estos. d. Al llegar al destino, se genera el aviso de la llegada cuya notificación se hacen por voz y pitidos.  e. El funcionamiento es el mismo con el mapa offline (siguiendo la propia ruta del mismo) como se ve en el inciso a.
--	--

En la tabla 4-4 se registran las capturas de pantallas respectivas a la navegación ofrecida por el aplicativo. En las capturas se evidencian la generación de instrucciones por parte del aplicativo siguiendo la lógica de la discretización en 5 sentidos, permitiendo seguir la ruta generada. Finalmente, al llegar al destino el aplicativo genera el aviso, permitiendo volver a seleccionar un nuevo destino si así se requiere.

4.3 PRUEBAS DE CAMPO

Las pruebas de campo se llevaron a cabo con diez miembros activos de la Universidad del Valle. Estas se realizaron con personas videntes, ya que el objetivo principal era determinar si el sistema filtra adecuadamente la información de

posición y con esto evitar saltos en la señal de GPS/GPRS. En la Figura 4.2 se muestra la evidencia de algunos participantes probando el aplicativo con los ojos vendados mientras recorren una ruta propuesta.



Figura 4.2 Pruebas de campo del aplicativo GuíaMe realizadas en la Universidad del Valle.

La prueba de campo consistió, en primera instancia, en familiarizar a los estudiantes con el aplicativo: su interfaz, la configuración, la gestión de destinos y las posibles señales e indicadores sonoros (tipos de pitidos) que podrían recibir durante la navegación. En una segunda fase, cada participante recorrió la ruta propuesta con los ojos vendados, teniendo como única guía las instrucciones proporcionadas por el aplicativo. Adicionalmente por cada trayecto se registraron los datos de navegación tales como el GPS crudo y filtrado, y la ruta calculada.

La Figura 4.3 muestra las rutas propuestas para la prueba de campo, para el mapa online el camino de ida y el camino de regreso para mapa offline.

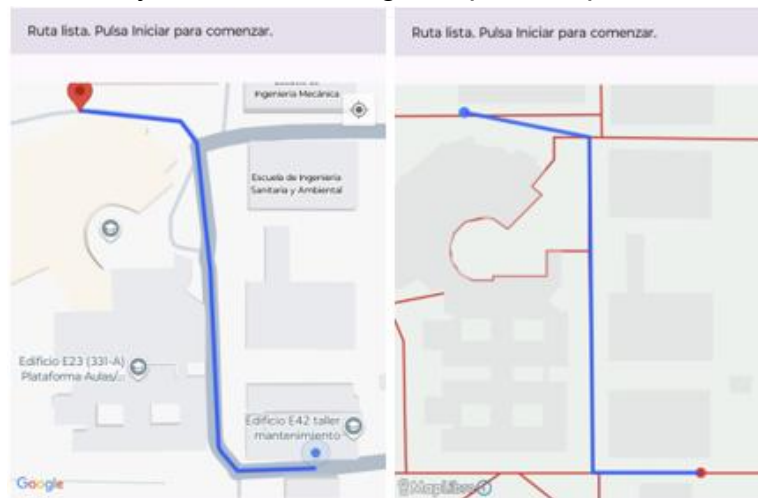


Figura 4.3 Ruta definida para la prueba de campo del mapa online (izquierda) y offline (derecha).

4.3.1 Resultados cuantitativos

En esta sección se analiza el comportamiento del módulo de filtrado de la señal de posición separando las trayectorias en dos grupos: ida, correspondientes a las pruebas realizadas con mapa online (Google Maps), y vuelta, asociadas a las pruebas con mapa offline (MBTiles). Para cada muestra se dispone de la posición GPS cruda y de la posición filtrada.

La Tabla 4-7 resume el error entre la posición GPS cruda y la posición estimada por el filtro, agrupando las muestras de ida y de vuelta. En ambos casos el RMSE se sitúa alrededor de 1.3 m y la media del error es cercana a 1.2 m, con percentiles 95 en torno a 2.2–2.3 m y valores máximos inferiores a 4.3 m.

Tabla 4-5 Resumen error crudo vs filtrado.

Trayectoria	No. muestras	RMSE (m)	Media (m)	p50 (m)	p95 (m)	Máx. (m)
Ida	1270	1.38	1.21	1.18	2.31	4.03
Vuelta	1628	1.29	1.14	1.08	2.20	4.20

La Figura 4.4 muestra la distribución de este error para las trayectorias de ida y de vuelta. Se observa que la mayoría de las muestras presentan diferencias inferiores a 2 m entre la posición cruda y la filtrada, y que las distribuciones de ida y vuelta prácticamente se solapan. Esto indica que el filtrado suaviza la trayectoria sin introducir modificaciones bruscas en la posición estimada y que su comportamiento es consistente tanto en la configuración con mapa online como en la configuración con mapa offline.

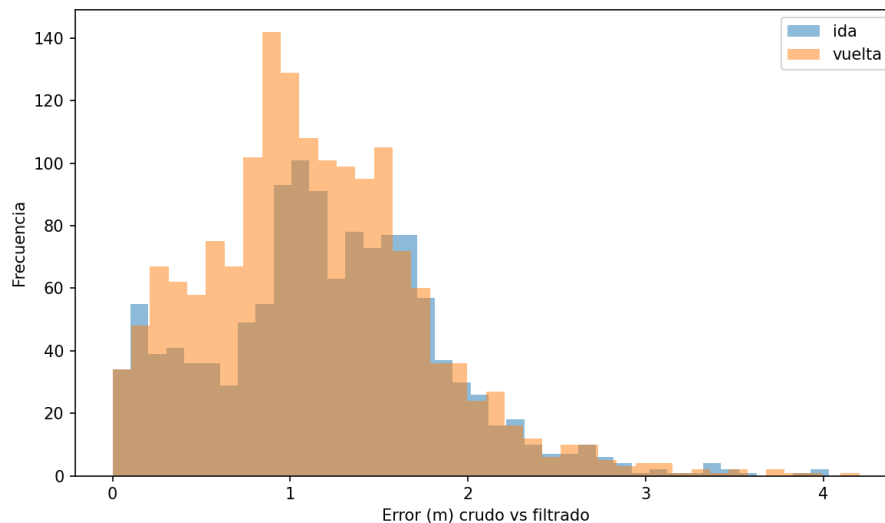


Figura 4.4 Histograma del error entre la posición GPS cruda y la posición estimada por el filtro.

Para complementar el análisis del error crudo–filtrado mostrado en la Figura 4.4, se realizó una evaluación punto a punto del desempeño del GNSS a lo largo de toda la trayectoria. La ruta peatonal se discretizó en puntos de referencia (waypoints) igualmente espaciados cada 2,5 m, coherente con la parametrización utilizada en el aplicativo, y para cada waypoint se agruparon todas las muestras de todas las pruebas cuya distancia acumulada a lo largo de la ruta se encontraba en un entorno de $\pm 1,25$ m alrededor de dicho punto. Con este conjunto de datos se calculó el RMSE, la media y la desviación estándar del error entre la posición GPS cruda y la posición filtrada. La Figura 4.5 presenta la curva del RMSE crudo–filtrado en función de la distancia recorrida, para las trayectorias de ida (online) y de vuelta (offline).

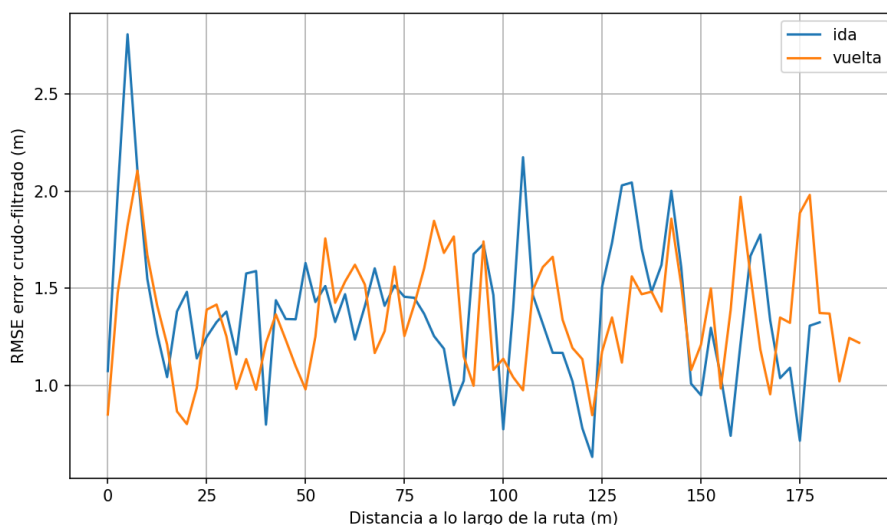


Figura 4.5 RMSE del error crudo–filtrado en función de la distancia a lo largo de la ruta para trayectorias de ida y de vuelta.

En términos globales, la mayor parte del recorrido presenta valores de RMSE entre aproximadamente 1,0 y 1,5 m tanto en ida como en vuelta, lo cual concuerda con los resultados agregados de la Tabla 4-7. El RMSE promedio por waypoint se sitúa en torno a 1,4 m para la ida y 1,35 m para la vuelta, con mínimos cercanos a 0,6–0,8 m, lo que indica que, en la mayoría de los segmentos, el filtro introduce correcciones moderadas sobre la señal GNSS sin desplazamientos excesivos de la posición.

No obstante, la curva permite identificar varios tramos donde el RMSE aumenta de manera localizada. En las trayectorias de ida se observa un pico pronunciado al inicio del recorrido, donde el RMSE se acerca a 2,8 m, así como incrementos adicionales en sectores intermedios de la ruta donde el error supera los 2 m. En las trayectorias de vuelta también se aprecia un aumento del error en el entorno del inicio y, especialmente, en el tramo final, donde se concentran picos con RMSE por

encima de 1,8–2,0 m. Estos máximos suelen corresponder a segmentos donde la señal GNSS es más inestable, obligando al filtro a aplicar correcciones mayores.

El patrón general de la curva confirma la presencia de zonas del recorrido en las que el GNSS presenta mayor variabilidad, posiblemente asociadas a efectos de multipath o sombra satelital por edificaciones y vegetación cercanas, mientras que en la mayor parte de la ruta las correcciones del filtro se mantienen dentro de rangos moderados.

En conjunto, la caracterización punto a punto del RMSE, junto con el análisis global de la Tabla 4-7 y el histograma del error crudo–filtrado, permite concluir que el filtrado de la señal de posición introduce correcciones moderadas (del orden de 1–2 m) y se comporta de manera estable a lo largo de la mayor parte de la ruta, por lo que no constituye la causa principal de las discrepancias observadas en el seguimiento del recorrido. Las diferencias entre las trayectorias de ida y de vuelta se explican principalmente por la forma en que se calcula la ruta en cada configuración de mapa, especialmente en el modo offline, de modo que los esfuerzos de mejora deberían centrarse en la generación de rutas peatonales más precisas y en el ajuste de la configuración del entorno de navegación en dicho modo.

4.3.2 Resultados de satisfacción

La evaluación del aplicativo móvil se realizó mediante una encuesta dirigida a los estudiantes que participaron en las pruebas. El instrumento de evaluación utilizado en esta sección fue diseñado específicamente para este trabajo de grado y estuvo conformado por 18 preguntas formuladas de manera ad hoc a partir de los requerimientos funcionales del aplicativo, de los criterios evaluados en las pruebas de integración y de campo, y de las sugerencias realizadas por el director del trabajo de grado. En consecuencia, la encuesta se empleó como una herramienta de valoración exploratoria de la percepción de uso, la claridad de las indicaciones, la utilidad de los módulos principales y el funcionamiento general del sistema, más que como una escala psicométrica estandarizada. Las respuestas se registraron mediante una escala tipo Likert, cuyas opciones iban desde “totalmente de acuerdo” hasta “totalmente en desacuerdo”. En la Tabla 4-6 se muestran las preguntas aplicadas.

Tabla 4-6 Preguntas de la encuesta.

#	Preguntas
1	Cuando me desvíe de la ruta, ¿el aplicativo detectó el desvío oportunamente?
2	Las indicaciones del aplicativo me ayudaron a volver con facilidad al camino correcto.

3	¿Sentí que el sistema evitó que me alejara demasiado de la ruta planificada?
4	¿La frecuencia de las indicaciones de corrección fue adecuada?
5	Durante la navegación, ¿sentí que las indicaciones se actualizaban de forma oportuna?
6	¿Las indicaciones de voz fueron claras y fáciles de entender?
7	¿Los pitidos lateralizados me ayudaron a identificar desde qué lado debía corregir mi trayectoria?
8	¿La combinación de voz y pitidos lateralizados fue clara?
9	Pude llegar al destino indicado utilizando únicamente las indicaciones del aplicativo.
10	¿Los gestos de navegación del aplicativo se detectaron de forma confiable?
11	¿Los cambios aplicados en la pantalla de configuración se mantuvieron de forma constante?
12	¿La adición de destinos funcionó de forma adecuada?
13	¿La edición de destinos funcionó de forma adecuada?
14	¿La eliminación de destinos funcionó de forma adecuada?
15	¿Los destinos adicionados se pudieron seleccionar en la pantalla de navegación?
16	¿El mapa online funcionó de forma adecuada?
17	¿El mapa offline funcionó de forma adecuada?
18	¿El mapa online funcionó mejor que el mapa offline?

La encuesta fue aplicada a los estudiantes una vez finalizada la actividad de pruebas de campo. Los resultados obtenidos se presentan en los diagramas circulares mostrados en la Figura 4.6, donde cada gráfico corresponde a una de las preguntas planteadas.

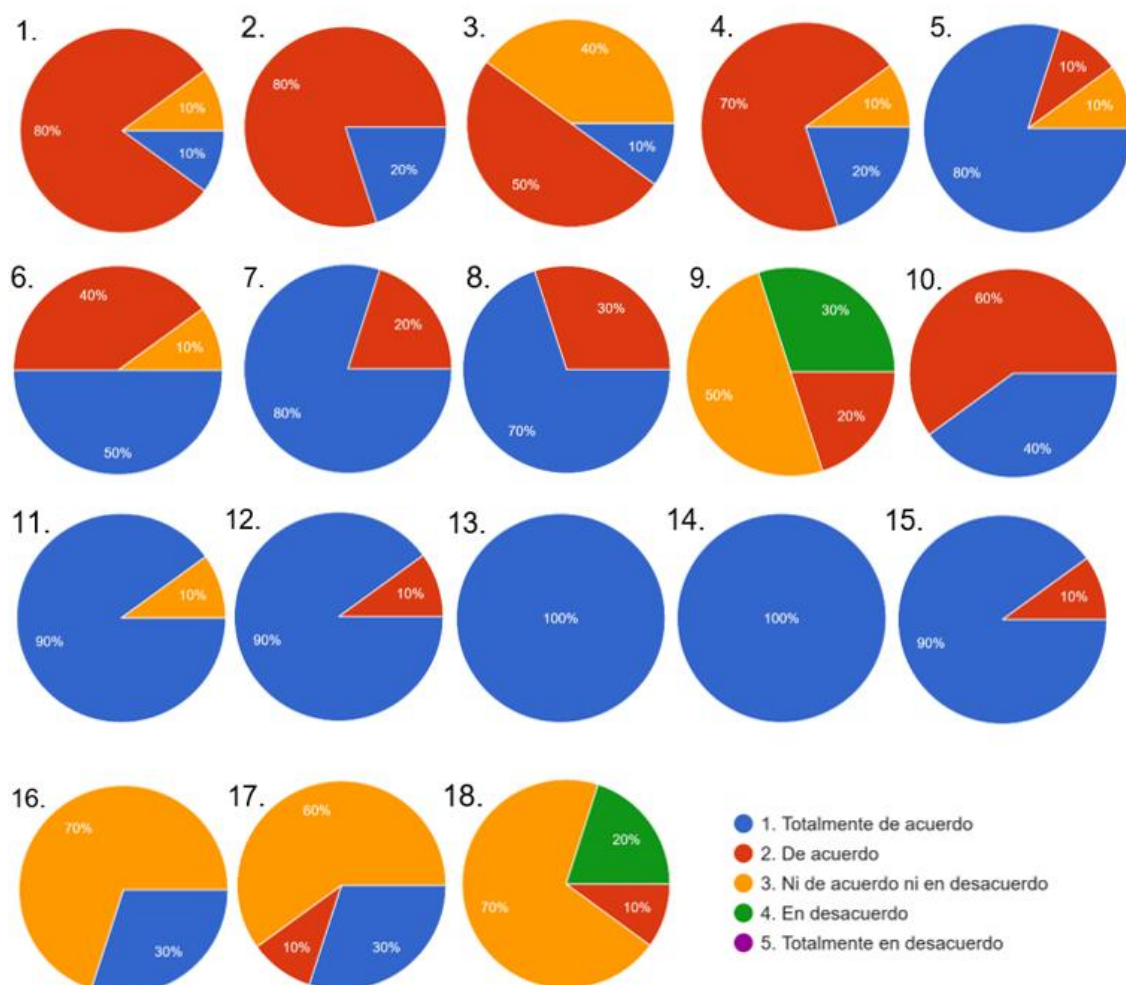


Figura 4.6 Resumen de resultados de la encuesta de evaluación del aplicativo GuiaMe.

Con base en las respuestas proporcionadas por los estudiantes en la encuesta, la Tabla 4-6 presenta un resumen de los resultados obtenidos en la evaluación del aplicativo. En la columna “Conclusión” se incluye un análisis derivado de dichas respuestas.

Tabla 4-7 Resumen de la encuesta sobre las pruebas de campo.

#	a	b	c	d	e	Conclusión
1	10%	80%	10%	0%	0%	La mayoría de los estudiantes consideró que el aplicativo detectó oportunamente los desvíos, lo que refleja una buena usabilidad en la práctica.
2	20%	80%	0%	0%	0%	La mayoría de los estudiantes consideró que las indicaciones del aplicativo ayudaron a regresar al camino correcto, lo que refleja una buena usabilidad en la práctica.

3	10%	50%	40%	0%	0%	La mayoría de los estudiantes percibió que el sistema evitó que se alejaran demasiado de la ruta, aunque una proporción importante se mantuvo neutral, lo que indica un buen desempeño general, pero con margen para reforzar la sensación de seguridad en la trayectoria.
4	20%	70%	10%	0%	0%	Casi todos los estudiantes consideraron adecuada la frecuencia de las indicaciones de corrección, lo cual sugiere que el ritmo de los avisos es apropiado para apoyar la navegación.
5	80%	10%	10%	0%	0%	La gran mayoría indicó que las indicaciones se actualizaban de forma oportuna durante la navegación, lo que muestra que el sistema responde con suficiente rapidez a los cambios de posición.
6	50%	40%	10%	0%	0%	La mayoría percibió las indicaciones de voz como claras y fáciles de entender, por lo que el canal de voz puede considerarse adecuado para apoyar la navegación.
7	80%	20%	0%	0%	0%	Todos los participantes señalaron que los pitidos lateralizados les ayudaron a identificar desde qué lado corregir su trayectoria, lo que confirma que esta estrategia sonora cumple su propósito orientador.
8	70%	30%	0%	0%	0%	La combinación de voz y pitidos fue comprendida claramente por la totalidad de los estudiantes, indicando que el uso conjunto de ambos tipos de realimentación es coherente y fácil de interpretar.
9	0%	20%	50%	30%	0%	Solo una parte de los participantes afirmó haber llegado al destino únicamente con las indicaciones del aplicativo, lo que evidencia que el sistema es útil, pero aún se puede mejorar para lograr una autonomía completa en la navegación.
10	40%	60%	0%	0%	0%	Todos los estudiantes consideraron que los gestos de navegación se detectaron de forma confiable, lo que refuerza la validez de los gestos como mecanismo de interacción accesible.
11	90%	0%	10%	0%	0%	La mayoría percibió que los cambios realizados en la pantalla de configuración se mantuvieron de

						forma constante durante la prueba, lo cual indica una gestión estable y confiable de las preferencias del usuario.
12	90%	10%	0%	0%	0%	Prácticamente todos los estudiantes indicaron que la adición de destinos funcionó de manera adecuada, lo que demuestra que este módulo cumple correctamente su función principal.
13	100%	0%	0%	0%	0%	La edición de destinos fue valorada como plenamente funcional por la totalidad de los participantes, lo que señala que las modificaciones sobre los destinos son sencillas y efectivas.
14	100%	0%	0%	0%	0%	De forma similar, la eliminación de destinos fue percibida como correcta por todos los estudiantes, confirmando que la gestión completa del ciclo de vida de los destinos es consistente.
15	90%	10%	0%	0%	0%	En general, los estudiantes no tuvieron dificultades para seleccionar en navegación los destinos previamente adicionados, lo que indica que la integración entre gestión de destinos y pantalla de navegación es adecuada.
16	30%	0%	70%	0%	0%	Predominaron respuestas neutras frente al funcionamiento del mapa online, con un porcentaje menor de opiniones positivas y sin respuestas claramente negativas, por lo que se considera aceptable para el objetivo de la prueba, aunque sin destacar frente al modo offline.
17	30%	10%	60%	0%	0%	El mapa offline fue valorado como adecuado por una parte de los estudiantes, mientras que la mayoría se mantuvo neutral, lo que sugiere que cumple su función básica, pero requiere más uso o mejoras para generar una percepción más claramente positiva.
18	0%	10%	70%	20%	0%	No hubo consenso claro sobre si el mapa online funciona mejor que el offline, ya que la mayoría se mantuvo neutral y una fracción no percibió ventajas del mapa online, lo que indica que, en las condiciones de prueba, ninguno de los dos modos se percibió como claramente superior y

						ambos resultaron suficientes para el recorrido propuesto.
--	--	--	--	--	--	---

4.4 DISCUSIÓN DE RESULTADOS FRENTE A LOS ANTECEDENTES

Los resultados obtenidos en las pruebas de campo permiten establecer una comparación directa entre la solución desarrollada en este trabajo y los antecedentes revisados en el Capítulo 2. En términos generales, GuiaMe muestra fortalezas importantes como aplicativo de navegación accesible para exteriores, especialmente por concentrar el guiado paso a paso en un teléfono inteligente y ofrecer retroalimentación auditiva como mecanismo principal de orientación. En este sentido, la propuesta desarrollada favorece la autonomía del usuario al no depender de hardware externo especializado ni de dispositivos complementarios para su operación básica, lo que constituye una ventaja frente a algunas soluciones reportadas en la literatura que requieren sensores adicionales o configuraciones más complejas.

Desde el punto de vista del desempeño de la localización, los resultados cuantitativos muestran que el filtrado implementado contribuye a suavizar la trayectoria y a reducir los sobresaltos de la señal GNSS, manteniendo en gran parte del recorrido correcciones moderadas entre la posición cruda y la posición estimada. Este comportamiento es coherente con la motivación planteada en el trabajo, donde se propone el uso de técnicas de filtrado e integración con sensores inerciales para mejorar la estabilidad de la navegación en escenarios con degradación de señal, como cañones urbanos o zonas con baja visibilidad satelital. Por tanto, frente a los antecedentes revisados, GuiaMe gana en la capacidad de ofrecer una solución funcional de navegación accesible con procesamiento local y seguimiento continuo de trayectoria sobre una plataforma móvil de uso común.

No obstante, la comparación también evidencia limitaciones. Aunque el filtrado mejora la continuidad de la trayectoria, los resultados obtenidos muestran que el desempeño final del sistema no depende únicamente del filtro, sino también de la calidad de la ruta peatonal generada y del comportamiento del modo de navegación empleado. Esto se hace más visible en el modo offline, donde todavía se identifican oportunidades de mejora para lograr trayectorias más consistentes y una planificación de ruta más precisa. En consecuencia, frente a algunos antecedentes que integran fuentes adicionales de contexto o estrategias más robustas de percepción del entorno, la propuesta presentada pierde en riqueza de información contextual y en capacidad para ampliar el apoyo más allá del guiado de trayectorias. Su principal fortaleza está en la navegación paso a paso y en la asistencia auditiva, pero no en la interpretación avanzada del entorno ni en la detección semántica de elementos del escenario.

Los resultados de satisfacción complementan esta discusión, ya que la valoración positiva de los participantes respecto a la claridad de las indicaciones y la facilidad de uso respalda la pertinencia de la solución desarrollada como herramienta de apoyo a la movilidad exterior. Sin embargo, al contrastar estos hallazgos con los antecedentes, se concluye que futuras mejoras deberían orientarse al fortalecimiento del modo offline, al refinamiento de la generación de rutas peatonales y, de ser posible, a la incorporación de información contextual adicional que permita enriquecer la experiencia de navegación. De esta manera, la comparación con la literatura no solo permite identificar en qué gana y en qué pierde la propuesta, sino también delimitar con mayor claridad el alcance real de los aportes del trabajo.

4.5 CONCLUSIONES Y OBSERVACIONES

En síntesis, el capítulo 4 muestra que el aplicativo Guíame alcanza un funcionamiento estable en las condiciones de prueba planteadas y cumple con los escenarios de validación definidos. En cuanto a los alcances logrados, las pruebas de integración confirman que la navegación entre las pantallas de inicio, configuración, gestión de destinos y navegación se realiza de manera coherente, que los parámetros de configuración se almacenan correctamente y que la selección de mapa online u offline, así como del idioma y del tipo de retroalimentación, se refleja en el comportamiento del sistema. Asimismo, las pruebas de campo evidencian que los usuarios logran seguir la ruta propuesta recibiendo instrucciones en tiempo, perciben que la aplicación les ayuda a corregir desvíos de la trayectoria y valoran positivamente la claridad de la retroalimentación auditiva, especialmente cuando se combinan mensajes de voz con pitidos lateralizados.

Desde el punto de vista cuantitativo, los resultados obtenidos permiten identificar el alcance real del filtrado de la señal de posición. El error entre la señal GPS/GPRS cruda y la señal filtrada se mantiene en el orden de 1–2 m, de manera consistente tanto en las trayectorias de ida como en las de vuelta, lo que indica que el filtro suaviza los sobresaltos de la trayectoria sin introducir desplazamientos excesivos. El análisis punto a punto del RMSE a lo largo de la ruta, calculado sobre waypoints separados cada 2,5 m, muestra que en la mayor parte del recorrido las correcciones se mantienen entre 1,0 y 1,5 m, pero pone de manifiesto la existencia de tramos específicos donde el RMSE se incrementa de forma localizada, con valores cercanos o superiores a 2 m. Estos picos sugieren que las principales limitaciones de precisión se relacionan más con la calidad de la señal y con la definición de la ruta en el modo offline que con el comportamiento intrínseco del filtro.

En cuanto a las limitaciones del estudio, se destaca que el protocolo de pruebas se aplicó a una única ruta en un entorno controlado del campus universitario y con un grupo reducido de participantes videntes vendados, por lo que los resultados no

pueden generalizarse de manera directa a otros contextos urbanos ni a toda la población con discapacidad visual. Adicionalmente, no se exploraron variaciones en las condiciones ambientales (tráfico intenso, lluvia, ruido) ni en la infraestructura peatonal, aspectos que pueden afectar de forma importante la experiencia de uso. Finalmente, aunque el filtrado mejora la estabilidad de la señal, persisten desviaciones de varios metros entre la posición estimada y la ruta propuesta en determinados segmentos, lo que señala la necesidad de trabajar en la generación de rutas peatonales más precisas y en la fusión de información con otros sensores para aumentar la robustez del sistema en escenarios reales más exigentes.

5 CAPÍTULO 5: CONCLUSIONES Y TRABAJOS FUTUROS

5.1 CONCLUSIONES GENERALES

La movilidad independiente de las personas con discapacidad visual continúa siendo un desafío tanto tecnológico como social, especialmente en entornos urbanos donde la señal GNSS se ve afectada por cañones urbanos, edificios altos y zonas de baja cobertura. Frente a esta problemática, el presente trabajo de grado se planteó como una propuesta orientada a fortalecer la autonomía de desplazamiento mediante el desarrollo de un prototipo de aplicación móvil accesible, denominada GuiaMe, que ofrece navegación paso a paso con retroalimentación auditiva y soporte para mapas online y offline. La solución se orientó al contexto de la Universidad del Valle, pero con la intención de sentar bases que puedan extenderse a otros escenarios urbanos.

El primer paso consistió en realizar una revisión bibliográfica sobre tecnologías de asistencia para personas con discapacidad visual, aplicaciones de navegación accesible, sistemas GNSS y métodos de filtrado de señales de localización. Esta revisión, desarrollada en el capítulo 2, permitió identificar las limitaciones de los sistemas de posicionamiento en entornos reales, así como las estrategias propuestas en la literatura para mitigar errores mediante filtros de Kalman, filtros de Kalman extendidos y filtros de partículas. Adicionalmente, se analizaron lineamientos de accesibilidad en Android y el papel de la retroalimentación multimodal (voz, pitidos, vibración) en aplicaciones dirigidas a usuarios con discapacidad visual. Este marco teórico proporcionó los elementos necesarios para definir los requerimientos funcionales y no funcionales del aplicativo y para seleccionar las tecnologías más adecuadas para su implementación.

Sobre esta base conceptual, en el capítulo 3 se llevó a cabo el diseño de la solución, organizando el sistema en tres componentes principales: configuración, gestión de destinos y navegación. El módulo de configuración permite seleccionar el tipo de mapa (online u offline), el idioma de la interfaz y el tipo de retroalimentación (voz, pitidos, vibración), almacenando estos parámetros para que el usuario no deba configurarlos en cada sesión. El módulo de gestión de destinos se encarga de registrar, editar y eliminar puntos de interés, aprovechando tanto mapas online como mapas offline en formato MBTiles para obtener las coordenadas de los destinos. El módulo de navegación integra el cálculo de ruta, el filtrado de la señal de posición y la emisión de instrucciones turn-by-turn mediante mensajes de voz y señales auditivas lateralizadas, siguiendo una lógica de interacción simplificada compatible con lectores de pantalla como TalkBack.

Posteriormente, se materializó el diseño conceptual en un prototipo funcional de la aplicación Guíame. La implementación incluyó la integración con servicios de mapas online, el uso de mapas offline en MBTiles para escenarios de conectividad limitada, la incorporación de un filtro de partículas para estabilizar la señal de localización y la definición de una interfaz accesible que separa el modo de acompañante (para la configuración inicial) del modo de usuario (para la navegación autónoma). En esta fase se prestó especial atención a la persistencia de parámetros mediante almacenamiento local, a la coherencia en la navegación entre pantallas y a la compatibilidad con tecnologías de asistencia integradas en Android, con el fin de que la aplicación pueda ser utilizada por personas con discapacidad visual en un entorno de uso real.

El capítulo 4 presentó el proceso de validación del sistema, que incluyó pruebas de integración de los módulos, pruebas cuantitativas sobre la señal de localización y pruebas de campo en una ruta peatonal dentro del campus universitario, con participantes videntes vendados. Los resultados obtenidos muestran que el filtro seleccionado introduce correcciones moderadas, del orden de 1–2 m, contribuyendo a suavizar los saltos de posición sin distorsionar de manera significativa la trayectoria. El análisis punto a punto del RMSE entre la señal GPS/GPRS cruda y la señal filtrada, calculado sobre waypoints separados cada 2,5 m, evidencia que en la mayor parte del recorrido las correcciones se mantienen entre 1,0 y 1,5 m, pero también permite identificar tramos específicos donde el error se incrementa hasta valores cercanos o superiores a 2 m, asociados a zonas de mayor variabilidad de la señal GNSS. Desde el punto de vista cualitativo, las pruebas de campo sugieren que la combinación de instrucciones de voz y señales auditivas apoya al usuario en el seguimiento de la ruta, aunque se requieren ajustes adicionales para reducir la carga cognitiva y mejorar la claridad de algunos mensajes en contextos más complejos.

En conjunto, este trabajo de grado demuestra que es posible diseñar, implementar y validar un prototipo de aplicación de navegación accesible que combina mapas online y offline, filtrado de la señal de localización y retroalimentación auditiva, orientado a apoyar la movilidad de personas con discapacidad visual. La solución propuesta no pretende sustituir otras ayudas existentes, sino complementar el ecosistema de tecnologías de asistencia mediante una herramienta adaptable al contexto universitario y potencialmente extensible a otros entornos urbanos. Aunque persisten limitaciones relacionadas con la precisión de la navegación, especialmente en modo offline y en los tramos del recorrido donde la señal GNSS presenta mayor variabilidad, así como en la cobertura de rutas evaluadas y el tamaño de la muestra de usuarios, los resultados obtenidos constituyen una base sólida sobre la cual continuar desarrollando y perfeccionando sistemas de guiado peatonal accesible en futuras investigaciones.

5.2 TRABAJOS FUTUROS

A partir de los resultados obtenidos y las limitaciones observadas en el presente trabajo, se identifican múltiples oportunidades de mejora y expansión que pueden guiar el desarrollo de futuras investigaciones o implementaciones del sistema propuesto. Por lo tanto, se proponen las siguientes líneas de acción:

1. Ampliar la validación con usuarios objetivo, realizando pruebas de campo con personas con discapacidad visual en diferentes contextos urbanos y rurales, con el fin de evaluar la usabilidad, la carga cognitiva y la percepción de seguridad en condiciones reales de uso.
2. Mejorar la precisión de la navegación en escenarios offline, revisando y refinando la generación de rutas peatonales a partir de mapas MBTiles, incluyendo procesos de corrección y ajuste de la red de caminos para reducir la desalineación entre la ruta calculada y el recorrido real del usuario.
3. Profundizar en la fusión sensorial, incorporando de manera más extensa la información de las unidades de medida inercial (IMU), técnicas de odometría por pasos y estrategias de map-matching, con el objetivo de reducir el impacto de la degradación de la señal GPS/GPRS en zonas de baja cobertura o alta interferencia.
4. Enriquecer y personalizar la retroalimentación multimodal, explorando nuevos patrones de voz, pitidos y vibración, así como mecanismos de ajuste automático según el entorno (ruido ambiental, velocidad de marcha) y las preferencias del usuario, manteniendo siempre la compatibilidad con lectores de pantalla como TalkBack y comandos de voz.
5. Escalar el prototipo hacia una solución desplegable, incorporando módulos de registro de eventos y telemetría que permitan analizar el comportamiento del sistema en uso prolongado, mejorar la gestión de consumo energético y estudiar su integración con servicios externos (por ejemplo, información de transporte público o infraestructura accesible).

Estas líneas de trabajo permitirán fortalecer la solución propuesta, incrementar su confiabilidad en escenarios más exigentes y acercarla a un producto que pueda ser transferido y adoptado por la comunidad de personas con discapacidad visual y por organizaciones dedicadas a su acompañamiento.

6 REFERENCIAS

- [1] “Panorama general de la discapacidad en Colombia,” 2020.
- [2] L. F. Rendón Campo, P. Morales, and O. Ramírez, “Caracterización de la población con discapacidad visual y auditiva matriculada en la Unidad de Rehabilitación del Instituto para Niños Ciegos y Sordos del Valle del Cauca.,” Oct. 31, 2014. Accessed: Feb. 17, 2025. [Online]. Available: <https://hdl.handle.net/10893/8227>
- [3] “Lazarillo - Inclusive Navigation and Accessible Mapping for Indoor and Outdoor Spaces.” Accessed: Feb. 17, 2025. [Online]. Available: <https://lazarillo.app/>
- [4] “Seeing Assistant – Move.” Accessed: Feb. 17, 2025. [Online]. Available: <https://seeingassistant.tt.com.pl/en/move/>
- [5] “Accesibilidad en Google Maps - Android - Ayuda de Google Maps.” Accessed: Feb. 17, 2025. [Online]. Available: <https://support.google.com/maps/answer/6396990?hl=es&co=GENIE.Platform%3DAndroid&oco=1>
- [6] “Inicio - Plan Integral de Expansión de Conectividad Digital.” Accessed: Oct. 17, 2025. [Online]. Available: <https://mintic.gov.co/micrositios/PlanConectividadDigital/870/w3-channel.html#intro>
- [7] A. Angrisano and S. Gaglione, “Smartphone GNSS Performance in an Urban Scenario with RAIM Application,” *Sensors 2022, Vol. 22, Page 786*, vol. 22, no. 3, p. 786, Jan. 2022, doi: 10.3390/S22030786.
- [8] S. Pundlik, P. Shivshanker, T. Traut-Savino, and G. Luo, “Field evaluation of a mobile app for assisting blind and visually impaired travelers to find bus stops,” Sep. 2023, Accessed: Feb. 17, 2025. [Online]. Available: <https://arxiv.org/abs/2309.10940v1>
- [9] S. C. Reina, Y. Hou, J. Kannala, and A. Solin, “Iterative Path Reconstruction for Large-Scale Inertial Navigation on Smartphones,” *FUSION 2019 - 22nd International Conference on Information Fusion*, Jun. 2019, doi: 10.23919/fusion43075.2019.9011414.
- [10] L. Tepelea, I. Gavriliuț, and A. Gacsádi, “Smartphone application to assist visually impaired people,” *2017 14th International Conference on Engineering of Modern Electric Systems, EMES 2017*, pp. 228–231, Jul. 2017, doi: 10.1109/EMES.2017.7980421.
- [11] H. Baskaran, R. L. M. Leng, F. A. Rahim, and M. E. Rusli, “Smart vision: Assistive device for the visually impaired community using online computer vision service,” *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*, pp. 730–734, Feb. 2019, doi: 10.1109/CCOMS.2019.8821635.
- [12] L. Árvai, “Mobile phone based indoor navigation system for blind and visually impaired people: VUK - Visionless supporting framework,” *Proceedings of the 2018 19th International Carpathian Control Conference, ICC 2018*, pp. 383–388, Jun. 2018, doi: 10.1109/CARPATHIANCC.2018.8399660.
- [13] C. P. D. Wong, K. Y. F. Wong, and V. W. L. Tam, “Adapting the Particle Filter Algorithm for the Street Navigation of the Visually Impaired,” *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, pp. 2918–2923, Dec. 2020, doi: 10.1109/SSCI47803.2020.9308237.
- [14] “Voice Helper: A Mobile Assistive System for Visually Impaired Persons | IEEE Conference Publication | IEEE Xplore.” Accessed: Feb. 17, 2025. [Online]. Available: <https://ieeexplore-ieee-org.bd.univalle.edu.co/document/7363253>

- [15] J. A. Costales, A. K. S. Vida, and M. G. Albino, "Mobile Based Navigation System for Visually Impaired Person Using Image Detection with Voice Translation," *2023 3rd International Conference on Information Communication and Software Engineering, ICICSE 2023*, pp. 32–36, 2023, doi: 10.1109/ICICSE58435.2023.10211761.
- [16] A. Ghosh, S. A. Al Mahmud, T. I. R. Uday, and D. M. Farid, "Assistive Technology for Visually Impaired using Tensor Flow Object Detection in Raspberry Pi and Coral USB Accelerator," *2020 IEEE Region 10 Symposium, TENSYP 2020*, pp. 186–189, Jun. 2020, doi: 10.1109/TENSYP50017.2020.9230630.
- [17] K. Soeda, S. Aoki, K. Yanashima, and K. Magatani, "Development of the visually impaired person guidance system using GPS," *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, vol. 26 VII, pp. 4870–4873, 2004, doi: 10.1109/IEMBS.2004.1404347.
- [18] D. P. Soukaras *et al.*, "Augmented Audio Reality Mobile Application Specially Designed for Visually Impaired People," *Panhellenic Conference on Informatics*, pp. 13–18, 2012, doi: 10.1109/PCI.2012.50.
- [19] D. H. Yunardi, K. Saputra, S. J. Akbar, and Y. Aditya, "Design and Development of Route Guidance Application on Android-Based Navigation System for Visually Impaired Individuals," *Proceedings of the International Conference on Electrical Engineering and Informatics*, pp. 147–152, 2024, doi: 10.1109/ICELTICS62730.2024.10776116.
- [20] S. Choudhary, V. Bhatia, and K. R. Ramkumar, "IoT Based Navigation System for Visually Impaired People," *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, pp. 521–525, Jun. 2020, doi: 10.1109/ICRITO48877.2020.9197857.
- [21] I. Maly, J. Balata, O. Krejcir, E. Fuzessery, and Z. Mikovec, "Qualitative measures for evaluation of navigation applications for visually impaired," *6th IEEE Conference on Cognitive Infocommunications, CogInfoCom 2015 - Proceedings*, pp. 223–228, Jan. 2016, doi: 10.1109/COGINFocom.2015.7390595.
- [22] H. E. Chen, Y. Y. Lin, C. H. Chen, and I. F. Wang, "BlindNavi: A navigation app for the visually impaired smartphone user," *Conference on Human Factors in Computing Systems - Proceedings*, vol. 18, pp. 19–24, Apr. 2015, doi: 10.1145/2702613.2726953.
- [23] J.-E. Kim, M. Bessho, N. Koshizuka, and K. Sakamura, "Mobile applications for assisting mobility for the visually impaired using IoT infrastructure," *Proceedings of 2014 TRON Symposium (TRONSHOW)*, 2014.
- [24] J. A. Costales, A. K. S. Vida, and M. G. Albino, "Mobile Based Navigation System for Visually Impaired Person Using Image Detection with Voice Translation," *2023 3rd International Conference on Information Communication and Software Engineering, ICICSE 2023*, pp. 32–36, 2023, doi: 10.1109/ICICSE58435.2023.10211761.
- [25] G. Olmschenk, C. Yang, Z. Zhu, H. Tong, and W. H. Seiple, "Mobile Crowd Assisted Navigation for the Visually Impaired," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, IEEE, Aug. 2015, pp. 324–327. doi: 10.1109/UIC-ATC-ScalCom-CBDCOM-LoP.2015.69.
- [26] "Blindness and vision impairment." Accessed: Oct. 18, 2025. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>

- [27] “La OMS presenta el primer Informe mundial sobre la visión.” Accessed: Feb. 17, 2025. [Online]. Available: <https://www.who.int/es/news/item/08-10-2019-who-launches-first-world-report-on-vision>
- [28] “UNFPA Colombia | Análisis de Situación de Población (ASP Colombia 2023) Capítulo Personas con Discapacidad.” Accessed: Feb. 17, 2025. [Online]. Available: <https://colombia.unfpa.org/es/publications/analisis-de-situacion-de-poblacion-asp-colombia-2023-capitulo-personas-con-discapacidad>
- [29] W. Health Organization, “World report on vision”.
- [30] “Visual Field Assessment and Disability Evaluation,” *Visual Field Assessment and Disability Evaluation*, Aug. 2025, doi: 10.17226/29124.
- [31] V. D. Vázquez Palacios, R. Mastachi Torres, and A. S. Sánchez Sánchez, “Dispositivos auxiliares para la discapacidad visual,” *Revista Digital Universitaria*, vol. 23, no. 5, Sep. 2022, doi: 10.22201/CUAIEED.16076079E.2022.23.5.1.
- [32] P. Theodorou, K. Tsiligkos, and A. Meliones, “Multi-Sensor Data Fusion Solutions for Blind and Visually Impaired: Research and Commercial Navigation Applications for Indoor and Outdoor Spaces,” *Sensors (Basel)*, vol. 23, no. 12, p. 5411, Jun. 2023, doi: 10.3390/S23125411.
- [33] J. M. Feghali, C. Feng, A. Majumdar, and W. Y. Ochieng, “Comprehensive Review: High-Performance Positioning Systems for Navigation and Wayfinding for Visually Impaired People,” *Sensors 2024, Vol. 24, Page 7020*, vol. 24, no. 21, p. 7020, Oct. 2024, doi: 10.3390/S24217020.
- [34] N. Tyagi, D. Sharma, J. Singh, B. Sharma, and S. Narang, “Assistive Navigation System for Visually Impaired and Blind People: A Review,” *Proceedings - 2021 1st IEEE International Conference on Artificial Intelligence and Machine Vision, AIMV 2021*, 2021, doi: 10.1109/AIMV53313.2021.9670951.
- [35] M. Hersh, “Wearable Travel Aids for Blind and Partially Sighted People: A Review with a Focus on Design Issues,” *Sensors 2022, Vol. 22, Page 5454*, vol. 22, no. 14, p. 5454, Jul. 2022, doi: 10.3390/S22145454.
- [36] “I+D y Aplicaciones GNSS | Portal Geodésico de Andalucía.” Accessed: Oct. 24, 2025. [Online]. Available: <https://www.juntadeandalucia.es/institutodeestadisticaycartografia/rap/id-y-aplicaciones-gnss>
- [37] “What are Global Navigation Satellite Systems? | NovAtel.” Accessed: Oct. 24, 2025. [Online]. Available: <https://novatel.com/tech-talk/an-introduction-to-gnss/what-are-global-navigation-satellite-systems-gnss>
- [38] “GPS Accuracy | GPS.gov.” Accessed: Nov. 03, 2025. [Online]. Available: <https://www.gps.gov/gps-accuracy-0>
- [39] “Filtros Kalman - MATLAB & Simulink.” Accessed: Nov. 03, 2025. [Online]. Available: <https://la.mathworks.com/discovery/kalman-filter.html>
- [40] G. Welch and G. Bishop, “An Introduction to the Kalman Filter”, Accessed: Nov. 03, 2025. [Online]. Available: <http://www.cs.unc.edu/~gb>
- [41] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 50, no. 2, 2002.
- [42] “Scrum: qué es y por qué es una de las metodologías ágiles.” Accessed: Feb. 17, 2025. [Online]. Available: <https://ausum.cloud/scrum-metodologia-agil-mas-popular-en-empresas/>

7 ANEXOS

En la Figura 7.1 se presenta cómo están organizados los documentos y archivos pertenecientes al desarrollo del trabajo de grado como: los documentos de la metodología SCRUM, pruebas de integración y el aplicativo móvil. Se puede acceder a esta documentación a través del siguiente enlace:

https://drive.google.com/drive/folders/1XhDhOFpkGoG5M1MrFH8hn-SW7fhpNQot?usp=drive_link

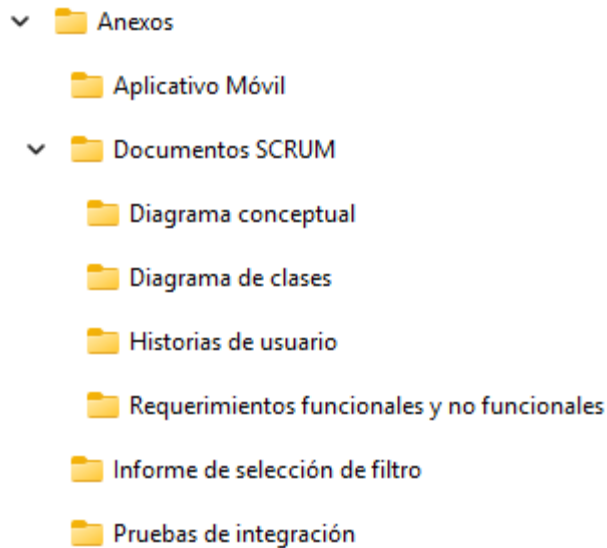


Figura 7.1 Árbol de directorio de los anexos.