



**Optimización de la Movilidad Urbana mediante una Plataforma Conversacional
Asistida por IA para la Planificación de Rutas**

Obando Buitron, Camila Antonela y Gualpa Rivera, Mathias Elian

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniería en
Tecnologías de la Información

Ph.D. Benavides Astudillo, Diego Eduardo

31 de julio de 2026



Certificado de análisis

Compilatio Magister+ | ESPE-ECU

Optimización de la Movilidad Urbana mediante una Plataforma Conversacional Asistida por IA

ID : b6b2ab5bcc0bee08dbdd99fab0bd4dc67e66e767

7%

Textos sospechosos

Nombre del fichero : Optimización de la Movilidad Urbana mediante una Plataforma Conversacional Asistida por IA.txt

Tamaño del archivo original : 3,09 MB

Número de palabras : 23.171

Depositante : DIEGO EDUARDO BENAVIDES ASTUDILLO

Fecha de depósito : 24 de agosto de 2026

Tipo de carga : interface

fecha de fin de análisis : 24 de agosto de 2026

 Resumen

(sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

 Similitudes

3%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



 Detección de IA

4%

Pasajes estilísticamente próximos a un texto generado por una IA.

Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.



 Idiomas no reconocidos

9%

Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.

Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.



No incluido en el porcentaje de textos sospechosos :

 Textos entre comillas

<1%

Pasajes entre comillas o entre comillas de comillas.



Ph.D. Benavides Astudillo, Diego Eduardo

Director

Código de documento: UDED-FOR-2026-V5-016

Código de proceso: GDOC-ATAD-1

Rev. UPDI:2025-Feb-25



Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Certificación

Certifico que el trabajo de integración curricular: “Optimización de la Movilidad Urbana mediante una Plataforma Conversacional Asistida por IA para la Planificación de Rutas” fue realizado por los señores Obando Buitrón Camila Antonela y Gualpa Rivera Mathias Elian, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Santo Domingo, 31 de Julio de 2026

Ph.D. Benavides Astudillo, Diego Eduardo

Director

C.C.: 1712883063



Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Responsabilidad de Autoría

Nosotros, Obando Buitrón, Camila Antonela y Gualpa Rivera, Mathias Elian, con cédulas de ciudadanía N° 2350442782 y 2350025140, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: Optimización de la Movilidad Urbana mediante una Plataforma Conversacional Asistida por IA para la Planificación de Rutas es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Santo Domingo, 31 de Julio de 2026

Obando Buitron, Camila Antonela

C.C.: 2350442782

Gualpa Rivera, Mathias Elian

C.C.: 2350025140



Departamento de Ciencias de la Computación

Carrera de Ingeniería en Tecnologías de la Información

Autorización de Publicación

Nosotros Obando Buitrón, Camila Antonela y Gualpa Rivera, Mathias Elian, con cédulas de ciudadanía N° 2350442782 y 2350025140, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: Optimización de la Movilidad Urbana mediante una Plataforma Conversacional Asistida por IA para la Planificación de Rutas en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Santo Domingo, 31 de Julio de 2026

Obando Buitron, Camila Antonela

C.C.: 2350442782

Gualpa Rivera, Mathias Elian

C.C.: 2350025140

Dedicatoria

Quiero agradecer, primeramente, a Dios por acompañarme en cada paso de este camino.

A mi madre, Ángela Cagua, por su apoyo incondicional durante todo mi proceso de formación, desde la escuela hasta ahora; por su amor, compromiso y protección hacia mí; por comprender mi proceso y acompañarme en cada una de las fases de mi vida. A ella no me alcanzaría la vida para demostrarle lo infinitamente agradecida que estoy, por tanto.

A mi tío Fernando Obando, quien ha sido desde siempre un gran ejemplo y un pilar fundamental en mi vida académica. Gracias por sus consejos, su orientación y por el esfuerzo dedicado hacia mí, este logro también es suyo.

A mi tío Jaime Obando y su esposa Érika Ásqui, quienes han sido un claro ejemplo de la persona que quiero llegar a ser algún día. Gracias por apoyarme siempre; mi tío ha sido alguien en quien siempre he podido confiar, y la señora Érika, a quien admiro por su dedicación y éxito en el ámbito laboral.

A mi padre, Anthony Obando, expreso mi más profundo agradecimiento por su amor y afecto, así como por estar siempre presente y brindarme su apoyo a lo largo de las distintas etapas de mi vida.

A toda mi familia, a quienes quizás ya no pueda nombrar uno a uno, pero a quienes estoy igualmente agradecida por impulsar mi crecimiento hasta este logro.

Gracias infinitas también a mis compañeros de universidad, por brindarme su apoyo y su amistad, por lo cual siempre los llevaré en mi corazón.

Camila Obando

En primer lugar, quiero agradecer a Dios por permitirme cumplir un logro más en mi vida y por siempre bendecirme y cuidarme en todo momento. Este trabajo va dedicado a todas las personas que formaron parte de mi vida durante toda esta etapa universitaria.

A mi madre, Tatiana Rivera quien sin duda ha sido una pieza fundamental a lo largo de toda mi carrera ya que, gracias a ella, a su apoyo, a su cariño y a sus palabras de aliento no lo hubiese logrado ya que estuvo en los momentos indicados. Por más estresado que estuviese, siempre tuvo esa paciencia y ese amor que solo una madre sabe brindar.

A mi padre, Franklin Gualpa, quien también ha sido una pieza fundamental, ya que él, a su manera supo brindarme su apoyo, supo guiarme por el camino del bien, y por él y sus enseñanzas soy la persona de ahora, ya que me enseñó que la vida no es fácil y toca dar todo de uno por más difícil que sea, y seguir adelante a pesar de todo.

A mis hermanos Derik y Daniela que a pesar de la lejanía siempre me brindaron su apoyo y ese cariño de hermanos, y a mis compañeros de universidad que de cada uno de ellos me llevo aprendizajes y recuerdos.

Finalmente, dedico este trabajo a mi mejor amiga Dayana Mendoza, quien también se convirtió en una pieza fundamental durante los últimos períodos de mi carrera universitaria. Quiero agradecerle de todo corazón por el apoyo que me brindó especialmente en los momentos más difíciles, cuando más necesitaba de alguien, ella siempre estuvo presente ofreciéndome palabras de aliento, aconsejándome y sobre todo demostrándome esa seguridad necesaria para seguir adelante sin darme por vencido. Su compañía, su lealtad y la confianza que siempre depositó en mí fueron un impulso invaluable para superar cada obstáculo y continuar luchando por este objetivo.

Gracias a todos por creer en mí, incluso cuando yo dudaba de mis capacidades. Sin duda, sus apoyos fueron una parte muy importante para que hoy este logro se haga realidad.

Mathias Gualpa

Agradecimiento

Expresamos nuestro más sincero agradecimiento al Ingeniero Cristian Coronel, director de Carrera, por su constante atención y seguimiento hacia cada uno de los estudiantes de la institución, así como por brindarnos, de manera oportuna, las soluciones necesarias ante los diversos inconvenientes presentados a lo largo de nuestra formación académica.

Al Doctor Eduardo Benavides, nuestro más profundo reconocimiento por haber participado como tutor del presente trabajo de titulación, y por habernos acompañado con dedicación y compromiso, a lo largo de todo el proceso que ha hecho posible alcanzar esta importante meta profesional.

Al Ingeniero Paulo Galarza, nuestra sincera gratitud por habernos proporcionado el dataset correspondiente al año 2024, insumo que constituyó el recurso principal para el desarrollo del presente trabajo de integración curricular implementado en esta investigación, cuyo aporte resultó determinante para el cumplimiento satisfactorio de los objetivos planteados.

Al Ingeniero Armando Ortiz y al Ingeniero Daniel Núñez, quienes fungieron como revisores del presente trabajo de titulación, extendemos nuestro reconocimiento por el tiempo y la dedicación brindados durante este proceso. Ambos fueron, en distintos momentos de nuestra formación, nuestros docentes, por lo que su aporte académico contribuyó significativamente a nuestro desarrollo profesional.

Finalmente, hacemos extensivo nuestro agradecimiento a todos los docentes de la Universidad de las Fuerzas Armadas ESPE, sede Santo Domingo, quienes, a lo largo de nuestra formación, nos brindaron los conocimientos, herramientas y valores necesarios para convertirnos en los profesionales que somos hoy en día.

INDICE

CERTIFICACIÓN	3
DEDICATORIA.....	6
AGRADECIMIENTO	8
RESUMEN	15
ABSTRACT	16
CAPÍTULO I	17
INTRODUCCIÓN	17
PLANTEAMIENTO DEL PROBLEMA.....	20
FORMULACIÓN DEL PROBLEMA O PREGUNTA DE INVESTIGACIÓN	22
OBJETIVOS	23
JUSTIFICACIÓN	24
CAPÍTULO II	27
ESTADO DEL ARTE	27
MARCO TEÓRICO	29
ARQUITECTURA DE SOFTWARE.....	29
METODOLOGÍA DE DESARROLLO ITERATIVO.....	29
ARQUITECTURA EN CAPAS	30
API REST	31
TECNOLOGÍAS BACKEND.....	32
Python	32
Gestión de paquetes PIP	32
Flask (Micro-Framework).....	33
PostgreSQL.....	33
SQLAlchemy	34
NetworkX.....	34
TECNOLOGÍAS FRONTEND.....	34

React.....	35
Vite	35
React Router DOM.....	36
Axios.....	36
SISTEMA DE INFORMACIÓN GEOGRÁFICA	37
Leaflet.....	37
React Leaflet	38
OpenStreetMap	38
Fórmula de Haversine	38
SEGURIDAD Y AUTENTICACIÓN	39
JWT (JSON Web Token)	39
Bcrypt	39
HERRAMIENTAS DE DESARROLLO	40
Git.....	40
GitHub	40
Postman	41
PgAdmin.....	42
CAPÍTULO III	43
METODOLOGÍA	43
ENFOQUE DE INVESTIGACIÓN	43
TIPO Y DISEÑO DE INVESTIGACIÓN	43
ANÁLISIS DE REQUERIMIENTOS	44
Requerimientos funcionales.....	45
Requerimientos no funcionales.....	46
Requerimientos tecnológicos	47
ARQUITECTURA DEL SISTEMA	48
Capa de presentación	49
Capa de aplicación.....	50

Capa de inteligencia artificial	50
Capa de servicios geográficos	50
Capa de procesamiento de rutas	51
Capa de datos	51
FLUJO DE PROCESAMIENTO DE UNA CONSULTA	52
METODOLOGÍA DE DESARROLLO ITERATIVA	53
Iteración 1 Análisis, planificación y preparación de la información.....	55
Iteración 2 Desarrollo de la infraestructura base	56
Iteración 3 Integración del procesamiento conversacional.....	58
Iteración 4 Construcción y optimización del motor de rutas	60
Iteración 5 Integración de servicios geográficos y visualización	62
Iteración 6 Pruebas, corrección y refinamiento.....	64
POBLACIÓN Y MUESTRA	65
TÉCNICAS E INSTRUMENTOS.....	66
VALIDACIÓN Y CONFIABILIDAD DEL INSTRUMENTO	67
CONSIDERACIONES ÉTICAS	68
CAPÍTULO IV	69
ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	70
DESARROLLO DE LA PLATAFORMA CONVERSACIONAL.....	70
IMPLEMENTACIÓN DE LA PLATAFORMA WEB	71
PROCESAMIENTO CONVERSACIONAL Y PLANIFICACIÓN DE RUTAS	75
Procesamiento mediante lenguaje natural.....	75
Localización de lugares.....	76
Cálculo de rutas	77
Generación del recorrido.....	79
VISUALIZACIÓN DE RUTAS E INTERACCIÓN CON EL USUARIO	80
PRUEBAS DEL SISTEMA	81
Pruebas funcionales.....	82

Pruebas de funcionamiento	83
Pruebas de usabilidad (SUS).....	85
Interpretación y análisis de usabilidad del chatbus	87
DISCUSIÓN DE RESULTADOS	88
CAPÍTULO V	90
CONCLUSIONES	90
RECOMENDACIONES	92
BIBLIOGRAFÍA	94

INDICE DE TABLAS

Tabla 1 Requerimientos Funcionales ChatBus.....	45
Tabla 2 Requerimientos No Funcionales ChatBus	46
Tabla 3 Requerimientos tecnológicos ChatBus	47
Tabla 4 Actividades de la primera iteración	56
Tabla 5 Ejemplos de interpretación de consultas	60
Tabla 6 Criterio de evaluación.....	61
Tabla 7 Elementos visibles en el mapa	63
Tabla 8 Resultados de las pruebas funcionales	82
Tabla 9 Resultados de las pruebas de funcionamiento	84

INDICE DE FIGURAS

Figura 1 Arquitectura general de la plataforma conversacional.....	49
Figura 2 Flujo de procesamiento de una consulta.....	53
Figura 3 Ciclo de vida de la metodología iterativa.....	54
Figura 4 Arquitectura del backend implementado con Flask.....	58
Figura 5 Visualización del recorrido	63
Figura 6 Pantalla principal de la plataforma conversacional.....	71
Figura 7 Pantalla de inicio de sesión de la plataforma.	72
Figura 8 Pantalla de registro de usuarios.....	73
Figura 9 Pantalla principal de la plataforma conversacional.....	73
Figura 10 Historial de conversaciones.	74
Figura 11 Perfil del usuario.	75
Figura 12 Procesamiento de una consulta mediante lenguaje natural.	76
Figura 13 Localización del origen y destino de la consulta.....	77

Figura 14 Resultado del cálculo de la ruta óptima.....	78
Figura 15 Recorrido generado por la plataforma conversacional.	79
Figura 16 Visualización del recorrido generado por la plataforma.....	81
Figura 17 Respuestas obtenidas del cuestionario SUS aplicado a los participantes.	85
Figura 18 Transformación de las respuestas del cuestionario SUS.	86
Figura 19 Cálculo del puntaje SUS y resultados estadísticos.	87
Figura 20 Interpretación de la puntuación de la escala System Usability Scale (SUS)..	88

Resumen

La planificación de las rutas del transporte público urbano a las que pueden acceder los ciudadanos presenta problemas en la resolución de la situación debido a la no existencia de herramientas que sean capaces de dar respuesta a las preguntas que se formulan en lenguaje natural. En este sentido, y partiendo de lo anterior, esta investigación se centró en el desarrollo de una plataforma web inteligente basada en un agente conversacional para la recomendación de rutas del transporte público urbano de Santo Domingo, a partir del procesamiento del lenguaje natural y los modelos lógicos de grafos. Este desarrollo se realizó mediante una metodología creciente iterativa y haciendo uso de una arquitectura cliente-servidor y React para el frontend, Flask para el backend, PostgreSQL como motor de gestión de datos y Google Gemini para dar respuestas a las preguntas, integrando el algoritmo de Dijkstra para el cálculo de las rutas. También se integraron las APIs de los servicios geográficos Nominatim, OSRM, Leaflet y OpenStreetMap para localizar los recorridos y mostrarlos en mapas. La Plataforma fue analizada mediante pruebas funcionales, de funcionamiento y de usabilidad con la aplicación de la escala System Usability Scale (SUS) e involucrando a quince usuarios en total. Los resultados de la evaluación mostraron que el sistema funciona correctamente en la interpretación de consultas, en la identificación de ubicaciones, en la planificación de rutas y en la representación mediante mapas, además de obtener una puntuación media de 80,33/100 en la evaluación de la escala SUS, ubicada dentro de la categoría Excelente, lo que demuestra la facilidad de uso de la plataforma y su viabilidad como herramienta para soportar la movilidad urbana en el cantón Santo Domingo.

Palabras clave: Inteligencia artificial, procesamiento de lenguaje natural, movilidad urbana.

Abstract

Planning urban public transport routes for citizens presents challenges due to the lack of tools capable of answering questions posed in natural language. Therefore, this research focuses on developing an intelligent web platform based on a conversational agent to recommend urban public transport routes in Santo Domingo, using natural language processing and logical graph models. This development was carried out using an iterative, incremental methodology and a client-server architecture. The platform utilizes React for the frontend, Flask for the backend, PostgreSQL as the data management engine, and Google Gemini to answer queries, integrating Dijkstra's algorithm for route calculation. The APIs of the geographic services Nominatim, OSRM, Leaflet, and OpenStreetMap were also integrated to locate routes and display them on maps. The platform was analyzed through functional, operational, and usability testing using the System Usability Scale (SUS), involving a total of fifteen users. The evaluation results showed that the system functions correctly in interpreting queries, identifying locations, planning routes, and displaying maps. Furthermore, it obtained an average score of 80.33 on the SUS scale, placing it in the Excellent category. This demonstrates the platform's ease of use and its usefulness as a tool to support urban mobility in the Santo Domingo canton.

Keywords: Artificial intelligence, natural language processing, urban mobility.

Capítulo I

Introducción

La transformación digital ha hecho posible la incorporación de tecnologías inteligentes como herramientas de apoyo a la planificación y optimización de la movilidad urbana de las ciudades. Las plataformas conversacionales apoyadas por inteligencia artificial permiten ofrecer experiencias de consulta accesibles desde un navegador web, de manera que los ciudadanos puedan identificar rutas de transporte público usando lenguaje natural, sin la necesidad de conocer previamente la nomenclatura oficial de las líneas o de las paradas. Estas soluciones aportan a reforzar la accesibilidad digital en los sistemas de transporte y a mitigar las barreras tecnológicas que enfrentan las personas usuarias a la hora de planificar los desplazamientos.

Las herramientas han comenzado a ser utilizadas todavía en un desarrollo limitado, donde predominan, en muchas ocasiones, aplicaciones comerciales como Moovit o Waze, que han tenido una eficacia demostrada al integrar geolocalización, tráfico y rutas en tiempo real [1], pero para las cuales es necesario contar con una estructura de datos con un alto grado de digitalización que no se encuentra presente en muchas ciudades intermedias. Esto se constata en la ciudad de Santo Domingo de los Tsáchilas, donde no contaba con una plataforma que permitiese a la ciudadanía planificar recorridos de transporte público urbano utilizando aplicaciones cotidianas como barrios, cooperativas o puntos emblemáticos de la ciudad. Esta situación se traduce en una baja capacidad de las herramientas de planificación y de acceso por parte de las personas que no prestan atención a las direcciones ni conocen la terminología técnica del sistema vial.

El presente proyecto se apoya en las limitaciones detectadas en otros estudios previos sobre movilidad urbana mediatizada por tecnología microscópica. Los estudios desarrollados en ciudades como Medellín y Río de Janeiro evidenciaron que el análisis de la información podría hacer más eficiente la gestión del tráfico, pero también expusieron una fuerte

dependencia de una infraestructura tecnológica avanzada que limita su escala o réplica para lugares en los que las condiciones digitales son escasas [2]. Las plataformas comerciales como Uber están transitando hacia modelos de movilidad multimodal [3], los cuales mitigan tiempos de viaje mediante un emparejamiento en tiempo real; otras propuestas académicas como el Prototipo de rutas de transporte público o el Plato – Planificador de viajes basado en IA han desarrollado sistemas automatizados para la generación de itinerarios. La lógica de estas propuestas va más allá de la eficiencia matemática del cálculo de las rutas, la cual ha sido automatizada, al tiempo de relegar el uso del lenguaje cotidiano en entornos como el ecuatoriano. A nivel local en Santo Domingo, las investigaciones sobre IA aplicada al transporte urbano público evidenciaron las altas capacidades de este tipo de sistemas, aunque también confirmaron la diferencia existente entre la disponibilidad de datos técnicos y la forma en que los ciudadanos se comunican [4].

La investigación pretende abordar el diseño de una plataforma web inteligente, sustentada por un agente conversacional que esté orientada a recomendar rutas del sistema de transporte urbano de Santo Domingo, aprovechando el procesamiento natural del lenguaje y modelos lógicos de grafos, con el módulo de mejorar la movilidad de la ciudadanía. Para lograr este objetivo se nos planteó, por un lado, identificar las tecnologías y enfoques más adecuados para la planificación de rutas inteligentes y en segundo lugar, el modelado de la arquitectura de datos y del motor del procesamiento del lenguaje natural, así como la implementación de una interfaz interactiva validada a partir de medidas de precisión y de usabilidad con usuarios de la localidad.

La investigación, en lo que respecta a la metodología queda encapsulada en un proyecto de carácter aplicado dentro de un tipo de diseño no experimental y transversal y de un tipo de enfoque cuantitativo, dentro del marco de la implementación de una plataforma de conversación que hace uso de Python y Flask para la implementación del backend, de React y

Leaflet para el frontend, de Google Gemini para el procesamiento de lenguaje natural y de NetworkX que hace uso del algoritmo de Dijkstra para calcular rutas mediante grafos; por lo cual la solución fue evaluada a partir de pruebas funcionales y pruebas de funcionamiento las cuales junto con la Escala de Usabilidad del Sistema (SUS) la cual permite obtener un proceso de validación a partir del comportamiento técnico y la percepción que puedan tener los usuarios de su uso.

La primera de las asignaturas de nuestra investigación consiste en proporcionar un soporte conversacional para mejorar la accesibilidad digital del transporte público dentro de las ciudades de la ciudad de Santo Domingo, siendo capaz de entender preguntas en lenguaje natural y enumeraciones locales y dar finalmente un acto de recomendación que se ajuste a la realidad del contexto ecuatoriano. Esta investigación, así como señalaremos más adelante, muestra la superposición de un gran modelo de lenguaje que engrana con un motor de grafos para dar lugar a un tratamiento geoespacial pero también aporta un flujo de trabajo adecuado para desarrollar plataformas de este tipo de avance sobre la movilidad inteligente en ciudades intermedias.

La presente tesis se estructura en cinco capítulos. En el Capítulo I, se describe el planteamiento del problema, se menciona la formulación de la pregunta de investigación, se exponen los objetivos que se persiguen y se ofrece la justificación de la investigación. El Capítulo II contiene el desarrollo del estado del arte y del marco teórico relativos a la arquitectura de software, a las tecnologías backend y frontend, a los sistemas de mapas y geolocalización y a los mecanismos de seguridad utilizados. El Capítulo III explica la metodología de la investigación utilizada, análisis de requerimientos, arquitectura del sistema y las iteraciones de desarrollo. El Capítulo IV nos muestra los resultados obtenidos y el correspondiente análisis, en términos del procesamiento conversacional, del cálculo de rutas y

de la usabilidad. El Capítulo V contiene las conclusiones a las que se ha llegado y las recomendaciones derivadas del desarrollo de la investigación.

Planteamiento del problema

En la actualidad, en la movilidad urbana, la utilización de plataformas digitales para la planificación de rutas se ha convertido en una práctica habitual en las ciudades que buscan avanzar hacia modelos de ciudades inteligentes. Aplicaciones como Moovit o Waze han demostrado ser eficaces, ya que incorporan información sobre datos de geolocalización, el tráfico y las rutas en tiempo real [1]. La implementación de dichas soluciones en ciudades intermedias de Ecuador, como Santo Domingo de los Tsáchilas, encuentras limitaciones muy importantes. Las grandes ciudades gozan de una información del transporte público estructurada y actualizada, pero en el caso de las ciudades intermedias, la información disponible sobre rutas proviene de referencias informales que utilizan los propios usuarios, lo que merma la efectividad de aquellos sistemas de planificación de rutas.

Estudios han demostrado que, si bien las aplicaciones de movilidad han conseguido mejorar la eficiencia en el transporte de las grandes ciudades, todavía presentan dificultades para adaptarse a contextos específicos. Investigaciones realizadas en Medellín y Río de Janeiro demuestran que el uso de datos masivos permite optimizar la gestión del tráfico, pero también evidencian una fuerte dependencia de la infraestructura tecnológica avanzada [2]. De manera complementaria, se ha identificado que estas herramientas pueden contribuir a redistribuir el flujo vehicular, aunque no resuelven completamente los problemas de congestión, y, en algunos casos, incentivan el uso de transporte privado. Estas condiciones limitantes se hacen todavía más evidentes en urbes como Santo Domingo, donde el transporte público se encuentra más atravesado por dinámicas no estandarizadas.

En este marco de la inteligencia artificial, los desarrollos de los últimos años han conformado soluciones elaboradas en los temas del transporte y la movilidad urbana.

Plataformas internacionales como Uber han ido avanzando hacia modelos multimodales que incorporan junto a la aplicación de un único soporte de transporte otros modos de transporte, acortando los tiempos de recorrido por medio de la implementación de algoritmos de matching en tiempo real [3]. No obstante, estas soluciones comerciales están diseñadas principalmente para los contextos con alta digitalización, la estructura homogénea y los datos abiertos, por lo que no consideran completamente las particularidades sociales, económicas y operativas de las ciudades ecuatorianas.

Como alternativa a estas limitaciones, en el ámbito académico se han desarrollado propuestas orientadas a la optimización de rutas mediante IA que buscan una mayor adaptabilidad. Por ejemplo, el Prototipo de rutas de transporte público usando inteligencia artificial plantea un modelo que transforma la planificación tradicional en un proceso dinámico basado en datos de demanda y tráfico local. De forma similar, el Planificador de viajes basado en inteligencia artificial demuestra cómo automatizar la generación de itinerarios personalizados [4]. Aunque estas propuestas académicas intentan acercarse a variables del entorno, se centran casi exclusivamente en la eficiencia matemática del cálculo de rutas, dejando de lado cómo el usuario interactúa con el sistema en entornos de alta incertidumbre, como el contexto ecuatoriano.

Investigaciones más recientes han explorado el uso de modelos avanzados como el aprendizaje profundo para mejorar la predicción y la optimización de rutas. De igual manera, propuestas orientadas a la gestión inteligente del transporte de pasajeros evidencian la importancia de integrar datos y algoritmos para mejorar la toma de decisiones [4]. Estas prácticas continúan enfatizando la precisión técnica, en detrimento, llevan a una limitación a su aplicación en contextos en que la experiencia de acceso a la tecnología o la propia comprensión de la misma es determinante.

A nivel local, estudios sobre la utilización de la inteligencia artificial, específicamente aquellos aplicados al transporte público urbano en el cantón Santo Domingo, evidencian que existe mucho potencial para implementar soluciones tecnológicas a la movilidad urbana [5]. Se manifiesta otra brecha entre la disposición de datos técnicos y la forma en la que los ciudadanos interactúan con el transporte. En la práctica, el usuario tiende a utilizar direcciones no exactas, sino referencias como barrios, cooperativas o puntos emblemáticos de la ciudad, lo que dificulta la utilización de herramientas de planificación de rutas tradicionales.

Al analizar este panorama expuesto se manifiesta una limitante clara: las soluciones propuestas están orientadas desde un enfoque técnico, relegando la forma en que el usuario se comunica. Es decir, los sistemas pueden calcular rutas eficientes, pero no están preparados para interpretar el lenguaje cotidiano propio del contexto local de Santo Domingo. Esta desconexión representa un vacío importante tanto en el ámbito tecnológico como en el académico.

En este ámbito, esta investigación propone el desarrollo de una plataforma conversacional asistida por inteligencia artificial que permita optimizar la planificación de rutas del transporte urbano de Santo Domingo. A diferencia de las soluciones existentes, esta propuesta fusiona modelos de grafos con técnicas de procesamiento de lenguaje natural, con el fin de interpretar las consultas en lenguaje natural y adaptarse a las referencias que utilizan los ciudadanos. Así, no solo se pretende optimizar el sistema de transporte sino también humanizar la forma de interacción, con lo cual se contribuye al desarrollo de una movilidad inteligente que se integre a la realidad ecuatoriana.

Formulación del problema o pregunta de investigación

- ¿Qué limitaciones tienen los sistemas de planificación de rutas actuales del transporte público urbano de Santo Domingo para poder ofrecer información que sea accesible, veraz y ajustada a los intereses de las personas que residen en la ciudad?

- ¿Cuál es el impacto de la inexistencia de plataformas de conversación inteligente que faciliten que las personas identifiquen las rutas urbanas mediante el uso de términos del lenguaje cotidiano y referencias geográficas locales?
- ¿Cuáles son las características técnicas y funcionales que debe tener una plataforma web desde la cual se utilice una inteligencia artificial que sea capaz de interpretar preguntas en lenguaje natural y guiar a las rutas más adecuadas dentro del transporte público urbano de Santo Domingo?
- ¿Cómo pueden contribuir el Procesamiento Natural de Lenguaje y los modelos lógicos de grafos a mejorar la planificación de rutas y mejorar la movilidad de las personas que residen en la ciudad?

Objetivos

Objetivo general:

Desarrollar una plataforma web inteligente basada en un agente conversacional para la recomendación de rutas del sistema de transporte público urbano de Santo Domingo, mediante el procesamiento de lenguaje natural y modelos lógicos de grafos para optimizar la movilidad ciudadana.

Objetivos específicos:

1. Identificar las tecnologías y enfoques para la planificación inteligente de rutas del transporte público, analizando el estado del arte y los trabajos relacionados para determinar las técnicas más adecuadas para el desarrollo de la plataforma.
2. Modelar la arquitectura de datos y el motor de procesamiento de lenguaje natural para la gestión inteligente de rutas en el entorno urbano Tsáchila.
3. Desplegar la interfaz web interactiva y validar el desempeño del sistema mediante métricas de precisión y usabilidad con usuarios de la localidad.

Justificación

La presente investigación se apoya en que los sistemas de planificación de transporte público que ofrece Santo Domingo poseen una importante falta de accesibilidad digital. La rigidez de las interfaces basadas en menús estáticos ignora la carga cognitiva que enfrentan los usuarios al navegar nomenclaturas viales oficiales que, a menudo, divergen del léxico urbano cotidiano.

Al proponer una interfaz conversacional, este estudio no solo busca una innovación estética, sino una reingeniería de la interacción humano-computador que permita al ciudadano transitar de una búsqueda paramétrica restrictiva a una consulta semántica fluida, reduciendo los tiempos de respuesta y eliminando las fricciones operativas que hoy limitan la autonomía en la movilidad urbana. Desde la perspectiva técnica, esta investigación se justifica en que es necesario integrar una arquitectura híbrida que contemple el procesamiento de lenguaje natural y la teoría de grafos.

Esta fusión de técnicas es clave para poder avanzar más allá de las limitaciones de las aplicaciones que existen y que permiten modelar la red de transporte y expresar la manera en que se trasladan las personas y los medios de transporte a través de nodos y aristas que difieren mediante pesos que pueden cambiar en el tiempo, para que la planificación de las rutas y de los transbordos no sea simplemente el resultado de una sugerencia lógica sino que sea una solución matemática adecuada. En este sentido surge la necesidad de crear artefactos tecnológicos que permitan acceder a información clara, comprensible y en el momento adecuado acerca del transporte urbano, que faciliten a los usuarios la planificación de los recorridos de forma más eficiente y ajustada a las condiciones que presente el medio. Con el objeto de satisfacer esta necesidad, esta investigación se enfoca en desarrollar una plataforma conversacional que permite mejorar la planificación de rutas mediante técnicas de procesamiento de lenguaje natural y mediante la aplicación de modelos de grafos. La

relevancia de esta propuesta radica en la democratización del acceso a la tecnología para la comunidad de Santo Domingo e los Tsáchilas y sus visitantes.

La implementación de un asistente conversacional que funcione con estas características facilitará a los usuarios la posibilidad de efectuar sus preguntas a través de referencias locales y expresiones cotidianas del medio, como, por ejemplo: El Bombolí, Círculo de los Continentes, etc., sin necesidad de conocer la dirección exacta o el nombre oficial del lugar. Esto facilitará una interacción más natural con la plataforma y mejorará la experiencia de planificación de rutas del transporte público urbano. Esto fomenta una inclusión digital efectiva, en la que el dominio de tecnicismos cartográficos deja de ser una barrera de entrada, permitiendo que sectores históricamente marginados por la brecha tecnológica [6], puedan navegar por la red de transporte con la misma agilidad que un usuario especializado.

El presente trabajo se ajusta a los objetivos de proporcionar acceso a sistemas de transporte, los cuales son seguros, asequibles y accesibles. El hecho de permitir que un asistente inteligente que optimiza los traslados logre una gestión a los aspectos de desplazamiento, mejorando la calidad de vida de los usuarios al eliminar la incertidumbre y motivar al uso del transporte público en detrimento del vehículo privado. De igual manera, transformar la actual red de buses en un sistema fácil de leer y predecible permite una movilidad sostenible que aligera los flujos en el tráfico de las principales arterias de la ciudad, y permite optimizar los flujos de tránsito en puntos neurálgicos como la Terminal Terrestre.

Por último, el trabajo supone un aporte metodológico nuevo porque hace avanzar un modelo de humanización de la interacción técnica, a través de una transformación de datos en bruto a diálogos naturales, un avance cualitativo hacia la movilidad inteligente en el sistema de ciudades intermedias; así, el trabajo investigativo no solo es el resultado de haber resuelto un problema logístico inmediato, sino que también es la base para la integración de inteligencia artificial en el acceso a las personas durante la gestión de servicios públicos en la región [7].

Lo que evidencia este trabajo, por tanto, es que la precisión científica del modelo de grafos y la cercanía del lenguaje natural no son elementos opuestos, sino cimientos complementarios que, en conjunto, contribuyen al bienestar de la comunidad.

Capítulo II

Estado del arte

El estudio de la movilidad urbana asistida por tecnología ha cobrado relevancia en la última década, particularmente en ciudades que buscan consolidarse como "ciudades inteligentes". Aplicaciones comerciales como Moovit y Waze han demostrado la viabilidad de integrar geolocalización, tráfico en tiempo real y planificación de rutas [1]. No obstante, su implementación en ciudades intermedias como Santo Domingo de los Tsáchilas enfrenta obstáculos derivados de la falta de estructuración y actualización de los datos del transporte público, así como de la fuerte dependencia del conocimiento informal que manejan los propios ciudadanos.

Estudios desarrollados en contextos como Medellín y Río de Janeiro han evidenciado que el análisis de datos masivos permite optimizar la gestión del tráfico urbano, aunque también revelan una alta dependencia de infraestructura tecnológica avanzada [2], lo cual limita su replicabilidad en ciudades con menores recursos digitales. Además, se ha documentado que este tipo de herramientas, si bien contribuyen a redistribuir el flujo vehicular, no logran resolver por completo los problemas de congestión y, en ciertos casos, terminan incentivando el uso del vehículo privado [2].

En paralelo, el desarrollo de la inteligencia artificial ha permitido a plataformas globales como Uber avanzar hacia modelos multimodales que combinan distintos medios de transporte en una sola aplicación, optimizando los tiempos de viaje mediante algoritmos de emparejamiento en tiempo real [3]. Estas soluciones de carácter comercial están desarrolladas para contextos con alta digitalización y datos abiertos en los sistemas públicos, lo que permite su implementación; sin embargo, no están diseñadas para adecuarse a las singularidades sociales y operativas de contextos como el ecuatoriano; ante estas limitaciones, el ámbito

académico se ha atrevido a presentar propuestas más adaptativas. Por ejemplo, el prototipo de rutas de transporte público utilizando inteligencia artificial plantea un modelo dinámico basado en datos de demanda y tráfico local, mientras que el planificador de viajes basado en inteligencia artificial automatiza la generación de itinerarios personalizados. Ambas propuestas, sin embargo, priorizan la eficiencia matemática del cálculo de rutas por sobre la manera en que el usuario interactúa con el sistema en contextos de alta incertidumbre.

Se ha investigado la posibilidad de un uso más intensivo de modelos de aprendizaje profundo para llevar a cabo la predicción y la optimización de las trayectorias de una forma más afinada [8], así como también el uso de métodos de optimización orientados a la gestión inteligente del transporte de pasajeros, pero en este caso, teniendo en cuenta la integración de datos y métodos de toma de decisiones [9]. En ambos casos, se presenta la misma limitación: predomina la excelencia técnica sobre la experiencia del usuario, lo que dificulta su adopción en entornos donde el acceso y la comprensión tecnológica juegan un papel predominante.

En el ámbito local, las investigaciones centradas en la inteligencia artificial en el transporte público del cantón Santo Domingo, corroboran que esta tecnología tiene un elevado potencial de aplicación [7], pero muestran a la vez una disparidad entre la disposición de datos técnicos y el tipo de intercambio que se produce entre los ciudadanos, puesto que las personas suelen hacer referencia a barrios, cooperativas o puntos de interés e importancia, en lugar de hacerlo mediante direcciones concretas [10]. En el ámbito local, las investigaciones centradas en la inteligencia artificial en el transporte público del cantón Santo Domingo, corroboran que esta tecnología tiene un elevado potencial de aplicación [7], pero muestran a la vez una disparidad entre la disposición de datos técnicos y el tipo de intercambio que se produce entre los ciudadanos, puesto que las personas suelen hacer referencia a barrios, cooperativas o puntos de interés e importancia, en lugar de hacerlo mediante direcciones concretas [10].

En definitiva, la literatura revisada pone de manifiesto que las soluciones existentes tanto comerciales como académicas se han construido mayoritariamente con una clara orientación hacia una visión técnica y no hacia el lenguaje cotidiano con el que se relacionan los usuarios. Esta desconexión constituye el vacío que la presente investigación busca atender mediante una plataforma conversacional que integra procesamiento de lenguaje natural y modelos de grafos, adaptada a las referencias locales de Santo Domingo.

Marco teórico

Arquitectura de Software

La arquitectura de software es el diseño y desarrollo a alto nivel de un sistema de software. Implica definir los componentes de software, su organización, sus relaciones, patrones, principios que se van a usar, etc. Se puede considerar una rama de la arquitectura de sistemas, que es más amplia. Mientras la arquitectura de sistemas incluye hardware, redes, dispositivos físicos y todo lo necesario para que el sistema global funcione, la arquitectura de software se concentra principalmente en los componentes de software [11]. En general, los diagramas y los modelos de descripción arquitectónica, como las vistas 4+1 de Kruchten y el modelo C4, son representaciones de la arquitectura que facilitan la comprensión de la estructura y el comportamiento del sistema.

Metodología de desarrollo iterativo

El desarrollo iterativo es un proceso de construcción de software en el que el proyecto se divide en ciclos cortos y repetitivos, dentro de los cuales se planifica, diseña, implementa y prueba una parte del sistema antes de avanzar hacia el siguiente ciclo este enfoque permite entregar el software para que sea probado y evaluado por los usuarios al final de cada ciclo [12]. Este modelo surgió como respuesta a las limitaciones del modelo en cascada tradicional, ya que el desarrollo iterativo se basa en conjuntos de tareas agrupadas en breves etapas

repetitivas que permiten que el software se implemente con éxito en diversas metodologías de trabajo [12].

La metodología de desarrollo iterativo permitió que el equipo de desarrollo fuese construyendo el sistema por fases. Se comenzó por las funcionalidades más básicas que incluían características como la autenticación de usuarios o la visualización del mapa para posteriormente ir añadiendo componentes más complejos, como pueden ser el motor de procesamiento de lenguaje natural o el módulo de recomendación de rutas mediante grafos. Este enfoque facilitó la corrección temprana de errores, la validación constante de cada módulo con los requerimientos del proyecto y la adaptación del diseño conforme se profundizaba en el conocimiento del dominio del transporte público de Santo Domingo y sus alrededores.

Arquitectura en Capas

La arquitectura en capas organiza una aplicación en niveles con responsabilidades diferenciadas, como: presentación, lógica de negocio, acceso a datos e infraestructura. Cada capa se comunica con otras mediante interfaces o mecanismos definidos, buscando reducir el acoplamiento y facilitar el mantenimiento del sistema. La dirección de las dependencias puede variar según el enfoque arquitectónico utilizado; en una arquitectura tradicional de N capas suele existir una comunicación secuencial entre niveles, mientras que, en propuestas como la arquitectura hexagonal, de cebolla o Clean Architecture las dependencias se orientan hacia los componentes centrales del dominio [13].

En una arquitectura en capas, todas las capas se colocan de forma horizontal, de tal forma que cada capa solo puede comunicarse con la capa que está inmediatamente por debajo, por lo que, si una capa quiere comunicarse con otras que están mucho más abajo, tendrán que hacerlo mediante la capa que está inmediatamente por debajo. Si la capa de presentación requiere consultar la base de datos, tendrá que solicitar la información a la capa de negocio, la cual, a su vez, la solicitará a la capa de persistencia, la que a su vez, la

consultará a la base de datos, finalmente, la respuesta retornará en el sentido inverso hasta llegar la capa de presentación [14].

Por lo general, la arquitectura en capas se compone de tres o más niveles, según las necesidades de la aplicación. De entre las capas más usadas se encuentran:

Capa de presentación: Encargado de la interacción entre el usuario y el sistema. Controla el flujo de datos de entrada y salida, y ofrece la interfaz mediante la que los usuarios pueden acceder a las funciones de la aplicación.

Capa de aplicación: Implementa las reglas de negocio y controla el procesamiento de la información. En esta capa se ejecutan los procesos necesarios para dar respuesta a las solicitudes formuladas por los usuarios y manejar el flujo de la aplicación.

Capa de datos: Responsable de gestionar el almacenamiento, consulta, actualización y control de la información persistente del sistema. Su propósito es asegurar que se pueda entrar a los datos usados en la aplicación de forma ordenada y segura.

Api Rest

La comunicación entre frontend y backend se realiza mediante una API RESTful utilizando el protocolo HTTP y datos en formato JSON.

Las API se han convertido en la forma más común de interacción entre programas y dispositivos en la informática moderna. Una API es un conjunto de reglas que describe cómo un programa puede conectarse y comunicarse con otro. El medio para establecer la comunicación correspondiente, entre las diferentes aplicaciones actuantes a través de la Internet, lo constituye una API REST, que facilita el intercambio de información, de forma que cada aplicación pueda solicitar información o la prestación de un servicio, para obtener una respuesta. Cada solicitud que se envía incorpora toda la información que sirve para que el servidor la procese, sin que exista dependencia alguna de invocaciones anteriores; y por lo tanto, las aplicaciones podrán funcionar de forma autónoma, resultando muy bien la

estructuración del sistema al poder conseguir la construcción de soluciones más escalables, eficientes y fáciles de mantener. REST se ha desarrollado durante más de dos décadas y es un enfoque muy común para arquitecturas distribuidas y basadas en servicios. Un concepto fundamental para las API REST es el recurso. Un recurso es un objeto con un tipo, datos asociados, relaciones con otros recursos y un conjunto de métodos que operan sobre él. Es muy similar a la idea de objetos en programación, aunque solo se definen unos pocos métodos estándar, como HTTP GET, POST, PUT y DELETE. Los recursos pueden existir por sí mismos o agruparse en colecciones que, a su vez, son recursos [15].

Tecnologías Backend

El backend es el encargado de gestionar la lógica del negocio de la aplicación y la gestión de los datos, siendo también la responsable de comunicarse inmediatamente con las bases de datos y con servicios ajenos totales, su función es recibir las solicitudes provenientes de la capa del frontend, procesarlas conforme a las reglas estipuladas por el sistema y aportar las respuestas correspondientes según lo esperado. Tecnologías como Python, Flask y PostgreSQL, entre otros, suelen utilizarse para el desarrollo del backend, herramientas para desarrollar aplicaciones escalables, eficientes y para poder integrar servicios ajenos, como por ejemplo APIs o modelos de IA.

Python

Python es un lenguaje de programación orientado a objetos de alto nivel y fácil de interpretar con una sintaxis fácil de leer. Ideal para crear prototipos y tareas ad hoc, Python tiene un amplio uso en computación científica, desarrollo web y automatización. Como lenguaje de programación para principiantes de uso general [16].

Gestión de paquetes PIP

PIP es el gestor de paquetes por defecto de Python. Las siglas PIP significan Pip Installs Packages o Pip Installs Python, aunque también es conocido por el nombre de Python

Package Index, ya que está directamente conectado al repositorio pypi.org, que es el repositorio de paquetes más grande del mundo para Python. PIP se instala automáticamente cuando instalas Python en tu servidor, hosting u ordenador, por lo que no necesitas instalarlo por separado. Básicamente, PIP es la herramienta que permite instalar, gestionar, actualizar y desinstalar paquetes o módulos de Python de forma simple [8].

Flask (Micro-Framework)

Flask es un framework para el desarrollo de proyectos web en Python. Muchos lo calificarían dentro del segmento de los microframework, ya que tiene un enfoque minimalista y está orientado para un tipo de trabajos más específicos [9]. Flask permite al programador incluir solamente las bibliotecas y herramientas que el proyecto requiera, añadiendo de este modo mayor flexibilidad en el desarrollo. Por esta razón, está orientado más a la creación de APIs REST, servicios web y microservicios que, a renderizar páginas HTML, como la mayoría de frameworks por ejemplo, como por ejemplo Django.

PostgreSQL

PostgreSQL, es una base de datos de código abierto con una sólida reputación por su confiabilidad, flexibilidad y soporte de estándares técnicos abiertos. A diferencia de otros RDBMS (sistemas de gestión de bases de datos relacionales), PostgreSQL admite tipos de datos relacionales y no relacionales. Esto lo convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles en la actualidad [10].

Brinda una muy buena compatibilidad con SQLAlchemy permitiendo que el backend desarrollado en Python y Flask se pueda integrar sin ningún tipo de problema. Asimismo, gracias a su escalabilidad, el sistema evoluciona y se adapta a futuras implementaciones sin que eso implique que el rendimiento se vea perjudicado. Hoy en día, PostgreSQL almacena a buen recaudo y con un resultado correcto a los usuarios ya registrados, a sus credenciales y, cómo no, a la información que está relacionada con la autenticación del sistema.

SQLAlchemy

SQLAlchemy es un conjunto de herramientas de código abierto construido para el lenguaje de programación Python. Está destinado para el óptimo acceso a bases de datos de forma eficiente y de alto rendimiento. Su función principal es ayudar a las aplicaciones Python a interactuar con bases de datos relacionales, facilitando la creación, actualización y consulta de tablas. Esto convierte a SQLAlchemy en una herramienta poderosa para cualquier desarrollador de Python que trabaje con almacenamiento o recuperación de datos [17].

SQLAlchemy fue utilizado como un ORM por sus siglas en inglés Object Relational Mapping para mejorar la interacción entre el backend desarrollado en Python y la base de datos PostgreSQL mediante clases y objetos, evitando el uso constante de consultas SQL tradicionales. Su implementación permite realizar consultas más seguras, reducir riesgos como Inyección SQL y simplificar la manipulación de tablas y registros dentro del sistema. Además, proporciona una integración eficiente con PostgreSQL, mejorando la organización y mantenibilidad del código backend.

NetworkX

NetworkX es una biblioteca de Python para el estudio de grafos y análisis de redes, publicada como software libre bajo licencia BSD-new, que ofrece clases para grafos y dígrafos, conversión de grafos entre distintos formatos, y funciones para explorar propiedades como adyacencia, grado, diámetro y centralidad [18]. Esta biblioteca es una herramienta que sirve tanto para grafos pequeños como para redes a gran escala, es decir, es una herramienta adecuada a la hora de modelar sistemas de transporte urbano.

Tecnologías Frontend

El frontend del sistema es la correspondiente a la capa visual e interactiva. La capa visual es la responsable de la comunicación directa con el usuario, no solamente tiene cabida la parte estética, sino que también, la interacción con los diferentes componentes que tiene una

aplicación para que pudiéramos necesitar para ello solamente la de poder llegar hasta tu funcionalidad de una manera intuitiva. Para hoy en día para la construcción de interfaces en la web es común que se utilicen tecnologías como React, Vite, React Router DOM y Axios ya que permiten tener herramientas para construir interfaces más dinámicas, gestionar la navegación entre vistas, optimizar el proceso de desarrollo y facilitar la comunicación con servicios o con APIs externas.

React

ReactJS es una de las bibliotecas más demandadas de JavaScript para el desarrollo de aplicaciones web y móviles. Diseñada por Facebook, React contiene una colección de trozos de código de JavaScript reutilizables utilizados para crear interfaces de usuario (UI) denominadas componentes [19]. La propiedad de que la arquitectura se basa en componentes ayuda a construir componentes que son reutilizables e independientes y que a largo plazo propician una mejor organización del código, una mejor reutilización de funcionalidades y un mejor mantenimiento de las aplicaciones, haciendo que el desarrollo de interfaces dinámicas e interactivas sea más fácil.

React hace uso del Virtual DOM para mejorar los cambios en la interfaz, ya que permite que los cambios sean dinámicos e instantáneos en el renderizado de dicha interfaz sin tener que refrescar por completo la página. Es una forma de mejorar drásticamente la experiencia de usuario y el rendimiento en global de una aplicación. React con el uso de Hooks hace que tratar estados, eventos y llamadas a API en componentes funcionales resulte más inmediato para pasar a ser implementado en la conexión con la interfaz y los servicios.

Vite

El Vite es un empaquetador y un servidor de desarrollo que sirve para trabajar con proyectos modernos basados en JavaScript o en frameworks como React, Vue, Svelte, Angular, etc. Su mayor ventaja es la velocidad que permite recargar módulos en caliente y no

reconstruir toda la aplicación para acelerar el proceso de desarrollo y aumentar la productividad [20].

Vite es un entorno de desarrollo del frontend debido a la velocidad de compilación, de optimización de aplicaciones modernas desarrolladas con React. Esto hace posible iniciar el servidor de desarrollo rápidamente, mejora el rendimiento en el transcurso de la programación y hace posible actualizar los cambios efectuados automáticamente a través de Hot Reload sin necesidad de recargar toda la aplicación. El Vite facilita una configuración más light y eficiente, optimiza el ciclo de desarrollo y mejora la productividad dentro del proyecto.

React Router DOM

Es una librería empleada en el desarrollo de aplicaciones en React que ayuda a gestionar la navegación entre cada una de las páginas de una aplicación web sin necesidad de realizar la recarga del navegador. Con un sistema de rutas, React Router DOM hace posible el pase entre páginas y eso mejora la experiencia de los usuarios, haciendo posible una navegación rápida y fluida. También, nos ayuda a estructurar nuestra aplicación, definir rutas públicas y privadas además de controlar el acceso a los distintos módulos de una forma eficaz.

React Router DOM está diseñado para aplicaciones web, proporcionando todas las características de react-router junto con componentes específicos del navegador como <BrowserRouter>, <Link>, y <NavLink>. Estos componentes simplifican el enrutamiento dinámico, lo que lo hace ideal para SPAs que se ejecutan en navegadores [21].

Axios

Axios es una librería JavaScript que puede ejecutarse en el navegador y que nos permite hacer sencillas las operaciones como cliente HTTP, por lo que podremos configurar y realizar solicitudes a un servidor y recibiremos respuestas fáciles de procesar [22]. Axios fue utilizado para gestionar la comunicación entre el frontend y el backend mediante peticiones HTTP hacia las APIs REST del sistema. Su implementación facilita el envío y recepción de

datos en formato JSON, permitiendo consumir los servicios desarrollados en Flask de manera eficiente. Axios ofrece la posibilidad de configurar una serie de interceptores para adjuntar de forma automática los tokens JWT en las peticiones siempre que estas sean autenticadas, lo que significa que se aporta un enfoque más seguro a la hora de hacer uso de la gestión de la autenticación dentro de la aplicación.

Sistema de Información Geográfica

Un Sistema de Información Geográfica (SIG) es una herramienta para trabajar con información georreferenciada. La información georreferenciada es aquella que viene acompañada de una posición geográfica.

Un SIG es un sistema que permite realizar una serie de operaciones:

- Lectura, edición, almacenamiento y gestión, de manera general, de datos espaciales.
- Análisis simples o complejos de datos espaciales. Este análisis puede llevarse a cabo sobre la componente espacial (la localización de cada valor o elemento) como sobre la componente temática (el valor o elemento en sí).
- Generación de resultados tales como mapas, informes, gráficos, etc.

Un Sistema de Información Geográfica es un sistema de información diseñado para trabajar con datos referenciados mediante coordenadas espaciales o geográficas [23]. En otras palabras, un SIG es tanto un sistema de base de datos con capacidades específicas para datos georreferenciados, como un conjunto de operaciones para trabajar con esos datos.

Leaflet

Leaflet es una librería JavaScript que permite crear mapas interactivos de una forma muy sencilla además es de código abierto y es mobile-friendly. Leaflet fue creada en 2011 por Volodymyr Agafonkin, un desarrollador ucraniano que actualmente está sufriendo la guerra en su país y que nos avisa de ello al entrar en la web [24].

Esta biblioteca de JavaScript tiene como principal objetivo, el de ir representando información geográfica de modo dinámico, en el que se pueden visualizar mapas; añadir marcadores de posicionamiento; representar recorridos a través de formatos geoespaciales como por ejemplo GeoJSON, y por último, la posibilidad adicional de poder realizar interacciones tales como el zoom, el movimiento de un mapa o la superposición de capas. Gracias a su ligereza, facilidad de integración y soporte extensivo, Leaflet se ha convertido en una de las bibliotecas más utilizadas en aplicaciones de navegación, sistemas de información geográfica y soluciones basadas en movilidad urbana.

React Leaflet

React Leaflet proporciona enlaces entre React y Leaflet. No reemplaza a Leaflet, sino que lo aprovecha para abstraer las capas de Leaflet como componentes de React. Por lo tanto, puede comportarse de manera diferente a como funcionan otros componentes de React [25].

OpenStreetMap

OpenStreetMap (OSM) es un conjunto de datos cartográficos de gran detalle que se utiliza en muchos productos Garmin® con funciones de cartografía. El OSM es desarrollado y ampliado por cartógrafos comunitarios de todo el mundo [26].

Su función consiste en ofrecer mapas digitales y la información geográfica que permite mostrar calles, avenidas, intereses y otros tipos de elementos del espacio geográfico. Y también ayuda a la visualización cartográfica y permite conectar con la visualización contextual de los mapas, hasta incluso las aplicaciones de navegación, los sistemas de información geográfica o las aplicaciones relacionadas con la movilidad, con bibliotecas como Leaflet para realizar la representación de mapas de forma interactiva.

Fórmula de Haversine

La fórmula de Haversine es una fórmula utilizada para calcular la distancia entre dos puntos de una esfera a partir de sus coordenadas de longitud y latitud, es decir, la distancia

más corta entre dos puntos sobre la superficie de un objeto esférico como la Tierra [27]. A diferencia del cálculo de distancia euclidiana en un plano, esta fórmula considera la curvatura terrestre, lo que la hace más precisa para aplicaciones de geolocalización.

Seguridad y Autenticación

La plataforma implementa mecanismos de seguridad y autenticación para proteger la información de los usuarios y garantizar el acceso seguro a las funcionalidades privadas del sistema.

JWT (JSON Web Token)

JWT (JSON Web Token) es un estándar que forma parte del RFC 7519. En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios. Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro del payload o cuerpo de un mensaje que va firmado digitalmente [28].

Su función: Gestionar la autenticación y autorización de los usuarios de manera segura. Cuando un usuario inicia sesión correctamente, el backend genera un token JWT que es enviado al frontend y almacenado temporalmente en el navegador.

Bcrypt

Bcrypt es un algoritmo de hashing de contraseñas diseñado para protegerlas frente a ataques maliciosos. A diferencia de otros métodos más antiguos como MD5 o SHA1, que son muy rápidos (y por lo tanto vulnerables a ataques de fuerza bruta), bcrypt está pensado para ser de forma intencional, lento y seguro. Este comportamiento hace que descifrar contraseñas sea un proceso mucho más costoso para un atacante [29].

Su funcionamiento es el de mantener la protección de las contraseñas de los usuarios a través de un algoritmo de hash seguro, antes de almacenarlas, en lugar de almacenarlas en texto plano. También permite comprobar las credenciales de los usuarios en el proceso de

autenticación sin que sea necesario conocer la contraseña original, lo que a su vez mejorará la seguridad de las aplicaciones y los sistemas ante un acceso no autorizado y posibles vulnerabilidades.

Herramientas de Desarrollo

Durante el proceso del desarrollo del sistema se han utilizado diferentes herramientas técnicas que han estado orientadas en función de facilitar la programación, la administración del proyecto, el control de versiones, las pruebas de funcionamiento y la gestión de la base de datos. Estas herramientas permitieron mantener una estructura de trabajo organizada, mejorar la productividad del equipo y garantizar un desarrollo más eficiente de la plataforma.

Git

Git es un sistema de control de versiones distribuido, lo cual quiere decir que un clon local del proyecto constituye un repositorio completo para el control de las versiones. Estos repositorios locales, que funcionan completamente, permiten trabajar de manera remota o sin conexión. Los programadores verifican su trabajo en el entorno local y, después de eso, sincronizan la copia local del repositorio con la que está en el servidor. Este paradigma difiere del control de versiones centralizado en el que los clientes deben sincronizar el código con un servidor antes de crear nuevas versiones de código [30].

Su función principal Git es mantener un sistema bajo un control de versiones para gestionar y registrar los cambios realizados durante el desarrollo de un proyecto.

GitHub

GitHub es una plataforma de desarrollo colaborativo que contiene proyectos en la nube el cual se maneja bajo un sistema de control de versiones llamado Git. Ayuda a los desarrolladores a almacenar y administrar el código llevando un registro de cambios. Generalmente el código es abierto, lo que permite realizar proyectos compartidos y mantener el

seguimiento detallado de su progreso. La plataforma GitHub también funciona como red social conectando a los desarrolladores con los usuarios [31].

Su función es ofrecer un lugar para el alojamiento, gestión y colaboración de los proyectos de software usando los repositorios remotos. GitHub permite almacenar el código fuente, disponer de un seguimiento de los cambios, gestionar las versiones del proyecto y colaborar en el trabajo entre varios desarrolladores. También permite integrar herramientas para la gestión de incidencias, revisión del código, gestión de proyectos, etc., convirtiéndose en la plataforma más utilizada para el desarrollo colaborativo y el control de las versiones de las aplicaciones.

Postman

Postman, en sus comienzos, se originó como una extensión que podía usarse con el navegador Chrome de Google. En esencia, nos posibilita hacer solicitudes sencillas para probar APIs REST propias o ajenas. Gracias a los avances tecnológicos, Postman ha evolucionado y ha pasado de ser de una extensión a una aplicación que dispone de herramientas nativas para diversos sistemas operativos como lo son Windows, Mac y Linux [32].

La intención de la misma consistía principalmente en ayudar no sólo al diseño, prueba, documentación y gestión de las APIs en todo su ciclo de desarrollo, sino que también Postman dirige las peticiones HTTP hacia los servicios webs mediante la comprobación de su funcionamiento, la posibilidad de agrupar las peticiones en colecciones, la gestión de los diferentes entornos de ejecución (pruebas, desarrollo, producción), así como en la generación de las especificaciones de las APIs. Por otro lado, Postman puede ayudar en la tarea de trabajo en equipo, pues permite compartir configuraciones, pruebas y recursos entre los miembros de un equipo de desarrollo, ayudando en la calidad y el mantenimiento de éstas.

PgAdmin

PgAdmin es la herramienta de código abierto más empleada para administrar bases de datos PostgreSQL. Está concebida para facilitar todas las labores vinculadas a la gestión de bases de datos, como visualizar datos, crear diagramas, establecer, poner al día o eliminar tablas, mediante una interfaz gráfica intuitiva [33].

Capítulo III

Metodología

Enfoque de investigación

Esta investigación se realizó desde una perspectiva cuantitativa, porque estuvo centrada en el diseño, la implementación y la evaluación de una plataforma conversacional que utiliza inteligencia artificial para la planificación de rutas del transporte urbano público de Santo Domingo. Esta perspectiva permitió la recolección y el análisis de datos objetivos acerca del desempeño del sistema, tomando en cuenta indicadores como los siguientes: la exactitud en el consejo de rutas, el reconocimiento de puntos de origen y destino, la producción de trayectos a través de grafos y la rapidez con que responde la plataforma. Los resultados conseguidos permitieron la evaluación del rendimiento general del sistema desarrollado y comprobar si se alcanzaron las metas propuestas.

Además, se utilizó la escala System Usability Scale (SUS) para medir cuantitativamente la percepción de los usuarios sobre cuán fácil es utilizar la plataforma. Esta herramienta otorgó una evaluación numérica acerca de elementos vinculados con la usabilidad, la interacción con el asistente conversacional y la interpretación de la interfaz. Adicionalmente, se llevaron a cabo exámenes funcionales con el fin de verificar la exactitud de las rutas sugeridas y el adecuado funcionamiento de los módulos de visualización cartográfica, cálculo de rutas y geolocalización, empleando diversos contextos de origen y destino, lo cual proporcionó evidencia para validar la solución propuesta.

Tipo y diseño de investigación

La investigación actual se clasifica como aplicada, ya que su objetivo fue crear y poner en funcionamiento una plataforma de conversación con asistencia de inteligencia artificial para planificar las rutas del transporte público urbano en la ciudad de Santo Domingo. El estudio se enfocó en ofrecer una solución tecnológica a un problema real vinculado con la movilidad

urbana, incorporando varias tecnologías para optimizar la experiencia de los usuarios al usar el transporte público y simplificar la consulta de rutas.

La investigación es de tipo descriptivo, porque tuvo la posibilidad de examinar y detallar cómo opera la plataforma, sus partes y los procedimientos que intervienen en la creación de sugerencias de rutas. Asimismo, se examinó el rendimiento del sistema a través de pruebas funcionales, comprobando que los módulos de procesamiento de consultas, cálculo de rutas, visualización cartográfica y geolocalización se integraban como era debido y que las sugerencias obtenidas eran precisas.

En lo que respecta al diseño metodológico, la metodología de la investigación es no experimental y de tipo transversal, ya que en el estudio las variables no fueron manipuladas y la información se recopiló en un solo periodo temporal, después de terminado el desarrollo de la plataforma. Se realizó la validación del sistema a través de evaluaciones técnicas y la utilización de la escala System Usability Scale (SUS), lo que posibilitó una valoración cuantitativa de la usabilidad y el punto de vista que tienen los usuarios sobre la solución desarrollada.

Análisis de requerimientos

Como primer paso de la creación de la plataforma conversacional se realizó el levantamiento, así como la valoración, de los requerimientos necesarios para que el sistema pudiera funcionar. Este levantamiento, además, llevó a determinar cuáles eran las necesidades de las personas usuarias del transporte público urbano de Santo Domingo y a especificar las funcionalidades que la plataforma debía aportar para poder ayudar en la planificación de rutas a partir de las consultas a un sistema en lenguaje natural. Asimismo, se definieron los requerimientos de calidad del sistema y las tecnologías a utilizar para ponerlo en funcionamiento. Los requerimientos establecidos fueron, por eso, el punto de partida para la

posterior elaboración, desarrollo e integración de la plataforma que constituía la solución propuesta.

Requerimientos funcionales

Los requerimientos funcionales abarcan las principales funciones y capacidades que ha de proporcionar la plataforma para llevar a cabo la interacción entre la aplicación, el usuario y el sistema, del mismo modo como se procesan las consultas y se generan las rutas de transporte. En la Tabla 1 se presentan los requerimientos funcionales definidos durante la etapa de análisis.

Tabla 1 *Requerimientos Funcionales ChatBus*

Código	Requerimiento	Descripción
RF-001	Registro e inicio de sesión	El sistema debe permitir que los usuarios se registren e inicien sesión para acceder a las funcionalidades de la plataforma.
RF-002	Consulta mediante lenguaje natural	El sistema debe permitir que el usuario consulte rutas utilizando lenguaje natural indicando el origen y el destino del recorrido.
RF-003	Obtención de ubicación actual	El sistema tiene que identificar la ubicación geográfica del usuario a través del GPS correspondencias concedidas por el dispositivo.
RF-004	Identificación de lugares	El sistema tiene que determinar automáticamente el origen y el destino de la consulta a partir de inteligencia artificial y de la base de lugares necesaria ya almacenada dentro de la propia plataforma.
RF-005	Cálculo de rutas óptimas	El sistema tiene que calcular la mejor ruta de transporte a partir de grafos y del algoritmo de Dijkstra para los casos de las caminatas y de los transbordos.

RF-006	Visualización de la ruta	El sistema tiene que representar sobre un mapa la mejor ruta de transporte, las caminatas y las líneas de transporte que debe utilizar el usuario.
RF-007	Gestión del historial	El sistema tiene que almacenar todas las conversaciones y permitir consultas sobre el historial de rutas realizadas.
RF-008	Presentación de recomendaciones	El sistema tiene que explicar las líneas de transporte, los puntos de transbordo y las indicaciones necesarias para realizar dicho recorrido.

Requerimientos no funcionales

Los requisitos no funcionales determinan las propiedades de calidad que la plataforma tiene que satisfacer para asegurar un desempeño apropiado, una experiencia satisfactoria para el usuario y la oportunidad de hacer mejoras en el futuro. Los requerimientos no funcionales más relevantes tenidos en cuenta durante el desarrollo del sistema se muestran en la Tabla 2.

Tabla 2 *Requerimientos No Funcionales ChatBus*

Código	Requerimiento	Descripción
RNF-001	Usabilidad	La plataforma debe proporcionar una interfaz intuitiva que facilite la interacción mediante lenguaje natural y un mapa interactivo.
RNF 002	Tiempo de respuesta	El sistema estará preparado para realizar la recomendación de rutas con tiempo suficiente para facilitar que el usuario mecanice la experiencia de usuario de forma continua y fluida.
RNF 003	Disponibilidad	La plataforma permanecerá accesible para atender al usuario cada vez que deba preparar un recorrido.
RNF 004	Compatibilidad	El sistema deberá interoperar de perfección en los navegadores web públicos más habituales y para distintas dimensiones de las pantallas.

RNF 005	Mantenibilidad	La aplicación está organizada en módulos autónomos para facilitar futuras versiones y correcciones.
RNF 006	Escalabilidad	El sistema estará preparado para la incorporación de nuevas rutas, lugares o líneas de transporte sin modificar lo que ahora está en funcionamiento.
RNF 007	Seguridad	El acceso al software de la aplicación se ejecutará mediante autenticación de los usuarios y almacenando de forma segura la información.
RNF 008	Precisión	La plataforma estará preparada para proporcionar recomendaciones de rutas a partir de información geográfica en tiempo real, mediante algoritmos de búsqueda potentes.

Requerimientos tecnológicos

Las herramientas, los lenguajes de programación y los servicios que se emplean en la creación de la plataforma son lo que se conoce como requisitos tecnológicos. Estos recursos posibilitaron la integración de servicios externos, el funcionamiento adecuado de la aplicación en la generación y visualización de rutas, así como en el procesamiento de consultas. La Tabla 3 muestra las tecnologías más importantes utilizadas para desarrollar el proyecto.

Tabla 3 *Requerimientos tecnológicos ChatBus*

Código	Requerimiento	Descripción
RT-001	Frontend	Realización de la interfaz de la web a través de React y Leaflet para el trato del usuario final y visualización del mapa.
RT -002	Backend	Implementación de los servicios de la plataforma a través de Python y el framewok Flask para tratar las peticiones.
RT -003	Base de datos	Almacén de usuarios, conversaciones e historial de las consultas a través de PostgreSQL.
RT -004	Inteligencia artificial	Integración del modelo Google Gemini para la interpretación de las consultas en lenguaje natural así como para

		determinar cuáles son el origen y destino solicitados por el usuario.
RT -005	Cálculo de rutas	Uso de la librería NetworkX para difundir la red de transporte mediante grafos y llevar a cabo el algoritmo de Dijkstra.
RT -006	Servicios geográficos	Integración de OpenStreetMap, Nominatim y OSRM para geocodificar lugares, obtener coordenadas y calcular recorridos para caminar.
RT -007	Control de versiones	Manejo del código fuente por medio de Git y GitHub para llevar el control de los cambios que se van realizando en el desarrollo.
RT -008	Entorno de desarrollo	Uso de Visual Studio Code como entorno de desarrollo y herramientas auxiliares para llevar a cabo la implementación y las pruebas del sistema.

Arquitectura del sistema

La plataforma fue diseñada empleando una arquitectura de capas, para dividir las responsabilidades de cada componente y hacer más fácil la integración, escalado y mantenimiento de nuevas funcionalidades. Desde la interacción del usuario hasta la creación de la ruta ideal y el almacenamiento de datos, cada capa desempeña un papel particular en el proceso de planificación de rutas.

Esta organización permite que los componentes trabajen de manera independiente, favoreciendo la reutilización del código y simplificando futuras modificaciones. Asimismo, la arquitectura integra tecnologías de inteligencia artificial, servicios geográficos y algoritmos de búsqueda de rutas para ofrecer recomendaciones precisas del transporte público urbano del cantón Santo Domingo como podemos observar en la Figura 1.

Figura 1 *Arquitectura general de la plataforma conversacional*



Capa de presentación

La interface de web que hace uso el usuario final es la correspondiente a la capa de presentación. Se utilizó la librería React como la base para la implementación de la aplicación, lo que dio lugar a una interfaz gráfica interactiva y fácil de navegar. En esta capa el usuario tiene la facultad de autenticarse, observar el mapa urbano, repasar las conversaciones anteriores y consultar rutas usando lenguaje natural.

Esta capa emplea la API de Geolocalización del navegador para determinar la ubicación en tiempo real del aparato y representa gráficamente las rutas calculadas a través de OpenStreetMap y Leaflet. De este modo, se visualiza el trayecto del autobús, los tramos iniciales y finales a pie, los lugares de partida y llegada y las paradas que se incluyen durante el recorrido.

Capa de aplicación

Utilizando Python y el marco Flask, la capa de aplicación se estableció como el núcleo del backend del sistema. Su función principal es gestionar la autenticación de usuarios, recibir las solicitudes que llegan desde la interfaz web y organizar el flujo de procesamiento necesario para dar respuesta a cada pregunta que haga un usuario.

Esta capa tiene la función de actuar como un intermediario entre la interfaz, los servicios geográficos, los de inteligencia artificial y el módulo de cálculo de rutas. Se encarga de organizar el intercambio de datos y asegurar que cada componente intervenga solo en la tarea para la cual fue creado.

Capa de inteligencia artificial

La API de Google Gemini, que tiene como objetivo entender las preguntas formuladas por el usuario en lenguaje natural, es utilizada por la capa de inteligencia artificial para incorporar un modelo de lenguaje grande (LLM). En vez de necesitar nombres precisos de ubicaciones, el modelo detecta automáticamente tanto la procedencia como el destino requerido, aun si el usuario utiliza expresiones cotidianas, abreviaturas o distintas maneras de nombrar un mismo lugar.

Más tarde, los datos adquiridos se normalizan y se mandan al sistema para identificar las coordenadas geográficas pertinentes. Esto posibilita que el resto del procedimiento de planificación de rutas pueda llevarse a cabo automáticamente sin necesidad de intervención adicional por parte del usuario.

Capa de servicios geográficos

La capa de servicios geográficos combina varias fuentes de información espacial que tienen la responsabilidad de determinar ubicaciones, conseguir coordenadas y estimar las rutas reales para peatones. En primer término, el sistema revisa una base de datos local que tiene los puntos de interés más importantes del cantón Santo Domingo. Se emplea el servicio

Nominatim de OpenStreetMap para efectuar la geocodificación y conseguir las coordenadas si una ubicación no se encuentra en esta base.

Tras determinar las ubicaciones, el sistema utiliza el servicio OSRM (Open Source Routing Machine) para calcular los recorridos a pie reales a lo largo de la red vial existente, en lugar de limitarse a emplear solamente distancias en línea recta. Este grupo de servicios opera como una capa especializada en el procesamiento geográfico, lo que posibilita que la plataforma cuente con información espacial precisa para calcular las rutas

Capa de procesamiento de rutas

Este componente es el que se encarga de establecer cuál es la opción de transporte público más apropiada para el usuario. Para ello, se establece un grafo dirigido con la funcionalidad NetworkX en el que cada estación se encuentra representada en forma de nodo y cada conexión representa las líneas de transporte bien diferenciadas que hay en la ciudad.

Con el objetivo de seleccionar la ruta más idónea, se utiliza en este grafo el algoritmo de Dijkstra añadiendo penalizaciones por trayectos en contra del sentido, por caminatas y por transbordos. Por lo tanto, nuestro sistema genera los segmentos de la ruta, excepcionalmente las líneas de transporte que se tienen que seguir, estima las caminatas precisas y genera el GeoJSON que posteriormente se utilizará para que se muestre en el mapa interactivo.

Capa de datos

Se guardan las líneas de transporte, los lugares de interés y las intersecciones, recursos geográficos que se utilizan en la construcción de rutas, mediante los archivos de tipo GeoJSON. El sistema de gestión de las conversaciones que se realizan, los usuarios que se encuentran registrados y el historial de consultas se administra a través de PostgreSQL.

El separar esta capa de las demás nos permite gestionar de una forma más independiente la información y facilita, a partir de este momento, el mantenimiento de los

cambios en la aplicación, el añadir nuevas rutas de transporte, y el mantenimiento de la plataforma de una forma global.

Flujo de procesamiento de una consulta

El flujo que sigue el procesamiento de una consulta se inicia cuando un usuario ingresa lo que es una solicitud mediante un lenguaje natural desde la interfaz web en donde daría su origen y destino además teniendo la posibilidad de activar la ubicación actual, la solicitud es enviada al servidor desarrollado con Flask el cual se encarga de procesar la información y comunicarse con los demás servicios.

Como primera parte el backend se encarga de enviar la consulta al modelo de lenguaje Google Gemini el cual es el encargado de identificar el origen y el destino solicitado por el usuario. Y una vez hecho eso el sistema verifica si dichas ubicaciones existen dentro de la base local de lugares e intersecciones y si tal caso no las encuentra se realiza una consulta al servicio de Nominatim de OpenStreetMap para obtener las coordenadas geográficas correspondiente.

Después de obtener las coordenadas de origen y destino, el sistema se encarga de identificar las paradas de transporte cercanas y calcula a su vez la caminata que debe realizar el usuario y esto gracias a un servicio OSRM. Con esta información se construye el conjunto de candidatos los cuales serán evaluados mediante el algoritmo de Dijkstra sobre el grafo del sistema el cual considera la distancia recorrida, el número de transbordo, el sentido de circulación y la distancia de caminata.

Y finalmente el sistema selecciona la ruta con menor costo y genera la representación geográfica en formato GeoJSON, se almacena la consulta y la respuesta en la base de datos de PostgreSQL y envía toda la información al usuario en donde la interfaz representara el recorrido sobre el mapa utilizando lo que es Leaflet y OpenStreetMap y gracias a ello se podrá

mostrar el origen, destino, caminata, línea de transbordo y paradas involucradas y todo este flujo se puede observar en la Figura 2.

Figura 2 Flujo de procesamiento de una consulta



Metodología de desarrollo iterativa

Para el desarrollo de la plataforma conversacional orientada a la planificación de rutas del transporte público urbano de Santo Domingo se empleó una metodología de desarrollo iterativa. Esta metodología posibilitó la construcción del sistema por medio de ciclos continuos de análisis, diseño, implementación, pruebas y optimización. En cada iteración se lograba una versión funcional que luego se evaluaba y mejoraba.

La selección de esta metodología respondió a la necesidad de integrar componentes tecnológicos con comportamientos diferentes, entre ellos la interfaz web, el backend, el procesamiento de lenguaje natural, los datos geográficos, el algoritmo de Dijkstra, los servicios

de geocodificación y la visualización cartográfica. A causa de esta necesidad, fue imprescindible comprobar de forma continua la comunicación entre los módulos, detectar fallos y adaptar poco a poco el funcionamiento de la plataforma.

Los resultados de la etapa anterior fueron el punto de partida para cada iteración. Se llevaron a cabo cambios antes de pasar a la versión siguiente cuando se encontraron problemas en las pruebas, como una selección de paradas que no eran convenientes, una inadecuada interpretación de lugares, la creación de itinerarios largos o un mapa con rutas incompletas.

Figura 3 Ciclo de vida de la metodología iterativa



Como se observa en la Figura 3, cada iteración comprendió las etapas de análisis, diseño, implementación, pruebas, evaluación y mejoras. Los resultados obtenidos permitieron identificar oportunidades de mejora que fueron incorporadas en el siguiente ciclo de desarrollo hasta alcanzar una versión funcional de la plataforma.

Iteración 1 Análisis, planificación y preparación de la información

En la primera iteración se estableció que el sistema tenía que posibilitar a los usuarios la consulta de rutas a través del lenguaje natural, el uso de su ubicación actual, la identificación de lugares emblemáticos, el cálculo de recorridos en transporte público y la visualización de los resultados con un mapa interactivo.

Durante esta etapa se definieron los requerimientos funcionales, no funcionales y tecnológicos del sistema. De igual forma, se creó la arquitectura inicial de la solución, diferenciando entre la capa de presentación, la lógica del backend, el procesamiento de consultas, el motor de rutas, los servicios geográficos y la capa de almacenamiento.

A la par, se reunió la información geográfica de las líneas de transporte urbano. Se revisaron y organizaron los recorridos disponibles para que luego se usaran en la creación del grafo. Además, se determinaron las zonas de interés, los puntos de intersección, las áreas y los lugares que son utilizados a menudo por los usuarios.

Las actividades principales de esta iteración fueron:

- Análisis de la problemática de movilidad urbana.
- Identificación de los usuarios potenciales.
- Definición de requerimientos del sistema.
- Selección de tecnologías.
- Diseño inicial de la arquitectura.
- Recopilación de recorridos de transporte.
- Organización de lugares e intersecciones.
- Conversión y revisión de información geográfica.

Como resultado se obtuvo la estructura inicial del proyecto, la arquitectura propuesta y el conjunto base de datos geográficos necesarios para comenzar la implementación, como podemos observar la Tabla 4 se presenta las actividades realizadas en esta primera iteración.

Tabla 4 Actividades de la primera iteración

Elemento	Descripción
Objetivo	Analizar el problema y definir la solución tecnológica
Entradas	Información de rutas, necesidades de movilidad y objetivos
Actividades	Requerimientos, arquitectura y preparación de datos
Resultado	Diseño inicial y conjunto de datos geográficos organizado
Validación	Revisión de estructura, rutas, lugares y tecnologías seleccionadas

Iteración 2 Desarrollo de la infraestructura base

Durante la segunda iteración se basó en la construcción de la infraestructura principal de la plataforma. Aquí el backend fue desarrollado con Python y Flask, organizando la aplicación mediante rutas, modelos, servicios y componentes de acceso a datos y gracias a esta separación permitió que cada módulo cumpliera una responsabilidad específica y con esto nos facilitó las posteriores modificaciones del sistema

Se realizó la configuración del PostgreSQL como sistema de gestión de base de datos para poder almacenar usuarios, conversaciones e historial de consultas. Se desarrollaron funciones de registro, inicio de sesión, consulta de información de usuario y gestión de conversaciones. A su vez se implementó mecanismo para crear, listar y eliminar conversaciones asociadas a cada usuario.

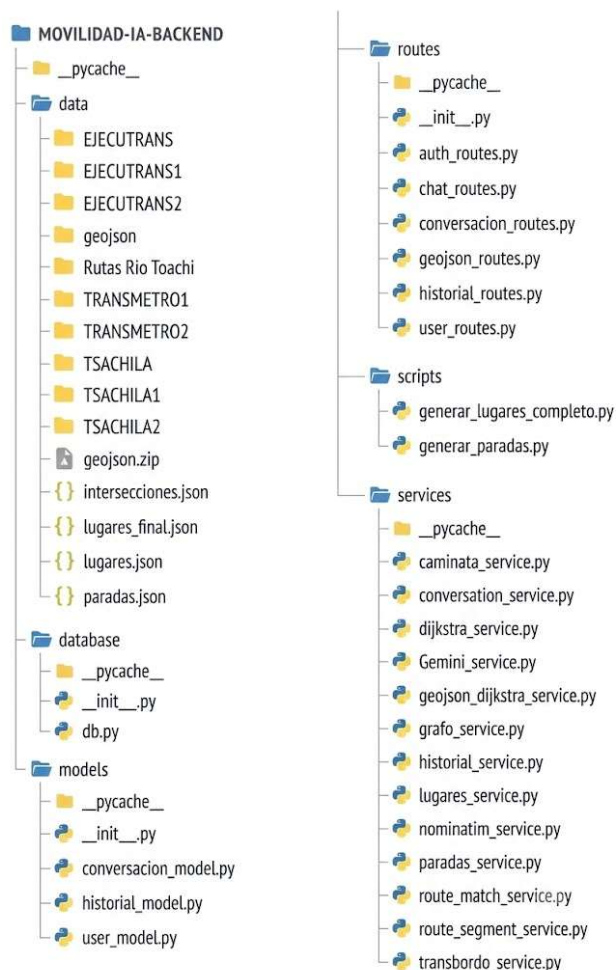
En el frontend se estableció la estructura inicial mediante React en donde se desarrollaron las interfaces de autenticación, página principal, chat conversacional y visualización del mapa, además se configuró la comunicación con el backend mediante solicitudes HTTP y se incorporó el almacenamiento de información necesaria para mantener la sesión del usuario.

- Las actividades más importantes desarrolladas durante esta iteración fueron las siguientes:

- Configuración del proyecto con Flask.
- Organización del backend a través de las rutas, los modelos y los servicios.
- Configuración de PostgreSQL.
- Implementación del registro e inicio de sesión.
- Desarrollo de la gestión de usuarios.
- Creación y recuperación de conversaciones.
- Almacenamiento en el historial de consultas.
- Configuración inicial del frontend con React.
- Comunicación entre frontend y backend.
- Pruebas de los endpoints mediante solicitudes controladas.

Esta segunda iteración produjo una versión funcional de la infraestructura base de la plataforma, que incorporó los siguientes elementos: el frontend realizado con React, el backend desarrollado con Flask, la base de datos PostgreSQL y los servicios responsables de gestionar las conversaciones y autenticar. Esta estructura posibilitó la división clara de responsabilidades entre cada parte del sistema, lo que simplificó su mantenimiento y la integración de nuevas funcionalidades en las iteraciones posteriores. La organización de los módulos, rutas, modelos y servicios que dan forma a la lógica de la aplicación se muestra en la Figura 4, la cual ilustra la estructura del backend implementado en esta fase.

Figura 4 *Arquitectura del backend implementado con Flask*



Iteración 3 Integración del procesamiento conversacional

La tercera iteración tuvo como propósito permitir que el usuario pueda realizar consultas mediante lenguaje natural, para ello se integro el modelo de lenguaje Google Gemini el cual fue encargado de analizar el mensaje enviado e identificar los datos relevantes para el recorrido como lo es el origen y el destino

El modelo de lenguaje no fue utilizado para calcular directamente las rutas, su funcionalidad se limitó a interpretar la consulta y convertirla en una estructura comprensible para los demás servicios como por ejemplo, ante una solicitud como “Estoy en el shopping y

quiero ir hasta la bengala”, el sistema debe obtener el origen que vendría a ser “el shopping” y el destino que vendría a ser “la bengala”.

Una vez que se extrae el origen y el destino, el backend procesa la información obtenida mediante un servicio local de búsqueda. Este servicio lo que hace es primero revisar los archivos de lugares_final.json e interacciones.json que se tiene registrados en el sistema y para su mejora en el reconocimiento se incorporaron nombres alternativos, alias y procesos de normalización con el fin de eliminar diferencias causadas por mayúsculas, tildes o expresiones adicionales.

Con el fin de mantener el funcionamiento si en tal caso el lugar que se solicitó no se encuentra en los archivos locales, se implementó una consulta a Nominatim que es un servicio de geocodificación basado en OpenStreetMap. De esta manera, el sistema puede obtener las coordenadas geográficas de ubicaciones no registrada previamente.

Las actividades principales de esta iteración fueron:

- Integración de la API de Google Gemini.
- Diseño de instrucciones para extraer origen y destino.
- Limpieza de expresiones utilizadas por los usuarios.
- Normalización de nombres de lugares.
- Creación de alias para ubicaciones conocidas.
- Integración de lugares_final.json.
- Integración de intersecciones.json.
- Consulta alternativa mediante Nominatim.
- Validación de coordenadas obtenidas.
- Pruebas con distintas formas de expresar una misma consulta.

Al finalizar esta iteración el usuario fue capaz de enviar una consulta mediante lenguaje natural para después ser procesada mediante el modelo de lenguaje de Google Gemini para

poder analizar lo que es el origen y destino y así convertirla en una estructura entendible para los servicios como podemos ver en la Tabla 5 cómo funciona la interpretación de la consulta.

Tabla 5 *Ejemplos de interpretación de consultas*

Consulta del usuario	Origen identificado	Destino identificado
Quiero ir desde el Terminal hasta el Shopping	Terminal Terrestre	Paseo Shopping
Quiero ir al Portón	Ubicación actual	Portón del Consejo Provincial
Estoy en Bombolí y quiero ir a al Portón	Santuario Bombolí	Portón del Consejo Provincial
Estoy en el shopping y quiero ir a la bengala	Paseo Shopping	Camino a la Bengala

Iteración 4 Construcción y optimización del motor de rutas

La cuarta iteración se concentró más en el desarrollo del componente que se encarga de calcular las rutas del transporte público. Los recorridos geográficos de las líneas fueron procesados para generar una estructura de paradas organizada por línea y sentido es decir cada parada contiene un nombre, una latitud, una longitud y la ruta de transporte a la que pertenece.

Con esta información se procedió con la construcción del grafo dirigido mediante NetworkX. En el grafo, cada parada representa un nodo y las conexiones consecutivas de una línea representa a las aristas, el peso de cada arista se determinó inicialmente mediante la distancia geográfica calculada con la fórmula de haversine.

También se incorporó conexiones de transbordo entre paradas pertenecientes a líneas diferentes que se encontraban dentro de un rango de distancia definida, a su vez estas conexiones se les asignó una penalización para así evitar que el algoritmo recomiende cambios de líneas de buses innecesarios. El algoritmo de Dijkstra fue utilizado para buscar caminos entre diferentes paradas candidatas cercanas a lo que es el origen y destino. El sistema no se

limitó a solo seleccionar la parada más cercana en línea recta. Si no que evaluó varias alternativas para así evitar recorridos poco convenientes.

Durante las pruebas realizadas se identificaron inconvenientes como

- Selección de la parada cercana pero difícil de alcanzar caminando
- Rutas que se excedía con los transbordos
- Se seleccionaba líneas correctas pero con el sentido contrario
- Preferencia con rutas cercanas pero nada eficiente

Para poder solucionar estos inconvenientes se realizaron ajustes sucesivos en algunos aspectos como lo fueron:

- Cantidad de paradas candidatas
- Se estableció un radio máximo de búsqueda
- Se realizó penalización por transbordos
- Se realizó una evaluación de distancia real caminando
- Se optimizó la ejecución del Dijkstra

Una vez realizado estos ajustes esta fase permitió que el motor de planificación de rutas capaz de evaluar múltiples alternativas antes de seleccionar el recorrido más adecuado para el usuario. Con estos ajustes el algoritmo dejó de basarse solo en las paradas más cercanas en línea recta y comenzó a considerar de manera conjunta diferentes criterios como lo son la distancia del recorrido, la caminata inicial y final, la cantidad de trasbordos, el sentido de la línea. Los criterios que se utilizó para la evaluación y selección se pueden observar en la Tabla 6.

Tabla 6 *Criterio de evaluación*

Criterio	Finalidad
Distancia del recorrido	Evitar trayectos excesivamente largos
Caminata inicial	Seleccionar una parada accesible desde el origen

Caminata final	Reducir la distancia entre la última parada y el destino
Transbordos	Evitar cambios de línea innecesarios
Sentido de la línea	Seleccionar el recorrido correspondiente

Iteración 5 Integración de servicios geográficos y visualización

Como quinta iteración se basó más en representar el resultado del motor de rutas dentro de la interfaz web y para lograr eso se utilizó Leaflet como biblioteca de mapas y OpenStreetMap como fuente para mostrar en el mapa, gracias a esto en el mapa se puede mostrar el origen, el destino, las paradas, los segmentos correspondientes de cada línea y la caminata.

Para el uso de la ubicación actual se utilizó una API de geolocalización del navegador, cuando el usuario autoriza el acceso, sus coordenadas son enviadas al backend como punto inicial y solamente debería indicar el destino a cuál ir, caso contrario si no autoriza el usuario deberá indicar su origen y su destino .

Para calcular la caminata se utilizó OSRM (Open Source Routing Machine). Inicialmente se consideraba la distancia geográfica entre el usuario y una parada, pero esta aproximación podía seleccionar puntos separados por vía sin acceso peatonal por ese motivo se incorporó para que siguiera una red vial disponible.

El resultado de Dijkstra fue convertido a GeoJSON. Cada segmento fue guardado como una entidad geográfica que incluye propiedades asociadas a la línea, así como al comienzo, al final y a la clase de recorrido. Se mandó esta estructura al frontend con el fin de trazar el recorrido utilizando Leaflet.

Las actividades principales fueron:

- Integración de Leaflet.
- Incorporación de OpenStreetMap.
- Obtención de ubicación mediante GPS.

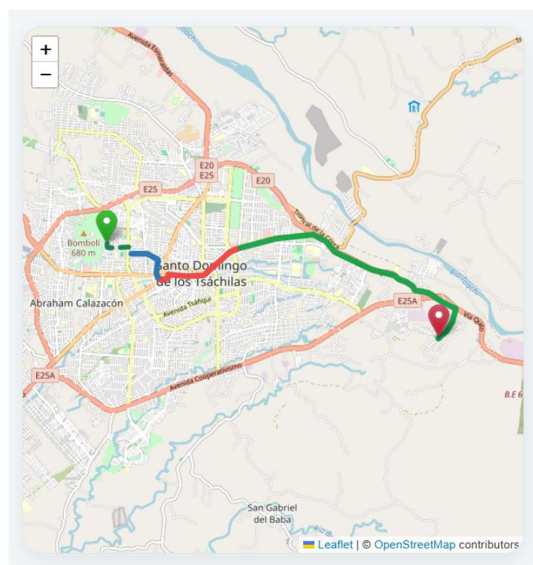
- Cálculo de caminatas con OSRM.
- Generación de GeoJSON.
- Representación de líneas y segmentos.
- Visualización de marcadores.
- Presentación de instrucciones textuales.
- Integración del historial con el mapa.
- Pruebas de adaptación de la interfaz.

Como resultado se obtuvo una plataforma capaz de mostrar de forma gráfica y textual una recomendación completa de movilidad como podemos ver en la Tabla 7 los elementos que muestra en el mapa y en la Figura 5 la visualización en el mapa.

Tabla 7 *Elementos visibles en el mapa*

Elemento	Representación
Origen	Marcador Inicial
Destino	Marcador final
Ruta del autobús	Línea correspondiente al recorrido
Caminata inicial	Trayecto desde el usuario hasta la primera parada
Caminata final	Trayecto desde la última parada hasta el destino
Transbordos	Puntos de cambio entre líneas
Paradas	Marcadores asociados al recorrido

Figura 5 *Visualización del recorrido*



Iteración 6 Pruebas, corrección y refinamiento

La sexta iteración se basó básicamente en la ejecución de pruebas integrales sobre la plataforma, se utilizaron distintos escenarios de origen y destino para verificar la interpretación de las consultas, la búsqueda de lugares, el funcionamiento del grafo, la selección de líneas, los transbordos y la visualización final.

Las pruebas permitieron detectar errores que no eran visible al evaluar los componentes de manera independientes, entre ellos se encontraron coincidencias incorrectas entre nombres de lugares, rutas que utilizaban un sentido poco conveniente, caminatas mayores a las esperadas y líneas que no ingresaban al destino seleccionado, cada error fue analizado para poder identificar al componente responsable.

Dependiendo al responsable se realizaba ajustes, cuando el problema era en la interpretación de lugares se ajustó la normalización y el sistema de coincidencias, si se relacionaba con la selección de rutas se modificaron los candidatos o la penalización y si el problema era en la representación visual se verificaba la generación y lectura del GeoJSON.

Los casos de prueba abarcaron rutas directas, itinerarios con transbordos, consultas que utilizan la ubicación actual, lugares registrados en el sistema localmente, intersecciones y

ubicaciones adquiridas a través de Nominatim. La recuperación de rutas que ya se habían consultado y el almacenamiento de las conversaciones también fueron verificados.

Como resultado de esta iteración se obtuvo una versión más estable y adecuada para ser sometida a la validación técnica y a la evaluación de usabilidad con usuarios de la localidad.

Población y muestra

La investigación tuvo como población objetivo a los usuarios del transporte público urbano del cantón Santo Domingo, debido a que constituyen los principales beneficiarios de la plataforma conversacional desarrollada para la planificación de rutas. Este grupo está conformado por ciudadanos que utilizan diariamente las diferentes líneas de transporte público para desplazarse hacia sus lugares de trabajo, estudio o actividades personales, siendo los usuarios potenciales del sistema propuesto.

De acuerdo con información publicada por Zaracay televisión, el Instituto Nacional de Estadística y Censos INEC presentó los resultados del Censo 2023 en Santo Domingo de los Tsáchilas. En esta provincia hay 492 969 habitantes. 441.583 en el cantón Santo Domingo, cifra la cual nos sirve como referencia para la población del estudio [34].

Considerando el tamaño de la población y las limitaciones de tiempo y recursos para la ejecución de la investigación, se empleó un muestreo no probabilístico por conveniencia. Según Hernández Sampieri et al., este tipo de muestreo permite seleccionar a los participantes de acuerdo con las características y necesidades de la investigación, priorizando aquellos casos que aporten información relevante para el estudio, sin pretender que sean estadísticamente representativos de toda la población [35].

La muestra estuvo compuesta por 15 participantes de la investigación, los cuales interactuaron con la plataforma conversacional llevando a cabo consultas relativas a la planificación de itinerarios de transporte urbano dentro de la ciudad de Santo Domingo.

Posteriormente a las consultas, los participantes respondieron un cuestionario apoyado en la System Usability Scale (SUS), providencial para evaluar los resultados relacionados con la percepción general sobre la usabilidad, la facilidad de aprendizaje y la satisfacción respecto al sistema desarrollado.

Técnicas e instrumentos

Para la recolección de información se utilizaron encuestas y observación sistemática para la recolección de datos, métodos que posibilitaron valorar el desempeño técnico de la plataforma y el punto de vista de los usuarios acerca de su facilidad para usarla al planificar rutas del transporte público urbano.

Para llevar a cabo la evaluación funcional de la plataforma, se empleó el método de observación sistemática utilizando pruebas de caja negra, las cuales tienen como objetivo comprobar que el sistema funcione adecuadamente sin examinar su estructura interna. Se empleó también una ficha de registro explícito que sirvió para ir anotando todos y cada uno de los resultados que se iban obteniendo de las pruebas que se llevaban a cabo con relación al reconocimiento de consultas de lenguaje natural, la obtención de origen y de destino, el reconocimiento de los sitios, el cálculo de rutas con la técnica de grafos, la elaboración de rutas a pie, la representación en mapas y el tiempo del sistema en respondernos. De igual manera se llevó a revisión la satisfacción de los requerimientos funcionales establecidos en el momento del desarrollo.

Se realizaron también pruebas de funcionamiento de los componentes tecnológicos más decisivos de la plataforma. Se verificó la correcta integración del modelo de lenguaje Google Gemini, la geocodificación de Nominatim, el servicio OSRM (OpenStreetMap Routing Machine) y del algoritmo de Dijkstra para la planificación de rutas. Mediante estas pruebas se ha podido verificar que se tienen que procesar correctamente las consultas y que sobre la información geográfica existente se tienen que obtener rutas coherentes.

La técnica de encuesta se utilizó para determinar cómo los usuarios perciben la facilidad de uso de la plataforma. Se empleó el cuestionario System Usability Scale (SUS), creado por Brooke en 1986, el cual es muy conocido para evaluar la usabilidad de sistemas interactivos [36]. El cuestionario se compuso de diez preguntas cerradas organizadas con una escala de Likert de cinco niveles, lo que posibilitó la evaluación de elementos vinculados a la sencillez de uso, la facilidad para aprender, la confianza al interactuar y el grado de satisfacción general que los usuarios tenían sobre la plataforma creada.

La encuesta se llevó a cabo digitalmente a través de Google Forms, justo después de que los participantes utilizaran la plataforma y formularan preguntas vinculadas con la planificación de rutas del transporte público en las ciudades. La información que se recopiló fue empleada solamente para fines académicos y fue manejada con principios de anonimato y confidencialidad.

Validación y confiabilidad del instrumento

Con el propósito de garantizar la calidad de los instrumentos empleados en la investigación, se aplicaron procedimientos de validación acordes con la naturaleza de cada uno de ellos. Se creó una ficha de observación con el fin de evaluar el desempeño de la plataforma conversacional, dicha ficha fue desarrollada conforme a los objetivos particulares del estudio y está destinada a registrar los resultados logrados en el transcurso de las pruebas funcionales. La ficha incluye indicadores que se vinculan con la identificación de consultas mediante lenguaje natural, la determinación de origen y destino, el hallazgo de ubicaciones, el cálculo de rutas, la visualización del trayecto en el mapa y el tiempo que tarda en reaccionar el sistema. Gracias a esta herramienta se pudo documentar de manera sistemática cómo la plataforma se comportó en cada uno de los casos de prueba que se llevaron a cabo.

El investigador principal, que elaboró la ficha de observación, llevó a cabo un proceso de validación de contenido, con un juicio de expertos. El juicio de expertos contempló la

participación de un grupo de expertos en metodología de la investigación, en inteligencia artificial e ingeniería de software. Los expertos valoraron, la claridad, la pertinencia, la coherencia y la suficiencia de los indicadores que se habían fijado, realizando observaciones que sirvieron para llevar a cabo las modificaciones pertinentes antes de su aceptación. Las observaciones realizadas permitieron verificar la pertinencia de los indicadores definidos para la evaluación del funcionamiento de la plataforma.

Para la evaluación de la usabilidad, se utilizó el System Usability Scale (SUS), una herramienta estandarizada que fue creada por Brooke en 1986 y que tiene un uso extendido para evaluar la percepción de usabilidad de sistemas interactivos [36]. Como el cuestionario es la versión original del instrumento y no se le hicieron cambios, no se requirió un procedimiento extra de validación de contenido. Además, el cuestionario SUS muestra una consistencia interna apropiada, con un coeficiente Alfa de Cronbach mayor a 0,80. Por lo tanto, es una herramienta confiable para valorar la usabilidad de sistemas y aplicaciones interactivas.

Consideraciones éticas

La investigación actual se llevó a cabo de acuerdo con los principios éticos que guían la investigación científica, asegurando en todo momento el respeto por los derechos, la privacidad y la participación voluntaria de las personas implicadas en el proceso de evaluación de la plataforma conversacional [37].

Consentimiento informado: Se les comunicó a los participantes el objetivo de la investigación y la finalidad de la encuesta, señalando que su participación era totalmente voluntaria y que podían abandonar el proceso en cualquier momento sin sufrir ningún tipo de repercusión. De igual manera, se les informó antes de contestar el cuestionario que los datos recogidos solo serían empleados para fines académicos e investigativos.

Confidencialidad y privacidad: Los datos que se adquirieron a través del cuestionario de la System Usability Scale (SUS) y de la ficha de observación se manejaron respetando el

anonimato y la confidencialidad, para impedir que se recojan datos personales que permitan identificar a los participantes. Los resultados se examinaron de forma agregada y solo se utilizaron para valorar el desempeño y la facilidad de uso de la plataforma creada.

No maleficencia y beneficencia: Por último, se hizo un esfuerzo para que las actividades de evaluación no implicaran peligros a nivel físico, psicológico o social para los participantes durante la realización del estudio. Se empleó la plataforma en un contexto controlado y se guardaron los datos adquiridos de manera segura, asegurando su integridad y aplicación responsable según los principios éticos de la investigación científica.

Capítulo IV

El presente capítulo expone el análisis y la discusión de los hallazgos logrados a lo largo del desarrollo de una plataforma conversacional asistida por inteligencia artificial para organizar las rutas del transporte público urbano en Santo Domingo se exponen en este capítulo. En esta fase se muestran los resultados que provienen de la aplicación del sistema, de la integración de sus elementos tecnológicos y de las pruebas efectuadas para corroborar su buen desempeño.

Los resultados se analizan en base a los objetivos específicos fijados en la investigación, lo que permite estudiar el grado en el que se han conseguido los requerimientos tecnológicos y funcionales fijados a lo largo de la metodología. Además se recoge el estado de los resultados obtenidos en la generación y procesamiento de las consultas mediante el lenguaje natural; en la evaluación de la usabilidad de la plataforma mediante la escala de System Usability Scale (SUS), así como el cálculo de rutas con grafos y el correspondiente mapa que permite la representación de las rutas.

Análisis y discusión de los resultados

Desarrollo de la plataforma conversacional

La plataforma conversacional basada en web con la que se planifican rutas urbanas para el uso de transporte público en el Cantón Santo Domingo surgió a partir de las acciones generadas en el proceso de desarrollo de este escenario que precede. El desarrollo de la solución contemplaba que un usuario tenga la posibilidad de buscar rutas a través del lenguaje natural; con esto se quería evitar que el usuario tuviera que conocer las líneas existentes o cuáles eran las paradas de transporte. Para lograr esto, se integraron varios componentes de software que trabajan en conjunto para interpretar la consulta, encontrar las ubicaciones deseadas, calcular la mejor ruta disponible y mostrar gráficamente la ruta en un mapa interactivo.

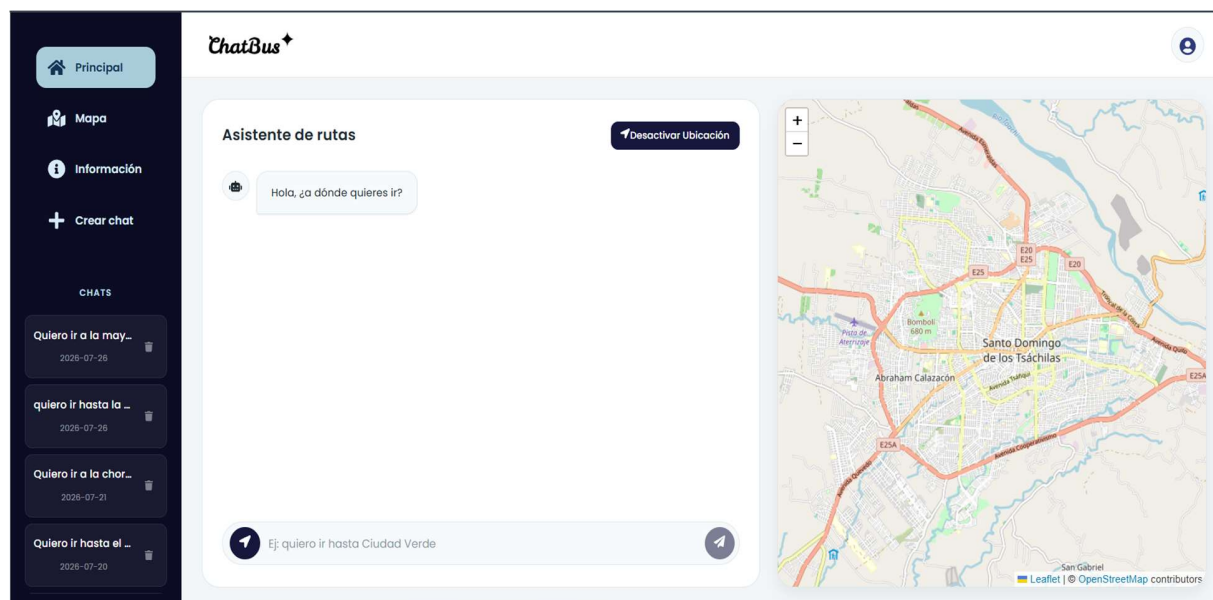
La plataforma fue desarrollada como aplicación web compuesta por un frontend y un backend que trabaja de forma integrada. El frontend es el encargado de proporcionar la interfaz con la que el usuario va a ser capaz de interactuar, mientras que el backend es el encargado de administrar el procesamiento de las consultas, la comunicación con los diferentes servicios y la generación de rutas recomendada.

La interfaz principal fue diseñada con el propósito de facilitar la interacción del usuario con la plataforma, incorporando un mapa interactivo, un asistente conversacional y herramientas que permiten visualizar de manera clara el recorrido recomendado, es decir mostrar las líneas de transporte utilizada, los tramos de caminata y los transbordos. Así mismo se implementó módulos adicionales para el registro e inicio de sesión de los usuarios, consulta de historial y administración del perfil de usuario.

Como resultados de esta etapa se obtuvo una aplicación web totalmente funcional que integra todos los componentes desarrollados durante la investigación, con la finalidad de poder

proporcionar una herramienta capaz de asistir a los usuarios en la planeación de rutas de transporte público urbano como podemos ver en la Figura 6.

Figura 6 Pantalla principal de la plataforma conversacional



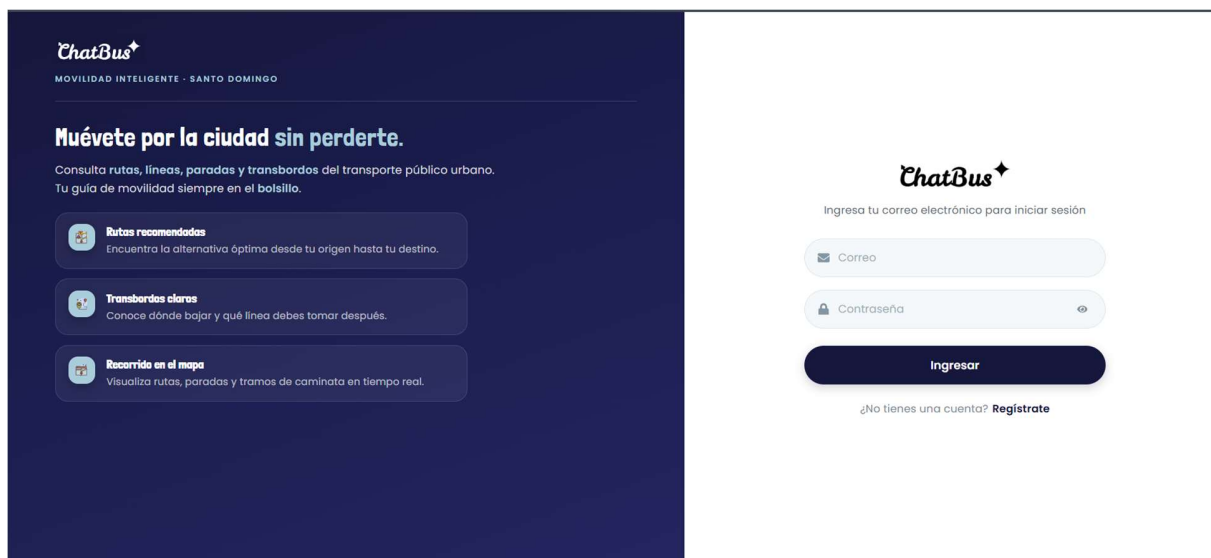
Implementación de la plataforma web

Como parte del desarrollo de la plataforma se implementó una interfaz web orientada a facilitar la interacción entre el usuario y el sistema de planificación de rutas. La aplicación fue diseñada con el fin de considerar criterios de simplicidad, accesibilidad y facilidad de uso, permitiendo así que el usuario pueda acceder a las principales funcionalidades mediante una navegación intuitiva.

La plataforma incorpora módulos de registro e inicio de sesión, administración de perfil, consulta de historial de conversaciones y acceso de asistente conversacional. Cada uno de estos componentes fueron creados con el propósito de mejorar la experiencia de uso y así permitir que el usuario pueda consultar rutas del transporte público urbano de manera organizada.

La Figura 7 muestra la pantalla de inicio de sesión implementada en la plataforma la cual permite autenticar a los usuarios mediante sus credenciales previamente registradas y controlar el acceso a las funcionalidades del sistema.

Figura 7 Pantalla de inicio de sesión de la plataforma.



La Figura 8 presenta el módulo de registro de usuarios, mediante el cual los nuevos usuarios pueden crear una cuenta para acceder a la plataforma. Aquí es donde se recopila la información básica requerida para la creación de una cuenta permitiendo así el inicio de sesión para tener acceso a las funcionalidades de planificación de rutas y almacenamiento de historial de rutas.

Figura 8 Pantalla de registro de usuarios.

La Figura 9 corresponde a la pantalla principal de la plataforma, donde se integra el asistente conversacional con el mapa interactivo utilizado para visualizar las rutas generadas. La interfaz concentra en una misma vista el mapa de la ciudad, el asistente conversacional y una barra lateral donde se encuentra el historial de chats y información de la aplicación

Figura 9 Pantalla principal de la plataforma conversacional.

La Figura 10 muestra el módulo de historial de conversaciones, encargado de almacenar las consultas realizadas por cada usuario gracias a esto permite consultar nuevamente recorridos previamente generados, evitando que el usuario tenga que repetir la misma consulta cada vez que requiera visualizar una ruta.

Figura 10 Historial de conversaciones.

Historial de rutas

Volver

Fecha	Origen	Destino	Transbordos	Favorito	Detalle
2026-07-26	Mi ubicación actual	maya moncayo	2	Agregar	Ver
2026-07-26	Mi ubicación actual	moncayo	1	Agregar	Ver
2026-07-26	Mi ubicación actual	maya	0	Agregar	Ver
2026-07-22	Mi ubicación actual	km 7 la quevedo	0	Agregar	Ver
2026-07-22	Mi ubicación actual	chorrera	0	Agregar	Ver
2026-07-21	Mi ubicación actual	el paseo shopping	2	Agregar	Ver
2026-07-21	Mi ubicación actual	camino a la bengala	1	Agregar	Ver

Finalmente, la Figura 11 presenta el módulo de perfil del usuario, donde se visualiza la información correspondiente a la cuenta autenticada y además permite visualizar cuantas conversaciones ah tenido, cuales son sus favoritas y funciones como actualizar su información o cerrar cesión.

Figura 11 *Perfil del usuario.*

Perfil de usuario

Gestiona tu información personal y preferencias de movilidad. [← Volver al inicio](#)

Sistema Inteligente de Movilidad Urbana
Consulta rutas óptimas, administra tus trayectos favoritos y recibe recomendaciones inteligentes mediante IA.

Mathias Gualpa
Usuario registrado
Activo

Correo
megualpa@espe.edu.ec

Ciudad
Santa Domingo

ID Usuario
1

Editar perfil

Historial de rutas

Estadísticas:

- 25** Viajes consultados
- 0** Rutas favoritas
- 25** Consultas con IA

Información personal

Nombre completo Mathias Gualpa	Correo electrónico megualpa@espe.edu.ec
Tipo de usuario Ciudadano	Última conexión Hoy

Acciones de cuenta

[Cambiar contraseña](#) [Cerrar sesión](#)

Procesamiento conversacional y planificación de rutas

La función principal de la plataforma desarrollada es brindar a los usuarios la posibilidad de consultar las rutas del transporte público mediante el uso de lenguaje natural. Para ello, el sistema incorpora diferentes elementos que funcionan de manera coordinada para analizar la consulta, ubicar los lugares requeridos, determinar la mejor ruta y mostrar el resultado en un mapa interactivo.

Cada consulta se procesa secuencialmente, empezando por la interpretación del mensaje por medio de inteligencia artificial y concluyendo con la representación visual de la ruta sugerida. Los resultados alcanzados en cada fase de este proceso se muestran a continuación.

Procesamiento mediante lenguaje natural

El procesamiento mediante lenguaje natural corresponde a la interpretación del mensaje que fue enviado por el usuario mediante un lenguaje natural para esto se integró el modelo de lenguaje de Google Gemini el cual nos ayuda a analizar la consulta e identifica

automáticamente el origen y el destino solicitado, transformando la información en una estructura comprensible para el resto de los componentes del sistema.

La aplicación web permite que el usuario pida sus consultas utilizando expresiones cotidianas con el fin de no tener que emplear una estructura específica o conocer previamente las líneas de transporte disponible. De esta manera consultas como Estoy en el shopping y quiero ir hasta la bengala son interpretadas correctamente por el modelo de lenguaje, identificando las ubicaciones que posteriormente serán utilizadas para el cálculo del recorrido.

La Figura 12 muestra el resultado obtenido durante el procesamiento de una consulta realizada por el usuario. En esta etapa, el modelo de lenguaje analiza el mensaje recibido e identifica automáticamente el origen y el destino que serán utilizados por el sistema para iniciar el proceso de planificación de la ruta.

Figura 12 *Procesamiento de una consulta mediante lenguaje natural.*

```
=====
PROCESAMIENTO DE CONSULTA
=====
Mensaje recibido: Quiero ir desde el shopping hasta la bengala
Ubicación GPS: No proporcionada

INTERPRETACIÓN MEDIANTE IA
-----
Origen identificado: shopping
Destino identificado: bengala
```

Como se observa en la Figura 12, el sistema recibió la consulta Quiero ir desde el shopping hasta la bengala y el modelo de lenguaje identificó correctamente el origen Shopping y el destino La Bengala. Una vez obtenida esta información, el backend inició automáticamente el proceso de búsqueda de las ubicaciones y el cálculo de la ruta correspondiente.

Localización de lugares

Después de que se ha determinado el origen y el destino a través del procesamiento del lenguaje natural, la plataforma comienza a localizar los lugares pedidos por el usuario. Para esto, el backend primero consulta los datos guardados en los archivos locales de

intersecciones y lugares. Los archivos que se han generado, que recogen los puntos de interés más importantes, las paradas y las referencias urbanas que utiliza el sistema.

En esta fase se lleva a cabo un proceso de normalización del texto que se ha recogido, de manera que se puede reconocer diferentes formas de referirse a un mismo lugar. Así, el sistema tiene la posibilidad de reconocer adecuadamente apodos, nombres alternativos y formas variadas de escribir, mejorando así la identificación de los lugares y reduciendo la posibilidad de error ante una búsqueda.

La Figura 13 presenta el resultado del proceso de localización correspondiente a la consulta realizada entre el Shopping y La Bengala. En esta etapa, el sistema resolvió los nombres detectados por el modelo de lenguaje y obtuvo las coordenadas geográficas asociadas a cada ubicación.

Figura 13 *Localización del origen y destino de la consulta.*

LOCALIZACIÓN DE LUGARES

Origen localizado: Paseo Shopping
 Coordenadas origen: -0.249245515318525, -79.16300454735978
 Destino localizado: La Bengala
 Coordenadas destino: -0.229402, -79.198529

Como se observa en la Figura 13, el término shopping fue relacionado con el lugar registrado como Paseo Shopping, mientras que el destino bengala fue identificado como La Bengala. La obtención de las coordenadas de ambos puntos permitió que el sistema continuara con la selección de las paradas cercanas y el posterior cálculo del recorrido.

Cálculo de rutas

Después de identificar las coordenadas del punto de partida y el de llegada, la plataforma comienza a calcular la mejor ruta empleando el grafo del transporte público urbano que fue previamente elaborado. En este momento se detectan las paradas más próximas a los

dos puntos y se escogen las que brindan el trayecto más adecuado, teniendo en cuenta la distancia de caminata, cuántos transbordos hay y si el trayecto es continuo.

La ruta se calcula utilizando el algoritmo de Dijkstra, que se ejecuta en la estructura de grafo creada con NetworkX. Cada una de las paradas del sistema es un nodo, y los enlaces entre ellas son las aristas del grafo. Asimismo, con el fin de prevenir cambios innecesarios de bus y generar trayectos más eficaces para el usuario, se tienen en cuenta las penalizaciones por cambiar de línea durante la búsqueda.

La Figura 14 muestra el resultado obtenido durante el cálculo de la ruta para la consulta realizada entre el Paseo Shopping y La Bengala. En esta etapa el sistema seleccionó la línea de transporte correspondiente, determinó el número de paradas que conforman el recorrido y calculó la cantidad de transbordos necesarios.

Figura 14 Resultado del cálculo de la ruta óptima.

```
=====
RESULTADOS DE CONSULTA
=====
Información del plan de viaje:
Ruta principal recomendada: LINEA_10_NORTE
Líneas alternativas disponibles: LINEA_05_SUR, LINEA_12_OESTE
Cobertura del servicio: ZONA METROPOLITANA COMPLETA
Tiempo promedio de viaje: 45 minutos (estimado)
Costo del boleto: $2.50 USD
Opciones de tarifa y pago:
  Tarifa estándar: $2.50 USD
  Tarifa reducida (estudiantes/seniors): $1.25 USD (requiere acreditación)
  Abonos mensuales disponibles.
Instrucciones de trasbordo (si aplica):
  Esta ruta no requiere trasbordos adicionales.
Puntos de interés cercanos al destino:
  Parque de la ciudad, Museo de Arte Moderno, Estación de tren.
Estado del sistema: Operando con normalidad.
Recomendaciones: Verifique horarios y disponibilidad antes del viaje.
=====
```

Como se observa en la Figura 14, la plataforma determinó que el recorrido puede realizarse utilizando la línea LINEA_21_RETORNO, recorriendo un total de 24 paradas y sin necesidad de efectuar transbordos. Asimismo, se identificaron las paradas de subida y bajada, junto con la distancia aproximada que el usuario debe recorrer caminando al inicio y al final del trayecto, permitiendo generar una recomendación de ruta acorde con las condiciones del sistema de transporte urbano.

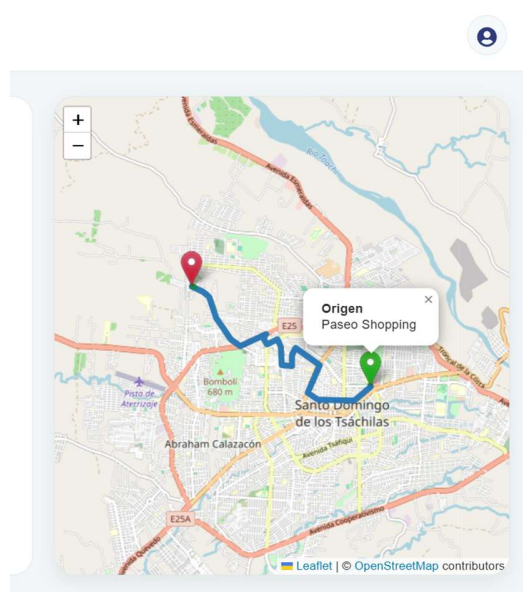
Generación del recorrido

La plataforma, después de calcular la ruta óptima, genera el trayecto que se le presentará al usuario. En esta etapa se genera la información geográfica respectiva al recorrido mediante el formato GeoJSON lo cual permite que sea representada en el mapa, las rutas de transporte seleccionadas, las paradas de subida/bajada y los segmentos del trayecto y los tramos de caminata requeridos para culminar el desplazamiento.

El sistema también realiza automáticamente la estimación de la caminata de inicio desde el lugar de procedencia hasta la parada de subida, la del final del trayecto desde la última parada hasta el destino requerido y también del trayecto en transporte público. Con este fin, se incorporó el servicio OpenStreetMap Routing Machine (OSRM), que tiene la responsabilidad de crear trayectos para peatones empleando la red vial existente.

La Figura 15 presenta el recorrido generado por la plataforma para la consulta realizada entre el Paseo Shopping y La Bengala. En la representación cartográfica es posible identificar la ruta del transporte público, los puntos de inicio y destino, así como los tramos de caminata calculados automáticamente por el sistema.

Figura 15 *Recorrido generado por la plataforma conversacional.*



Como se observa en la Figura 15, la plataforma representa de forma integrada todos los elementos que conforman el recorrido recomendado. La visualización permite al usuario identificar fácilmente el punto de inicio, la ruta del transporte público, las paradas utilizadas y el destino final, facilitando la comprensión del recorrido antes de iniciar el desplazamiento

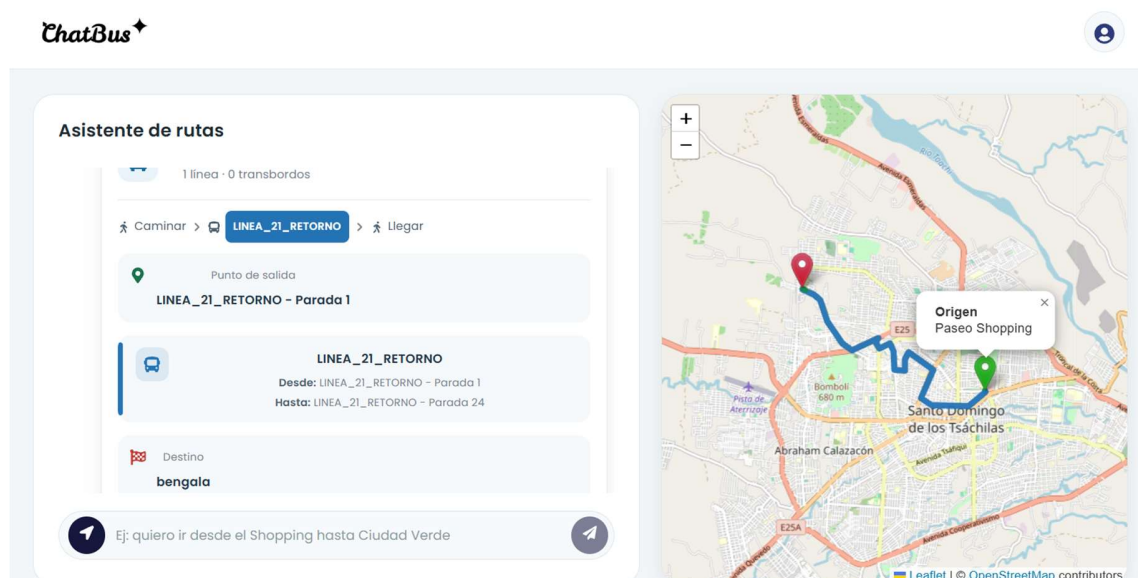
Visualización de rutas e interacción con el usuario

Después de que se genera el trayecto, la aplicación ofrece al usuario una respuesta que incluye información textual y la representación gráfica del camino sugerido. El recorrido se puede entender con facilidad gracias a la interfaz, que posibilita reconocer las líneas de transporte empleadas, los puntos de subida y bajada, las posibles transferencias y las partes en que se debe caminar para finalizar el desplazamiento.

La respuesta conversacional ofrece indicaciones organizadas de manera secuencial, señalando al usuario las acciones que tiene que llevar a cabo durante la ruta. Al mismo tiempo, el recorrido se ilustra visualmente en el mapa interactivo mediante varios componentes gráficos: líneas que indican la ruta del transporte público y marcadores que señalan el inicio y el final de la ruta, además de segmentos que corresponden a las caminatas inicial y final. El cruce entre la respuesta textual y la representación cartográfica simplifica el análisis de la información y optimiza el uso que se hace de la plataforma.

La Figura 16 muestra la visualización del recorrido generado por la plataforma después de procesar una consulta realizada por el usuario. En ella se observa la respuesta emitida por el asistente conversacional junto con la representación gráfica del trayecto sobre el mapa interactivo.

Figura 16 Visualización del recorrido generado por la plataforma.



La plataforma ofrece al usuario una recomendación integral del recorrido, fusionando datos escritos con componentes visuales que facilitan la comprensión del trayecto propuesto, como se puede apreciar en la Figura 16. Al mostrar al mismo tiempo las instrucciones y el mapa, se facilita la identificación de la ruta sugerida y se posibilita que el usuario entienda con más facilidad el trayecto propuesto, lo que hace que la experiencia de navegación sea más intuitiva y comprensible.

Pruebas del sistema

Se llevaron a cabo las pruebas sobre el sistema para asegurar que la plataforma conversacional que sirve para planear las rutas del transporte público urbano funcionaba como era esperado. Amanera que en esta etapa se comprobaron las funciones más importantes que se implementaron, se verificó que el sistema respondiera correctamente a las cuestiones que hacían los usuarios y que se generaran recorridos coherentes con la información geográfica existente para el sistema.

Por tanto, se llevaron a cabo pruebas que eran funcionales y operativas para comprobar la conducta de la plataforma y, por tanto, se quería constatar si se podía procesar las preguntas

en lenguaje natural, la identificación de ubicaciones, la planificación de rutas y la presentación del trayecto en el mapa interactivo. Posteriormente, se utilizó el cuestionario System Usability Scale (SUS) para evaluar la percepción de los usuarios respecto al uso fácil de la plataforma.

Pruebas funcionales

Las pruebas funcionales se llevaron a cabo usando la técnica de caja negra, analizando las funciones principales implementadas en la plataforma sin tener en cuenta su estructura interna. Para ello, se empleó la ficha de observación elaborada durante la investigación, que permitió el registro de la conducta del sistema ante diversos contextos de uso y corroborar que los requisitos funcionales preestablecidos estuviesen siendo cumplidos. Las pruebas de validación consisten en la ejecución de consultas sobre distintas rutas del tráfico público urbano; comprobándola con la correcta interpretación de consultas con lenguaje natural, la identificación del origen y destino, la localización de lugares, el cálculo de rutas, la generación del recorrido y la gráfica sobre el mapa interactivo. Para los resultados obtenidos ha permitido comprobar que la plataforma se comportó bien en todos los casos que tratado.

La Tabla 8 presenta el resumen de las pruebas funcionales realizadas sobre la plataforma conversacional.

Tabla 8 *Resultados de las pruebas funcionales*

Caso de prueba	Resultado esperado	Resultado obtenido	Estado
Inicio de sesión	Permitir el acceso al sistema	Acceso realizado correctamente	Cumple
Registro de usuarios	Registrar una nueva cuenta	Usuario registrado correctamente	Cumple
Procesamiento de consultas	Identificar origen y destino	Identificación correcta	Cumple
Localización de lugares	Encontrar ubicaciones registradas	Lugares localizados correctamente	Cumple

Cálculo de rutas	Generar la mejor ruta disponible	Ruta calculada correctamente	Cumple
Generación del recorrido	Construir el recorrido completo	Recorrido generado correctamente	Cumple
Visualización en el mapa	Mostrar la ruta sobre el mapa	Ruta visualizada correctamente	Cumple
Historial de conversaciones	Guardar consultas realizadas	Historial almacenado correctamente	Cumple

Como se observa en la Tabla 8, todas las funcionalidades evaluadas obtuvieron resultados satisfactorios, evidenciando que la plataforma cumple con los requerimientos funcionales definidos durante el desarrollo de la investigación. No se identificaron errores que impidieran el funcionamiento normal del sistema durante la ejecución de las pruebas realizadas.

Pruebas de funcionamiento

Para asegurar que la plataforma funciona de manera integral, se llevaron a cabo pruebas enfocadas en confirmar que los distintos componentes del sistema se comunican adecuadamente. Estas pruebas posibilitaron la evaluación de la comunicación entre el frontend y el backend, además de la integración de los servicios que se emplean para determinar ubicaciones, planear rutas y procesar conversaciones.

Durante la realización de las pruebas, se comprobó que la autenticación de usuarios, el procesamiento de consultas a través de Google Gemini, la comunicación con la base de datos PostgreSQL, el cálculo de rutas por medio del algoritmo de Dijkstra y NetworkX, así como la obtención de coordenadas geográficas mediante Nominatim y la creación de rutas para peatones gracias a OpenStreetMap Routing Machine (OSRM), funcionaban correctamente. Además, se verificó que los resultados conseguidos estuvieran adecuadamente representados en el mapa interactivo que se implementó en la plataforma.

El resumen de las pruebas funcionales efectuadas en los componentes más relevantes del sistema se encuentra en la Tabla 9.

Tabla 9 *Resultados de las pruebas de funcionamiento*

Componente evaluado	Resultado esperado	Resultado obtenido	Estado
Frontend	Visualizar correctamente la interfaz	Funcionamiento correcto	Cumple
Backend	Procesar consultas del usuario	Funcionamiento correcto	Cumple
PostgreSQL	Almacenar usuarios e historial	Funcionamiento correcto	Cumple
Google Gemini	Interpretar consultas en lenguaje natural	Funcionamiento correcto	Cumple
Nominatim	Obtener coordenadas geográficas	Funcionamiento correcto	Cumple
NetworkX y Dijkstra	Calcular la mejor ruta disponible	Funcionamiento correcto	Cumple
OSRM	Generar recorridos peatonales	Funcionamiento correcto	Cumple
Leaflet	Representar la ruta sobre el mapa	Funcionamiento correcto	Cumple

Como se observa en la Tabla 9, todos los componentes evaluados funcionaron correctamente durante las pruebas realizadas, evidenciando una adecuada integración entre los servicios que conforman la plataforma. La comunicación entre el frontend y el backend permitió procesar las consultas de los usuarios de forma satisfactoria, mientras que los servicios externos y las herramientas empleadas para el cálculo y representación de rutas respondieron conforme a lo esperado, garantizando el correcto funcionamiento del sistema.

Pruebas de usabilidad (SUS)

Con el propósito de evaluar la usabilidad de la plataforma conversacional desarrollada, se implementó el cuestionario de la Escala de Usabilidad del Sistema (SUS) a un grupo de 15 participantes que habían interactuado con el sistema antes, haciendo preguntas sobre cómo planificar rutas para el transporte público urbano en Santo Domingo. El cuestionario contaba con diez preguntas cerradas que empleaban una escala de Likert de cinco niveles, lo cual permitió evaluar la percepción de los usuarios con relación a la sencillez del uso, la facilidad para aprender, la integración de las funciones y el grado de satisfacción general en el momento de interactuar con la plataforma.

Para obtener el puntaje de usabilidad, las respuestas fueron procesadas siguiendo el procedimiento establecido por Brooke. En las preguntas impares se restó una unidad al valor seleccionado por el participante, mientras que en las preguntas pares se restó la respuesta del valor cinco. Posteriormente, la suma de los valores obtenidos se multiplicó por 2,5, obteniendo un puntaje comprendido entre 0 y 100 para cada participante [38].

La Figura 17 presenta las respuestas obtenidas por los quince participantes para cada una de las diez preguntas que conforman el cuestionario System Usability Scale (SUS). Estas respuestas constituyen la base para el cálculo del puntaje de usabilidad de la plataforma.

Figura 17 Respuestas obtenidas del cuestionario SUS aplicado a los participantes.

	1. Creo que me gustaría utilizar este sistema con frecuencia.	2. Encontré el sistema innecesariamente complejo.	3. Pensé que el sistema era fácil de usar.	4. Creo que necesitaría el apoyo de un técnico para poder usar este sistema.	5. Encontré que las diversas funciones del sistema estaban bien integradas.	6. Pensé que había demasiada inconsistencia en este sistema.	7. Imagino que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.	8. Encontré el sistema muy incómodo de usar.	9. Me sentí muy seguro(a) usando el sistema.	10. Necesité aprender muchas cosas antes de poder empezar a usar este sistema.	
2	4	2	4	1	5	1	5	1	5	1	
3	5	1	5	2	4	2	4	2	4	2	
4	4	1	4	2	5	1	5	1	4	1	
5	3	2	3	2	5	2	3	1	4	1	
6	5	1	4	1	5	2	4	1	3	1	
7	4	1	4	2	3	1	4	2	4	1	
8	5	1	5	1	5	1	5	1	5	1	
9	3	2	4	1	3	1	3	1	3	1	
10	5	1	4	1	4	1	4	2	4	2	
11	4	1	5	2	5	1	5	1	4	1	
12	5	2	4	1	5	2	4	1	5	1	
13	5	2	4	1	4	1	5	1	4	1	
14	3	3	3	3	3	3	3	3	3	3	
15	1	4	2	4	2	3	3	3	2	3	
16	5	2	4	1	4	2	4	2	4	1	

Con base en las respuestas originales, se realizó la transformación de cada uno de los ítems de acuerdo con la metodología propuesta por Brooke. En las preguntas impares se restó una unidad al valor seleccionado por el participante, mientras que en las preguntas pares se aplicó la operación cinco menos la respuesta obtenida. La Figura 18 muestra los valores transformados correspondientes a los quince participantes.

Figura 18 Transformación de las respuestas del cuestionario SUS.

Form_Responses										
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
1										
2	3	3	3	4	4	4	4	4	4	4
3	4	4	4	3	3	3	3	3	3	3
4	3	4	3	3	4	4	4	4	3	4
5	2	3	2	3	4	3	2	4	3	4
6	4	4	3	4	4	3	3	4	2	4
7	3	4	3	3	2	4	3	3	3	4
8	4	4	4	4	4	4	4	4	4	4
9	2	3	3	4	2	4	2	4	2	4
10	4	4	3	4	3	4	3	3	3	3
11	3	4	4	3	4	4	4	4	3	4
12	4	3	3	4	4	3	3	4	4	4
13	4	3	3	4	3	4	4	4	3	4
14	2	2	2	2	2	2	2	2	2	2
15	0	1	1	1	1	2	2	2	1	2
16	4	3	3	4	3	3	3	3	3	4

Posteriormente, se realizó la suma de los diez valores transformados para cada participante y el resultado fue multiplicado por 2,5, obteniendo el puntaje SUS individual. La Figura 19 presenta los puntajes obtenidos por cada participante, así como el promedio general y la desviación estándar calculados para la muestra evaluada.

Figura 19 Cálculo del puntaje SUS y resultados estadísticos.

Suma	Puntaje SUS	PROMEDIO	DESVIACION ESTANDAR
37	92,5		
33	82,5		
36	90		
30	75		
35	87,5		
32	80		
40	100		
30	75	80,33333333	17,54755444
34	85		
37	92,5		
36	90		
36	90		
20	50		
13	32,5		
33	82,5		

Interpretación y análisis de usabilidad del chatbus

Antes de analizar los resultados obtenidos, es necesario considerar los criterios de interpretación de la escala System Usability Scale (SUS). De acuerdo con Sauro y Lewis , la puntuación SUS puede clasificarse en diferentes niveles de usabilidad, desde la categoría F, que representa una baja percepción de usabilidad, hasta la categoría A, que corresponde a sistemas con una excelente experiencia de uso y que se ubican dentro del 15 % con mejor valoración por parte de los usuarios [38]. La Figura 20 presenta la clasificación propuesta por Sauro y Lewis para interpretar los puntajes obtenidos mediante la escala SUS.

Figura 20 Interpretación de la puntuación de la escala System Usability Scale (SUS).

Puntuación SUS	Calificación	Percentil
78,9 - 100	A	Top 15%
72,6 - 78,8	B	Top 35%
62,7 - 72,5	C	Top 65%
51,7 - 62,6	D	Top 85%
0 - 51,6	F	-

Como se observa en la Figura 19, la plataforma obtuvo un puntaje promedio de 80,33 y una desviación estándar de 17,55. De acuerdo con la clasificación presentada en la Figura 20, este resultado ubica a la plataforma dentro de la categoría A, lo que indica que los usuarios perciben el sistema como una herramienta fácil de utilizar, intuitiva y adecuada para la planificación de rutas del transporte público urbano. Igualmente, también se observa que la mayor parte de los participantes que llevaron a cabo la prueba del sistema alcanzaron puntuaciones superiores a 75 puntos, lo cual deja entrever una buena puntuación en la usabilidad del sistema. Es verdad que se generaron puntuaciones inferiores a las medias, pero estas corresponden a casos concretos, que en ningún caso varían el resultado global del sistema.

Los resultados que se han obtenido en este trabajo, finalmente, hacen ver que el sistema proporciona la satisfacción para el usuario en cuanto a la experiencia de uso, alcanzando de esta forma el objetivo de ofrecer una herramienta de planeación de trayectos ante el lenguaje natural que sea accesible e intuitiva.

Discusión de resultados

Los resultados obtenidos a lo largo de la creación de la plataforma conversacional demuestran que incorporar métodos de procesamiento del lenguaje natural y modelos de

grafos es una opción factible para planificar rutas del transporte público en las ciudades. El estudio del estado de la técnica reveló que la mayor parte de las plataformas existentes se enfocan en optimizar matemáticamente los recorridos y requieren información con un alto grado de estructura, lo cual restringe su uso en ciudades donde los usuarios utilizan referencias cotidianas para relatar sus desplazamientos. La propuesta elaborada en este sentido incluye un modelo de conversación basado en Google Gemini, que posibilita la interpretación de consultas formuladas en lenguaje natural y su adecuación al contexto urbano de Santo Domingo, lo cual permite superar una de las limitaciones más importantes que se encontraron en los estudios analizados.

Desde el punto de vista técnico, la implementación del sistema permitió integrar de manera coordinada múltiples tecnologías como lo fue Google Gemini para el procesamiento de lenguaje natural, NetworkX y el algoritmo de Dijkstra para el cálculo de rutas, Nominatim para la geocodificación de ubicaciones y el OSRM para la generación de recorridos peatonales. Las pruebas de funcionamiento y funcionales señalaban que estos elementos actúan de forma integrada, lo que permite localizar de manera precisa el origen y el destino de las consultas, así como la localización de los lugares necesarios, el cálculo de rutas y transbordos de acuerdo con los cambios de línea y el mapeo de los trayectos en un mapa interactivo, lo que permitirá concluir que se han conseguido las metas asociadas con la arquitectura del modelo y el progreso de la plataforma.

La valoración de la experiencia de los usuarios, ejecutada con la escala System Usability Scale (SUS), alcanzó un puntaje medio de 80,33. A seguir las guías interpretativas recomendadas por Sauro y Lewis, la plataforma fue colocada en la categoría A. Estos resultados ponen de manifiesto que los participantes consideraban la plataforma una solución intuitiva, fácil de aprender y adecuada para el planeamiento de las rutas del transporte público urbano. La evaluación alcanzada confirma que el uso de un agente conversacional favorece la

interacción con el sistema, disminuyendo así la necesidad de que los usuarios tengan información previa sobre las líneas o paradas de transporte que hay disponibles.

Los resultados obtenidos son, en líneas generales, una muestra de que la plataforma que se ha desarrollado cumple el objetivo general de la investigación, ya que la plataforma constituye una solución inteligente para recomendar rutas del transporte público urbano de Santo Domingo. De la unión de la modelación de grafos y el procesamiento del lenguaje natural se garantizó poder ofrecer una solución adecuada al contexto local que puede interpretar las referencias que los habitantes usan y generar trayectos directamente con una interfaz web fácil de usar. Con miras al futuro, la plataforma tiene la posibilidad de mejorar incluyendo datos sobre el transporte en tiempo real (por ejemplo, las condiciones del tráfico o los tiempos estimados de llegada) y así lograr que las recomendaciones generadas sean más útiles y precisas.

Capítulo V

Conclusiones

Se ha podido llevar a cabo satisfactoriamente el desarrollo de una plataforma web inteligente fundamentada en un agente conversacional para la recomendación de rutas en un sistema de transporte público urbano de la ciudad de Santo Domingo, haciendo uso del procesamiento de lenguaje natural de Google Gemini y de la modelación lógica de grafos a través de NetworkX. Los resultados presentados indican que la propuesta tiene un carácter transferible para contribuir a la movilidad del ciudadano habilitándole a los usuarios consultar las rutas de manera natural y ayudarse con referencias espaciales locales sin tener conocimiento previo para identificar las líneas del transporte o las paradas del mismo que tiene que usar. De esta manera, la investigación sostiene que la combinación heterogénea del procesamiento del lenguaje natural y la teoría de grafos representaría una alternativa

tecnológica factible para el tratamiento de la planificación informática de rutas urbanas intermedias como la de Santo Domingo de los Tsáchilas.

La identificación de las tecnologías y enfoques existentes para la planificación de rutas inteligentes a partir del estado del arte permitió conocer cómo las soluciones comerciales y académicas priorizan la eficiencia matemática en el cálculo de rutas en detrimento del modo de comunicación del usuario, y fue a partir de aquí que se tuvo la oportunidad de realizar una identificación de la situación que intentó responder esta investigación. Por otro lado, como consecuencia de ese análisis anterior, se modeló una arquitectura por capas (relativa a la presentación, a la aplicación, a la inteligencia artificial, a los servicios geográficos, al proceso de rutas y al almacenamiento) que permitió organizar el flujo de procesamiento de cada consulta de forma ordenada, del aterrizaje del lenguaje natural a la obtención de la ruta final.

La implementación de la plataforma realizando el uso de las tecnologías Python, Flask, PostgreSQL, React y Leaflet, permitió la creación de una aplicación web que es completamente apta y orientada a la práctica, recolectando registros y logueos nuevos de los usuarios, obtener las rutas en un lenguaje natural, obtener la posición del usuario en tiempo real, la obtención de las rutas óptimas mediante el algoritmo de Dijkstra, la visualización mediante mapas digitales de tipo interactivo y el control del historial de las conversaciones mantenidas. Así mismo, el uso de servicios geo codificadores geográficos externos como Nominatim y OSRM permitió obtener una información geográfica del sistema enriquecida mediante la geocodificación de lugares no registrados, el cálculo de rutas peatonales reales, lo que permitieron obtener una calidad enriquecida de las rutas que se le proponen al usuario.

La validación, tanto la funcional como la técnica, confirmó que la plataforma consigue en su globalidad resultados satisfactorios para aquellos aspectos evaluados en la misma. Las pruebas funcionales corroboraron el 100% de los requisitos planteados durante el proceso de desarrollo. Las pruebas funcionales corroboraron la correcta interrelación de todos los

elementos: frontend, backend, base de datos y servicios externos utilizados por la plataforma para el procesamiento de lenguaje natural, la geocodificación, y el cálculo de rutas. Por último, la evaluación practicada mediante la escala System Usability Scale (SUS), dio como resultado una media de 80,33, lo que corresponde a la categoría de A, es decir, una aceptación y gran percepción de facilidad de uso por parte del usuario que participó en la validación del sistema.

Recomendaciones

Se sugiere realizar una extensión gradual de la cobertura de la plataforma hacia la totalidad de las líneas y cooperativas que prestan el servicio de transporte urbano en el cantón Santo Domingo, así como hacia cantones o parroquias cercanas, con el objeto de establecer una herramienta de movilidad global de la comunidad Tsáchila. La arquitectura en capas adoptada en el proceso de desarrollo permite la inclusión de nuevas rutas de transporte, paradas y lugares de interés sin que sea necesario modificar de forma excesiva la estructura general del sistema.

El motor de procesamiento de rutas de NetworkX y el núcleo del algoritmo de Dijkstra son en efecto capaces de calcular adecuadamente las rutas, se recomienda que se avance en la calibración de la ponderación de las características del grafo e introducir información sobre la situación del tráfico en tiempo real y también habitual de las unidades de transporte y los tiempos de llegada previstos, para que las recomendaciones que emergen de su ejecución estén adaptadas de manera activa a las condiciones de uso del transporte urbano y no sólo a determinada distancia o a los penalizadores de manera estática.

Se recomienda seguir robusteciendo las propiedades de la accesibilidad y de la inclusión de la plataforma al implementar nuevas funcionalidades de modo que se pueda hacer las consultas, por ejemplo, el uso de comandos de voz, las opciones de alto contraste en los gráficos, compatibilidad con lectores de pantalla, las opciones para personas con deficiencia

visual o motora, etc., con el fin de incrementar la inclusión digital de la aplicación e incentivar un uso más masivo de ella por parte de una población más heterogénea del cantón.

Se sugiere la posibilidad de hacer la integración de la aplicación de la plataforma conversacional con otras aplicaciones de la ciudad, como sistemas municipales de información del tráfico, aplicaciones de cobro electrónico del transporte público, calendarios de fiestas, plataformas de gestión del tráfico gobernadas por parte del Gobierno Autónomo Descentralizado; de esta manera se realiza la transición de transformar la herramienta desarrollada en un servicio permanente de consulta para la ciudadanía, dando así más rendimiento de lo que permite la recomendación de trayectos.

Se sugiere realizar la integración de la aplicación de la plataforma conversacional con otras aplicaciones de la ciudad, como los sistemas municipales de información del tráfico, aplicaciones de pago electrónico del transporte público, calendarios de fiestas, plataformas de gestión del tráfico pertenecientes al Gobierno Autónomo Descentralizado; de este modo se llevaría a cabo la transición de transformar la herramienta desarrollada en un servicio permanente de consulta para la ciudadanía, extrayendo así más rendimiento de lo que permite la recomendación de trayectos.

En relación con el trabajo futuro se propone desarrollar investigaciones comparativas para evaluar el rendimiento de los diferentes modelos de lenguaje y algoritmos de búsqueda de rutas en situaciones de mayor complejidad de la geografía y también cuando hay un mayor número de usuarios. También sería interesante hacer un análisis del rendimiento de la incorporación del aprendizaje profundo en la predicción de la demanda, sobre recorridos adaptados en función del historial de trayectos del usuario o sobre mecanismos de retroalimentación de la ciudadanía en tiempo real, para hacer más extensa la evidencia en relación con las ventajas y las limitaciones en relación con la aplicación de plataformas conversacionales a la movilidad urbana en el caso de ciudades intermedias

Bibliografía

- [1] J. Z. Chacón, P. D. Sejas Churco y A. Sanchez Salces, «Optimización del transporte público en Santa Cruz mediante la aplicación de geolocalización Mango Street,» *Revista 3I Ingeniería, Innovación, Investigación*, 2022.
- [2] A. Costa, «Sistemas inteligentes de movilidad urbana en Río de Janeiro: una evaluación crítica,» *CVU*, vol. 15, p. 21, 2022.
- [3] I. Reyes, «vectorial,» *vectorial*, 14 Junio 2023. [En línea]. Available: <https://vectorial.substack.com/p/empresas-data-driven-1-uber-y-la>. [Último acceso: 26 Julio 2026].
- [4] M. A. A. Montoya, «repositorio,» 20 Octubre 2024. [En línea]. Available: <https://repositorio.utp.edu.co/server/api/core/bitstreams/63afedbb-df9e-48ba-92a3-8ab28d30aecf/content>. [Último acceso: 26 Julio 2026].
- [5] L. M. Vaca Bustos y L. V. Sánchez Parrales, «revistacodigocientifico,» 2024. [En línea]. Available: <https://revistacodigocientifico.itslosandes.net/index.php/1/article/view/693>. [Último acceso: 26 Julio 2026].
- [6] ibm, «¿Qué es la arquitectura de tres niveles?,» *ibm*, [En línea]. Available: <https://www.ibm.com/mx-es/think/topics/three-tier-architecture>. [Último acceso: 17 Mayo 2026].
- [7] C. Mayén, «ARQUITECTURA MODULAR,» *jgarqs*, 28 Agosto 2020. [En línea]. Available: <https://www.jgarqs.com/blog/2020/8/28/arquitectura-modular>. [Último acceso: 18 Mayo 2026].

- [8] raiolanetworks, «Que es PIP en Python,» raiolanetworks, [En línea].
Available: <https://raiolanetworks.com/ayuda/que-es-pip-python/>. [Último acceso: 18 Mayo 2026].
- [9] arsys, «¿Qué es Flask (Python) y cuáles son sus ventajas?,» arsys, [En línea]. Available: <https://www.arsys.es/blog/que-es-flask-python-y-cuales-son-sus-ventajas>. [Último acceso: 18 Mayo 2026].
- [10] ibm, «¿Qué es PostgreSQL?,» ibm, [En línea]. Available:
<https://www.ibm.com/mx-es/think/topics/postgresql>. [Último acceso: 18 Mayo 2026].
- [11] unbosque, «¿Qué es arquitectura de software?,» unbosque, 16 Enero 2026. [En línea]. Available: <https://www.unbosque.edu.co/educacion-continua/blog-educacion-continua/que-es-arquitectura-de-software>. [Último acceso: 17 Mayo 2026].
- [12] D. iterativo, «Desarrollo iterativo,» appmaster, [En línea]. Available:
<https://appmaster.io/es/glossary/desarrollo-iterativo-es>. [Último acceso: 26 07 2026].
- [13] S. Hudge, «medium,» medium, 16 Octubre 2024. [En línea]. Available:
<https://medium.com/@sagar.hudge/layers-in-software-architecture-c8cc16329ff6>.
[Último acceso: 20 Julio 2026].
- [14] O. Blancarte, «reactiveprogramming,» reactiveprogramming, [En línea].
Available: <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/capas>.
[Último acceso: 26 Julio 2026].

- [15] ovhcloud, «What is a REST API?,» ovhcloud, [En línea]. Available: <https://www.ovhcloud.com/en/learn/what-is-rest-api/>. [Último acceso: 18 Mayo 2026].
- [16] oracle, «¿Qué es Python?,» oracle, [En línea]. Available: <https://www.oracle.com/latam/developer/what-is-python-for-developers/>. [Último acceso: 18 Mayo 2026].
- [17] D. Datta, «What is SQLAlchemy Used For? An Overview with Practical Examples,» cdata, 28 Marzo 2024. [En línea]. Available: <https://www.cdata.com/blog/what-is-sqlalchemy>. [Último acceso: 2026 Mayo 19].
- [18] nekomath, «Uso básico de NetworkX,» nekomath, [En línea]. Available: <https://madi.nekomath.com/P5/NetworkX.html>. [Último acceso: 25 07 2026].
- [19] D. A., «Qué es React: definición, características y funcionamiento,» hostinger, 28 Marzo 2025. [En línea]. Available: <https://www.hostinger.com/es/tutoriales/que-es-react>. [Último acceso: 19 Mayo 2026].
- [20] P. Jose, «¿Qué es Vite?,» pablogarciajc, 06 Agosto 2025. [En línea]. Available: https://pablogarciajc.com/blog/frontend/que-es-vite/#1_%C2%BFQue_es_Vite_y_por_que_es_importante. [Último acceso: 2026 Mayo 19].
- [21] N. Dissanayake, «React Router vs. React Router DOM: Understanding the Differences,» syncfusion, 30 Abril 2026. [En línea]. Available: <https://www.syncfusion.com/blogs/post/react-router-vs-react-router-dom>. [Último acceso: 19 Mayo 2026].

- [22] arsys, «Axios Javascript: analizamos las características de este ligero cliente HTTP,» arsys, [En línea]. Available: <https://www.arsys.es/blog/axios>. [Último acceso: 19 Mayo 2026].
- [23] «geoinnova,» geoinnova, 11 Agosto 2021. [En línea]. Available: <https://geoinnova.org/blog-territorio/que-es-un-sig-gis-o-sistema-de-informacion-geografica/>. [Último acceso: 25 Julio 2026].
- [24] J. Yesares, «¿Qué es Leaflet?,» jesusyesares, 29 Junio 2022. [En línea]. Available: <https://jesusyesares.com/visualizacion-datos/que-es-leaflet-libreria-javascript/>. [Último acceso: 2026 Mayo 19].
- [25] react-leaflet, «react-leaflet,» react-leaflet, [En línea]. Available: <https://react-leaflet.js.org/docs/start-introduction/>. [Último acceso: 19 Mayo 2026].
- [26] garmin, «What is OpenStreetMap?,» support.garmin, [En línea]. Available: <https://support.garmin.com/en-US/?faq=mSWyDE91q874qY0e9JkL1A>. [Último acceso: 19 Mayo 2026].
- [27] locationiq, «Fórmula Haversine,» locationiq, [En línea]. Available: <https://es.locationiq.com/glossary/haversine-formula>. [Último acceso: 25 07 2026].
- [28] L. M. L. Mangaña, «Qué es Json Web Token y cómo funciona,» openwebinars, 17 Enero 2020. [En línea]. Available: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>. [Último acceso: 19 Mayo 2026].
- [29] Argenis, «bcrypt, ¿que es y para que funciona?,» webempresa, 02 Mayo 2025. [En línea]. Available: <https://www.webempresa.com/blog/bcrypt-que-es-y-para-que-funciona.html>. [Último acceso: 19 Mayo 2026].

- [30] learn.microsoft, «¿Qué es Git?,» learn.microsoft, [En línea]. Available: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>. [Último acceso: 19 Mayo 2026].
- [31] ebac, «Qué es Github y para qué sirve: una guía para principiantes,» ebac, 04 Junio 2023. [En línea]. Available: <https://ebac.mx/blog/que-es-github>. [Último acceso: 19 Mayo 2026].
- [32] Y. Muradas, «Qué es Postman y primeros pasos,» openwebinars, 03 Junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-postman/>. [Último acceso: 19 Mayo 2026].
- [33] L. L. Cuerva, «pgAdmin para novatos: Cómo gestionar bases de datos sin complicaciones,» openwebinars, 03 Septiembre 2025. [En línea]. Available: <https://openwebinars.net/blog/pgadmin-para-novatos-como-gestionar-bases-de-datos-sin-complicaciones/>. [Último acceso: 19 Mayo 2026].
- [34] J. Vega, «EL INEC PRESENTÓ LOS RESULTADOS DEL CENSO 2023 EN SANTO DOMINGO DE LOS TSÁCHILAS,» zaracaytv, 19 Octubre 2023. [En línea]. Available: <https://zaracaytv.com/?p=9319>. [Último acceso: 25 Julio 2026].
- [35] R. H. SAMPIERI, «¿Cómo y cuáles son las muestras no probabilísticas?,» de *Metodología de la investigacion*, México, booksmedicos.org, 2014, p. 189.
- [36] J. Sauro, «Medición de la usabilidad con la Escala de Usabilidad del Sistema (SUS),» measuringu, 03 Febrero 2011. [En línea]. Available: <https://measuringu.com/sus/>. [Último acceso: 25 Julio 2026].
- [37] G. D. Miteu, «Ética en la investigación científica: una mirada a su importancia, historia y futuro.,» pmc, 21 Marzo 2024. [En línea]. Available:

<https://pmc.ncbi.nlm.nih.gov/articles/PMC11060189/>. [Último acceso: 25 Julio 2026].

[38] S. Serafinelli, «Qué es la escala SUS y cómo usarla para medir la usabilidad,» teacuplab, 08 Marzo 2024. [En línea]. Available: <https://www.teacuplab.com/es/blog/que-es-la-escala-sus-y-como-usarla-para-medir-la-usabilidad/>. [Último acceso: 25 Julio 2026].

[39] supermicro, «¿Qué es la arquitectura cliente-servidor?,» supermicro, [En línea]. Available: <https://www.supermicro.com/es/glossary/client-server-architecture>. [Último acceso: 17 Mayo 2026].